Documentation 8.0

ZABBIX

02.04.2026

Contents

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 Prägnante, aufgabenorientierte Anleitungen, die Sie durch die Grundlagen führen – von Ihren ersten Konfigurationsschritten bis zum Erhalt Ihrer ersten Problemwarnung.

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.

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.

1 Installation und erste Schritte

Um Zabbix von Grund auf einzurichten, folgen Sie diesem Ablauf:

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.

Weitere Details zu jedem Schritt folgen.

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, z. B. zu unterstützten Softwareversionen, finden Sie unter Erforderliche Software.

Zabbix installieren Der **einfachste** Weg, Zabbix zu installieren, ist über **Distributionspakete**. Die Pakete installieren kleinere Softwareabhängigkeiten.

Genaue Installationsanweisungen, zugeschnitten auf Ihr Betriebssystem, finden Sie auf der **Zabbix-Website**.

Wählen Sie Ihre Plattform aus und folgen Sie den 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 Agenten hinzufügen.

Ein Teil des Installationsprozesses ist das Erstellen einer Datenbank (zum Speichern von Konfigurations- und Verlaufsdaten) und das Importieren von Anfangsdaten in diese Datenbank.

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

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

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

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

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

1 Installation der Weboberfläche

Dieser Abschnitt enthält Schritt-für-Schritt-Anleitungen zur Installation der Zabbix-Weboberfläche. Führen Sie diese Schritte aus, nachdem Sie das Zabbix-Backend – Zabbix-Server, Agent und Datenbank – installiert haben.

Das Zabbix-Frontend ist in PHP geschrieben, daher wird für den Betrieb ein von PHP unterstützter Webserver benötigt.

> **Note:**
> Weitere Informationen zum Einrichten von SSL für das Zabbix Frontend finden Sie in diesen Best Practices.

Willkommensbildschirm

Öffnen Sie die URL des Zabbix Frontend im Browser. Wenn Sie Zabbix aus Paketen installiert haben, lautet die URL:

- für Apache: `http://<server_ip_or_name>/zabbix`
- für Nginx: `http://<server_ip_or_name>`

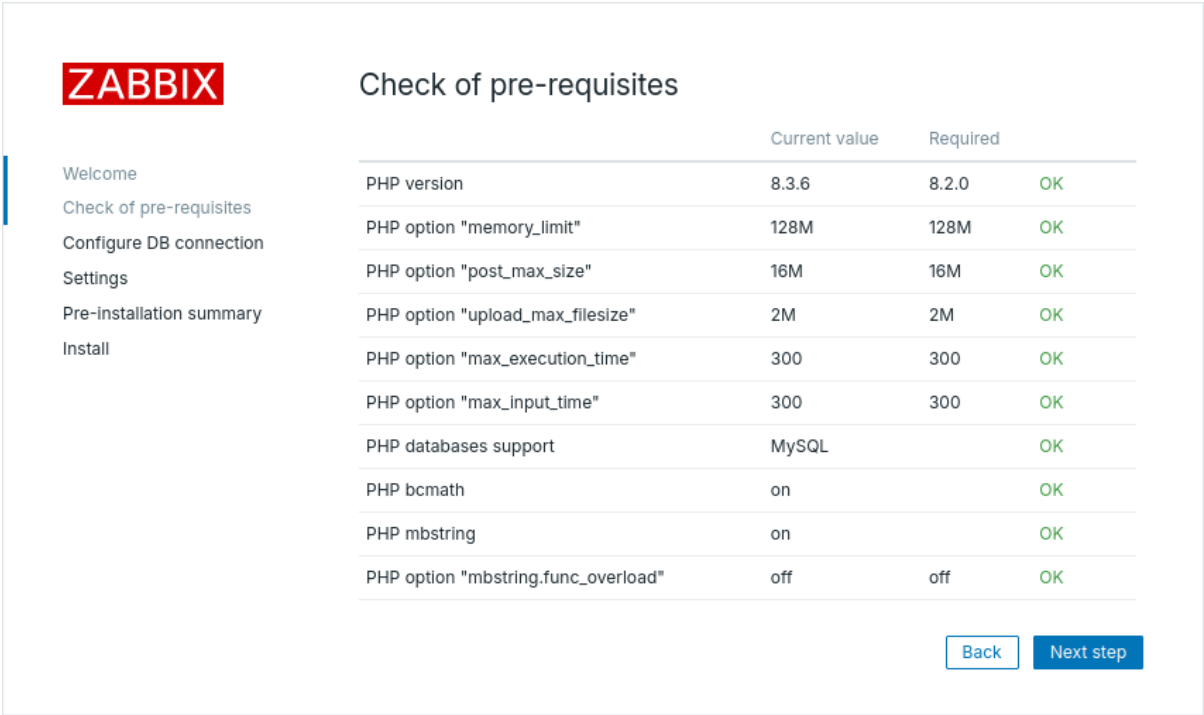
Sie sollten den ersten Bildschirm des Installationsassistenten des Frontend sehen.

Verwenden Sie das Dropdown-Menü *Default language*, um die Standardsprache des Systems zu ändern und den Installationsprozess in der ausgewählten Sprache fortzusetzen (optional). Weitere Informationen finden Sie unter Installation zusätzlicher Frontend-Sprachen.

Beachten Sie, dass das Festlegen der Sprache auf *English (en_US)* auch das US-Zeit-/Datumsformat im Frontend aktiviert.

Überprüfung der Voraussetzungen

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



Voraussetzung	Mindestwert	Beschreibung
PHP-Version	8.2.0	
PHP-Option "memory_limit"	128MB	In php.ini: memory_limit = 128M
PHP-Option "post_max_size"	16MB	In php.ini: post_max_size = 16M
PHP-Option "upload_max_filesize"	2MB	In php.ini: upload_max_filesize = 2M
PHP-Option "max_execution_time"	300 Sekunden	In php.ini: max_execution_time = 300 (die Werte 0 und -1 sind ebenfalls zulässig)
PHP-Option "max_input_time"	300 Sekunden	In php.ini: max_input_time = 300 (die Werte 0 und -1 sind ebenfalls zulässig)

Voraussetzung	Mindestwert	Beschreibung
Unterstützung von PHP-Datenbanken	Eines von: MySQL, PostgreSQL	Siehe Anforderungen für die Liste aller obligatorischen und optionalen PHP-Erweiterungen. Beachten Sie, dass nicht erfüllte optionale Voraussetzungen mit dem roten Status *Warnung* angezeigt werden und der Einrichtungsprozess auch dann fortgesetzt werden kann, wenn sie nicht erfüllt sind.
PHP bcmath	muss aktiviert sein	
PHP mbstring	muss aktiviert sein	
PHP-Option "mbstring.func_overload"	muss deaktiviert sein	In php.ini: mbstring.func_overload = 0
PHP-Option "session.auto_start"	muss deaktiviert sein	In php.ini: session.auto_start = 0
PHP-Option "arg_separator.output"	&	In php.ini: arg_separator.output = "&" (der Wert "&" ist ebenfalls zulässig)

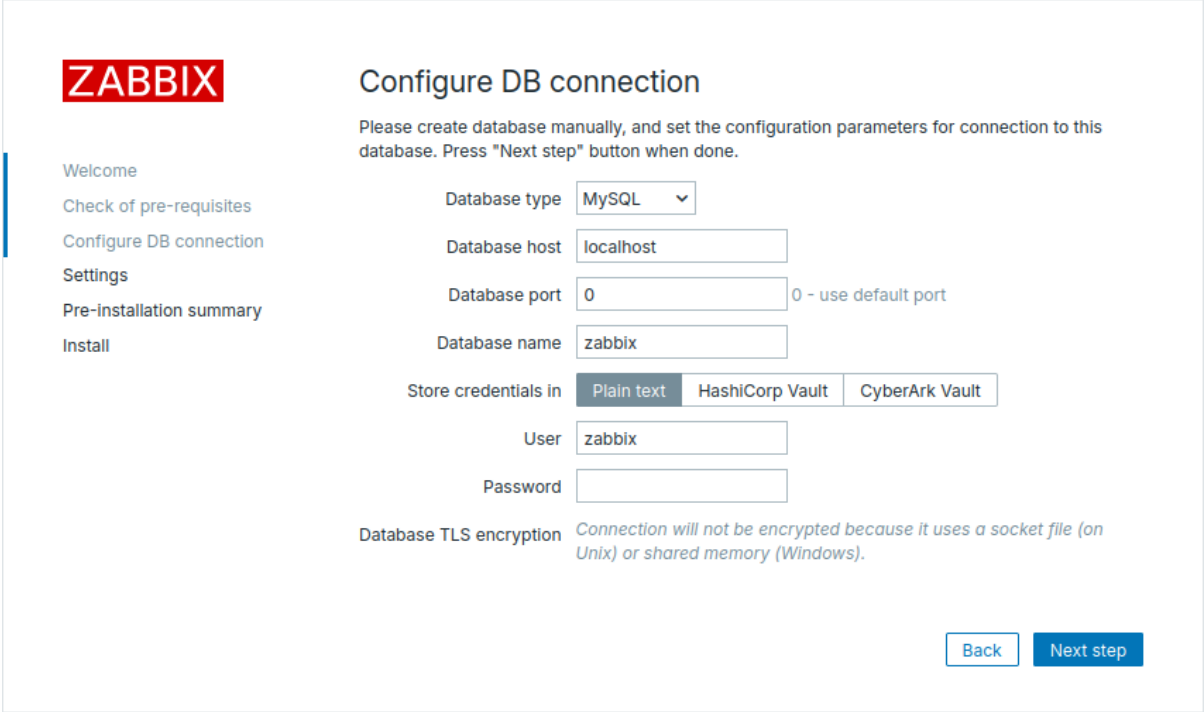
Attention:
Wenn der Apache-Benutzer oder die Apache-Benutzergruppe geändert werden muss, überprüfen Sie die Berechtigungen für den Sitzungsordner; andernfalls kann die Zabbix-Einrichtung möglicherweise nicht fortgesetzt werden.

DB-Verbindung konfigurieren

Geben Sie die Details für die Verbindung zur Datenbank ein. Die Zabbix-Datenbank muss bereits erstellt worden sein.

Bei MySQL führt die Eingabe von `localhost` oder das Leerlassen des Feldes *Database host* zu einer Verbindung über den standardmäßigen Unix-Socket. Das Einrichtungsformular bietet kein separates Feld *Database socket*, daher konfigurieren Sie für die Verwendung eines benutzerdefinierten Sockets diesen in den Zabbix-Server-Einstellungen (zum Beispiel mit `DBSocket=` in zabbix_server.conf). Dadurch bleibt das Frontend mit den Verbindungs-Einstellungen zwischen Server und Datenbank abgestimmt.

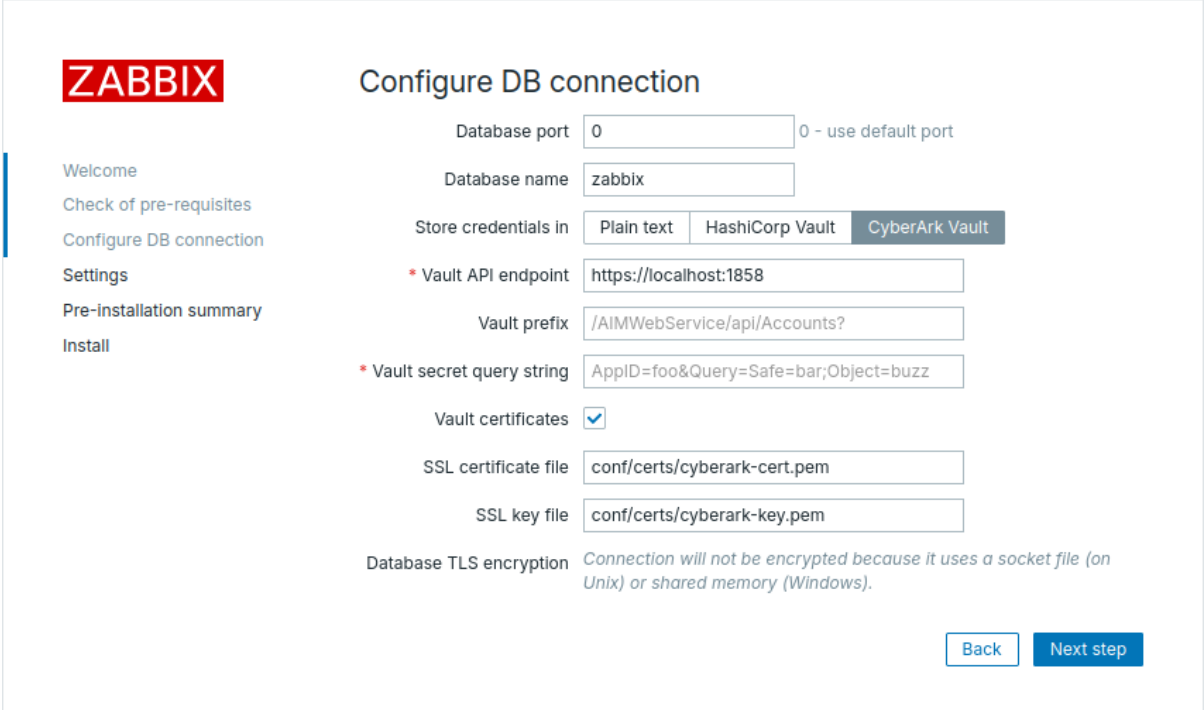
Bei PostgreSQL wird der standardmäßige Unix-Domain-Socket verwendet, wenn das Feld *Database host* leer gelassen wird. Wenn stattdessen ein Socket-Pfad eingegeben wird (zum Beispiel `/var/run/pgbouncer`), wird dieser Unix-Domain-Socket verwendet.



Wenn die Option *Database TLS encryption* aktiviert ist, erscheinen im Formular zusätzliche Felder zum Konfigurieren der TLS-Verbindung zur Datenbank (nur MySQL oder PostgreSQL).

Wenn *Store credentials in* auf HashiCorp Vault oder CyberArk Vault gesetzt ist, werden zusätzliche Parameter verfügbar:

- für HashiCorp Vault: Vault-API-Endpunkt, Vault-Präfix, Secret-Pfad und Authentifizierungs-Token;
- für CyberArk Vault: Vault-API-Endpunkt, Vault-Präfix, Secret-Abfragezeichenfolge und Zertifikate. Nach dem Aktivieren des Kontrollkästchens *Vault certificates* erscheinen zwei neue Felder zur Angabe der Pfade zur SSL-Zertifikatsdatei und zur SSL-Schlüsseldatei.

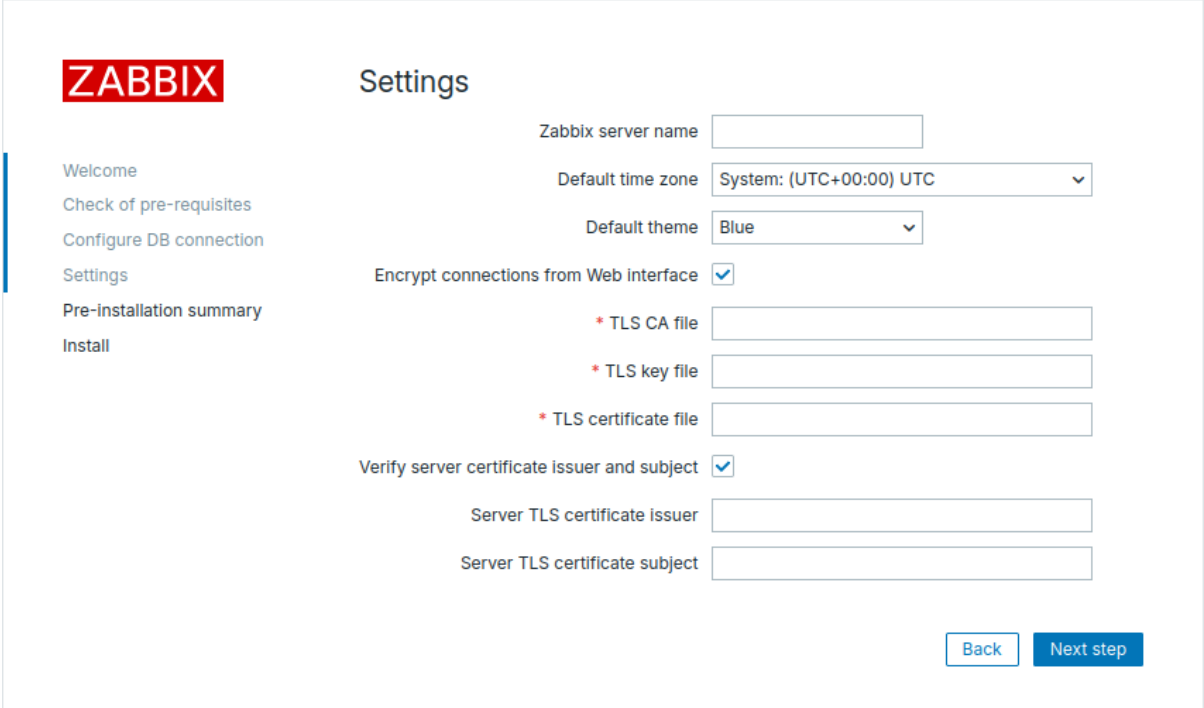


Einstellungen

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

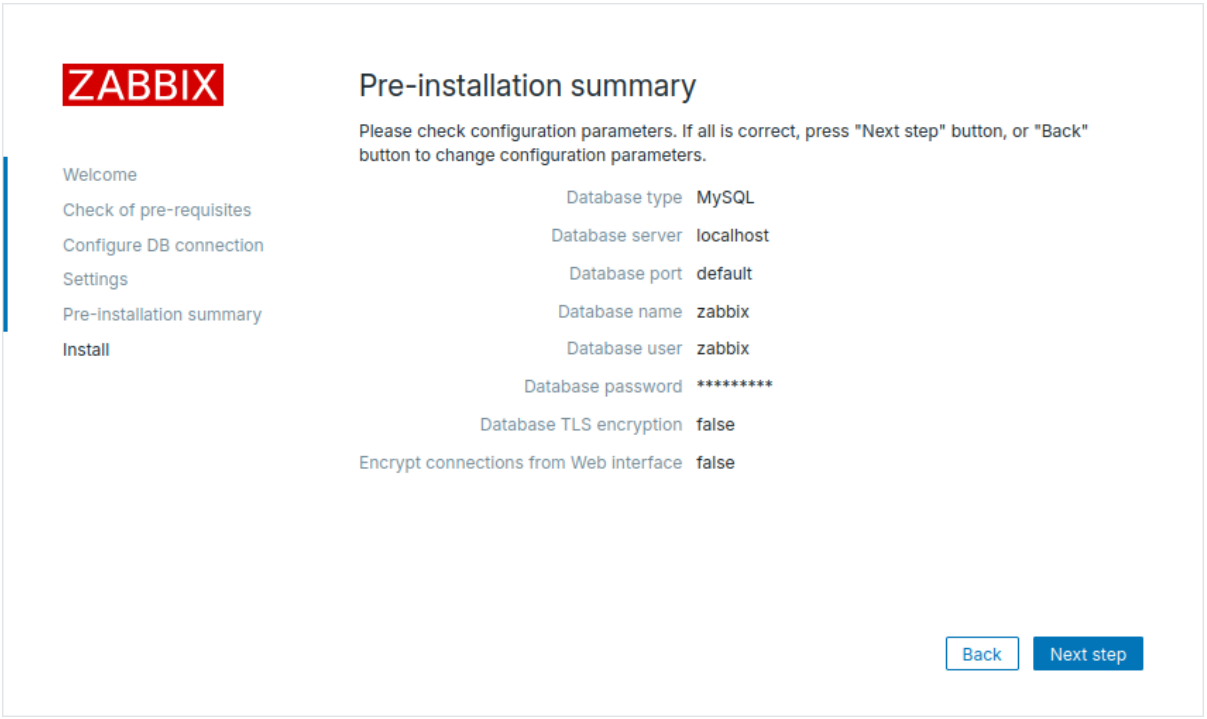
Legen Sie die Standard-Zeitzone und das Theme für das Frontend fest.

Wenn die Option *Verbindungen von der Weboberfläche verschlüsseln* aktiviert ist, werden im Formular zusätzliche Felder für die Konfiguration der TLS-Verbindung zwischen Zabbix Server und Frontend angezeigt.

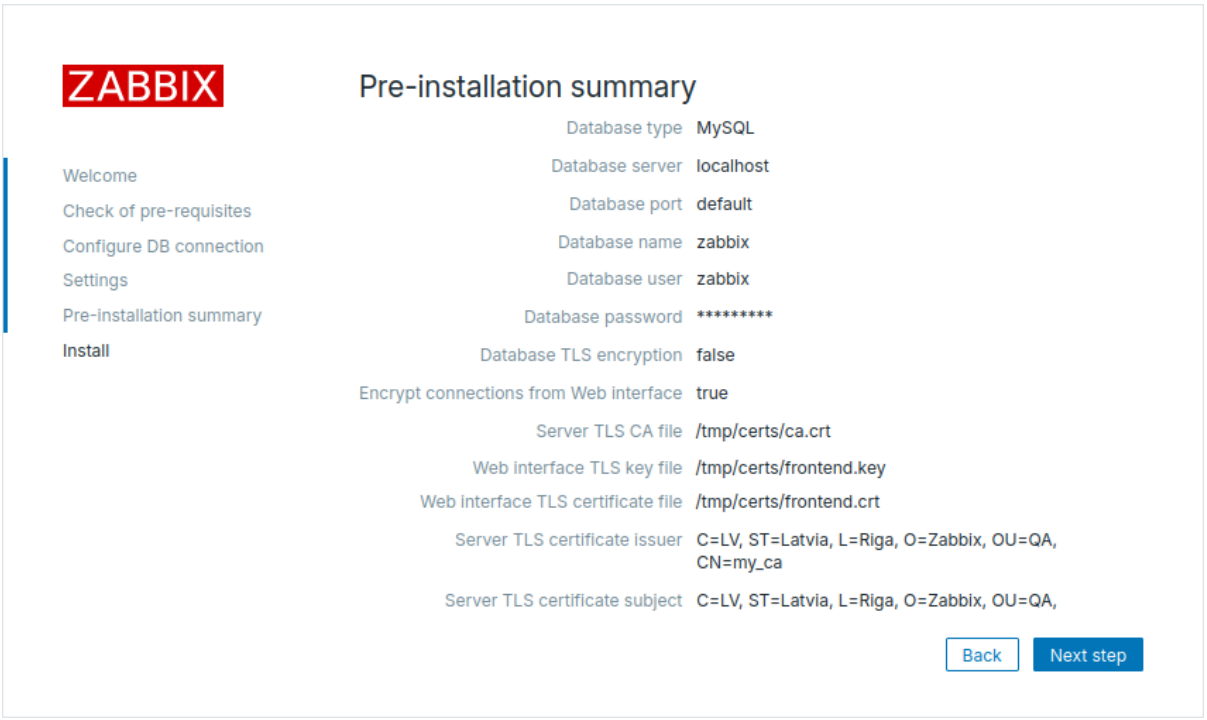


Zusammenfassung vor der Installation

Überprüfen Sie eine Zusammenfassung der Einstellungen.

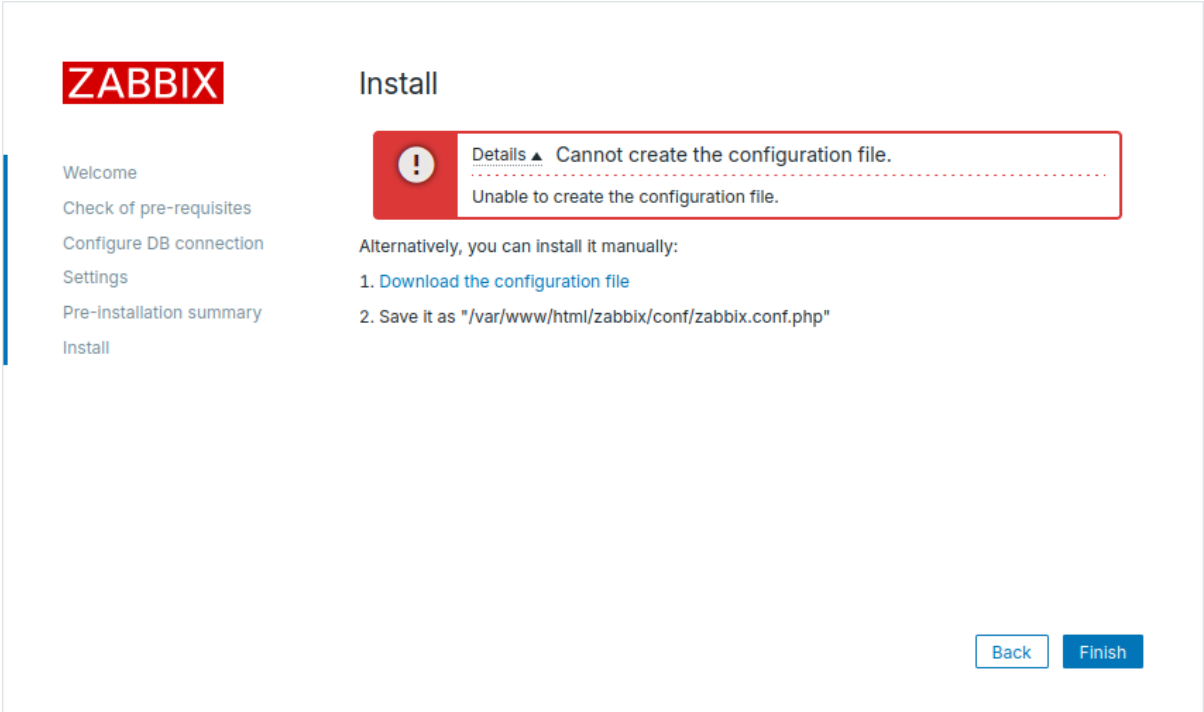


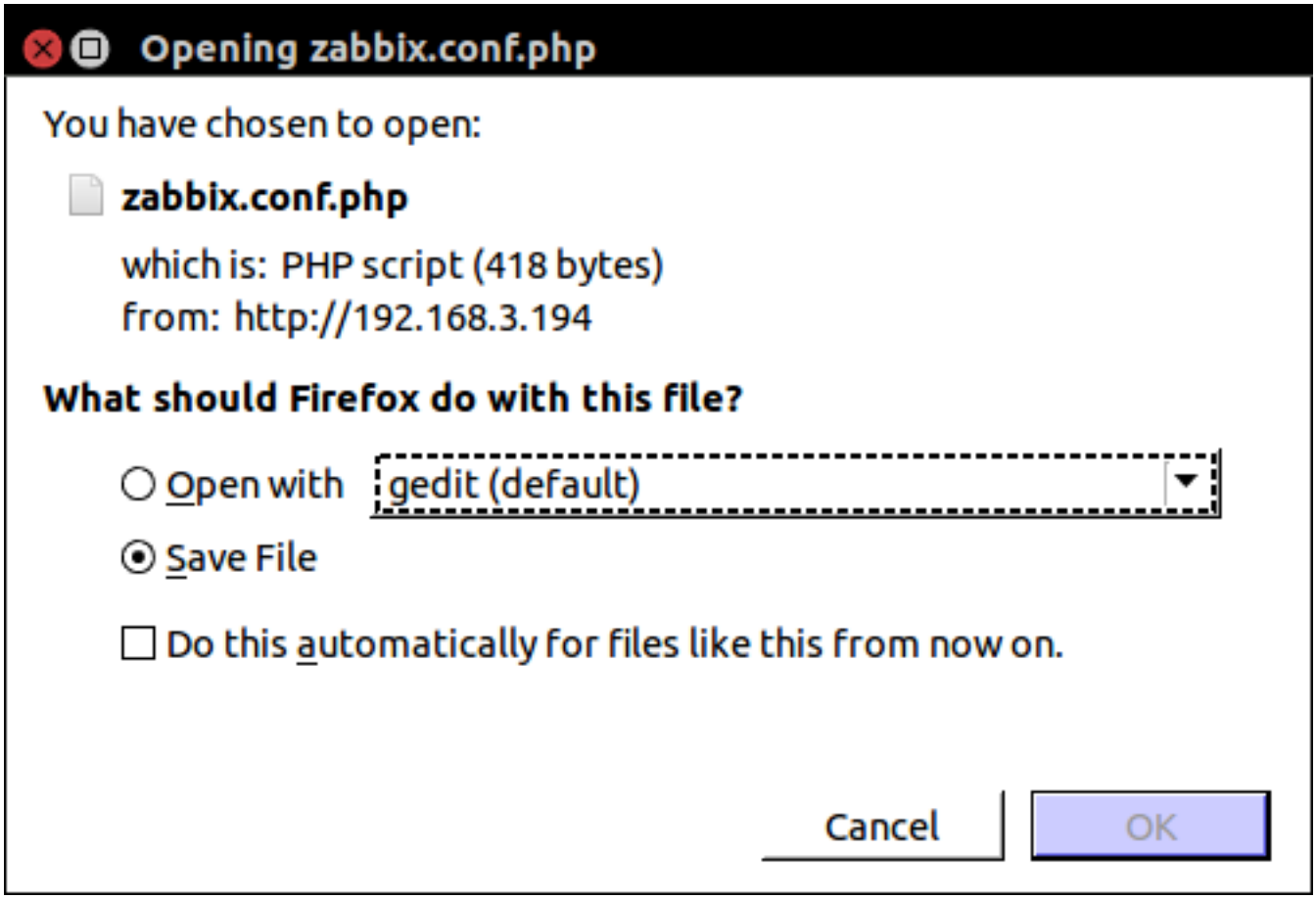
Die Unterseite zeigt die Daten an, wenn eine TLS-Konfiguration hinzugefügt wurde.



Installation

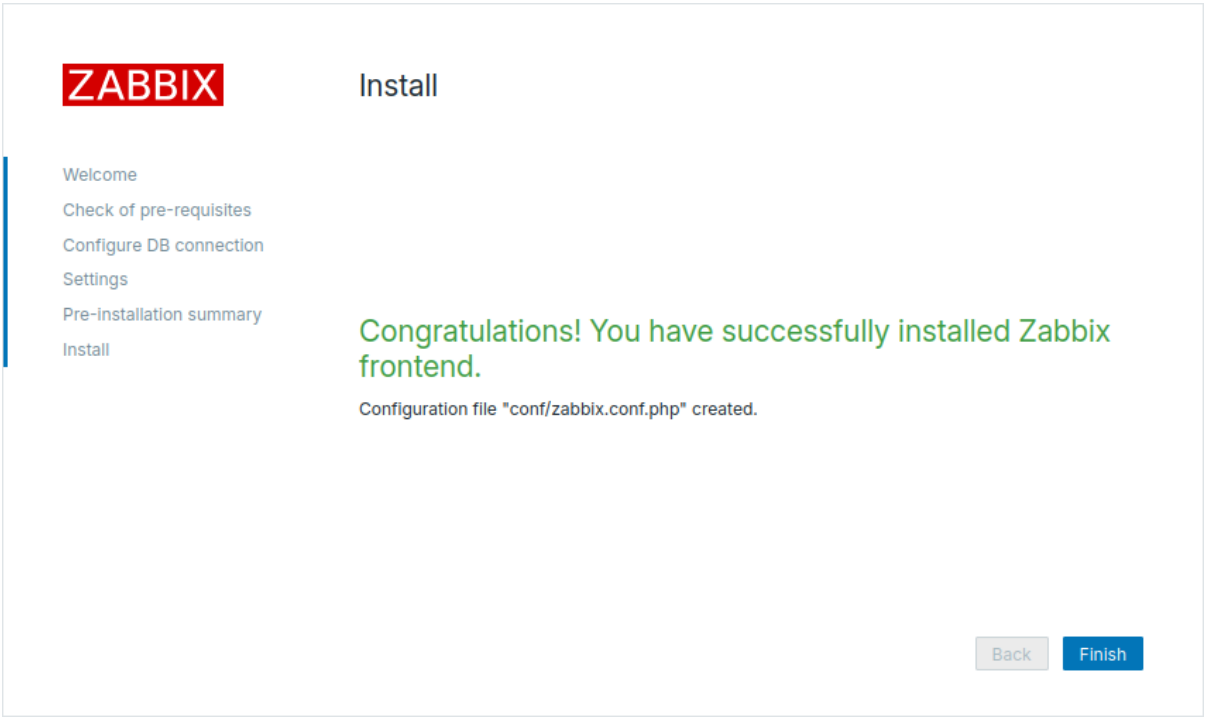
Wenn Sie Zabbix aus den Quellen installieren, laden Sie die Konfigurationsdatei herunter und legen Sie sie unter conf/ im Unterverzeichnis der HTML-Dokumente des Webservers ab, in das Sie die Zabbix-PHP-Dateien kopiert haben.





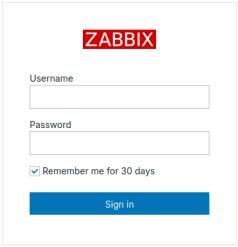
Note:
Wenn der Webserver-Benutzer Schreibzugriff auf das Verzeichnis conf/ hat, wird die Konfigurationsdatei automatisch gespeichert und Sie können sofort mit dem nächsten Schritt fortfahren.

Schließen Sie die Installation ab.



Anmelden

Das Zabbix Frontend ist bereit! Der Standardbenutzername ist **Admin**, das Passwort **zabbix**.



Fahren Sie mit der Anleitung zur Linux-Überwachung fort.

2 Schnellstartanleitungen

Nachdem Sie Zabbix und sein Frontend installiert haben, ist es an der Zeit, mit dem Monitoring zu beginnen.

Verwenden Sie diese Kurzanleitungen, um mit dem Monitoring gängiger Ressourcen zu beginnen:

- Monitoring von Linux-Servern
- Monitoring von Apache-Webservern
- Monitoring von MySQL-Servern

Schnellstartanleitungen konzentrieren sich auf die Verwendung offizieller Zabbix-Vorlagen für das Monitoring.
Eine Vorlage ist ein vorkonfigurierter Satz von Monitoring-Elementen.
Mit einer Vorlage lässt sich das Monitoring skalieren — Sie können eine Vorlage an einem einzelnen Monitoring-Ziel (Gerät, App usw.) testen, sie anpassen und dann auf mehrere ähnliche Monitoring-Ziele anwenden.

Vorlagen eignen sich zwar gut für sofort einsatzbereites Monitoring, aber möglicherweise möchten Sie auch deren Elemente anpassen.

Um die grundlegenden Elemente, aus denen eine Vorlage besteht, besser zu verstehen, lesen Sie Grundkonfiguration.

Grundkonfiguration

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

- Anmeldung und Benutzerkonfiguration
- Neuer Host
- Neuer Datenpunkt

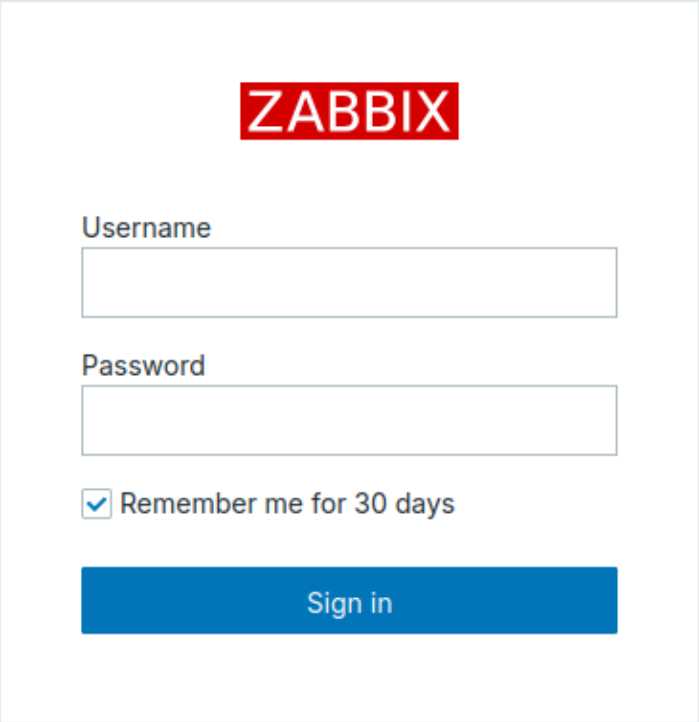
- Neuer Auslöser
- Empfang von Problembenachrichtigungen
- Neue Vorlage

1 Anmeldung und Konfiguration des Benutzers

Übersicht

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

Anmeldung



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 Standard-Passwort für das Admin-Konto unmittelbar nach der ersten Anmeldung zu ändern.

Dauerhafte Anmeldung

Um bis zu 30 Tage angemeldet zu bleiben, wählen Sie *30 Tage merken*, bevor Sie auf *Anmelden* klicken.

30 Tage merken aktiviert:

- Ihre Sitzung bleibt 30 Tage lang aktiv.
- *Automatische Abmeldung* wird außer Kraft gesetzt, sodass Sie bis zum Ende dieses 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 aufgehoben.
- Die Sitzung läuft gemäß dem konfigurierten Intervall für die *Automatische Abmeldung* ab.

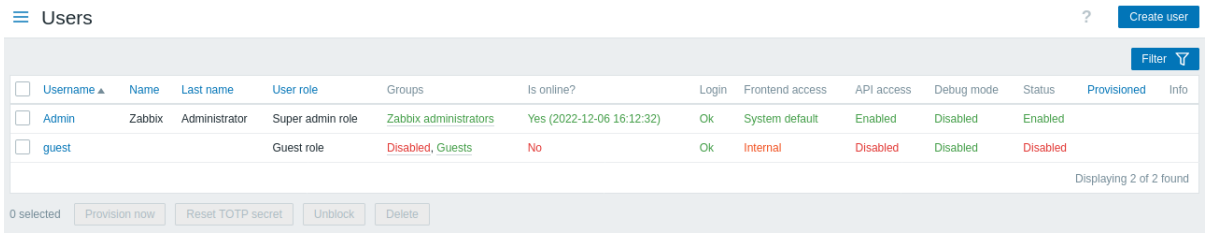
Schutz vor Brute-Force-Angriffen

Bei fünf aufeinanderfolgenden fehlgeschlagenen Anmeldeversuchen pausiert das Zabbix-Frontend für 30 Sekunden, um Brute-Force- und Wörterbuchangriffe zu verhindern.

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

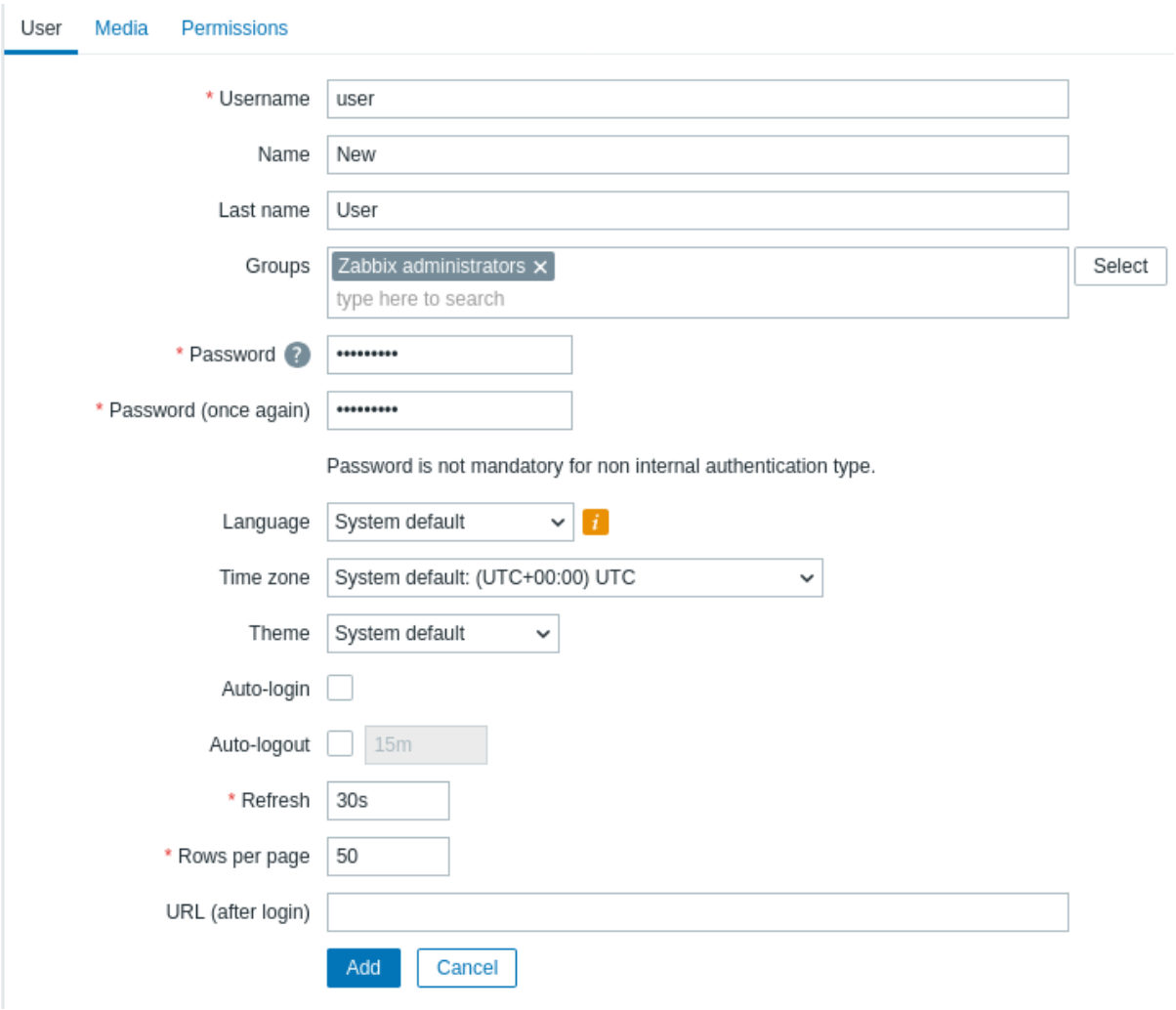
Benutzer hinzufügen

Um Informationen über Benutzer anzuzeigen, gehen Sie im vertikalen Menü der Seitenleiste zu *Benutzer* > *Benutzer*.



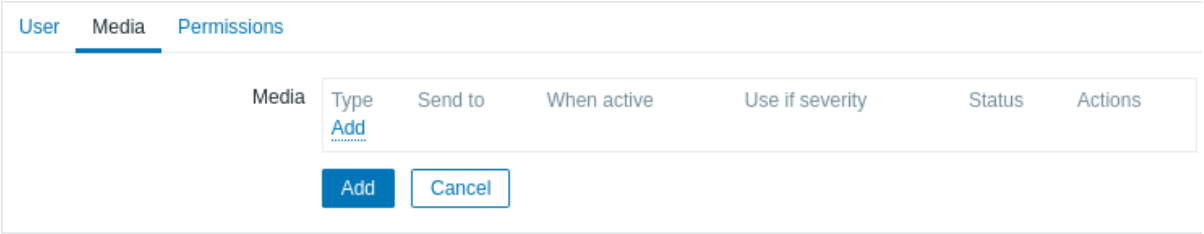
Um einen neuen Benutzer hinzuzufügen, wählen Sie oben rechts *Benutzer erstellen* aus.

Stellen Sie im Formular für den neuen Benutzer sicher, dass Sie Ihren Benutzer zu einer der vorhandenen Benutzergruppen hinzufügen, zum Beispiel „Zabbix-Administratoren“.



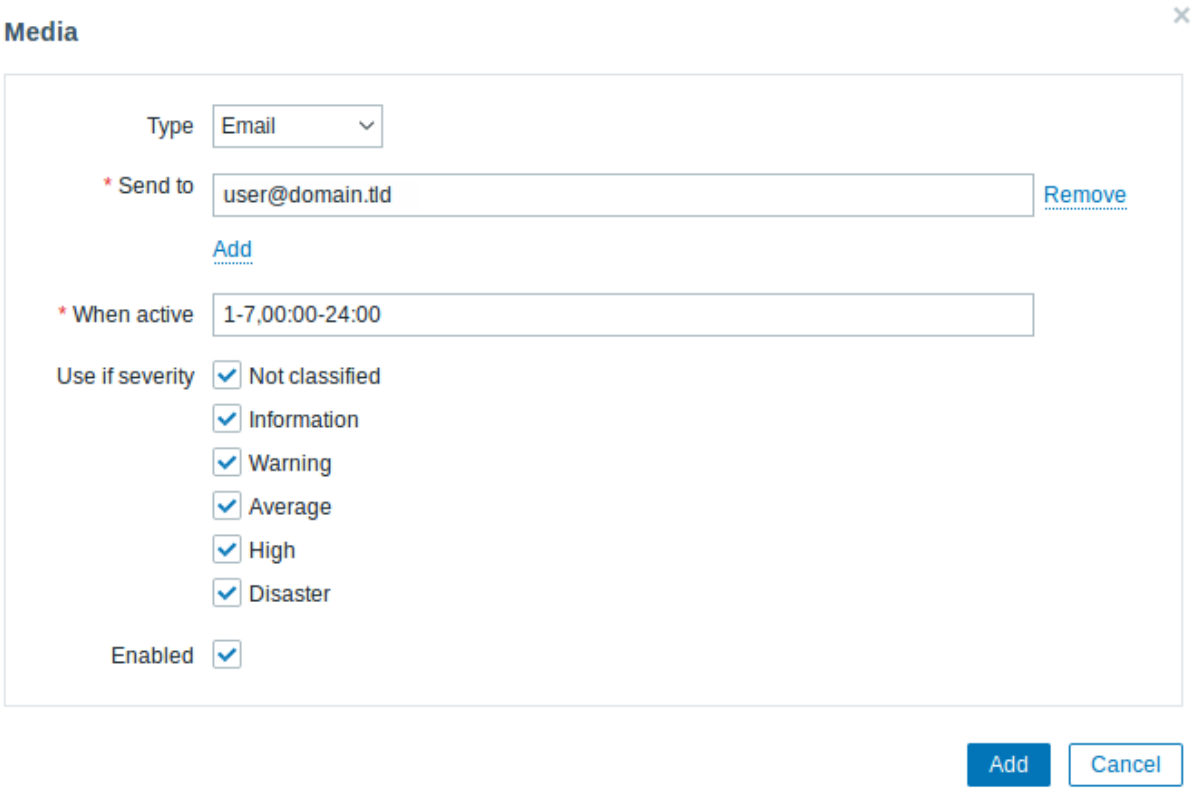
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert. Ausführliche Informationen zu den Eingabefeldern in diesem Konfigurationsformular finden Sie auf der Seite Benutzereinstellungen.

Standardmäßig sind für neue Benutzer keine Medien (Benachrichtigungszustellungsmethoden) definiert. Um eines zu erstellen, gehen Sie zur Registerkarte „Medien“ und klicken Sie auf *Hinzufügen*.



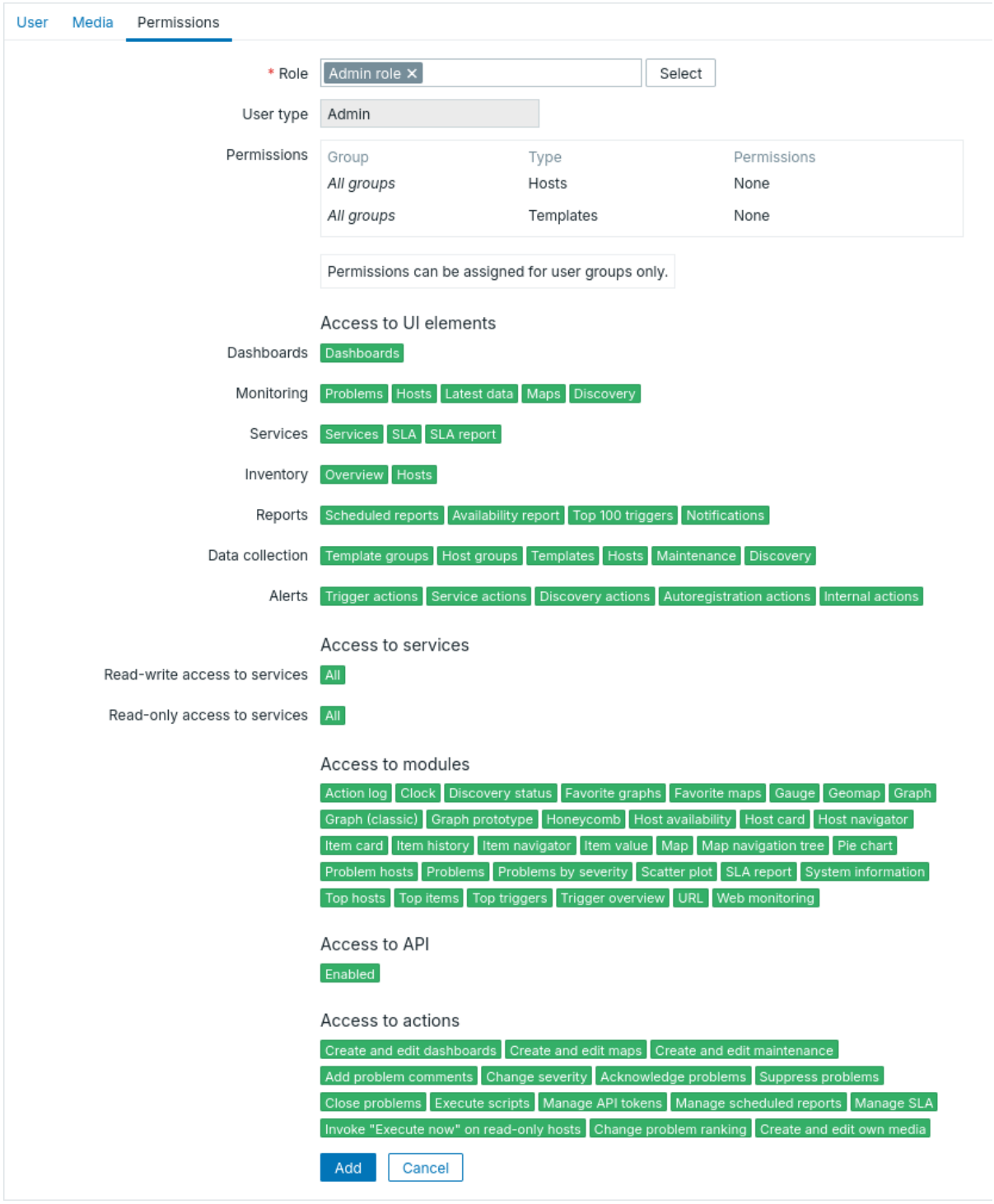
Geben Sie im Pop-up die E-Mail-Adresse des Benutzers ein.

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



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

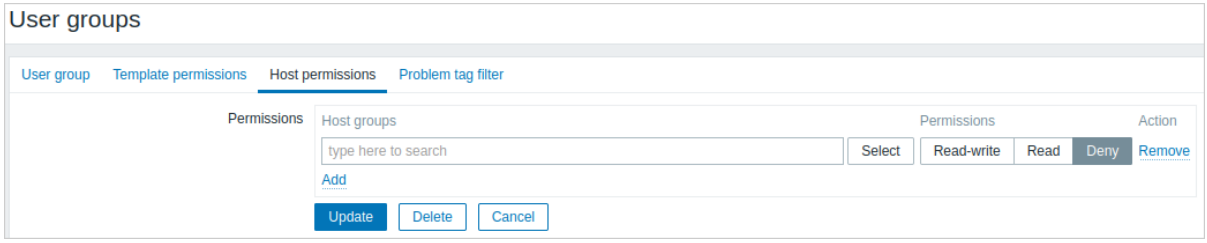
Die Registerkarte Berechtigungen enthält das Pflichtfeld *Rolle*. Die Rolle bestimmt, welche Frontend-Elemente der Benutzer sehen kann und welche Aktionen er ausführen darf. Klicken Sie auf Auswählen und wählen Sie eine der Rollen aus der Liste aus. Wählen Sie beispielsweise *Admin-Rolle*, um Zugriff auf alle Bereiche des Zabbix-Frontend zu gewähren, mit Ausnahme von Administration. Später können Sie Berechtigungen ändern oder weitere Benutzerrollen erstellen. Nach Auswahl einer Rolle werden die Berechtigungen auf derselben Registerkarte angezeigt:



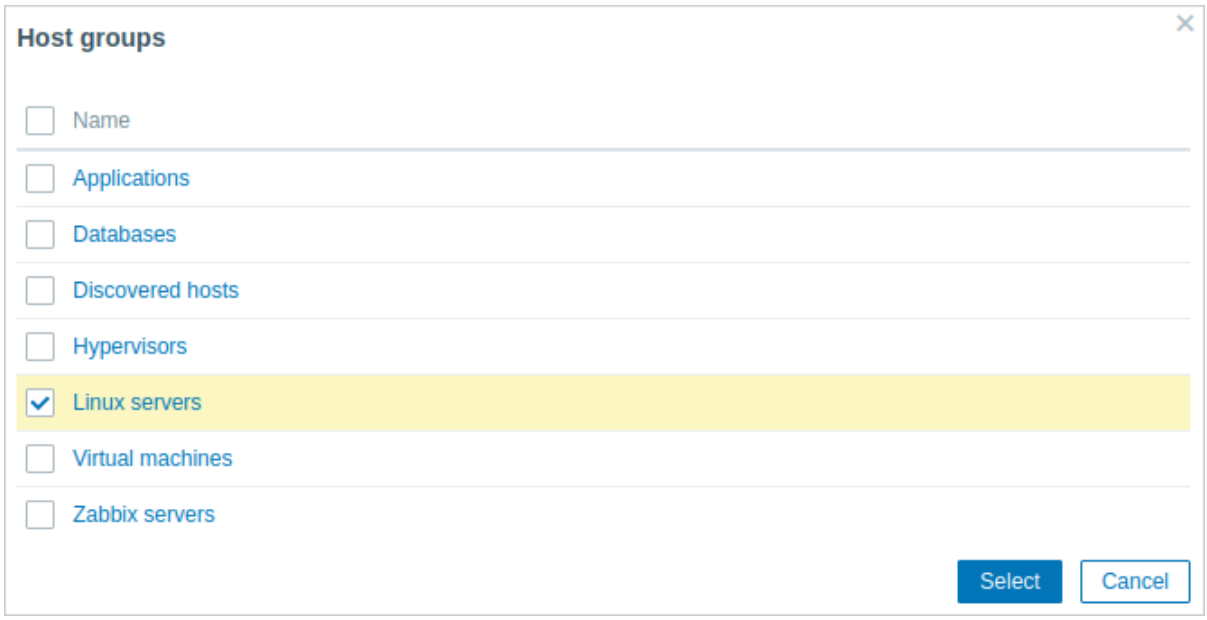
Klicken Sie im Formular mit den Benutzereigenschaften auf *Hinzufügen*, um den Benutzer zu speichern. Der neue Benutzer erscheint in der Benutzerliste.

Berechtigungen hinzufügen

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



Klicken Sie dann neben dem Feld auf *Select*, um die Liste der Host-Gruppen anzuzeigen. Dieser Benutzer soll schreibgeschützten Zugriff auf die Gruppe *Linux servers* erhalten. Aktivieren Sie daher das entsprechende Kontrollkästchen in der Liste und klicken Sie auf *Select*, um Ihre Auswahl zu bestätigen.



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 zu erteilen, müssen Sie zur Registerkarte *Template permissions* wechseln und Vorlagengruppen angeben. Die Schritte sind identisch mit der Zuweisung von Berechtigungen für Host-Gruppen. Eine Übersicht über Vorlagen finden Sie im Abschnitt New template dieser Schnellstartanleitung.

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

Fertig! Sie können nun versuchen, sich mit den Zugangsdaten des neuen Benutzers anzumelden.

2 Neuer Host

Übersicht

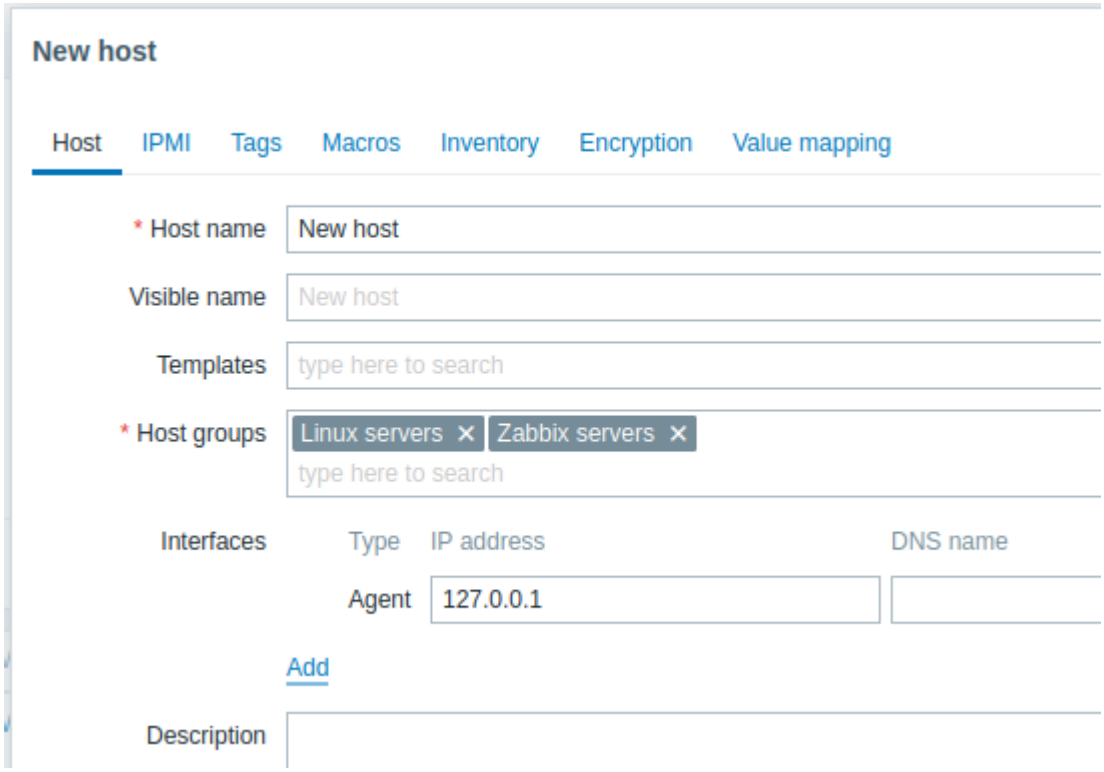
In diesem Abschnitt erfahren Sie, wie Sie einen neuen Host einrichten.

Ein Host in Zabbix ist eine vernetzte Entität (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 Netzwerk-Switch, eine virtuelle Maschine oder eine Anwendung handeln.

Host hinzufügen

Informationen über konfigurierte Hosts in Zabbix sind sowohl unter *Datensammlung* > *Hosts* als auch in den Menüabschnitten *Monitoring* > *Hosts* verfügbar. Es gibt bereits einen vordefinierten Host namens „Zabbix server“, aber wir möchten das Hinzufügen eines weiteren Hosts kennenlernen.

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



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Das absolute Minimum, das hier eingegeben werden muss, ist:

Host-Name

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

Host-Gruppen

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

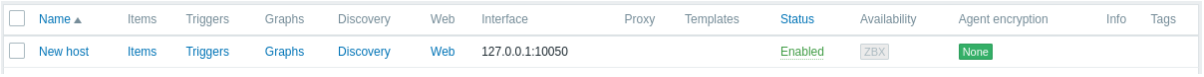
Note:
Alle Zugriffsberechtigungen werden Host-Gruppen zugewiesen, nicht einzelnen Hosts. Deshalb muss ein Host zu mindestens einer Gruppe gehören.

Schnittstellen: IP-Adresse

- Obwohl dies technisch gesehen kein Pflichtfeld ist, ist eine Host-Schnittstelle für die Erfassung bestimmter Metriken erforderlich. Um passive Prüfungen des Zabbix Agent zu verwenden, geben Sie in diesem Feld die IP-Adresse oder den DNS-Namen des Agent an. Beachten Sie, dass Sie auch die IP-Adresse oder den DNS-Namen des Zabbix Server in der Direktive 'Server' der Zabbix-Agent-Konfigurationsdatei angeben sollten. Wenn Zabbix Agent und Zabbix Server auf derselben Maschine installiert sind, müssen Sie an beiden Stellen dieselbe IP/DNS angeben.

Andere Optionen können unverändert mit ihren Standardwerten belassen werden.

Wenn Sie fertig sind, klicken Sie auf *Hinzufügen*. Ihr neuer Host sollte in der Host-Liste sichtbar sein.



Die Spalte „Verfügbarkeit" enthält Anzeigen für die Verfügbarkeit des Hosts pro Schnittstelle. Wir haben eine Zabbix-Agent-Schnittstelle definiert, daher können wir das Symbol für die Agent-Verfügbarkeit (mit „ZBX" darauf) verwenden, um die Verfügbarkeit des Hosts zu erkennen:

- ZBX - Der Host-Status wurde noch nicht festgestellt; es hat noch keine Metrikprüfung stattgefunden.
- 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 eventuell durch falsche

Zugangsdaten für die Schnittstelle verursacht wurde. Prüfen Sie, ob der Zabbix Server läuft, und versuchen Sie später auch, die Seite zu aktualisieren.

3 Neuer Datenpunkt

Übersicht

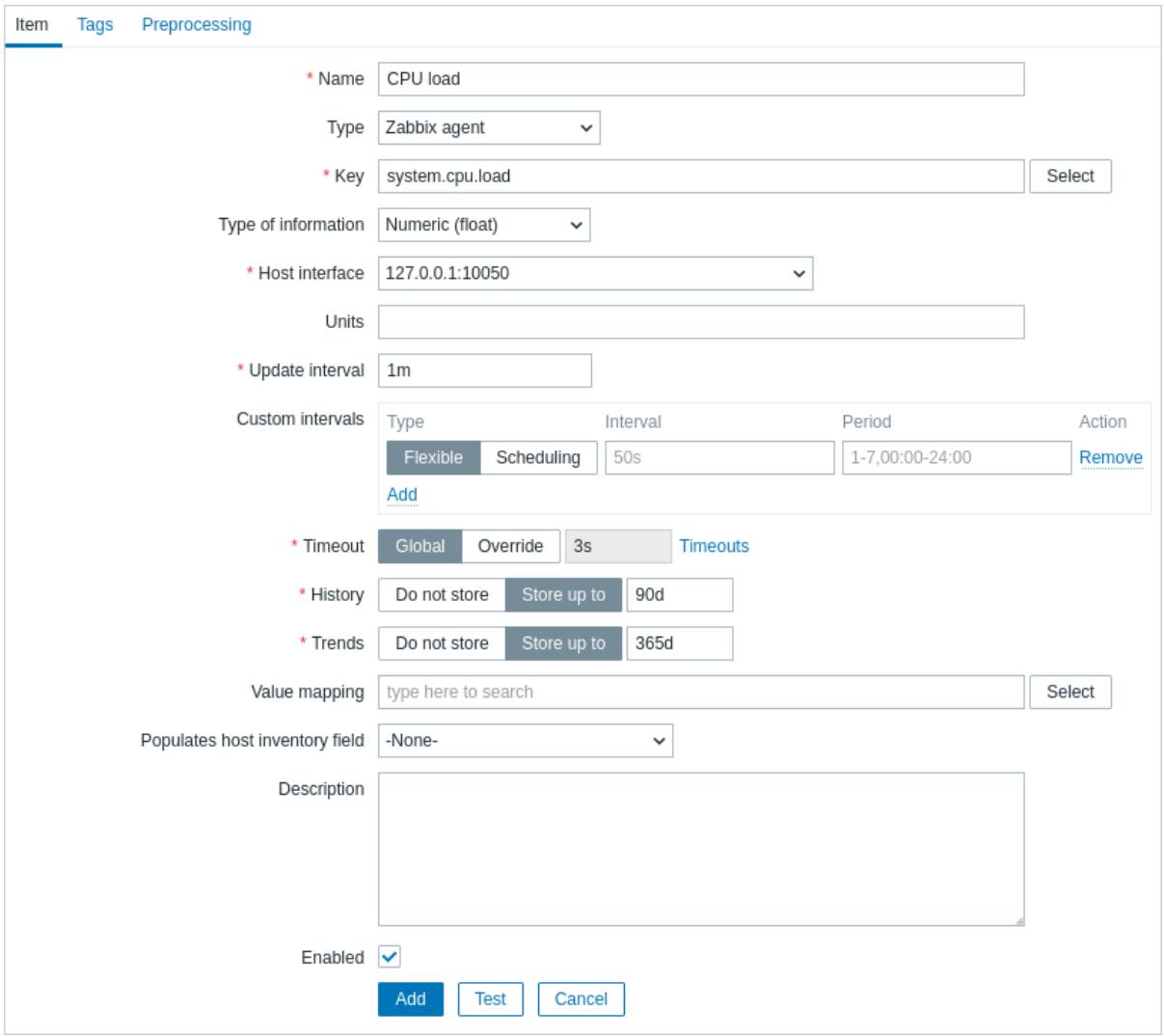
In diesem Abschnitt erfahren Sie, wie Sie einen Datenpunkt einrichten.

Datenpunkte bilden die Grundlage der Datenerfassung in Zabbix. Ohne Datenpunkte gibt es keine Daten – denn nur ein Datenpunkt definiert eine einzelne Metrik bzw. welche Art von Daten von einem Host erfasst werden soll.

Datenpunkt hinzufügen

Alle Datenpunkte sind um Hosts gruppiert. Deshalb gehen wir zum Konfigurieren eines Beispiel-Datenpunkts zu *Datensammlung* > *Hosts* und suchen den zuvor erstellten „New host“.

Klicken Sie in der Zeile von „New host“ auf den Link *Datenpunkte* und dann auf *Datenpunkt erstellen*. Daraufhin wird ein Formular zur Datenpunkt-Konfiguration geöffnet.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Für unseren Beispiel-Datenpunkt müssen im Wesentlichen folgende Informationen eingegeben werden:

Name

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

Schlüssel

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

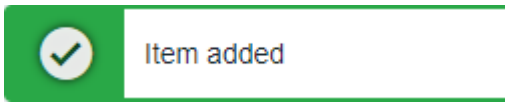
Informationstyp

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

Note:
Möglicherweise möchten Sie auch die Anzahl der Tage reduzieren, für die die Datenpunkt-Historie aufbewahrt wird, auf 7 oder 14. Dies ist eine bewährte Vorgehensweise, um die Datenbank zu entlasten, damit sie nicht so viele historische Werte speichern muss.

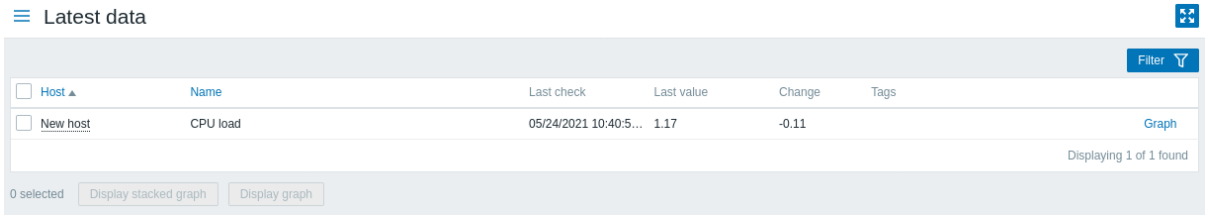
Andere Optionen können unverändert mit ihren Standardwerten belassen werden.

Klicken Sie anschließend auf *Hinzufügen*. Der neue Datenpunkt sollte in der Datenpunktliste erscheinen, und Sie sollten eine Erfolgsmeldung sehen.



Daten anzeigen

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



Beachten Sie, dass es bis zu 60 Sekunden dauern kann, bis die ersten Daten eintreffen. Standardmäßig liest der Server in diesem Intervall Konfigurationsänderungen ein und übernimmt neue Datenpunkte zur Ausführung.

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

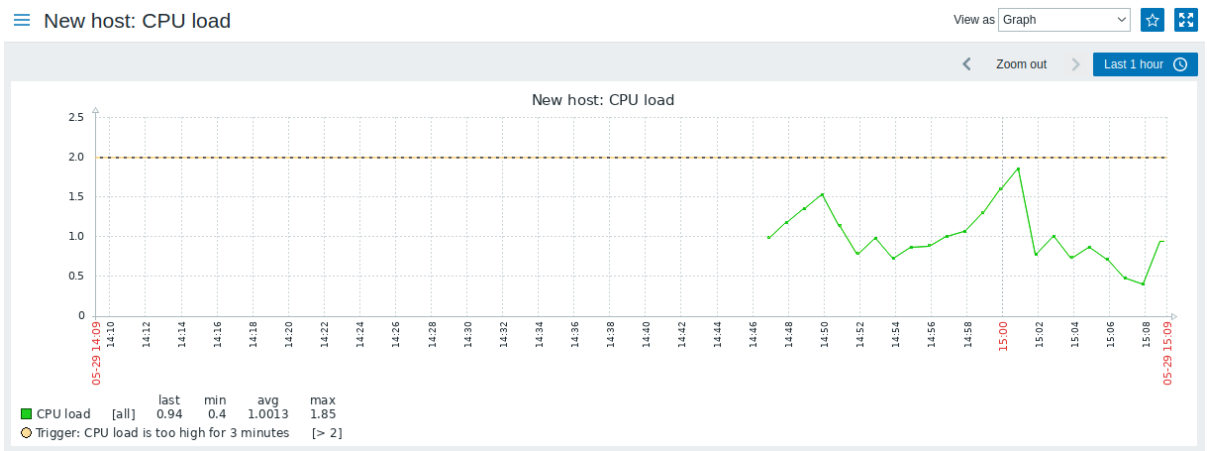
Wenn Sie keine Informationen über den Datenpunkt wie im Screenshot sehen, stellen Sie sicher, dass:

- 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 auf 'Monitored' gesetzt ist und sein Verfügbarkeitssymbol grün ist;
- der im Host-Filter ausgewählte Host korrekt ist;
- der Datenpunkt aktiviert ist.

Diagramme

Wenn der Datenpunkt eine Weile funktioniert hat, ist es möglicherweise an der Zeit, sich etwas Visuelles anzusehen. Einfache Diagramme sind für jeden überwachten numerischen Datenpunkt ohne zusätzliche Konfiguration verfügbar. Diese Diagramme werden zur Laufzeit erzeugt.

Um das Diagramm anzuzeigen, gehen Sie zu *Überwachung* > *Letzte Daten* und klicken Sie auf den Link „Graph" neben dem Datenpunkt.



4 Neuer Auslöser

Übersicht

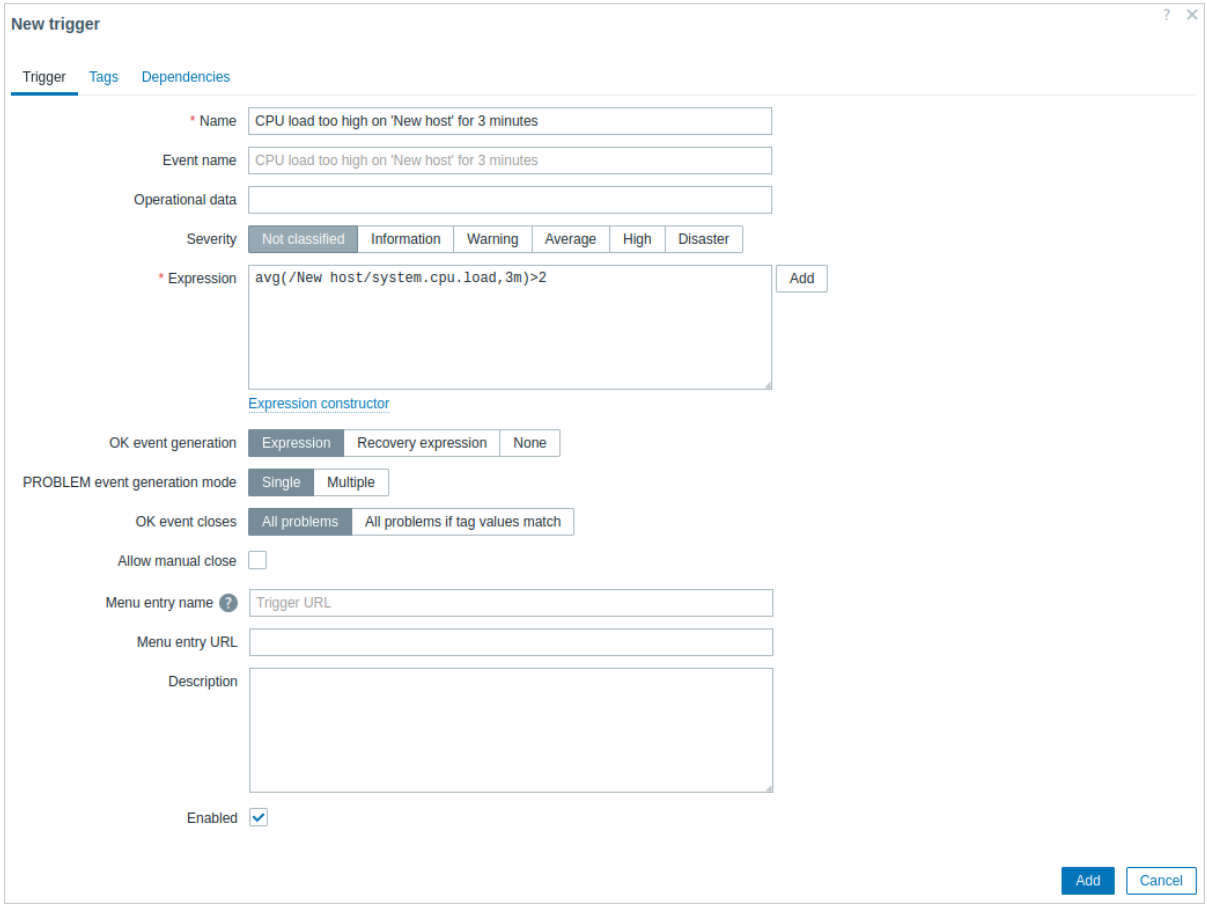
In diesem Abschnitt erfahren Sie, wie Sie einen Auslöser einrichten.

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

Wird dieser Wert durch die eingehenden Daten überschritten, wird ein Auslöser „ausgelöst“ oder wechselt in den Status „Problem“ – was darauf hinweist, dass etwas geschehen ist, das möglicherweise Aufmerksamkeit erfordert. Ist der Wert wieder akzeptabel, kehrt der Auslöser in den Status „Ok“ zurück.

Hinzufügen eines Auslösers

Um einen Auslöser für unseren Datenpunkt zu konfigurieren, gehen Sie zu *Datenerfassung* > *Hosts*. Klicken Sie auf den Link *Auslöser* in der Zeile von ’New host’ und dann auf *Auslöser erstellen*. Ein Formular zur Auslöserkonfiguration wird geöffnet.



Für unseren Auslöser müssen hier folgende wesentliche Informationen eingegeben werden:

Name

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

Expression

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

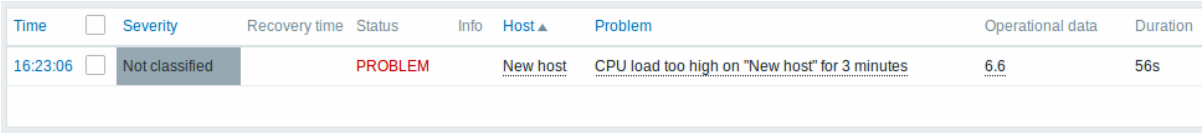
Dies ist der Auslöserausdruck. Stellen Sie sicher, dass der Ausdruck korrekt eingegeben wird, bis hin zum letzten Symbol. Der Datenpunktschlüssel hier (system.cpu.load) wird verwendet, um auf den Datenpunkt zu verweisen. Dieser spezielle Ausdruck besagt im Wesentlichen, dass der Problemschwellenwert überschritten wird, wenn der durchschnittliche CPU-Lastwert über 3 Minuten größer als 2 ist. Weitere Informationen finden Sie unter Syntax von Auslöserausdrücken.

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 sehen.

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



Time	Severity	Recovery time	Status	Info	Host	Problem	Operational data	Duration
16:23:06	Not classified		PROBLEM		New host	CPU load too high on "New host" for 3 minutes	6.6	56s

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

5 Empfangen von Problembenachrichtigungen

Übersicht

In diesem Abschnitt erfahren Sie, wie Sie die Alarmierung in Form von Benachrichtigungen in Zabbix einrichten.

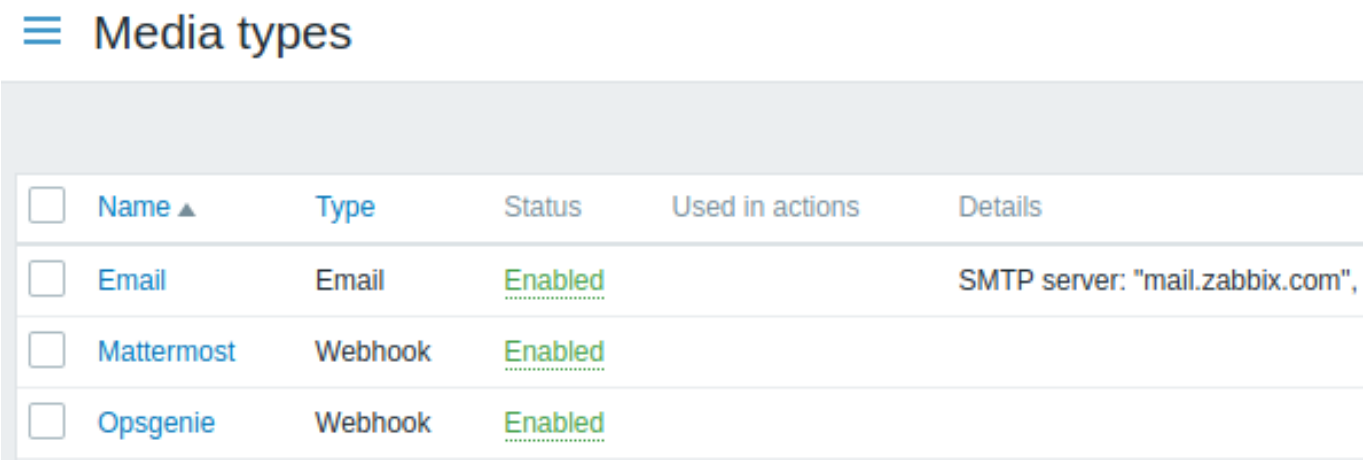
Da Datenpunkte Daten erfassen und Auslöser so konzipiert sind, dass sie in Problemsituationen „auslösen“, ist es außerdem sinnvoll, einen Alarmierungsmechanismus einzurichten, der über wichtige Ereignisse benachrichtigt, auch wenn das Zabbix Frontend nicht aktiv überwacht wird.

Genau das leisten Benachrichtigungen. Da E-Mail die beliebteste Zustellmethode für Problembenachrichtigungen ist, lernen wir, wie Sie eine E-Mail-Benachrichtigung einrichten.

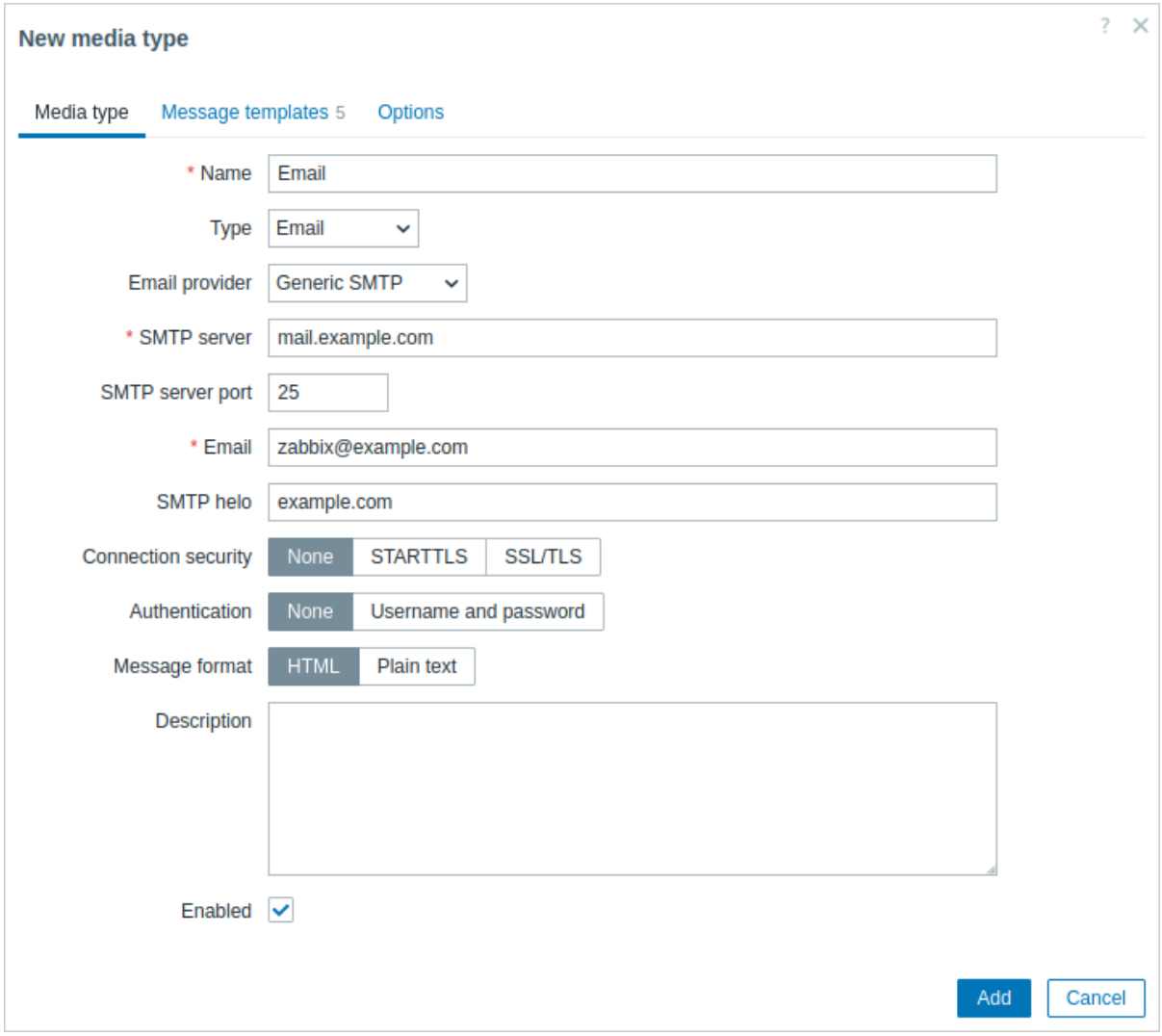
E-Mail-Einstellungen

In Zabbix gibt es zunächst mehrere vordefinierte Benachrichtigungs-Übertragungsmethoden. E-Mail ist eine davon.

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



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



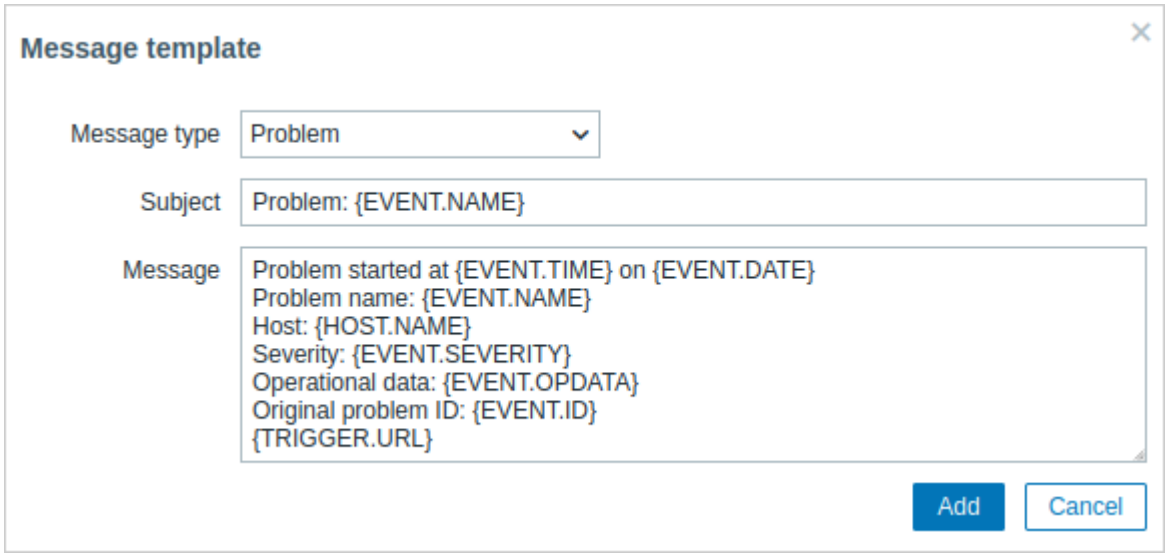
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Legen Sie auf der Registerkarte *Medientyp* die Werte für *SMTP-Server*, *SMTP-Helo* und *E-Mail* entsprechend Ihrer Umgebung fest.

Note:
Der Wert im Feld *E-Mail* wird als Absenderadresse ('From') für die von Zabbix gesendeten Benachrichtigungen verwendet.

Als Nächstes muss der Inhalt der Problemmeldung definiert werden. Der Inhalt wird mithilfe einer Nachrichtenvorlage festgelegt, die auf der Registerkarte *Nachrichtenvorlagen* konfiguriert wird.

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

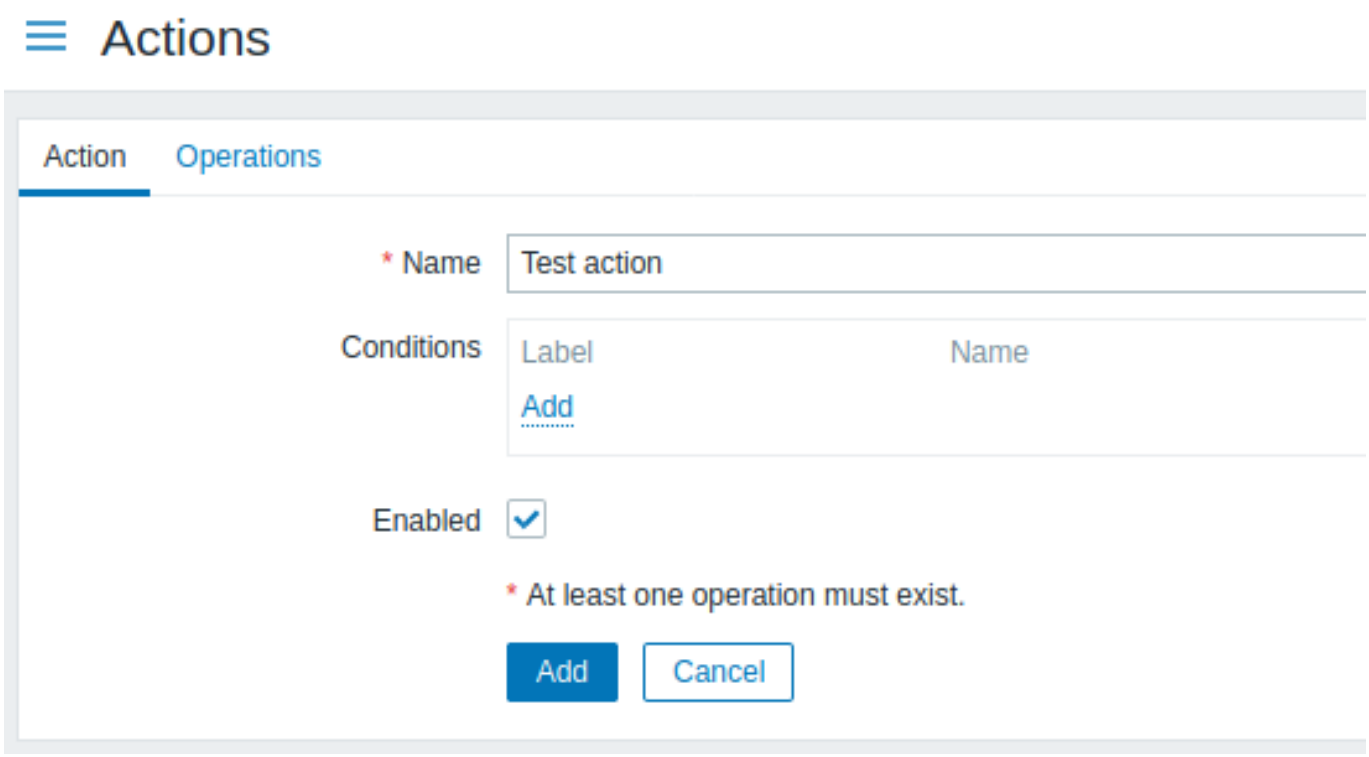


Klicken Sie auf *Hinzufügen*, wenn Sie fertig sind, und speichern Sie das Formular.

Nun haben Sie *E-Mail* als funktionierenden Medientyp konfiguriert. Der Medientyp muss außerdem mit Benutzern verknüpft werden, indem spezifische Zustelladressen definiert werden (wie bei der Konfiguration eines neuen Benutzers), andernfalls wird er nicht verwendet.

Neue Aktion

Das Zustellen von Benachrichtigungen ist eine der Aufgaben, die Aktionen in Zabbix ausführen. Um daher eine Benachrichtigung einzurichten, gehen Sie zu *Warnungen* > *Aktionen* > *Auslöser-Aktionen* und klicken Sie auf *Aktion erstellen*.

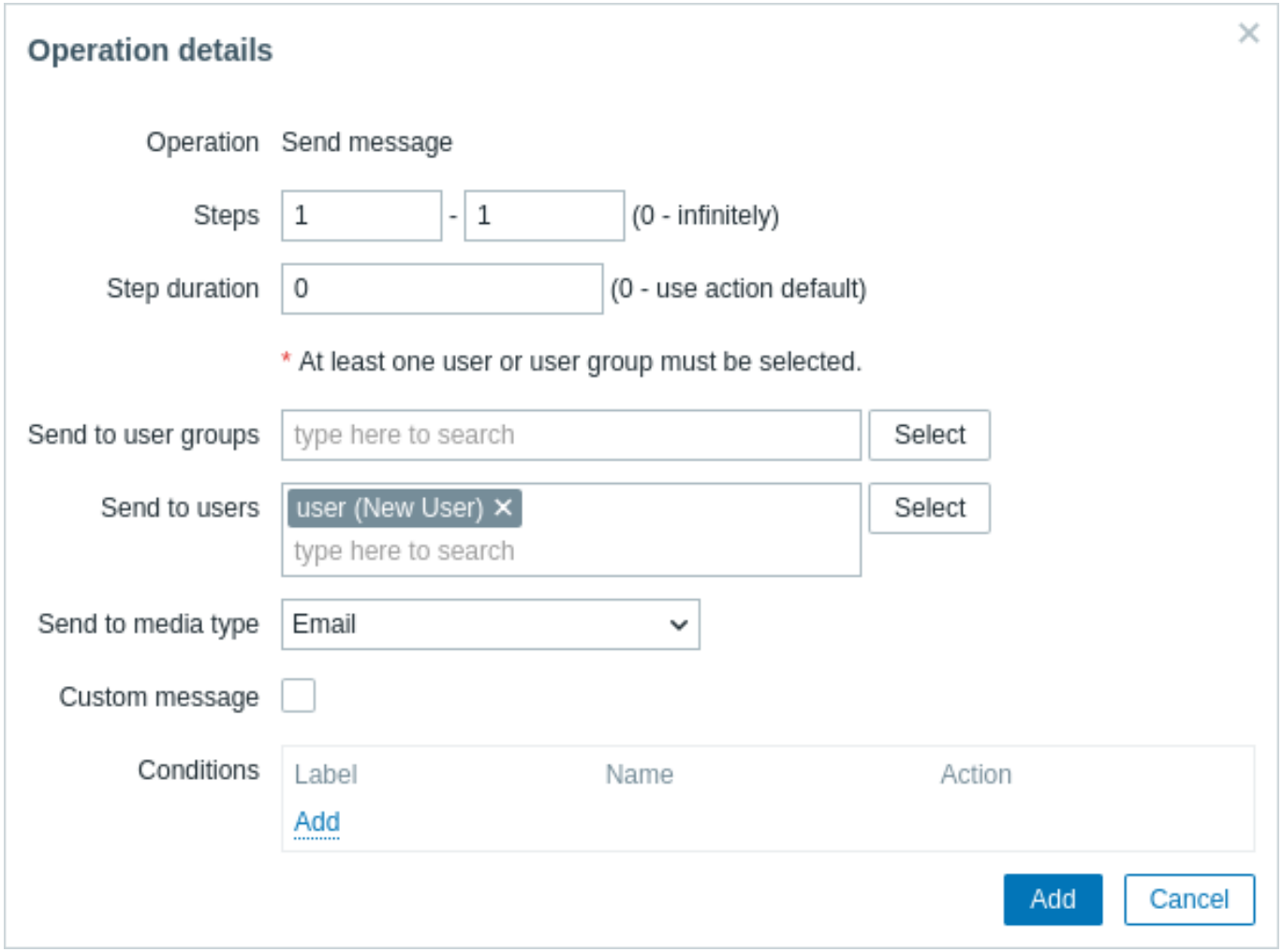


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

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

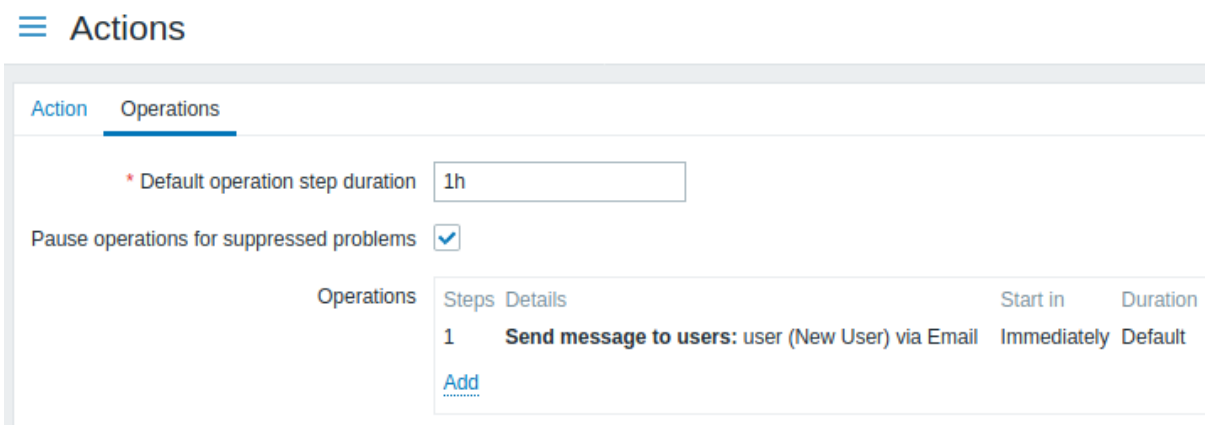
Im einfachsten Fall wird die Aktion bei jeder Änderung eines Auslösers von „Ok“ zu „Problem“ ausgeführt, wenn wir keine weiteren spezifischen Bedingungen hinzufügen.

Wir sollten noch festlegen, was die Aktion tun soll – das wird auf der Registerkarte *Operationen* definiert. Klicken Sie im Block *Operationen* auf *Hinzufügen*; dadurch wird ein neues Operationsformular geöffnet.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Klicken Sie hier im Block *An Benutzer senden* auf *Auswählen* und wählen Sie den von uns definierten Benutzer („user“) aus. Wählen Sie „Email“ als Wert für *An Medientyp senden*. Wenn Sie damit fertig sind, klicken Sie auf *Hinzufügen*; die Operation sollte dann hinzugefügt werden:



Damit ist die einfache Konfiguration einer Aktion abgeschlossen; klicken Sie nun im Aktionsformular auf *Hinzufügen*.

Benachrichtigung empfangen

Nachdem das Versenden von Benachrichtigungen nun konfiguriert ist, wäre es doch schön, auch tatsächlich eine zu erhalten. Um dabei zu helfen, können wir die Last auf unserem Host absichtlich 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 *Überwachung* > *Letzte Daten* und sehen Sie nach, wie die Werte von „CPU Load“ gestiegen sind. Denken Sie daran: Damit unser Auslöser „auslöst“, muss der Wert „CPU Load“ 3 Minuten lang über „2“ liegen. Sobald dies der Fall ist:

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

Attention:
Falls Benachrichtigungen nicht funktionieren:

- prüfen Sie noch einmal, dass 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 muss als Teil der Benutzergruppe „Zabbix administrators“ mindestens Lesezugriff auf die Host-Gruppe „Linux servers“ haben, zu der unser Host gehört.
- Zusätzlich können Sie das Aktionsprotokoll unter *Berichte* > *Aktionsprotokoll* prüfen.

6 Neue Vorlage

Überblick

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.

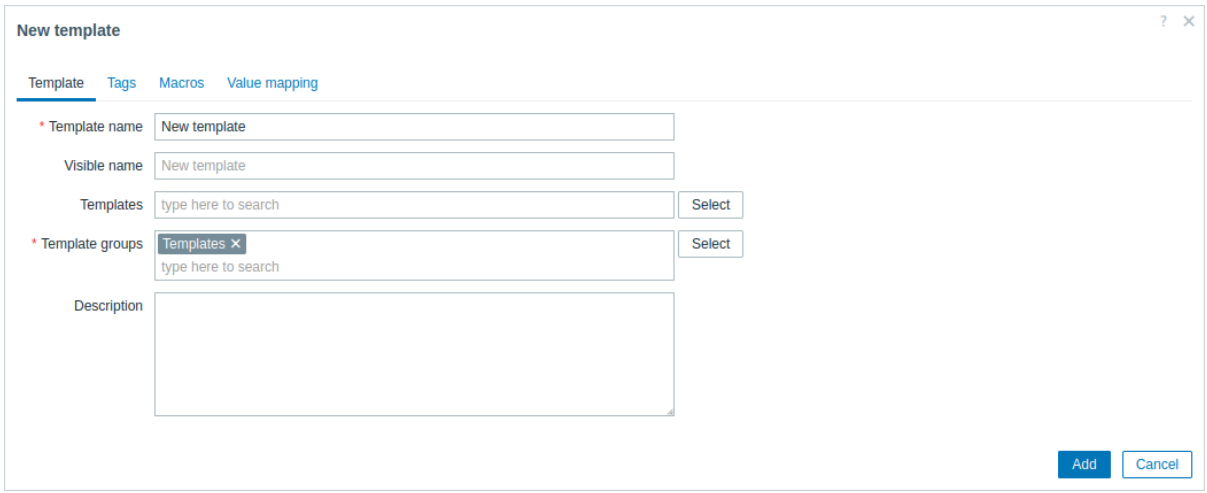
Während all diese Schritte für sich genommen ein hohes Maß an Flexibilität bieten, kann es wie eine große Anzahl von Schritten erscheinen, wenn dies beispielsweise für tausend Hosts erforderlich ist. Etwas Automatisierung wäre hier hilfreich.

Hier kommen Vorlagen ins Spiel. Vorlagen ermöglichen es, nützliche Datenpunkte, Auslöser und andere Entitäten zu gruppieren, sodass diese durch Anwenden auf Hosts in einem einzigen Schritt immer wieder wiederverwendet werden können.

Wenn eine Vorlage mit einem Host verknüpft wird, erbt der Host alle Entitäten der Vorlage. Im Grunde kann also ein vorgefertigtes Bündel von Prüfungen sehr schnell angewendet werden.

Vorlage hinzufügen

Um mit Vorlagen zu arbeiten, müssen wir zunächst eine erstellen. Gehen Sie dazu in *Datenerfassung* > *Vorlagen* und klicken Sie auf *Vorlage erstellen*. Dadurch wird ein Formular zur Vorlagenkonfiguration geöffnet.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Die hier einzugebenden erforderlichen Parameter sind:

Vorlagenname

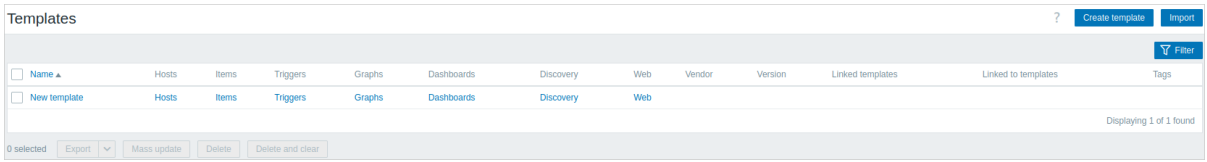
- Geben Sie einen Vorlagennamen ein. Alphanumerische Zeichen, Leerzeichen und Unterstriche sind zulässig.

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 in mindestens eine Gruppe aufzunehmen.

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.



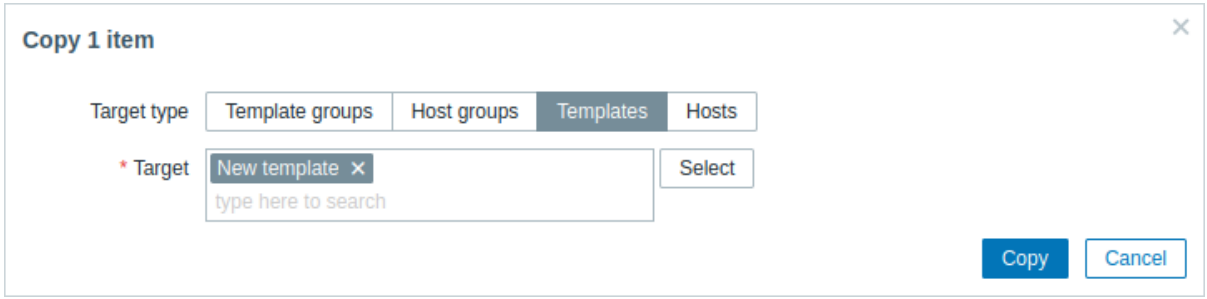
Wie Sie sehen können, ist die Vorlage vorhanden, enthält jedoch noch nichts – keine Datenpunkte, Auslöser oder andere Entitäten.

Datenpunkt zur Vorlage hinzufügen

Um einen Datenpunkt zur Vorlage hinzuzufügen, öffnen Sie die Datenpunktliste für 'New host', indem Sie zu *Datensammlung* → *Hosts* navigieren und neben 'New host' auf *Datenpunkte* klicken.

Dann:

- Aktivieren Sie das Kontrollkästchen des Datenpunkts 'CPU Load' in der Liste.
- Klicken Sie unterhalb der Liste auf *Kopieren*.
- Wählen Sie die Registerkarte *Vorlagen* aus.
- Wählen Sie die Vorlage aus, in die der Datenpunkt kopiert werden soll.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

- Klicken Sie auf *Kopieren*.

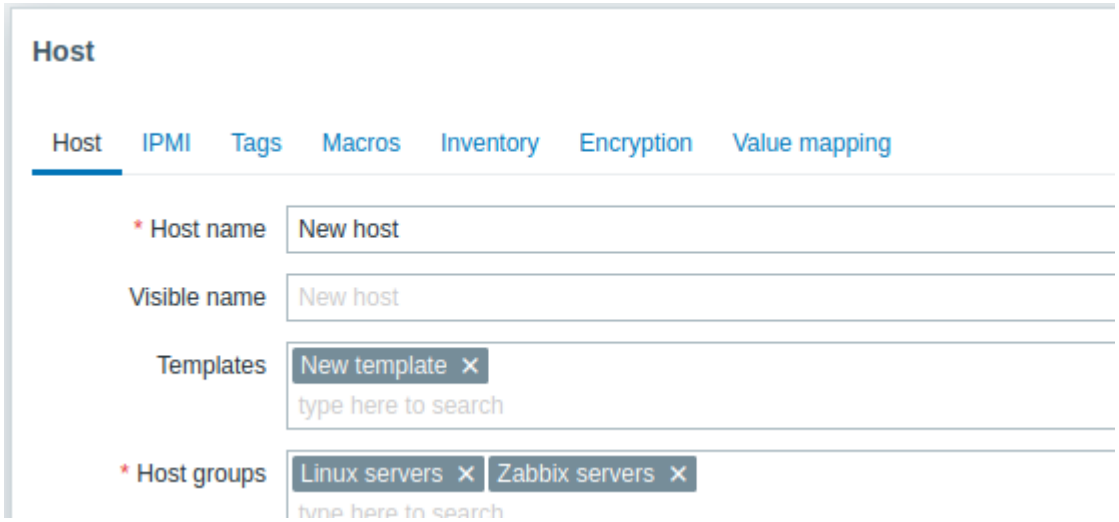
Wenn Sie nun zu *Datensammlung* > *Vorlagen* gehen, sollte 'New template' einen neuen Datenpunkt enthalten.

Wir belassen es vorerst bei nur einem Datenpunkt, aber auf ähnliche Weise können Sie auch beliebige andere Datenpunkte, Auslöser oder andere Entitäten zur Vorlage hinzufügen, bis sie einen recht vollständigen Satz von Entitäten für den jeweiligen Zweck enthält (Überwachung des Betriebssystems, Überwachung einer einzelnen Anwendung).

Vorlage mit Host verknüpfen

Sobald eine Vorlage bereit ist, muss sie nur noch zu einem Host hinzugefügt werden. Gehen Sie dazu zu *Datenerfassung* > *Hosts*, klicken Sie auf „Neuer Host", um das Konfigurationsformular zu öffnen, und suchen Sie das Feld **Vorlagen**.

Beginnen Sie im Feld *Vorlagen* mit der Eingabe von *New template*. Der Name der von uns erstellten Vorlage sollte in der Dropdown-Liste erscheinen. Scrollen Sie nach unten, um sie auszuwählen. Vergewissern Sie sich, dass sie im Feld *Vorlagen* angezeigt wird.



Klicken Sie im Formular auf *Aktualisieren*, um die Änderungen zu speichern. Die Vorlage ist nun dem Host hinzugefügt, zusammen mit allen Entitäten, die sie enthält.

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

Vordefinierte Vorlagen mit Hosts verknüpfen

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. Beachten Sie jedoch, dass diese Vorlagen für Ihre Umgebung feinabgestimmt werden müssen. Einige Prüfungen werden möglicherweise nicht benötigt, und die Abfrageintervalle können viel zu häufig sein.

Weitere Informationen zu Vorlagen sind verfügbar.

Apache-Webserver-Überwachung

Um mit der Überwachung des Apache-Webservers zu beginnen, folgen Sie diesem Ablauf:

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

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

Zugriff auf Apache-Statistiken

Das Ziel dieses Schritts ist es, sicherzustellen, dass Apache-Webserver-Statistiken 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`), die Leistungs- und Aktivitätsdaten anzeigt.

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

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

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

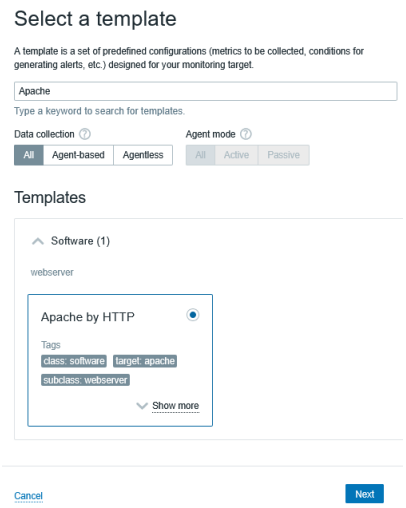
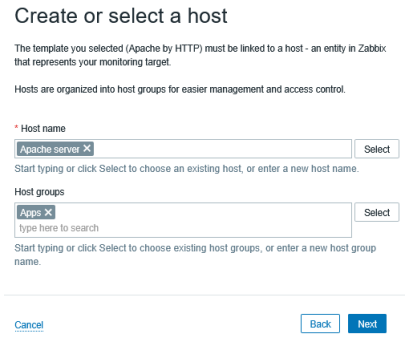
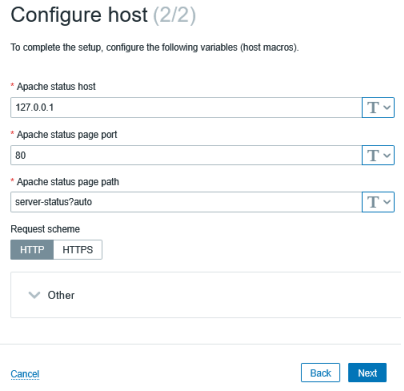
Apache-Monitoring-Vorlage anwenden

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

Um die Vorlage anzuwenden, starten Sie den Host Wizard (unter *Datenerfassung* > *Hosts*):

- **Wählen Sie die Vorlage aus** („Apache by HTTP“)

- Erstellen Sie einen Apache-Host und fügen Sie eine Gruppe dafür hinzu. Dieser virtuelle Host enthält Apache-bezogene Daten.
- Konfigurieren Sie den Host. Geben Sie dabei insbesondere die Adresse des Apache-Webservers ein (127.0.0.1 auf einem lokalen Rechner, IP-Adresse auf einem entfernten Rechner).

Step 1	Step 2	Step 3
		
Vorlage auswählen.	Host/Host-Gruppe erstellen.	Host konfigurieren.

Sie können die anderen Einstellungen auf den Standardwerten belassen.

Um den Host Wizard abzuschließen, klicken Sie auf *Create* und *Finish*.

Neueste Daten ansehen

Host	Name ▲	Last check	Last value	Change	Tags
Apache server	Bytes per second	21s	2.37 KBps		component: network
Apache server	Get status	21s	{"Date":"Mon, 24 Nov …		component: raw
Apache server	Requests per second	21s	0.5501		component: network
Apache server	Service ping	1m 19s	Up (1)		component: application
Apache server	Service response time	20s	0.11ms	-0.026ms	component: application
Apache server	Total bytes	21s	2.5 MB	+142 KB	component: network
Apache server	Total requests	21s	580	+33	component: network
Apache server	Total workers busy	21s	2		component: system
Apache server	Total workers idle	21s	8		component: system
Apache server	Uptime	21s	00:31:21	+00:01:00	component: system
Apache server	Version	1m 21s	Apache/2.4.58 (Ubuntu)		component: system

Herzlichen Glückwunsch, die Überwachung des Apache-Webservers wurde gestartet!

Zabbix bietet Visualisierungsoptionen und sendet Warnmeldungen, wenn Probleme auftreten. Informationen zur Konfiguration von Warnmeldungen per E-Mail finden Sie unter Empfangen von Problembenachrichtigungen.

Überwachung von Linux-Servern

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

1. Wenden Sie die Vorlage für die Linux-Überwachung 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 sowie die IP-Adresse des Zabbix-Servers kennen.

Auf dem Linux-Server wird ein Zabbix Agent zur Überwachung verwendet. Anweisungen zur Installation des Agent werden im Ablauf bereitgestellt.

Um diese Anleitung zu verwenden, müssen Zabbix Server und das Web-Interface installiert sein. Sie müssen als Standardbenutzer *Admin* bei Zabbix angemeldet sein.

Linux-Monitoring-Vorlage anwenden

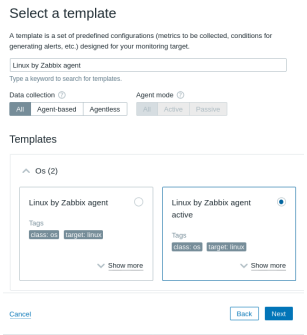
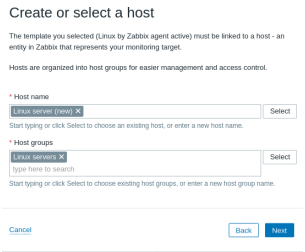
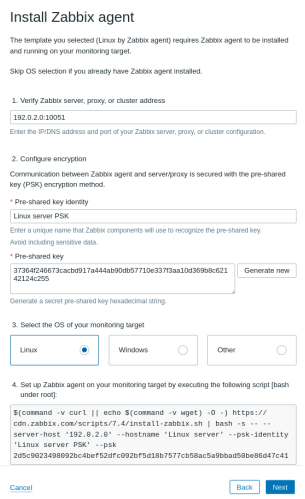
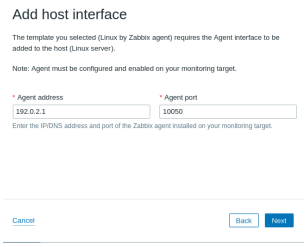
In Zabbix ist diese Vorlage sofort einsatzbereit. Sie enthält vorkonfigurierte Elemente für Datenerfassung, Visualisierung und Analyse.

Um die Vorlage anzuwenden, starten Sie den Host Wizard (unter *Data collection* > *Hosts*):

- **Wählen Sie die Vorlage aus** („Linux by Zabbix agent active“)
- Erstellen Sie einen Host für den Linux-Server und fügen Sie ihn einer Gruppe hinzu. Dieser virtuelle Host enthält alle Daten, die mit der Überwachung des Linux-Servers zusammenhängen.
- 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 zum Zabbix-Server mit Verschlüsselung konfiguriert.

Wenn Sie den Agent bereits installiert haben, können Sie diesen Schritt überspringen (klicken Sie auf *Next*). Sie werden jedoch aufgefordert, die Identität des vorinstallierten Schlüssels und den vorinstallierten Schlüssel anzugeben, um eine Verbindung zum Agent herzustellen.

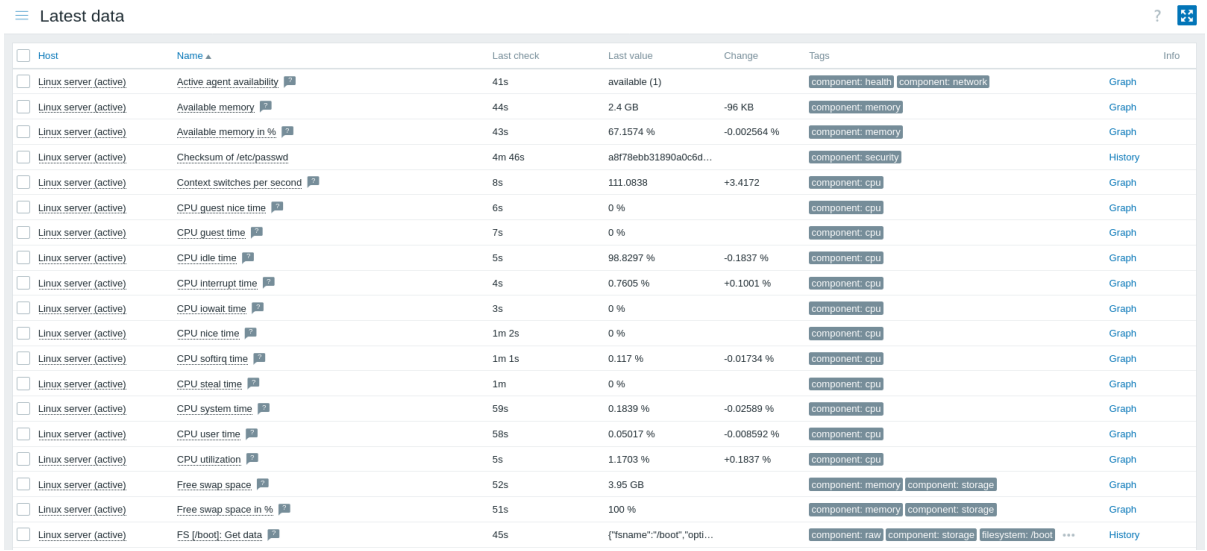
- Schließen Sie die Konfiguration des Host ab. Geben Sie die IP-Adresse und den Port des Agent (Linux-Server) ein.

Schritt 1	Schritt 2	Schritt 3	Schritt 4
			
Vorlage auswählen.	Host/Host-Gruppe erstellen.	Agent auf dem Host konfigurieren und installieren.	IP-Adresse/Port des Linux-Servers als Agent-Schnittstelle hinzufügen.

Sie können die anderen Einstellungen auf den Standardwerten belassen.

Um den Host Wizard abzuschließen, klicken Sie auf *Create* und *Finish*.

Neueste Daten anzeigen



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

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

Überwachung von MySQL-Servern

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

1. Erstellen Sie einen MySQL-Benutzer mit eingeschränkten Rechten.
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.

Zur 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 Zabbix-Server und die Weboberfläche installiert sein. Sie müssen in Zabbix als Standardbenutzer *Admin* angemeldet sein.

Eingeschränkten MySQL-Benutzer erstellen

Es wird empfohlen, einen MySQL-Benutzer zu erstellen, dessen Berechtigungen nur auf die für die Überwachung von MySQL erforderlichen Rechte beschränkt sind.

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

MySQL-Überwachungsvorlage anwenden

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

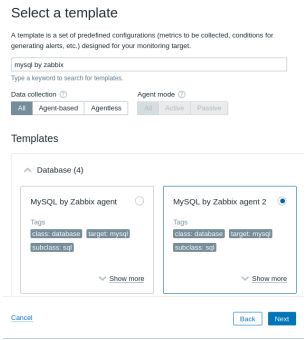
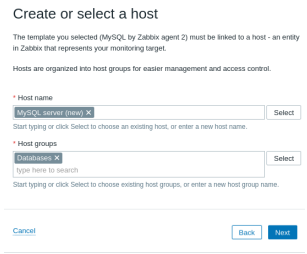
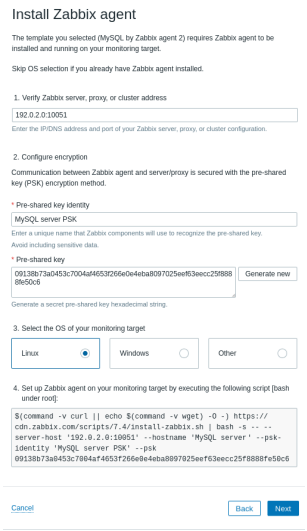
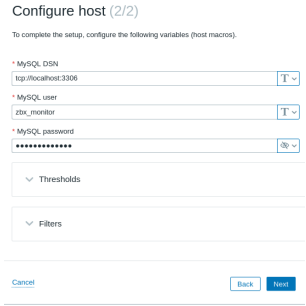
Um die Vorlage anzuwenden, starten Sie den Host Wizard (unter *Datenerfassung* > *Hosts*):

- **Wählen Sie die Vorlage aus** („MySQL by Zabbix agent 2“)
- Erstellen Sie einen MySQL-Server-Host und fügen Sie eine Gruppe dafür hinzu. Dieser virtuelle Host enthält alle Daten zur Überwachung des MySQL-Servers.
- Konfigurieren und installieren Sie Agent 2 (Zabbix agent 2 bietet MySQL-Überwachung sofort einsatzbereit).

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 Agent 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 vorinstallierten Schlüssels und den vorinstallierten Schlüssel anzugeben, um eine Verbindung zum Agent herzustellen.

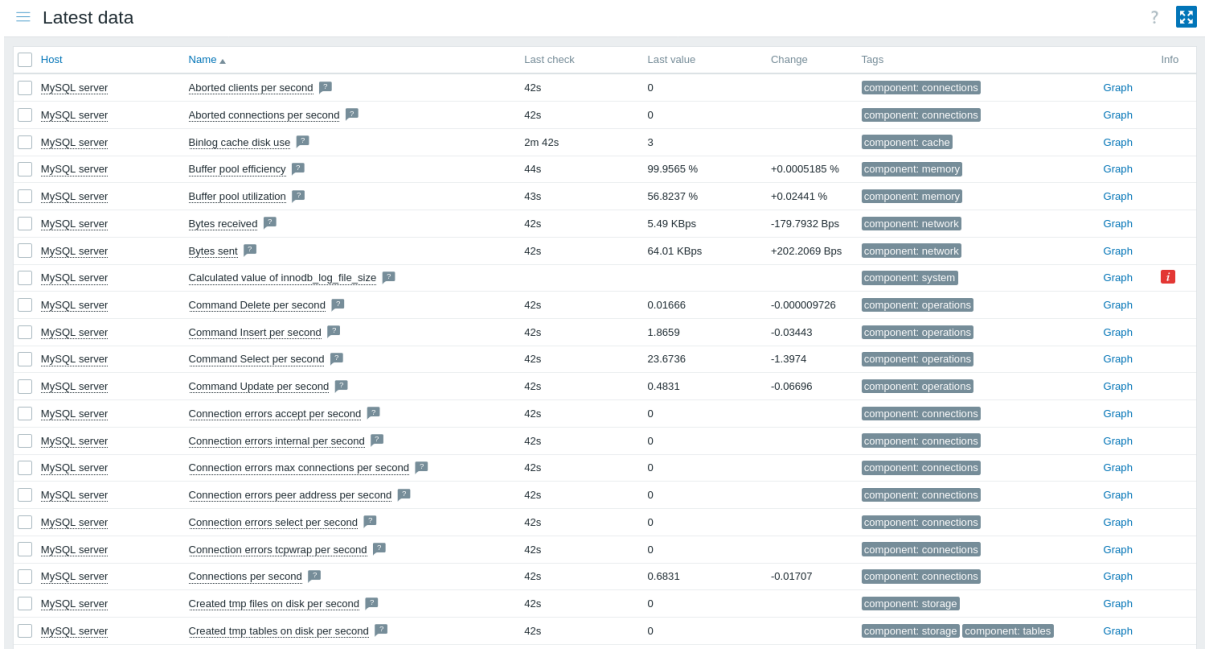
- Schließen Sie die Konfiguration des Hosts ab. Wichtig ist, dass Sie aufgefordert werden, den MySQL-Server-DSN, den MySQL-Benutzer und das Passwort einzugeben. Geben Sie den Benutzer `zbx_monitor` mit dem von Ihnen erstellten Passwort an.

Schritt 1	Schritt 2	Schritt 3	Schritt 4
			
Vorlage auswählen.	Host/Host-Gruppe erstellen.	Agent auf dem Host konfigurieren und installieren.	MySQL-Server-DNS, Benutzer und Passwort hinzufügen.

Sie können die anderen Einstellungen auf den Standardwerten belassen.

Um den Host Wizard abzuschließen, klicken Sie auf *Erstellen* und *Fertigstellen*.

Neueste Daten ansehen



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

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

3 Installation: zusätzliche Anleitungen

Wenn Sie nach der einfachsten Möglichkeit suchen, Zabbix zu installieren, lesen Sie bitte Installation and first steps.

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

- Installation aus Containern
- Installation aus Quellen
- Hinweise zur Installation aus Paketen
- Installation des Zabbix Agent aus Windows MSI
- Erstellen von Zabbix Agent 2 unter Windows
- Erstellen von Zabbix Agent unter Windows
- Installation des macOS-Agent aus PKG
- Erstellen von Zabbix Agent unter macOS
- Zabbix Appliance

Installation aus Containern

Überblick

Dieser Abschnitt beschreibt, wie Zabbix mit Docker oder Docker Compose bereitgestellt wird.

Zabbix stellt offiziell Folgendes bereit:

- Separate Docker-Images für jede Zabbix-Komponente, die als portable und eigenständige Container ausgeführt werden.
- Compose-Dateien zum Definieren und Ausführen von Zabbix-Komponenten mit mehreren Containern in Docker.

Attention:
Seit Zabbix 6.0 müssen deterministische Auslöser während der Installation 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 Database creation scripts.
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.

Quelldateien

Die Dockerfile-Quellen werden im Zabbix-offiziellen Repository auf GitHub gespeichert. Dort können Sie die neuesten Dateiänderungen verfolgen oder das Projekt forken, um Ihre eigenen Images zu erstellen.

Docker

Zabbix stellt Images bereit, die auf einer Vielzahl von Basis-OS-Images basieren. Um die Liste der unterstützten Basis-Betriebssystem-Images für eine bestimmte Zabbix-Komponente zu erhalten, siehe die Beschreibung der Komponente in Docker Hub. Alle Zabbix-Images sind so konfiguriert, dass die neuesten Images neu erstellt werden, wenn Basis-Images aktualisiert werden.

Installation

Um ein Zabbix-Komponenten-Image zu erhalten, führen Sie Folgendes aus:

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

Ersetzen Sie `zabbix/zabbix-server-mysql` durch den Namen des erforderlichen Docker-Repositorys.

Dieser Befehl lädt die neueste stabile Version der Zabbix-Komponente herunter, die auf dem Alpine-Linux-OS basiert. Sie können Tags an den Repository-Namen anhängen, um ein Image zu erhalten, das auf einem anderen Betriebssystem basiert oder einer bestimmten Zabbix-Haupt- oder Nebenversion entspricht.

Die folgenden Repositorys sind in Docker Hub verfügbar:

Komponente	Docker-Repository
Zabbix Agent	zabbix/zabbix-agent
Zabbix Server	
mit MySQL-Unterstützung	zabbix/zabbix-server-mysql
mit PostgreSQL-Unterstützung	zabbix/zabbix-server-pgsql

Komponente	Docker-Repository
Zabbix-Weboberfläche	
basierend auf dem Apache2-Webserver mit MySQL-Unterstützung	zabbix/zabbix-web-apache-mysql
basierend auf dem Apache2-Webserver mit PostgreSQL-Unterstützung	zabbix/zabbix-web-apache-pgsql
basierend auf dem Nginx-Webserver mit MySQL-Unterstützung	zabbix/zabbix-web-nginx-mysql
basierend auf dem Nginx-Webserver mit PostgreSQL-Unterstützung	zabbix/zabbix-web-nginx-pgsql
Zabbix Proxy	
mit SQLite3-Unterstützung	zabbix/zabbix-proxy-sqlite3
mit MySQL-Unterstützung	zabbix/zabbix-proxy-mysql
Zabbix Java gateway	zabbix/zabbix-java-gateway

Note:
Die Unterstützung für SNMP-Traps wird in einem separaten Repository zabbix/zabbix-snmptraps bereitgestellt. Es kann mit Zabbix Server und Zabbix Proxy verknüpft werden.

Tags

Offizielle Zabbix-Komponenten-Images können die folgenden Tags enthalten:

Tag	Beschreibung	Beispiel
latest	Die neueste stabile Version einer Zabbix-Komponente basierend auf einem Alpine-Linux-Image.	zabbix-agent:latest
<OS>-trunk	Der neueste Nightly-Build der Zabbix-Version, die derzeit auf einem bestimmten Betriebssystem entwickelt wird. **<OS>** - das Basisbetriebssystem. Unterstützte Werte: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (nur Agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (nur Agent); *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	Die neueste stabile Version einer Zabbix-Komponente auf einem bestimmten Betriebssystem. **<OS>** - das Basisbetriebssystem. Unterstützte Werte: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (nur Agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (nur Agent); *ubuntu* - Ubuntu	zabbix-agent:ol-latest
<OS>-X.X-latest	Die neueste Nebenversion einer Zabbix-Komponente einer bestimmten Hauptversion und eines bestimmten Betriebssystems. **<OS>** - das Basisbetriebssystem. Unterstützte Werte: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (nur Agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (nur Agent); *ubuntu* - Ubuntu **X.X** - die Zabbix-Hauptversion (d. h. *6.0*, *7.4*, *8.0*).	zabbix-agent:alpine-8.0-latest

Tag	Beschreibung	Beispiel
<OS>-X.X.*	Die spezifische Nebenversion einer Zabbix-Komponente einer bestimmten Hauptversion und eines bestimmten Betriebssystems. **<OS>** - das Basisbetriebssystem. Unterstützte Werte: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (nur Agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (nur Agent); *ubuntu* - Ubuntu **X.X** - die Zabbix-Hauptversion (d. h. *6.0*, *7.4*, *8.0*). * - die Zabbix-Nebenversion	zabbix-agent:alpine-8.0.1

Erstkonfiguration

Nach dem Herunterladen der Images starten Sie die Container, indem Sie den Befehl `docker run` ausführen, gefolgt von zusätzlichen Argumenten zur Angabe der erforderlichen Umgebungsvariablen und/oder Mount-Punkte. Einige Konfigurationsbeispiele werden unten bereitgestellt.

Note:
Um die Kommunikation zwischen Zabbix-Komponenten zu ermöglichen, werden einige Ports für einen Host-Rechner freigegeben, z. B. 10051/TCP für den Zabbix Server (trapper), 10050/TCP für den Zabbix Agent, 162/UDP für SNMP-Traps und 80/TCP für die Zabbix-Weboberfläche. Eine vollständige Liste der von Zabbix-Komponenten standardmäßig verwendeten Ports finden Sie auf der Seite Requirements. Für Zabbix Server und Agent kann der Standardport durch Setzen der Umgebungsvariablen ZBX_LISTENPORT environment variable geändert werden.

Umgebungsvariablen

Alle Images der Zabbix-Komponenten stellen Umgebungsvariablen zur Steuerung der Konfiguration bereit. Unterstützte Umgebungsvariablen sind im Komponenten-Repository aufgeführt.

Diese Umgebungsvariablen sind Optionen aus den Zabbix-Konfigurationsdateien, jedoch mit einer anderen Benennungsmethode. Zum Beispiel entspricht `ZBX_LOGSLOWQUERIES` dem Wert `LogSlowQueries` aus den Konfigurationsdateien von Zabbix Server oder Zabbix Proxy.

Attention:
Einige Konfigurationsoptionen (z. B. `PIDFile` und `LogType`) können nicht geändert werden.

Die folgenden Umgebungsvariablen sind spezifisch für Docker-Komponenten und existieren nicht in den Zabbix-Konfigurationsdateien:

Variable	Komponenten	Standardwert	Beschreibung
`DB_SERVER_HOST`	Server Proxy Weboberfläche	`mysql-server` für MYSQL `postgres-server` für PostgreSQL	IP- oder DNS-Name des MySQL- oder PostgreSQL-Servers.
`DB_SERVER_PORT`	Server Proxy Weboberfläche	3306 für MYSQL 5432 für PostgreSQL	Port des MySQL- oder PostgreSQL-Servers.
`MYSQL_USER`	Server Proxy Weboberfläche	`zabbix`	MySQL-Datenbankbenutzer.
`MYSQL_PASSWORD`	Server Proxy Weboberfläche	`zabbix`	MySQL-Datenbankpasswort.
`MYSQL_DATABASE`	Server Proxy Weboberfläche	`zabbix` für Zabbix-Server `zabbix_proxy` für Zabbix-Proxy	Name der Zabbix-Datenbank.
`POSTGRES_USER`	Server Weboberfläche	`zabbix`	PostgreSQL-Datenbankbenutzer.

Variable	Komponenten	Standardwert	Beschreibung
POSTGRES_PASSWORD	Server Weboberfläche	zabbix	PostgreSQL-Datenbankpasswort.
POSTGRES_DB	Server Weboberfläche	zabbix für Zabbix-Server zabbix_proxy für Zabbix-Proxy	Name der Zabbix-Datenbank.
PHP_TZ	Weboberfläche	Europe/Riga	Zeitzone im PHP-Format. Eine vollständige Liste der unterstützten Zeitzonen ist auf php.net verfügbar.
ZBX_SERVER_NAME	Weboberfläche	Zabbix Docker	Sichtbarer Name der Zabbix-Installation unterhalb des Zabbix-Logos im vertikalen Menü der Weboberfläche.
ZBX_JAVAGATEWAY_ENABLE	Server Proxy	false	Aktiviert die Kommunikation mit dem Zabbix Java gateway, um Java-bezogene Prüfungen zu erfassen.
ZBX_ENABLE_SNMP_TRAPS	Server Proxy	false	Aktiviert die SNMP-Trap-Funktion. Sie erfordert eine **zabbix-snmptraps**-Instanz und das gemeinsam genutzte Volume */var/lib/zabbix/snmptraps* für den Zabbix-Server oder Zabbix-Proxy.

Volumes

Die Images ermöglichen das Einhängen von Volumes unter Verwendung der folgenden Einhängepunkte:

Volume	Beschreibung
Zabbix Agent	
/etc/zabbix/zabbix_agentd.d	Das Volume ermöglicht das Einbinden von **.conf*-Dateien und die Erweiterung des Zabbix Agent mithilfe der UserParameter-Funktion
/var/lib/zabbix/modules	Das Volume ermöglicht das Laden zusätzlicher Module und die Erweiterung des Zabbix Agent mithilfe der Funktion LoadModule
/var/lib/zabbix/enc	Das Volume wird zum Speichern TLS-bezogener Dateien verwendet. Diese Dateinamen werden mithilfe der Umgebungsvariablen ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE und ZBX_TLSPSKFILE angegeben
Zabbix Server	
/usr/lib/zabbix/alertscripts	Das Volume wird für benutzerdefinierte Alarmierungsskripte verwendet. Es ist der Parameter AlertScriptsPath in zabbix_server.conf
/usr/lib/zabbix/externalscripts	Das Volume wird von externen Prüfungen verwendet. Es ist der Parameter ExternalScripts in zabbix_server.conf
/var/lib/zabbix/modules	Das Volume ermöglicht das Laden zusätzlicher Module und die Erweiterung des Zabbix Server mithilfe der Funktion LoadModule
/var/lib/zabbix/enc	Das Volume wird zum Speichern TLS-bezogener Dateien verwendet. Diese Dateinamen werden mithilfe der Umgebungsvariablen ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE und ZBX_TLSPSKFILE angegeben
/var/lib/zabbix/ssl/certs	Das Volume wird als Speicherort für SSL-Clientzertifikatsdateien zur Client-Authentifizierung verwendet. Es ist der Parameter SSLCertLocation in zabbix_server.conf
/var/lib/zabbix/ssl/keys	Das Volume wird als Speicherort für SSL-Dateien mit privaten Schlüsseln zur Client-Authentifizierung verwendet. Es ist der Parameter SSLKeyLocation in zabbix_server.conf
/var/lib/zabbix/ssl/ssl_ca	Das Volume wird als Speicherort für Dateien von Zertifizierungsstellen (CA) zur Verifizierung von SSL-Serverzertifikaten verwendet. Es ist der Parameter SSLCALocation in zabbix_server.conf
/var/lib/zabbix/snmptraps	Das Volume wird als Speicherort für die Datei snmptraps.log verwendet. Es kann vom Container zabbix-snmptraps gemeinsam genutzt und beim Erstellen einer neuen Instanz von Zabbix Server mithilfe der Docker-Option volumes_from übernommen werden. Die Funktion zur Verarbeitung von SNMP-Traps kann durch Verwendung eines gemeinsam genutzten Volumes und Setzen der Umgebungsvariablen ZBX_ENABLE_SNMP_TRAPS auf 'true' aktiviert werden
/var/lib/zabbix/mibs	Das Volume ermöglicht das Hinzufügen neuer MIB-Dateien. Unterverzeichnisse werden nicht unterstützt, alle MIBs müssen in /var/lib/zabbix/mibs abgelegt werden
Zabbix Proxy	
/usr/lib/zabbix/externalscripts	Das Volume wird von externen Prüfungen verwendet. Es ist der Parameter ExternalScripts in zabbix_proxy.conf

Volume	Beschreibung
/var/lib/zabbix/db_data	Das Volume ermöglicht das Speichern von Datenbankdateien auf externen Geräten. Wird nur für Zabbix Proxy mit SQLite3 unterstützt
/var/lib/zabbix/modules	Das Volume ermöglicht das Laden zusätzlicher Module und die Erweiterung des Zabbix Server mithilfe der Funktion LoadModule
/var/lib/zabbix/enc	Das Volume wird zum Speichern TLS-bezogener Dateien verwendet. Diese Dateinamen werden mithilfe der Umgebungsvariablen `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` und `ZBX_TLSPSKFILE` angegeben
/var/lib/zabbix/ssl/certs	Das Volume wird als Speicherort für SSL-Clientzertifikatsdateien zur Client-Authentifizierung verwendet. Es ist der Parameter `SSLCertLocation` in zabbix_proxy.conf
/var/lib/zabbix/ssl/keys	Das Volume wird als Speicherort für SSL-Dateien mit privaten Schlüsseln zur Client-Authentifizierung verwendet. Es ist der Parameter `SSLKeyLocation` in zabbix_proxy.conf
/var/lib/zabbix/ssl/ssl_ca	Das Volume wird als Speicherort für Dateien von Zertifizierungsstellen (CA) zur Verifizierung von SSL-Serverzertifikaten verwendet. Es ist der Parameter `SSLCALocation` in zabbix_proxy.conf
/var/lib/zabbix/snmptraps	Das Volume wird als Speicherort für die Datei snmptraps.log verwendet. Es kann vom Container zabbix-snmptraps gemeinsam genutzt und beim Erstellen einer neuen Instanz von Zabbix Server mithilfe der Docker-Option volumes_from übernommen werden. Die Funktion zur Verarbeitung von SNMP-Traps kann durch Verwendung eines gemeinsam genutzten Volumes und Setzen der Umgebungsvariablen `ZBX_ENABLE_SNMP_TRAPS` auf 'true' aktiviert werden
/var/lib/zabbix/mibs	Das Volume ermöglicht das Hinzufügen neuer MIB-Dateien. Unterverzeichnisse werden nicht unterstützt, alle MIBs müssen in `/var/lib/zabbix/mibs` abgelegt werden
Zabbix-Weboberfläche basierend auf Apache2-Webserver	
/etc/ssl/apache2	Das Volume ermöglicht die Aktivierung von HTTPS für die Zabbix-Weboberfläche. Das Volume muss die beiden für Apache2-SSL-Verbindungen vorbereiteten Dateien `ssl.crt` und `ssl.key` enthalten
Zabbix-Weboberfläche basierend auf Nginx-Webserver	
/etc/ssl/nginx	Das Volume ermöglicht die Aktivierung von HTTPS für die Zabbix-Weboberfläche. Das Volume muss die für Nginx-SSL-Verbindungen vorbereiteten Dateien `ssl.crt`, `ssl.key` und `dhparam.pem` enthalten
Zabbix snmptraps	
/var/lib/zabbix/snmptraps	Das Volume enthält die Protokolldatei `snmptraps.log` mit den empfangenen SNMP-Traps
/var/lib/zabbix/mibs	Das Volume ermöglicht das Hinzufügen neuer MIB-Dateien. Unterverzeichnisse werden nicht unterstützt, alle MIBs müssen in `/var/lib/zabbix/mibs` abgelegt werden

Weitere Informationen finden Sie in den offiziellen Zabbix-Repositories auf Docker Hub.

Beispiele

Beispiel 1

Das Beispiel zeigt, wie Zabbix Server mit Unterstützung für eine MySQL-Datenbank, einer auf dem Nginx-Webserver basierenden Zabbix-Weboberfläche und dem Zabbix Java gateway ausgeführt wird.

1. Erstellen Sie ein Netzwerk, das für Zabbix-Komponenten-Container vorgesehen ist:

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

2. Starten Sie eine leere MySQL-Server-Instanz:

```
docker run --name mysql-server -t \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
```

```
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    --network=zabbix-net \
    --restart unless-stopped \
    -d mysql:8.4-oracle \
    --character-set-server=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

3. Starten Sie eine Zabbix Java gateway-Instanz:

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

4. Starten Sie eine Zabbix-Server-Instanz und verknüpfen Sie diese mit der erstellten MySQL-Server-Instanz:

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

5. Starten Sie die Zabbix-Weboberfläche und verknüpfen Sie diese mit den erstellten MySQL-Server- und Zabbix-Server-Instanzen:

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

Beispiel 2

Das Beispiel zeigt, wie Zabbix Server mit Unterstützung für eine PostgreSQL-Datenbank, einer auf dem Nginx-Webserver basierenden Zabbix-Weboberfläche und der SNMP-Trap-Funktion ausgeführt wird.

1. Erstellen Sie ein Netzwerk, das für Zabbix-Komponenten-Container vorgesehen ist:

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

2. Starten Sie eine leere PostgreSQL-Server-Instanz:

```
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 eine Zabbix-snmptraps-Instanz:

```
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 eine Zabbix-Server-Instanz und verknüpfen Sie diese mit der erstellten PostgreSQL-Server-Instanz:

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

5. Starten Sie die Zabbix-Weboberfläche und verknüpfen Sie diese mit den erstellten PostgreSQL-Server- und Zabbix-Server-Instanzen:

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

Beispiel 3

Das Beispiel zeigt, wie Zabbix Server mit Unterstützung für eine MySQL-Datenbank, einer auf dem Nginx-Webserver basierenden Zabbix-Weboberfläche und dem Zabbix Java gateway unter Verwendung von `podman` auf Red Hat 8 ausgeführt wird.

1. Erstellen Sie einen neuen Pod mit dem Namen `zabbix` und freigegebenen Ports (Weboberfläche, Zabbix-Server-Trapper):

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

2. (optional) Starten Sie den 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-74:latest
```

3. Erstellen Sie auf dem Host das Verzeichnis `./mysql/` und starten Sie Oracle MySQL Server 8.4:

```
podman run --name mysql-server -t \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    -v ./mysql/:/var/lib/mysql/:Z \
    --restart=always \
    --pod=zabbix \
    -d mysql:8.4 \
    --character-set-server=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

4. Starten Sie den Zabbix-Server-Container:

```
podman run --name zabbix-server-mysql -t \
                -e DB_SERVER_HOST="127.0.0.1" \
```

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

5. Starten Sie den Zabbix Java Gateway-Container:

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

6. Starten Sie den Zabbix-Weboberflächen-Container:

```
podman run --name zabbix-web-mysql -t \
                -e ZBX_SERVER_HOST="127.0.0.1" \
                -e DB_SERVER_HOST="127.0.0.1" \
                -e MYSQL_DATABASE="zabbix" \
                -e MYSQL_USER="zabbix" \
                -e MYSQL_PASSWORD="zabbix_pwd" \
                -e MYSQL_ROOT_PASSWORD="root_pwd" \
                --restart=always \
                --pod=zabbix \
                -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-74
```

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

Docker Compose

Alternativ kann Zabbix mit dem Docker-Compose-Plugin installiert werden. Compose-Dateien zum Definieren und Ausführen von Zabbix-Komponenten mit mehreren Containern sind im offiziellen Zabbix-Docker-Repository auf GitHub verfügbar.

Attention:
Offizielle Zabbix-Compose-Dateien unterstützen Version 3 von Docker Compose.

Diese Compose-Dateien werden als Beispiele hinzugefügt; sie sind überladen. Sie enthalten zum Beispiel Proxys mit Unterstützung sowohl für MySQL als auch für SQLite3.

Um die von Zabbix bereitgestellten Docker-Compose-Dateien zu erhalten, klonen Sie das Repository:

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

Wechseln Sie zur erforderlichen Version:

```
git checkout 8.0
```

Konfigurieren Sie die Compose-Dateien und erstellen und starten Sie die Container:

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

Ersetzen Sie `docker-compose_v3_alpine_mysql_latest.yaml` im obigen Befehl durch die erforderliche Konfigurationsdatei.

Die folgenden Optionen sind verfügbar:

Dateiname	Beschreibung
`docker-compose_v3_alpine_mysql_latest.yaml`	Die Compose-Datei stellt die neueste Version der Zabbix-8.0-Komponenten unter Alpine Linux mit Unterstützung für die MySQL-Datenbank aus.
`docker-compose_v3_alpine_mysql_local.yaml`	Die Compose-Datei stellt die neueste Version von Zabbix 8.0 lokal und führt Zabbix-Komponenten unter Alpine Linux mit Unterstützung für die MySQL-Datenbank aus.
`docker-compose_v3_alpine_pgsql_latest.yaml`	Die Compose-Datei stellt die neueste Version der Zabbix-8.0-Komponenten unter Alpine Linux mit Unterstützung für die PostgreSQL-Datenbank aus.

Dateiname	Beschreibung
docker-compose_v3_alpine_pgsql_latest.yaml	Die Compose-Datei stellt die neueste Version von Zabbix 8.0 lokal und führt Zabbix-Komponenten unter Alpine Linux mit Unterstützung für die PostgreSQL-Datenbank aus.
docker-compose_v3_ol_mysql_latest.yaml	Die Compose-Datei führt die neueste Version der Zabbix-8.0-Komponenten unter Oracle Linux mit Unterstützung für die MySQL-Datenbank aus.
docker-compose_v3_ol_mysql_local.yaml	Die Compose-Datei stellt die neueste Version von Zabbix 8.0 lokal und führt Zabbix-Komponenten unter Oracle Linux mit Unterstützung für die MySQL-Datenbank aus.
docker-compose_v3_ol_pgsql_latest.yaml	Die Compose-Datei führt die neueste Version der Zabbix-8.0-Komponenten unter Oracle Linux mit Unterstützung für die PostgreSQL-Datenbank aus.
docker-compose_v3_ol_pgsql_local.yaml	Die Compose-Datei stellt die neueste Version von Zabbix 8.0 lokal und führt Zabbix-Komponenten unter Oracle Linux mit Unterstützung für die PostgreSQL-Datenbank aus.
docker-compose_v3_ubuntu_mysql_latest.yaml	Die Compose-Datei führt die neueste Version der Zabbix-8.0-Komponenten unter Ubuntu 24.04 (noble) mit Unterstützung für die MySQL-Datenbank aus.
docker-compose_v3_ubuntu_mysql_local.yaml	Die Compose-Datei stellt die neueste Version von Zabbix 8.0 lokal und führt Zabbix-Komponenten unter Ubuntu 24.04 (noble) mit Unterstützung für die MySQL-Datenbank aus.
docker-compose_v3_ubuntu_pgsql_latest.yaml	Die Compose-Datei führt die neueste Version der Zabbix-8.0-Komponenten unter Ubuntu 24.04 (noble) mit Unterstützung für die PostgreSQL-Datenbank aus.
docker-compose_v3_ubuntu_pgsql_local.yaml	Die Compose-Datei stellt die neueste Version von Zabbix 8.0 lokal und führt Zabbix-Komponenten unter Ubuntu 24.04 (noble) mit Unterstützung für die PostgreSQL-Datenbank aus.

Speicher

Compose-Dateien sind so konfiguriert, dass sie lokalen Speicher auf einem Host-Rechner unterstützen. Docker Compose erstellt ein Verzeichnis `zbx_env` in dem Ordner mit der Compose-Datei, wenn Sie Zabbix-Komponenten mit der Compose-Datei ausführen. Das Verzeichnis enthält dieselbe Struktur wie im Abschnitt Volumes beschrieben sowie ein Verzeichnis für die Datenbankspeicherung.

Es gibt außerdem Volumes im Nur-Lese-Modus für die Dateien `/etc/localtime` und `/etc/timezone`.

Umgebungsvariablen

Die Variablendateien haben die folgende Namensstruktur: `.env_<Typ der Komponente>` und befinden sich im Verzeichnis *env_vars* directory. Siehe Umgebungsvariablen für Details zur Benennung der Variablen und zur verfügbaren Auswahl.

Beispiele

Beispiel 1

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

Der Befehl lädt die neuesten Zabbix-8.0-Images für jede Zabbix-Komponente herunter und startet sie im Detached-Modus.

> **Attention:**
> Vergessen Sie nicht, die Dateien `.env_<type of component>` aus dem offiziellen Zabbix-Repository auf github.com zusammen mit den Compose-Dateien herunterzuladen.

Beispiel 2

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

Der Befehl lädt das Basis-Image Ubuntu 24.04 (noble) herunter, erstellt dann die Zabbix-8.0-Komponenten lokal und startet sie im Detached-Modus.

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 Konfigurationsdatei 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

C99 mit GNU-Erweiterungen ist für das Erstellen von Zabbix Server, Zabbix Proxy oder Zabbix Agent erforderlich. Diese Version kann explizit angegeben werden, indem CFLAGS=”-std=gnu99” gesetzt wird:

```
export CFLAGS="-std=gnu99"
```

Note:
Wenn die Installation aus dem Zabbix Git repository erfolgt, muss zuerst Folgendes ausgeführt werden:

```
./bootstrap.sh
```

Beim Konfigurieren der Quellen für einen Zabbix Server oder Proxy müssen Sie den zu verwendenden Datenbanktyp angeben. Es kann jeweils nur ein Datenbanktyp mit einem Server- oder Proxy-Prozess gleichzeitig 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 zum Beispiel 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 zum Beispiel Folgendes ausführen:

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

Um die Quellen für einen Zabbix Proxy (mit SQLite usw.) zu konfigurieren, können Sie zum Beispiel 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 Kompilierungsoptionen:

- `--enable-agent` - kompiliert Zabbix Agent sowie die Befehlszeilenwerkzeuge Zabbix get und Zabbix sender.
- `--with-libcurl` - erforderlich für die Überwachung virtueller Maschinen, SMTP-Authentifizierung und `web.page.*`-Zabbix Agent-Datenpunkte. 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 nur die Angabe eines benutzerdefinierten Installationspfads für PCRE2.
- `--with-mysql=/path/to/mysql_config` - gibt den Pfad zu einer bestimmten Konfiguration der MySQL-Client-Bibliothek an. Nützlich, wenn mehrere Versionen von MySQL oder MariaDB installiert sind.
- `--enable-static` - verknüpft Bibliotheken statisch (nicht unterstützt unter Solaris). Verwenden Sie dies, wenn Sie kompilierte Binärdateien auf Systeme ohne die erforderlichen Bibliotheken verteilen möchten. Nicht empfohlen beim Erstellen von Zabbix Server. Um den Server statisch zu erstellen, ist eine statische Version jeder externen Bibliothek erforderlich. Das configure-Skript prüft dies nicht automatisch.
- `--with-stacksize=<value>` - legt die Stack-Größe pro Thread in Kilobyte fest (z. B. `--with-stacksize=512`). Sie können diesen Wert erhöhen, wenn Zabbix aufgrund von Stack-Überläufen abstürzt oder hängen bleibt (z. B. während der Vorverarbeitung auf Systemen mit niedrigen Standardgrenzen für den Thread-Stack).

Attention:
Wenn `./configure` aufgrund fehlender Bibliotheken oder anderer Probleme fehlschlägt, prüfen Sie bitte die Datei `config.log` auf detaillierte Fehlerinformationen.
Wenn zum Beispiel `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:**
> Wenn die Installation aus dem Zabbix Git repository erfolgt, muss zunächst Folgendes ausgeführt werden:
>
> ```
> $ make dbschema
> ```

```
make install
```

Dieser Schritt sollte von einem Benutzer mit ausreichenden Berechtigungen ausgeführt werden (üblicherweise 'root' oder unter Verwendung von `sudo`).

Durch Ausführen von `make install` 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 Quellcode-Konfiguration den Schlüssel --prefix, zum Beispiel --prefix=/home/zabbix. In diesem Fall werden die Daemon-Binärdateien unter <prefix>/sbin und die Dienstprogramme unter <prefix>/bin installiert. Die Manpages werden unter <prefix>/share installiert.

6 Konfigurationsdateien überprüfen und bearbeiten

- Bearbeiten Sie die Zabbix-Agent-Konfigurationsdatei **/usr/local/etc/zabbix_agentd.conf**

Sie müssen diese Datei für jeden Host konfigurieren, auf dem zabbix_agentd installiert ist.

Sie müssen die **IP-Adresse** des Zabbix-Servers in der Datei angeben. Verbindungen von anderen Hosts werden abgelehnt.

- Bearbeiten Sie die Zabbix-Server-Konfigurationsdatei **/usr/local/etc/zabbix_server.conf**

Sie müssen den Datenbanknamen, den Benutzer und das Passwort angeben (falls verwendet).

Die übrigen Parameter sind mit ihren Standardwerten für eine kleine Installation (bis zu zehn überwachte Hosts) ausreichend. Sie sollten die Standardparameter jedoch ändern, wenn Sie die Leistung des Zabbix-Servers (oder Proxys) maximieren möchten.

- Wenn Sie einen Zabbix-Proxy installiert haben, bearbeiten Sie die Proxy-Konfigurationsdatei **/usr/local/etc/zabbix_proxy.conf**

Sie müssen die IP-Adresse des Servers und den Proxy-Hostnamen angeben (muss dem Server bekannt sein), ebenso wie den Datenbanknamen, den Benutzer und das Passwort (falls verwendet).

> **Note:**
> Bei SQLite muss der vollständige Pfad zur Datenbankdatei angegeben werden; DB-Benutzer und 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) Shared Memory 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 passieren.

Führen Sie zabbix_agentd auf allen überwachten Rechnern aus.

```
zabbix_agentd
```

> **Note:**
> Stellen Sie sicher, dass Ihr System die Zuweisung von 2 MB Shared Memory 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 passieren.

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 für den Betrieb ein PHP-unterstützender Webserver benötigt. Die Installation erfolgt einfach durch Kopieren der PHP-Dateien aus dem Verzeichnis `ui` in das HTML-Dokumentenverzeichnis des Webservers.

Gängige Speicherorte für HTML-Dokumentenverzeichnisse von Apache-Webservern sind:

- /usr/local/apache2/htdocs (Standardverzeichnis bei der Installation von Apache aus dem Quellcode)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

Es wird empfohlen, ein Unterverzeichnis anstelle des HTML-Wurzelverzeichnisses zu verwenden. Um ein Unterverzeichnis zu erstellen und die Dateien des Zabbix Frontends hineinzukopieren, führen Sie die folgenden Befehle aus und ersetzen Sie dabei `<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 Anweisungen unter Installation zusätzlicher Frontend-Sprachen.

Installation des Frontends

Informationen zum Installationsassistenten des Zabbix-Frontends finden Sie auf der Seite Installation der Weboberfläche.

Installation ladbarer Plugins für Zabbix Agent 2

Die Installation ladbarer Plugins für Zabbix Agent 2 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.). Eine vollständige Liste der ladbaren Plugins und integrierten Plugins finden Sie hier.

> **Attention:**
> Bevor Sie ein Plugin installieren, prüfen Sie bitte dessen README-Datei. Sie kann spezifische Anforderungen und Installationsanweisungen enthalten.

Um aus den Quellen zu installieren, laden Sie zunächst das Quellarchiv des ladbaren Plugins 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 Plugins für Zabbix Agent 2 muss eine unterstützte Go-Version installiert sein.

Die ausführbare Plugin-Datei kann an einem beliebigen Ort 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 Plugin PostgreSQL:

```
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.

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 werden
- `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` - Go-Code mit `go fmt` formatieren
- `make dist` - ein Quellarchiv einschließlich aller Abhängigkeiten erstellen

Installation des Java gateway

Die Installation des Java gateway ist nur erforderlich, wenn Sie JMX-Anwendungen überwachen möchten. Der Java gateway ist schlank und benötigt keine Datenbank.

Für die Installation aus den Quellen müssen Sie zunächst das Quellarchiv herunterladen und entpacken.

Um den Java gateway zu kompilieren, führen Sie das Skript `./configure` mit der Option `--enable-java` aus. Es wird empfohlen, zusätzlich die Option `--prefix` anzugeben, um einen anderen Installationspfad als den Standardpfad /usr/local festzulegen, da bei der Installation des Java gateway ein kompletter Verzeichnisbaum und nicht nur eine einzelne ausführbare Datei erstellt wird.

```
./configure --enable-java --prefix=$PREFIX
```

Um den Java gateway zu kompilieren und in eine 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 eine Datei zabbix-java-gateway-$VERSION.jar in src/zabbix_java/bin. Wenn Sie den Java gateway direkt aus src/zabbix_java im Distributionsverzeichnis ausführen möchten, können Sie mit den Anweisungen zur Konfiguration und Ausführung des 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 Ausführung des Java gateway finden Sie unter Einrichtung.

Installation des Zabbix-Web-Service

Die Installation des Zabbix-Web-Service ist nur erforderlich, wenn Sie geplante Berichte verwenden möchten.

Für die Installation aus den Quellen müssen Sie zunächst das Quellarchiv herunterladen und entpacken.

Um den Zabbix-Web-Service zu kompilieren, führen Sie das Skript `./configure` mit der Option `--enable-webservice` aus.

> **Note:**
> Zum Erstellen des Zabbix-Web-Service muss eine unterstützte Go-Version installiert sein.

Führen Sie zabbix_web_service auf dem Rechner aus, auf dem der Web-Service installiert ist:

```
zabbix_web_service
```

Weitere Informationen zur Konfiguration der Generierung geplanter Berichte finden Sie unter Einrichtung.

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).

Stack-Größe auf einigen Systemen zu klein

Wenn Zabbix aufgrund von Stack-Überläufen abstürzt oder einfriert, können Sie die Stack-Größe pro Thread mit der Option `--with-stacksize` erhöhen, wenn Sie die Quellen konfigurieren. Dieses Problem kann auf Systemen mit niedrigen Standardgrenzen für den Thread-Stack auftreten, insbesondere während der Vorverarbeitung, bei der mehrere Threads erstellt werden.

Das folgende Beispiel setzt die Stack-Größe auf 512 KB pro Thread:

```
./configure --enable-server --with-mysql --with-stacksize=512
```

Sie können die Systemgrenzen für den Thread-Stack zur Laufzeit mit dem Befehl `ulimit -s` auf Linux-basierten Systemen prüfen.

Hinweise zur Installation aus Paketen

Übersicht

Offizielle **Vorabversion**-Pakete von Zabbix 8.0 sind auf der Zabbix-Website verfügbar.

Wählen Sie Ihr Betriebssystem und die Zabbix-Komponente aus, um Installationsanweisungen zu erstellen, die für Ihre Umgebung geeignet sind. Siehe auch die Hinweise zur Paketinstallation auf dieser Seite für wichtige zusätzliche Informationen sowie den Abschnitt Installation and setup.

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 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 offiziellen Zabbix-Repository zu verwenden. Wenn Sie Zabbix zuvor aus dem Repository Ihres Betriebssystems installiert haben, siehe die Schritte für das Upgrade von Zabbix-Paketen aus Betriebssystem-Repositories.

Die Pakete unterstützen MySQL-/PostgreSQL-Datenbanken und Apache-/Nginx-Webserver. Beachten Sie, dass Zabbix Server und Proxy nicht dieselbe Datenbank gemeinsam nutzen 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.

Zabbix stellt außerdem vorkompilierte Binärdateien für Zabbix Agent für Nicht-Linux-Betriebssysteme bereit; siehe:

- Installation des Windows-Agenten aus MSI
- Installation des macOS-Agenten aus PKG
- Legacy-Binärdateien (für ältere/weniger verbreitete Systeme wie HP-UX, NetBSD, Tru64 und ältere Versionen von SLES)

Hinweise zur Paketinstallation

Die folgenden Hinweise gelten für alle Systeme:

- Wenn PostgreSQL verwendet wird, bewirkt `DBHost=localhost` (oder eine IP-Adresse) in der Konfiguration von Zabbix-Server/Proxy, dass PostgreSQL einen Netzwerk-Socket anstelle eines lokalen UNIX-Sockets verwendet; siehe SELinux-Konfiguration für entsprechende Einrichtungsanweisungen.
- Wenn TimescaleDB verwendet wird, siehe die zusätzlichen Informationen zur Einrichtung von TimescaleDB.
- Wenn Zabbix Java gateway installiert wird (zur Überwachung von JMX-Anwendungen), siehe die zusätzlichen Einrichtungsanweisungen für RHEL-basierte Systeme und Debian-basierte Systeme.
- Für den Betrieb des Zabbix Agent als root siehe Agent als root ausführen.

Die folgenden Hinweise gelten für RHEL und seine Derivate:

- Wenn Sie das EPEL-Repository für EL9 aktiviert haben, das ebenfalls Zabbix-Pakete bereitstellt, muss es vor der Installation offizieller Zabbix-Pakete von der Paketauflösung ausgeschlossen werden; siehe Versehentliche 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 socket-basierte 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/verwenden kann. Socket-Dateien werden vom Zabbix Server (Alerter, Preprocessing, IPMI) und vom Zabbix Proxy (IPMI) verwendet und sind vorhanden, solange der Prozess läuft.

Wenn SELinux im Enforcing-Modus aktiviert ist, führen Sie die folgenden Befehle aus, um die Kommunikation zwischen Zabbix Frontend und Server zu aktivieren:

Für RHEL 7 (und höher), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (und höher):

```
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, eine Verbindung zur Datenbank herzustellen:

```
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 Apache nach dem Anwenden der SELinux-Einstellungen neu:

```
systemctl restart httpd
```

Optional können Sie ein vordefiniertes Paket `zabbix-selinux-policy` aus dem offiziellen Zabbix-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.

> **Attention:**
> Für maximale Sicherheit wird empfohlen, benutzerdefinierte SELinux-Einstellungen festzulegen.

Das Paket `zabbix-selinux-policy` enthält eine grundlegende SELinux-Richtlinie, die es Zabbix ermöglicht, Sockets zu erstellen und zu verwenden, und die HTTPd-Verbindung zu PostgreSQL aktiviert (vom Frontend verwendet).

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 Betrieb nicht erforderlich, sind jedoch für die erweiterte Fehlerbehebung nützlich.

So aktivieren Sie das Repository `zabbix-debuginfo`:

- Bearbeiten Sie unter RHEL 7 `/etc/yum.repos.d/zabbix.repo` und setzen Sie `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
```

- Bearbeiten Sie unter SUSE `/etc/zypp/repos.d/zabbix.repo` und setzen Sie `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/stable/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Nach der Aktivierung installieren Sie die Pakete:

- Installieren Sie unter RHEL ein einzelnes Paket mit Debuginformationen für alle Zabbix-Komponenten:

```
dnf install zabbix-debuginfo
```

- Installieren Sie unter SUSE komponentenspezifische Debuginfo-Pakete:

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

Aktivieren instabiler Release-Repositories

Die folgenden Anweisungen dienen zum Aktivieren instabiler Zabbix-Release-Repositories (standardmäßig deaktiviert), die für Release-Kandidaten von kleineren Zabbix-Versionen verwendet werden.

Installieren Sie zunächst das neueste Paket zabbix-release oder aktualisieren Sie darauf. 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.

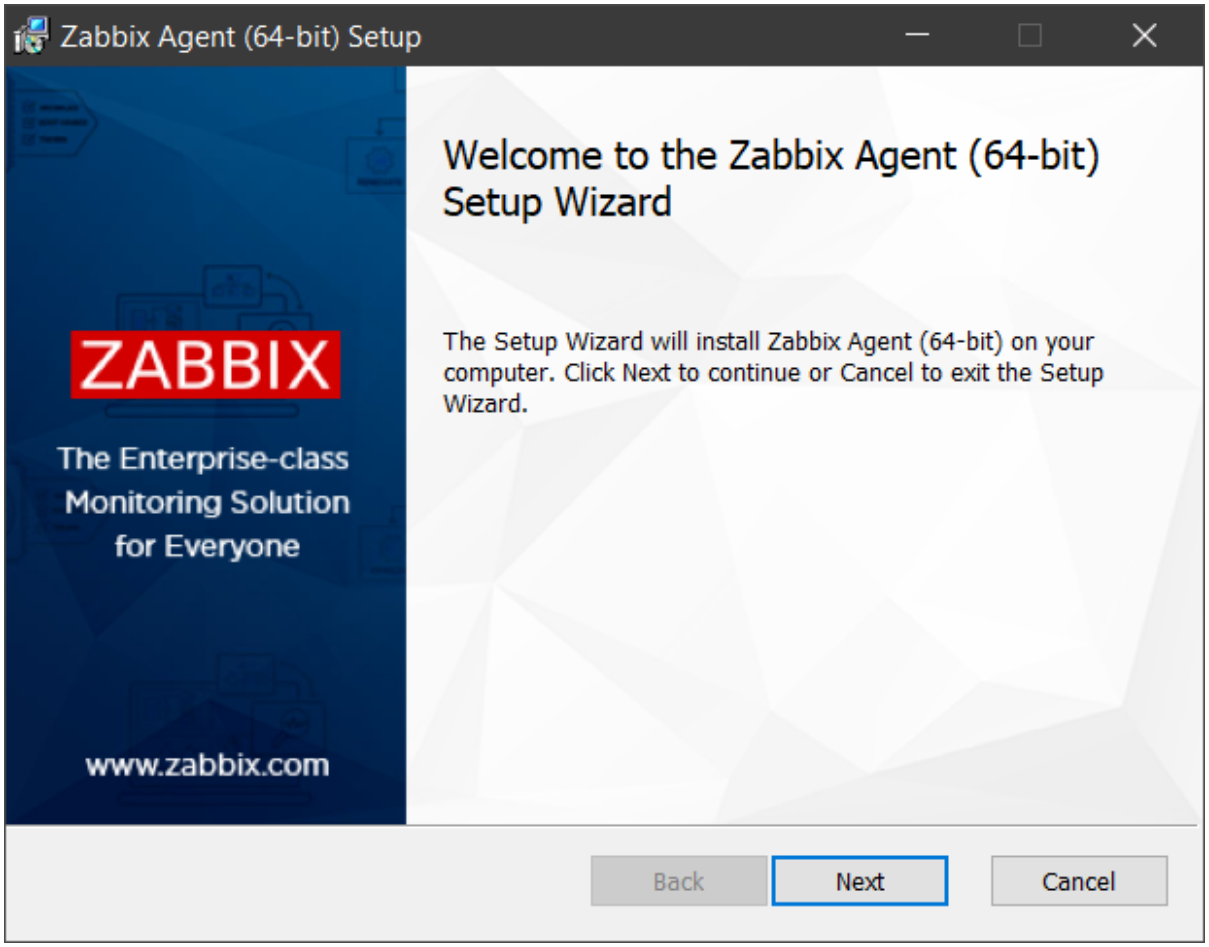
Obwohl die Installation mit MSI-Paketen vollständig unterstützt wird, wird für eine ordnungsgemäße Fehlerbehandlung empfohlen, mindestens Microsoft .NET Framework 2 zu installieren.

> **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 mit dem Setup-Assistenten

Die folgenden Installationsschritte gelten sowohl für Zabbix Agent als auch für Zabbix Agent 2.

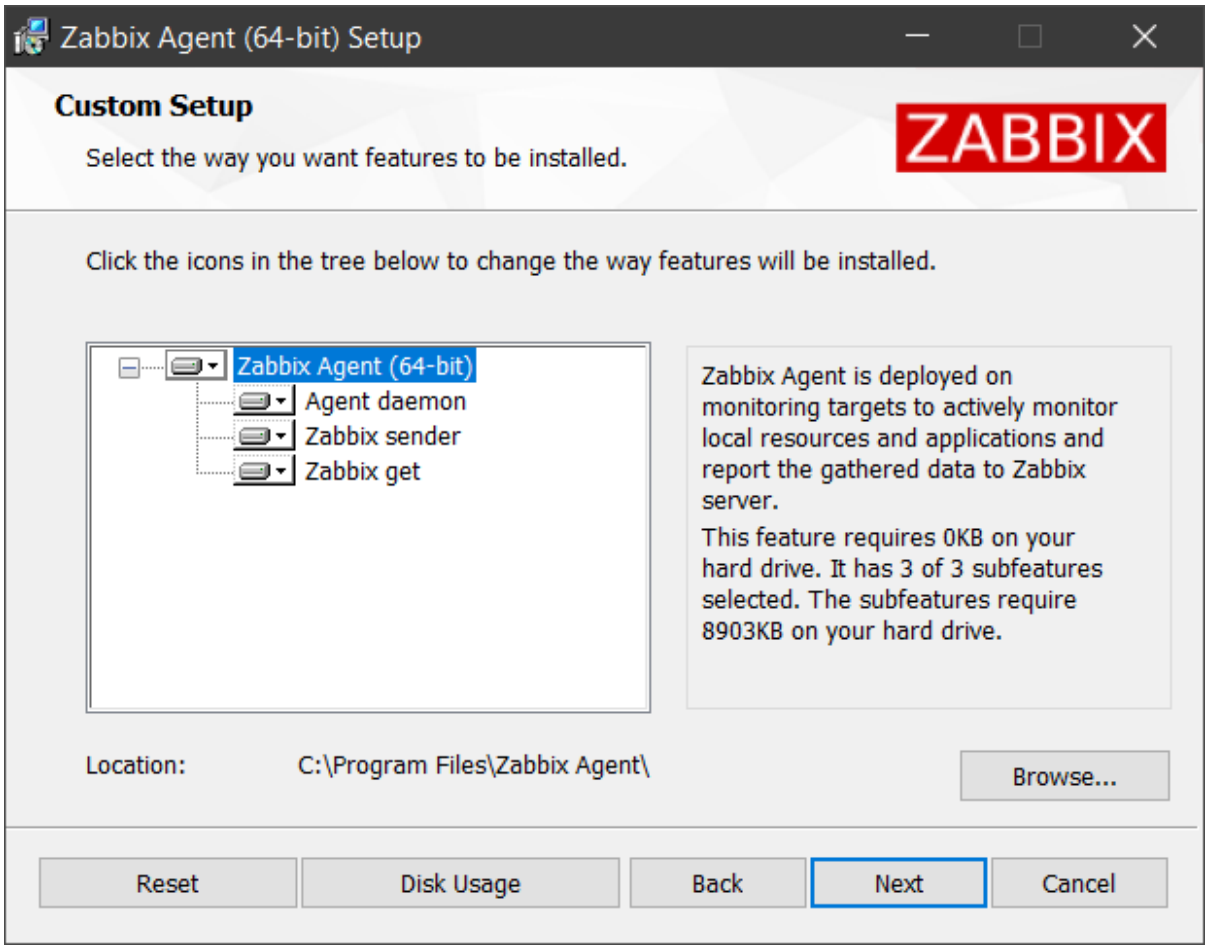
1. Doppelklicken Sie auf die heruntergeladene MSI-Datei, um die Installation zu starten:



2. Akzeptieren Sie die Endbenutzer-Lizenzvereinbarung:

3. Wählen Sie die zu installierenden Zabbix-Komponenten aus (Agent-Daemon, Zabbix sender, Zabbix get):

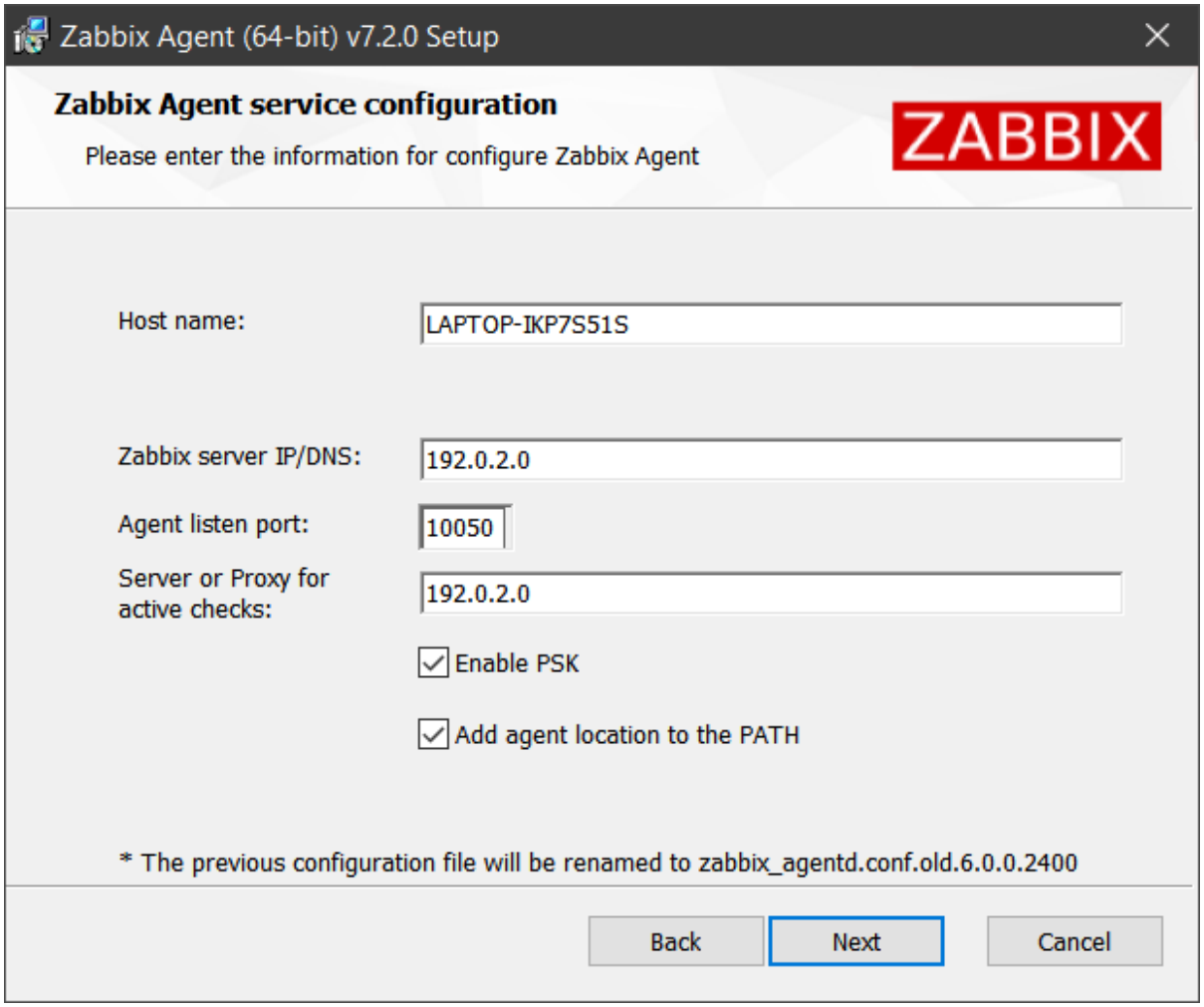
Attention:
Es wird empfohlen, den vom Installationsprogramm vorgegebenen Standard-Installationspfad zu verwenden. Die Verwendung eines benutzerdefinierten Pfads ohne die erforderlichen Berechtigungen kann die Sicherheit der Installation beeinträchtigen.



4. Konfigurieren Sie die folgenden Parameter. Ihre Werte werden in der Zabbix-Agent-Konfigurationsdatei festgelegt:

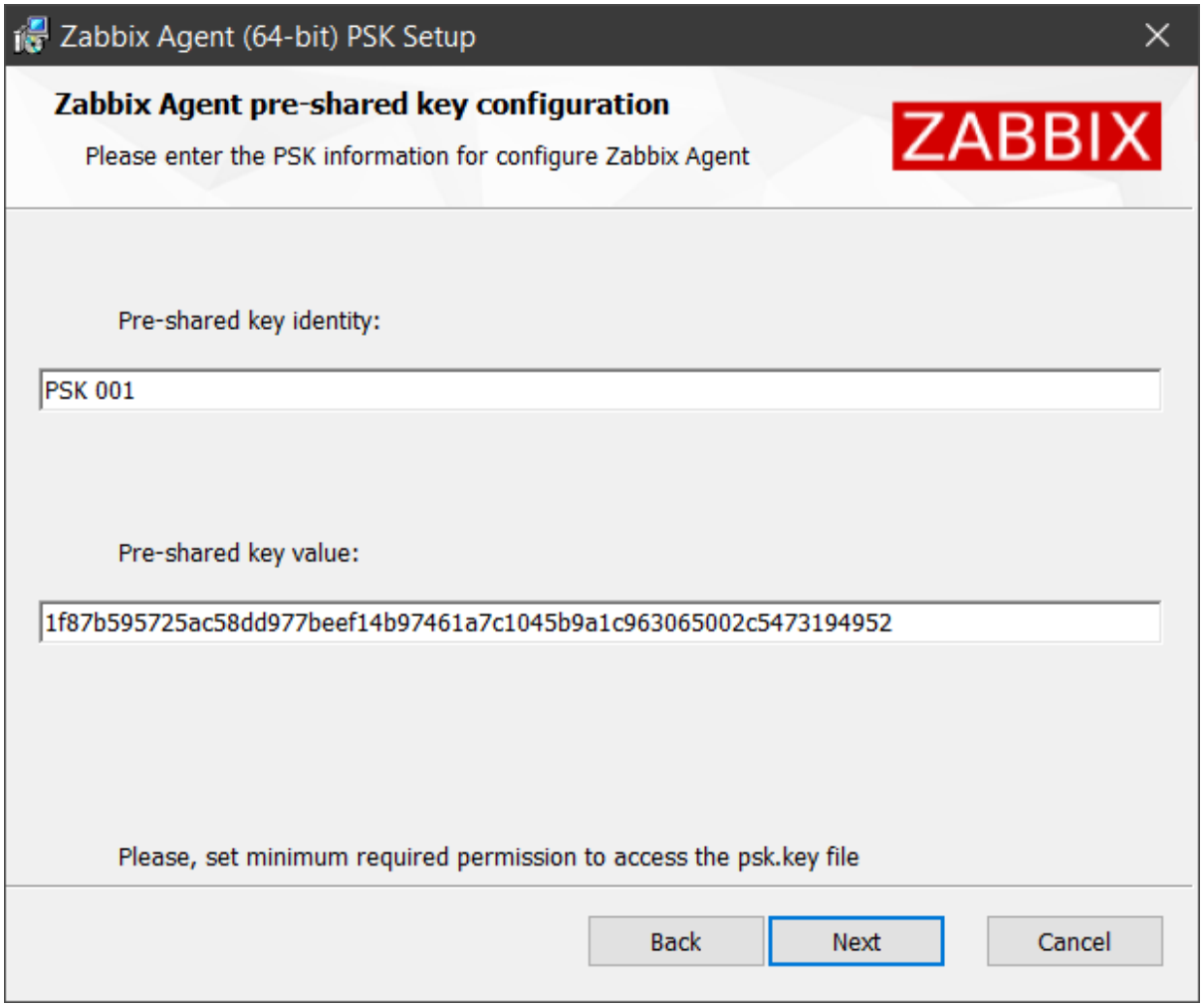
Parameter	Beschreibung
Host name	Der Hostname des Rechners, auf dem Zabbix Agent installiert wird. Setzt den Parameter Hostname.
Zabbix server IP/DNS	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern oder Zabbix-Proxys. Dieser Parameter ist **verpflichtend**. Setzt den Parameter Server.
Agent listen port	Der Agent lauscht auf diesem Port auf Verbindungen vom Server. Setzt den Parameter ListenPort.
Server or Proxy for active checks	Die Adresse des Zabbix-Servers/Proxys oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden. Die Server-/Proxy-Adresse ist eine IP-Adresse oder ein DNS-Name mit optionalem, durch Doppelpunkt getrenntem Port. Setzt den Parameter ServerActive.
Enable PSK	Aktivieren Sie das Kontrollkästchen, um TLS-Unterstützung mit vorinstallierten Schlüsseln zu aktivieren. Setzt die Parameter TLSConnect und TLSAccept auf psk.
Add agent location to the PATH	Aktivieren Sie das Kontrollkästchen, um den Installationspfad von Zabbix Agent zur Systemvariablen PATH hinzuzufügen.

Note:
Wenn ein vorhandener Zabbix Agent erkannt wird, werden die Parameterwerte aus dessen Konfigurationsdatei angezeigt. Außerdem 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 Zabbix-Agent-Konfigurationsdatei festgelegt:

Parameter	Beschreibung
Pre-shared key identity	Die Identitätszeichenfolge des vorinstallierten Schlüssels. Setzt den Parameter TLSPSKIdentity.
Pre-shared key value	Der Zeichenfolgenwert des vorinstallierten Schlüssels. Erstellt die Datei psk.key, die den Schlüssel enthält, und setzt den Parameter TLSPSKFile auf den Speicherort des Schlüssels (Standard: `C:\Program Files\Zabbix Agent\psk.key`). Es wird **empfohlen**, den Zugriff auf die Datei mit dem vorinstallierten Schlüssel durch Anpassen der Sicherheitseinstellungen der Datei einzuschränken, sodass nur Zabbix Agent (oder der Benutzer, unter dem der Agent ausgeführt wird) sie lesen kann.

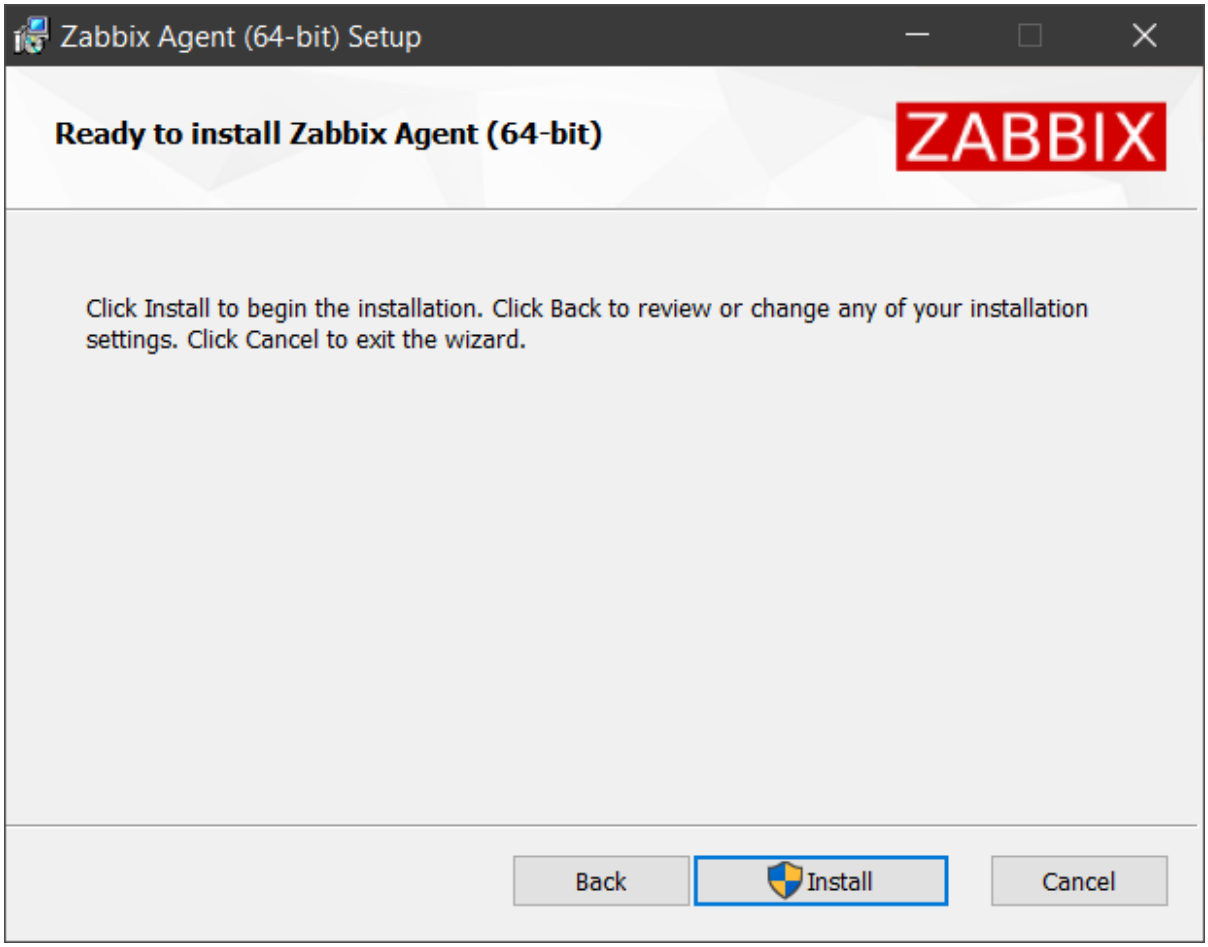


6. Klicken Sie auf *Install*, um die Installation zu starten.

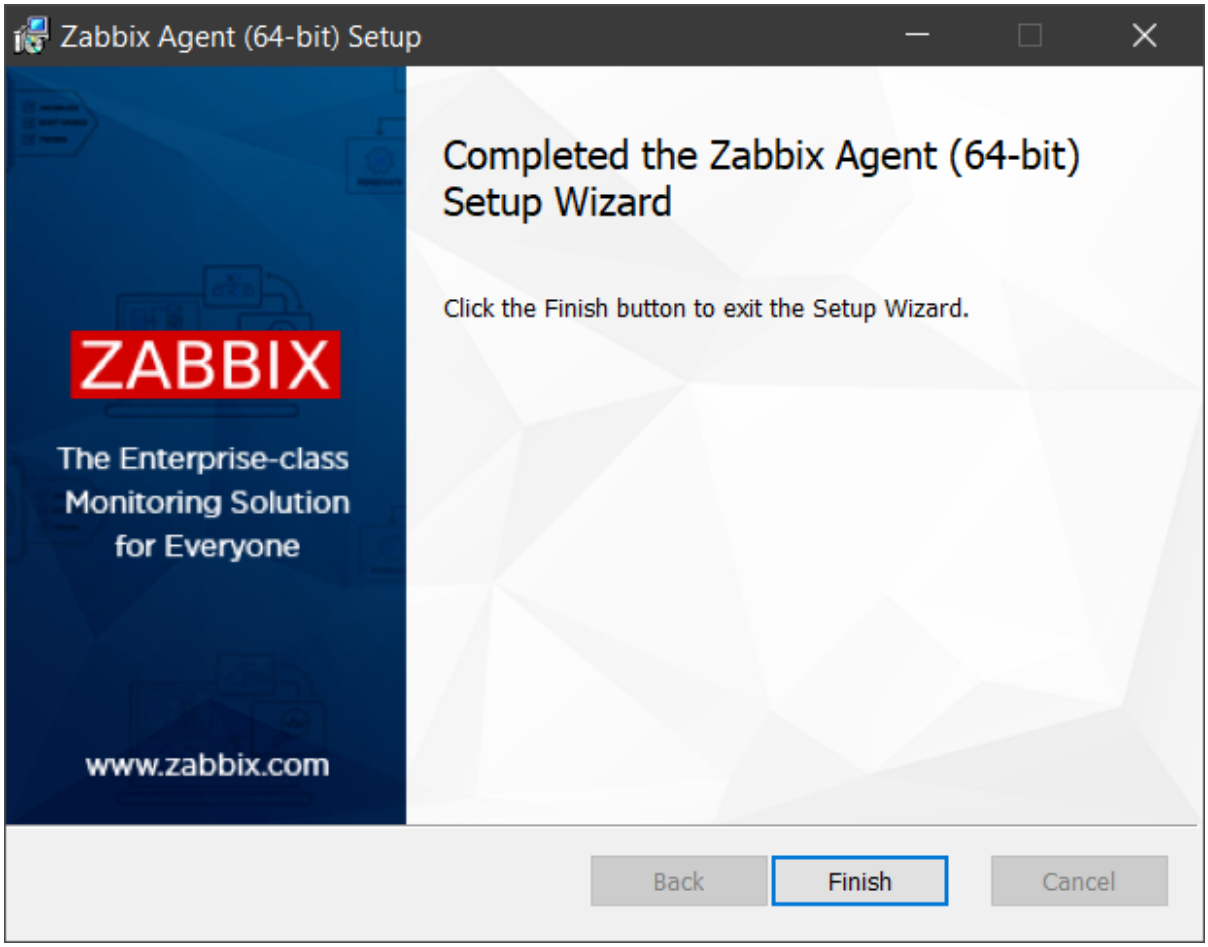
Alle ausgewählten Zabbix-Komponenten und die Zabbix-Agent-Konfigurationsdatei werden an dem von Ihnen angegebenen Speicherort installiert (Standard: `C:\Program Files\Zabbix Agent`). Dasselbe gilt für Zabbix Agent 2, mit der Ausnahme, dass zusätzliche Konfigurationsdateien für seine integrierten Plugins 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 auf Windows-Versionen vor Windows Vista/Server 2008).

Wenn während der Installation eine andere Version von 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 des Systems 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

Die MSI-Installer-Pakete des Zabbix Agent 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 gesetzt. 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 mit 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, um Zabbix-Agent-Prüfungen einzuschränken. Falls erforderlich, verwenden Sie einen Backslash, um das Trennzeichen zu maskieren (`\;`). Setzt die Parameter AllowKey und DenyKey in der Konfigurationsdatei des Agent. Beispiel: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	Der vollständige Pfadname zu einer Vorlagen-Konfigurationsdatei für Zabbix Agent. Während der Installation wird diese Datei zur Konfigurationsdatei des Agent. 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 der MSI-Installer den Zabbix-Agent-Dienst startet.
ENABLEPATH	Verwenden Sie `ENABLEPATH=1`, um den Speicherort von Zabbix Agent zur Systemvariablen PATH hinzuzufügen.
ENABLEPERSISTENTBUFFER	Nur Zabbix Agent 2. Aktiviert die Verwendung eines 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 des Zabbix Agent aufgenommen werden sollen.
INSTALLFOLDER	Der vollständige Pfadname zu einem Ordner, in dem Zabbix-Komponenten und die Konfigurationsdatei des 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 des Zabbix Agent.
LOGTYPE	Der Typ der Protokollausgabe.

Parameter	Beschreibung
NONMSICONFNAME	Der vollständige Pfadname 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 Konfigurationsdatei des Agent 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 Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden.
SKIP	Verwenden Sie `SKIP=fw`, um zu verhindern, dass der MSI-Installer eine Ausnahmeregel in der Windows-Firewall für Zabbix Agent hinzufügt.
SOURCEIP	Die Quell-IP-Adresse für ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy oder für Verbindungen, die bei der Ausführung einiger Datenpunkte hergestellt werden (web.page.get, net.tcp.port usw.).
STARTAGENTS	Die Anzahl der vorab geforkten 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** - den Dienst beim Windows-Start automatisch starten; **delayed** - *(Standard)* den Start des Dienstes verzögern, bis die automatisch gestarteten Dienste vollständig gestartet wurden (verfügbar unter Windows Vista/Server 2008 und neueren Versionen); **manual** - den Dienst manuell starten (durch einen Benutzer oder eine Anwendung); **disabled** - den Dienst deaktivieren, 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 (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Proxy oder Server gewartet werden soll.
TLSACCEPT	Die zu akzeptierenden eingehenden Verbindungen (verwendet für passive Prüfungen). Wenn auf `psk` gesetzt, wird TLSCONNECT ebenfalls auf `psk` gesetzt (sofern nicht anders angegeben).
TLSCAFILE	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) zur Verifizierung des Zertifikats der Gegenstelle enthält.
TLSCERTFILE	Der vollständige Pfadname einer Datei, die das Zertifikat oder die Zertifikatskette des Agent enthält.
TLSCONNECT	Wie sich der Agent mit Zabbix Server oder Proxy verbinden soll (verwendet für aktive Prüfungen). Wenn auf `psk` gesetzt, wird TLSACCEPT ebenfalls auf `psk` gesetzt (sofern nicht anders angegeben).
TLSCRLFILE	Der vollständige Pfadname einer Datei, die widerrufene Zertifikate enthält.
TLSKEYFILE	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Zabbix Agent enthält.
TLSPSKFILE	Der vollständige Pfadname einer Datei, die den Pre-Shared Key des Zabbix Agent enthält. 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 Pre-Shared-Key-Datei durch Anpassen der Sicherheitseinstellungen der Datei einzuschränken, sodass nur Zabbix Agent (oder der Benutzer, der den Agent ausführt) sie lesen kann.
TLSPSKIDENTITY	Die Identitätszeichenfolge des Pre-Shared Key.
TLSPSKVALUE	Der Zeichenfolgenwert des Pre-Shared Key. 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, alle Zeichen in Argumenten für benutzerdefinierte Parameter zu übergeben.

Beispiele

Das folgende Beispiel installiert den Zabbix Agent mit benutzerdefinierter Konfiguration. Es aktiviert außerdem die TLS-Unterstützung mithilfe von Pre-Shared Keys.

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.0^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

Das nächste Beispiel installiert eine neuere Version des Zabbix Agent und verwendet eine **Vorlagen-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 Parameter-Platzhalter (z. B. `[AllowDenyKey]`).

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### Beispiel für agent-template.conf:
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 Agent-Konfigurationsparameter, die in dieser Datei vorhanden sind (beschränkt auf die in der obigen Tabelle aufgeführten), in die neu erstellte Konfigurationsdatei des Agent geschrieben. Um die bestehende Agent-Konfiguration beizubehalten, definieren Sie die Parameter, die beibehalten werden sollen.

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### Beispiel für agent-custom.conf:
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
```

Zabbix Agent 2 ladbare Plugins

Ladbare Plugins für Zabbix Agent 2 loadable plugins können unter Windows mit 64-Bit-MSI-Installer-Paketen installiert werden, die zum Download verfügbar sind.

Die Mindestanforderungen an das Betriebssystem für die MSI-Installation sind Windows 10 (64-Bit) oder Windows Server 2016.

> **Attention:**
> Bitte prüfen Sie vor der Installation eines Plugins 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

Ladbare Plugins für Zabbix Agent 2 können 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_agent2_plugins-8.0.0-windows-amd64.msi" /qn+
```

MSI-Installationspakete für ladbare Plugins von Zabbix Agent 2 unterstützen die folgenden Parameter.

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: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Beispiel: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Eine durch Kommas getrennte Liste von Komponenten, die lokal installiert werden sollen. 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; ihre 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 Zabbix Agent 2 unter Windows 10 (64-Bit oder 32-Bit) aus den Quellen erstellt wird.

Sowohl 32-Bit- als auch 64-Bit-Versionen können auf einer 64-Bit-Plattform erstellt werden, auf einer 32-Bit-Plattform jedoch nur die 32-Bit-Version.

Zum Erstellen von Zabbix Agent 2 werden folgende Komponenten benötigt:

- MinGW-Build-Tools
- Programmiersprache Go
- OpenSSL (für Verschlüsselungs-Funktionen in Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; für Funktionen zum Musterabgleich mit regulären Ausdrücken in Zabbix)

Sie können Zabbix Agent 2 mit einer der folgenden Methoden erstellen:

- Mit vcpkg – ein automatisierter Ansatz, der die Verwaltung von Abhängigkeiten mithilfe eines C++-Paketmanagers vereinfacht.
- Manueller Build – ein manueller Ansatz, bei dem vor dem Kompilieren des Agent alle Abhängigkeiten installiert werden müssen.

Attention:
Bevor Sie mit dem Build-Prozess 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. Bei der Installation von OpenSSL und PCRE2 verwenden Sie jedoch das MSYS2-Terminal.
- Es wird empfohlen, ein Arbeitsverzeichnis unter `C:\Zabbix` für alle Quelldateien und Build-Ordner zu erstellen. Kompilierte Komponenten sollten jedoch in `C:\Zabbix\x64` installiert werden (bzw. in `C:\Zabbix\x86` für 32-Bit-Builds).

Erstellen von Zabbix Agent 2 mit vcpkg

Dieser Abschnitt enthält Anweisungen zum Erstellen von Zabbix Agent mit vcpkg, einem Paketmanager, der die Verwaltung von Abhängigkeiten und die Integration in C++-Projekte vereinfacht.

1. Laden Sie 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, die den vcpkg-Paketmanager enthält.

2. Laden Sie Go herunter und installieren Sie es (verfügbar als MSI-Installer). 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`

Extrahieren Sie sie anschließend nach `C:\Zabbix\mingw64` (oder `C:\Zabbix\mingw32` für 32-Bit-Builds).

4. Initialisieren Sie vcpkg und installieren Sie die für das Erstellen von 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

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

#### For 32-bit builds:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
vcpkg install --triplet x86-mingw-static --x-install-root=x86
```

5. Laden Sie das Zabbix-Quellarchiv herunter und extrahieren 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 Skript `build.bat`:

   - Für 64-Bit-Builds:

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

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

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

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

- Für 32-Bit-Builds:

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

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

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

:: Run the build process:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Kompilieren Sie Zabbix Agent 2, indem Sie das Skript ausführen:

```
build.bat
```

Nach der Kompilierung befindet sich die Binärdatei von Zabbix Agent 2 in `C:\Zabbix\zabbix-8.0.0\bin\win64` (für 64-Bit-Builds) oder `C:\Zabbix\zabbix-8.0.0\bin\win32` (für 32-Bit-Builds). Die Konfigurationsdateien von Zabbix Agent 2 befinden sich in `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Um den Agent zu starten, kopieren Sie `zabbix_agent2.exe` und die zugehörigen Konfigurationsdateien in einen dedizierten Ordner (z. B. `C:\Zabbix\agent2`) und starten Sie dann den Agent:

```
mkdir C:\Zabbix\agent2

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

#### For 32-bit builds:
copy C:\Zabbix\zabbix-8.0.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

C:\Zabbix\agent2\zabbix_agent2.exe -c C:\Zabbix\agent2\zabbix_agent2.win.conf
```

Falls erforderlich, fahren Sie mit dem Kompilieren von Zabbix Agent 2-loadable plugins fort.

Kompilieren von ladbaren Plugins für Zabbix Agent 2

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 dessen 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 seine Konfigurationsdatei im selben Plugin-Verzeichnis.

Die ausführbare Plugin-Datei kann an einem beliebigen Ort 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 Plugin PostgreSQL:

```
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.

Build-Tools einrichten

1. Laden Sie MSYS2 herunter und installieren Sie es (als MSI-Installer verfügbar). Stellen Sie während der Installation sicher, dass `C:\Zabbix\msys64` als Installationsordner angegeben ist.

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 `C:\Zabbix\Go` als Installationsordner angegeben ist.

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`

Extrahieren Sie sie dann nach `C:\Zabbix\mingw64` (oder nach `C:\Zabbix\mingw32` für 32-Bit-Builds).

Installation von OpenSSL

> **Note:**
> Um den Zabbix Agent ohne TLS-Unterstützung zu kompilieren, fahren Sie mit dem Abschnitt Installation von PCRE2 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 entpacken Sie es nach `C:\Zabbix\openssl-3.5.0`.

3. Wechseln Sie in das entpackte OpenSSL-Verzeichnis und erstellen Sie das folgende Skript `build.sh`:

- 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 Sie Nicht-Administrator-Benutzern den Schreibzugriff auf das Verzeichnis `C:\Zabbix\x64\OpenSSL-Win64-350-static` entziehen. Andernfalls lädt der Agent SSL-Einstellungen aus einem Pfad, der von nicht privilegierten Benutzern geändert werden kann, was zu einer potenziellen Sicherheitslücke führt.

- Die Option `no-shared` sorgt dafür, dass die statischen OpenSSL-Bibliotheken libcrypto.lib und libssl.lib in sich geschlossen sind, sodass Zabbix-Binärdateien OpenSSL enthalten, ohne externe DLLs zu benötigen. Das bedeutet, dass Zabbix-Binärdateien auf andere Windows-Rechner kopiert werden können, ohne OpenSSL-Bibliotheken mitzukopieren; wenn jedoch eine neue OpenSSL-Bugfix-Version veröffentlicht wird, muss der Zabbix Agent neu kompiliert werden.
- Ohne die Option `no-shared` verwendet Zabbix zur Laufzeit OpenSSL-DLLs. Das bedeutet, dass OpenSSL-Updates möglicherweise keine Neukompilierung des Zabbix Agent erfordern; beim Kopieren auf andere Rechner müssen jedoch auch die OpenSSL-DLLs mitkopiert 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
```

Zabbix Agent 2 kompilieren

1. Laden Sie das Zabbix-Quellarchiv herunter und entpacken Sie es nach `C:\Zabbix\zabbix-8.0.0`.

Wenn Sie ein Quellarchiv aus dem rohen Quellcode-Repository erzeugen müssen (z. B. um benutzerdefinierte Patches anzuwenden oder aus dem neuesten Quellcode zu bauen), führen Sie die folgenden Befehle auf einem **Linux**-Rechner 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 --enable-agent2 --enable-ipv6 --prefix=`pwd`
make dist
```

Dadurch wird ein Quellarchiv erstellt, das anschließend auf einen Windows-Rechner kopiert werden kann.

2. Öffnen Sie die Eingabeaufforderung mit Administratorrechten. Navigieren Sie dann zum Zabbix-Build-Verzeichnis und kompilieren Sie den Zabbix Agent; achten Sie darauf, die Verzeichnisse, in denen OpenSSL und PCRE2 installiert sind, korrekt anzugeben:

- 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-Win64-350-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 `C:\Zabbix\zabbix-8.0.0\bin\win32` für 32-Bit-Builds). Die Konfigurationsdateien von Zabbix Agent 2 befinden sich in `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Um den Agent zu starten, kopieren Sie die Binärdatei `zabbix_agent2.exe` und die zugehörigen Konfigurationsdateien in einen dedizierten Ordner (z. B. `C:\Zabbix\agent2`) und starten Sie dann 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.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

C:\Zabbix\agent2\zabbix_agent2.exe -c C:\Zabbix\agent2\zabbix_agent2.win.conf
```

Falls erforderlich, fahren Sie 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.

Zum Erstellen des Zabbix Agent werden folgende Komponenten benötigt:

- C-Compiler (in den Build Tools für Visual Studio 2022 enthalten)
- OpenSSL (für Verschlüsselungs-Funktionen in Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; für Funktionen zum Musterabgleich mit regulären Ausdrücken in Zabbix)

Sie können den Zabbix Agent mit einer der folgenden Methoden erstellen:

- Mit vcpkg – ein automatisierter Ansatz, der die Verwaltung von Abhängigkeiten mithilfe eines C++-Paketmanagers vereinfacht.
- Manueller Build – ein manueller Ansatz, bei dem vor dem Kompilieren des Agent alle Abhängigkeiten installiert werden müssen.

Abhängig von Ihren Monitoring-Anforderungen können zusätzliche Bibliotheken erforderlich sein. Weitere Informationen finden Sie unter Anforderungen.

Attention:
Bevor Sie mit dem Build-Prozess beginnen, beachten Sie bitte Folgendes:

- Verwenden Sie zum Ausführen von Befehlen die x64 Native Tools Command Prompt (in den Build Tools für Visual Studio 2022 enthalten), gestartet von einem Benutzer mit ausreichenden Berechtigungen zum Schreiben in geschützte Ordner.
- Es wird empfohlen, ein Arbeitsverzeichnis unter `C:\Zabbix` für alle Quelldateien und Build-Ordner zu erstellen. Kompilierte Komponenten sollten jedoch unter `C:\Program Files\Zabbix\x64` installiert werden.

Zabbix Agent 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 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, die die für das Erstellen des Agent mit vcpkg erforderlichen Werkzeuge enthält:

- C-Compiler (Microsoft Visual C++)
- NMake-Befehlszeilentool
- vcpkg-Paketmanager
- x64 Native Tools Command Prompt

2. 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 entpacken 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 Skript `build.bat`; stellen Sie sicher, dass Sie die Verzeichnisse, in denen OpenSSL und PCRE2 installiert sind, korrekt angeben:

```
:: Setzen Sie den vcpkg-Installationspfad:
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: Führen Sie den Build-Prozess aus:
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. Kompilieren Sie den Zabbix Agent, indem Sie das Skript ausführen:

```
build.bat
```

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_agent.exe` und seine Konfigurationsdatei in einen eigenen Ordner (z. B. `C:\Zabbix\agent`) und starten Sie dann den Agent:

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agent.win.conf C:\Zabbix\agent\

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

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.

Installation der Build-Tools

1. Laden Sie 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, die die für das manuelle Erstellen des Agent erforderlichen Werkzeuge enthält:

- C-Compiler (Microsoft Visual C++)
- NMake-Befehlszeilentool
- x64 Native Tools Command Prompt

Installation von OpenSSL

Note:
Um den Zabbix Agent ohne TLS-Unterstützung zu kompilieren, fahren Sie mit dem Abschnitt Installing PCRE2 fort.

1. Laden Sie Strawberry Perl herunter und installieren Sie es (als MSI-Installer verfügbar). Stellen Sie während der Installation sicher, dass Sie `C:\Zabbix\Strawberry` als Installationsordner angeben.

2. Installieren Sie das Perl-Modul `Text::Template`:

```
cpanm Text::Template
```

3. Vergewissern Sie sich, dass der Netwide Assembler (NASM; erforderlich zum Kompilieren von OpenSSL) während der Installation von Strawberry Perl kompiliert wurde:

```
nasm -v
#### NASM version 2.16.01 compiled on May  3 2024
```

Falls NASM nicht kompiliert wurde, installieren Sie es manuell. Weitere Informationen finden Sie in der NASM-Dokumentation.

4. Laden Sie das OpenSSL-Quellarchiv herunter und entpacken Sie es nach `C:\Zabbix\openssl-3.5.0`.

5. Wechseln Sie in das entpackte 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 wählen (z. B. `C:\zabbix` oder `C:\openssl-64bit`), stellen Sie sicher, dass Sie Nicht-Administrator-Benutzern den Schreibzugriff auf dieses Verzeichnis entziehen. Andernfalls lädt der Agent SSL-Einstellungen aus einem Pfad, der von nicht privilegierten Benutzern geändert werden kann, was zu einer potenziellen Sicherheitslücke führt.

- Die Option `no-shared` sorgt dafür, dass die statischen OpenSSL-Bibliotheken libcrypto.lib und libssl.lib in sich geschlossen sind, sodass Zabbix-Binärdateien OpenSSL enthalten, ohne externe DLLs zu benötigen. Das bedeutet, dass Zabbix-Binärdateien ohne OpenSSL-Bibliotheken auf andere Windows-Rechner kopiert werden können; wenn jedoch eine neue OpenSSL-Bugfix-Version veröffentlicht wird, muss der Zabbix Agent neu kompiliert werden.
- Ohne die Option `no-shared` verwendet Zabbix zur Laufzeit OpenSSL-DLLs. Das bedeutet, dass OpenSSL-Updates möglicherweise keine Neukompilierung des Zabbix Agent erfordern; beim Kopieren auf andere Rechner müssen jedoch auch die OpenSSL-DLLs mitkopiert werden.

Weitere Informationen zu anderen OpenSSL-Konfigurationsoptionen finden Sie in der OpenSSL-Dokumentation.

6. Kompilieren Sie OpenSSL und führen Sie 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. Falls einige Tests fehlschlagen, finden Sie Hinweise 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, Header-Dateien, aber 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 .
```

Zabbix Agent kompilieren

1. Laden Sie das Zabbix-Quellarchiv herunter und entpacken Sie es nach `C:\Zabbix\zabbix-8.0.0`.

Wenn Sie ein Quellarchiv aus dem rohen Quellcode-Repository erzeugen müssen (z. B. um benutzerdefinierte Patches anzuwenden oder aus dem neuesten Quellcode zu bauen), führen Sie die folgenden Befehle auf einem **Linux**-Rechner aus:

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

Dadurch wird ein Quellarchiv erstellt, das anschließend auf einen Windows-Rechner kopiert werden kann.

2. Wechseln Sie in das Zabbix-Build-Verzeichnis 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:

```
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
```

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 zu starten, kopieren Sie `zabbix_agent.exe` und seine Konfigurationsdatei in einen dedizierten Ordner (z. B. `C:\Zabbix\agent`) und starten Sie dann 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.win.conf C:\Zabbix\agent\

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

macOS-Agent-Installation aus PKG

Übersicht

Der Zabbix Agent kann unter macOS mit PKG-Installer-Paketen installiert werden, die hier heruntergeladen werden können.

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:

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

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 gemeinsam genutzte Version der 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 den 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
```

4 Upgrade

Vor dem Upgrade wird dringend empfohlen, alle relevanten Hinweise zum Upgrade zu lesen.

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 (mit Paketen)
- Debian/Ubuntu (mit Paketen)
- Upgrade mit Containern
- Upgrade aus Quellen

Verwandte Anleitungen:

- Für Server in einem Hochverfügbarkeits-Cluster (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 aber 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 auch Proxys, die **nicht älter** sind als die vorherige LTS-Version des Zabbix-Servers, jedoch mit eingeschränkter Funktionalität (Datenerfassung, Ausführung von Remote-Befehlen, sofortigen Datenpunkt-Wertprüfungen). Die Konfigurationsaktualisierung ist ebenfalls deaktiviert, und veraltete Proxys funktionieren nur mit alter Konfiguration.

> **Attention:**
> Proxys, die älter sind als die vorherige LTS-Release-Version des Zabbix-Servers oder neuer als die Major-Version des Zabbix-Servers, werden nicht unterstützt. Der Zabbix-Server ignoriert Daten von nicht unterstützten Proxys, und jede Kommunikation mit dem Zabbix-Server schlägt mit einer Warnung fehl. Weitere Informationen finden Sie unter Versionskompatibilität.

Um Ausfallzeiten und Datenverlust während des Upgrades zu minimieren, wird empfohlen, zuerst den Zabbix-Server zu stoppen, zu aktualisieren und wieder zu starten und anschließend die Zabbix-Proxys nacheinander zu stoppen, zu aktualisieren und wieder zu starten. Während der Ausfallzeit des Servers setzen laufende Proxys die Datenerfassung fort. Sobald der Server wieder läuft, senden veraltete Proxys die Daten an den neueren Server (die Proxy-Konfiguration wird jedoch nicht aktualisiert) und bleiben teilweise funktionsfähig. Benachrichtigungen über Probleme während der Ausfallzeit des Zabbix-Servers werden erst nach dem Start des aktualisierten Servers erzeugt.

Wenn der Zabbix Proxy zum ersten Mal gestartet wird und die SQLite-Datenbankdatei fehlt, erstellt der Proxy sie automatisch.

Beachten Sie, dass der Zabbix Proxy, wenn er SQLite3 verwendet und beim Start erkennt, dass die Version der vorhandenen Datenbankdatei älter ist als erforderlich, **die Datenbankdatei automatisch löscht** und eine neue erstellt. Daher gehen die in der SQLite-Datenbankdatei gespeicherten Verlaufsdaten verloren. Wenn die Version des Zabbix Proxy älter ist als die Version der Datenbankdatei, protokolliert Zabbix einen Fehler und wird beendet.

Abhängig von der Größe der Datenbank kann das Upgrade der Datenbank auf Version 8.0 lange dauern.

Hinweise zum Upgrade

Ein direktes Upgrade auf Zabbix 8.0.x wird ab Version **2.0**.x unterstützt. Informationen zum Upgrade von früheren Versionen finden Sie in der Zabbix-Dokumentation für 2.0 und frühere Versionen.

> **Note:**
> Bitte beachten Sie, dass nach dem Upgrade einige Integrationen von Drittanbieter-Software in Zabbix beeinträchtigt sein könnten, 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	Mindestanforderungen an Datenbankversionen erhöht. Erforderliche Mindestversion von PHP von 8.0.0 auf 8.2.0 erhöht. Das Zeichen % wurde zur Liste `UnsafeUserParameters` für Zabbix Agent und Zabbix Agent 2 hinzugefügt.
7.2.x	Für: Zabbix 7.4 Zabbix 8.0	Unterstützung der PCRE-Bibliothek eingestellt.
7.0.x	Für: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Unterstützung von Oracle DB eingestellt.
6.4.x	Für: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Erforderliche Mindestversion von PHP von 7.4.0 auf 8.0.0 erhöht. Asynchrone Poller für Agent-, HTTP-Agent- und SNMP walk[oid]-Prüfungen. Separate Datenbanktabelle für Proxys. Standardpfad für die Konfigurationsdatei des Windows-Agent geändert. Oracle DB als veraltet markiert. Alter numerischer (Float-)Werttyp entfernt.
6.2.x	Für: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Erforderliche Mindestversion von MySQL von 8.0.0 auf 8.0.30 erhöht. Die Bibliothek 'libevent_pthreads' ist für Zabbix Server/Proxy erforderlich. Beim ersten Start nach einem Upgrade verwirft Zabbix Proxy mit SQLite3 automatisch die alte Version der Datenbank (mit der gesamten Historie) 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	Erforderliche Mindestversion von PHP von 7.2.5 auf 7.4.0 erhöht. Service-Monitoring grundlegend überarbeitet. Deterministische Auslöser müssen während des Upgrades erstellt werden. Wenn binäres Logging 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 Database creation scripts.
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	Mindestanforderungen an Datenbankversionen erhöht. Server/Proxy startet nicht bei veralteter Datenbank. Audit-Log-Einträge gehen aufgrund einer Änderung der Datenbankstruktur 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	Mindestanforderungen an Datenbankversionen erhöht. Aggregierte Datenpunkte als separater Typ entfernt.
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	Erforderliche Mindestversion von PHP von 7.2.0 auf 7.2.5 erhöht. Passwort-Hashing-Algorithmus von MD5 auf bcrypt geändert.

Upgrade von	Vollständige Upgrade-Hinweise lesen	Wichtigste Änderungen zwischen den Versionen
4.4.x	Für:	Unterstützung von IBM DB2 eingestellt.
	Zabbix 5.0	Erforderliche Mindestversion von PHP von 5.4.0 auf 7.2.0 erhöht.
	Zabbix 5.2	Mindestanforderungen an Datenbankversionen erhöht.
	Zabbix 5.4	Zabbix-PHP-Dateiverzeichnis geändert.
	Zabbix 6.0	
	Zabbix 6.2	
	Zabbix 6.4	
	Zabbix 7.0	
	Zabbix 7.2	
	Zabbix 7.4	
	Zabbix 8.0	
4.2.x	Für:	Medientypen Jabber und Ez Texting entfernt.
	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	
4.0.x LTS	Für:	Ältere Proxys können keine Daten mehr an einen aktualisierten Server melden.
	Zabbix 4.2	
	Zabbix 4.4	Neuere Agenten können nicht mehr mit einem älteren Zabbix Server arbeiten.
	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	
3.4.x	Für:	Die Bibliotheken ’libpthread’ und ’zlib’ sind jetzt obligatorisch.
	Zabbix 4.0	Unterstützung für das Plain-Text-Protokoll eingestellt, und ein Header ist obligatorisch.
	Zabbix 4.2	
	Zabbix 4.4	Zabbix Agenten vor Version 1.4 werden nicht mehr unterstützt.
	Zabbix 5.0	Der Parameter Server in der Konfiguration eines passiven Proxys ist jetzt obligatorisch.
	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	

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	Unterstützung von SQLite als Backend-Datenbank für Zabbix Server/Frontend eingestellt. Perl Compatible Regular Expressions (PCRE) werden anstelle von erweitertem POSIX unterstützt. Die Bibliotheken 'libpcre' und 'libevent' sind für Zabbix Server obligatorisch. Prüfungen von Exit-Codes hinzugefügt für Benutzerparameter, Remote-Befehle und system.run[]-Datenpunkte ohne das Flag 'nowait' sowie für vom Zabbix Server ausgeführte Skripte. Zabbix Java gateway muss aktualisiert werden, um die neue Funktionalität 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 je nach Größe der History-Tabelle 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	Erforderliche Mindestversion von PHP von 5.3.0 auf 5.4.0 erhöht. 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	Knotenbasiertes verteiltes Monitoring 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	Erforderliche Mindestversion von PHP von 5.1.6 auf 5.3.0 erhöht. Für den ordnungsgemäßen Betrieb des Servers ist eine MySQL-Datenbank mit Groß-/Kleinschreibung erforderlich; der Zeichensatz utf8 und die Sortierung utf8_bin sind erforderlich, damit Zabbix Server mit einer MySQL-Datenbank ordnungsgemäß funktioniert. Siehe database creation scripts. 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 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/apache2/conf-enabled/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

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 herunterladen, damit der Paketmanager während des Upgrade-Vorgangs nicht versehentlich eine veraltete Version wiederverwendet.

Installieren Sie anschließend das neueste Repository-Konfigurationspaket, um die Kompatibilität mit den neuesten Paketen sicherzustellen und aktuelle Sicherheitspatches oder Fehlerbehebungen einzuschließen.

Unter **Debian 12** führen Sie Folgendes 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:
Ersetzen Sie bei älteren Debian-Versionen den obigen Link durch den passenden aus dem Zabbix repository. Beachten Sie jedoch, dass Pakete für diese Versionen möglicherweise nicht alle Zabbix-Komponenten enthalten. Wenn Sie diese Komponenten aus Paketen aktualisieren möchten, sollten Sie ein Upgrade Ihres Betriebssystems in Betracht ziehen. Eine Liste der enthaltenen Komponenten finden Sie unter Zabbix packages.

Unter **Ubuntu 24.04** führen Sie Folgendes 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 Folgendes 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:
Ersetzen Sie bei älteren Ubuntu-Versionen den obigen Link durch den passenden aus dem Zabbix repository. Beachten Sie jedoch, dass Pakete für diese Versionen möglicherweise nicht alle Zabbix-Komponenten enthalten. Wenn Sie diese Komponenten aus Paketen aktualisieren möchten, sollten Sie ein Upgrade Ihres Betriebssystems in Betracht ziehen. Eine Liste der enthaltenen Komponenten finden Sie unter Zabbix packages.

Möglicherweise wird Ihnen eine Eingabeaufforderung zur Konfiguration des Zabbix-Repositorys 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 der Zabbix-Repository-Konfiguration des Paketbetreuers zu installieren.

Aktualisieren Sie dann die Repository-Informationen:

```
apt update
```

5 Zabbix-Komponenten aktualisieren

Um Zabbix-Komponenten zu aktualisieren, führen Sie Folgendes aus:

```
apt install --only-upgrade zabbix-server-mysql zabbix-frontend-php 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 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 Server (oder Proxy) 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 zum Beispiel `D` ein, um die aktuelle und die neue Konfiguration zu vergleichen, und entscheiden Sie dann, ob Sie die Version des Paketbetreuers installieren möchten (`Y` oder `I`).

Führen Sie dann Folgendes aus, um das Zabbix Frontend mit Apache zu aktualisieren und Apache neu zu starten:

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

6 Konfigurationsparameter der Komponente überprüfen

Prüfen Sie die relevanten Upgrade-Hinweise, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Neue optionale Parameter 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 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

Bevor Sie mit dem Upgrade fortfahren, aktualisieren Sie Ihr aktuelles Repository-Paket auf die neueste Version, um die Kompatibilität mit den neuesten Paketen sicherzustellen und aktuelle Sicherheitspatches oder Fehlerbehebungen einzuschließen.

Unter **RHEL 10** führen Sie Folgendes aus:

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

Unter **RHEL 9** führen Sie Folgendes aus:

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/9/noarch/zabbix-release-latest.el9.noarch.rpm
```

> **Note:**
> Für ältere RHEL-Versionen oder deren Derivate ersetzen Sie den obigen Link durch den passenden aus dem Zabbix-Repository. Beachten Sie jedoch, dass Pakete für diese Versionen möglicherweise nicht alle Zabbix-Komponenten enthalten. Wenn Sie diese Komponenten aus Paketen aktualisieren möchten, ziehen Sie ein Upgrade Ihres Betriebssystems in Betracht. Eine Liste der enthaltenen Komponenten finden Sie unter Zabbix-Pakete.

Bereinigen Sie anschließend den Cache des Paketmanagers `dnf` (einschließlich Headern, Metadaten und Paketdateien, die bei früheren Installationen oder Aktualisierungen heruntergeladen wurden):

```
dnf clean all
```

Beim nächsten `dnf`-Vorgang lädt `dnf` frische Metadaten aus den Repositories herunter, da die alten Metadaten gelöscht wurden.

Siehe auch: Bekannte Probleme zum Aktualisieren des Repository-Konfigurationspakets unter RHEL.

5 Zabbix-Komponenten aktualisieren

Um Zabbix-Komponenten zu aktualisieren, führen Sie Folgendes aus:

```
dnf install zabbix-server-mysql zabbix-web-mysql 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 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 `dnf install zabbix-agent2` kann zu einem Fehler führen. Weitere Informationen finden Sie unter *Bekannte Probleme*.

Führen Sie dann Folgendes aus, um das Zabbix Frontend mit Apache zu aktualisieren und Apache neu zu starten:

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

6 Konfigurationsparameter der Komponente überprüfen

Prüfen Sie die entsprechenden Upgrade-Hinweise, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Neue optionale Parameter 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

Überblick

Dieser Abschnitt beschreibt die Schritte, die für ein erfolgreiches Upgrade von Zabbix **7.4**.x auf Zabbix **8.0**.x unter Verwendung der offiziellen Zabbix-Quellen erforderlich sind.

> **Warning:**
> Lesen Sie vor dem Upgrade unbedingt die entsprechenden **Hinweise zum Upgrade**!

Möglicherweise möchten Sie auch die Anforderungen für 8.0 prüfen.

> **Note:**
> Es kann hilfreich sein, während des Upgrades zwei parallele SSH-Sitzungen zu verwenden: Führen Sie in einer die Upgrade-Schritte aus und überwachen Sie in der anderen die Server-/Proxy-Protokolle. Führen Sie beispielsweise `tail -f zabbix_server.log` oder `tail -f zabbix_proxy.log` in der zweiten SSH-Sitzung aus, damit Ihnen die neuesten Protokolleinträge und mögliche Fehler in Echtzeit angezeigt werden. Dies kann für Produktivumgebungen 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 bestehende Zabbix-Datenbank

Dies ist ein sehr wichtiger Schritt. Stellen Sie sicher, dass Sie eine Sicherungskopie Ihrer Datenbank haben. Sie hilft, falls das Upgrade fehlschlägt (zu wenig Speicherplatz, Stromausfall, ein anderes unerwartetes Problem).

3 Konfigurationsdateien, PHP-Dateien und Zabbix-Binärdateien sichern

Erstellen Sie eine Sicherungskopie der Zabbix-Binärdateien, der Konfigurationsdateien und des PHP-Dateiverzeichnisses.

4 Neue Server-Binärdateien installieren

Verwenden Sie diese Anweisungen, um den Zabbix Server aus den Quellen zu kompilieren.

5 Konfigurationsparameter des Servers überprüfen

Stellen Sie sicher, dass Sie die Upgrade-Hinweise prüfen, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Neue optionale Parameter finden Sie auf der Seite Was ist neu.

6 Neue Zabbix-Binärdateien starten

Starten Sie die neuen Binärdateien. Prüfen Sie die Logdateien, um festzustellen, ob die Binärdateien erfolgreich gestartet wurden.

Der Zabbix Server aktualisiert die Datenbank automatisch. Beim Start meldet der Zabbix Server die aktuelle (obligatorische und optionale) sowie die erforderliche Datenbankversion. Wenn die aktuelle obligatorische Version älter ist als die erforderliche Version, führt der Zabbix Server automatisch die erforderlichen Datenbank-Upgrade-Patches aus. Der Beginn und der Fortschrittsstand (Prozentsatz) des Datenbank-Upgrades werden in die Logdatei des Zabbix Server geschrieben. Wenn das Upgrade abgeschlossen ist, wird eine Meldung „database upgrade fully completed“ in die Logdatei geschrieben. Falls einer der Upgrade-Patches fehlschlägt, startet der Zabbix Server nicht. Der Zabbix Server startet auch dann nicht, wenn die aktuelle obligatorische Datenbankversion neuer ist als die erforderliche. Der Zabbix Server startet nur, wenn die aktuelle obligatorische Datenbankversion der erforderlichen obligatorischen Version entspricht.

```
8673:20161117:104750.259 aktuelle Datenbankversion (obligatorisch/optional): 03040000/03040000
8673:20161117:104750.259 erforderliche obligatorische 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 genügend freier Speicherplatz vorhanden ist.

7 Neue Zabbix-Weboberfläche installieren

Die mindestens erforderliche PHP-Version ist 8.2.0. Aktualisieren Sie sie bei Bedarf und folgen Sie den Installationsanweisungen.

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.

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.

Neue optionale Parameter 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 wird die Datenbank automatisch aktualisieren. Die Datenbankaktualisierung erfolgt ähnlich wie beim Starten des Zabbix Server.

Agent-Aktualisierungsprozess

> **Attention:**
> Die Aktualisierung von Agenten ist nicht zwingend erforderlich. Sie müssen Agenten nur aktualisieren, wenn dies erforderlich ist, um auf die neuen Funktionen zuzugreifen.

Das in diesem Abschnitt beschriebene Aktualisierungsverfahren kann sowohl für die Aktualisierung des Zabbix-Agenten als auch des Zabbix-Agenten 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 Agenten von der Zabbix-Download-Seite herunterladen.

4 Konfigurationsparameter des Agent überprüfen

Stellen Sie sicher, dass Sie die Upgrade-Hinweise prüfen, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Neue optionale Parameter 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 Nebenversionen

Beim Upgrade zwischen Nebenversionen 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 Hauptversionen. Der einzige Unterschied besteht darin, dass beim Upgrade zwischen Nebenversionen 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.

Für das Upgrade einzelner Zabbix-Komponenten-Images und von Docker-Compose-Dateien stehen separate Anleitungen zur Verfügung.

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 zum Zweck des Upgrades 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 Database creation scripts.
Beachten Sie, dass die Variable bei Ausführung über eine Konsole nur vorübergehend gesetzt wird und beim Neustart eines Docker verworfen wird. Lassen Sie in diesem Fall Ihren SQL-Dienst weiterlaufen und stoppen Sie nur den zabbix-server-Dienst, 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 ein Upgrade auf Version 8.0 recht lange dauern.

Upgrade des Zabbix-Images

Die unten aufgeführten Schritte können verwendet werden, um jede Zabbix-Komponente zu aktualisieren. Ersetzen Sie `zabbix-server-mysql` durch den Namen des erforderlichen Komponenten-Images.

1. Aktuelle Image-Version prüfen:

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

2. Gewünschte Image-Version abrufen, zum Beispiel:

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

`zabbix/zabbix-server-mysql:alpine-8.0-latest` lädt die neueste veröffentlichte Minor-Version von Zabbix server 8.0 mit MySQL-Unterstützung auf Basis von Alpine Linux herunter. Ersetzen Sie sie durch die Kombination aus Docker-Repository-Namen und Tags, die Sie benötigen. Eine Liste der verfügbaren Optionen finden Sie unter Installation from containers.

3. Container stoppen:

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

4. Container entfernen:

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

5. Starten Sie den aktualisierten Container, indem Sie den Befehl `docker run` ausführen, gefolgt von zusätzlichen Argumenten zur Angabe der erforderlichen environment variables und/oder mount points.

Konfigurationsbeispiele

Zabbix server mit MySQL:

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

Zabbix server mit PostgreSQL:

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

MySQL-Server:

```
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 \
```

Weitere Konfigurationsbeispiele, einschließlich Beispielen für andere Zabbix-Komponenten, sind auf der Seite Installation from containers verfügbar.

6. Aktualisierung überprüfen:

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

Compose-Dateien

Befolgen Sie die Upgrade-Anweisungen in diesem Abschnitt, wenn Sie Zabbix mit einer Compose-Datei installiert haben.

1. Prüfen Sie die aktuelle Image-Version:

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

2. Rufen Sie die neuesten Aktualisierungen aus dem GitHub-Repository ab und wechseln Sie zum erforderlichen Branch:

```
git pull
git checkout 8.0
```

3. Starten Sie die Zabbix-Komponenten mit der neuen Compose-Datei:

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

4. Überprüfen Sie das Update:

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

Weitere Informationen, einschließlich Listen der unterstützten Umgebungsvariablen und Volume-Mount-Points, finden Sie unter Installation aus Containern.

5 Upgrade-Hinweise für Zabbix 8.0

Diese Hinweise gelten für das Upgrade von Zabbix 7.4.x auf Zabbix 8.0.0.

Alle Hinweise sind in folgende Gruppen unterteilt:

- **Inkompatible Änderungen** - Änderungen, die bestehende Installationen beeinträchtigen können, sowie andere wichtige Informationen im Zusammenhang mit dem Upgrade-Prozess
- **Sonstiges** - alle übrigen Informationen, die die Änderungen der Zabbix-Funktionalität beschreiben

Siehe auch:

- Upgrade-Verfahren für alle relevanten Informationen zum Upgrade von Versionen vor Zabbix 8.0.0;
- Upgrade eines HA-Clusters für Anweisungen zum Upgrade von Servern in einem **High-Availability**-(HA)-Cluster.

Inkompatible Änderungen Datenbankversionen

Die minimal erforderlichen Datenbankversionen wurden angehoben:

- MySQL/Percona: 8.0.30 → 8.4.0
- MariaDB: 10.5.00 → 10.11.00
- PostgreSQL: 13.0 → 15.0
- TimescaleDB: 2.13.0 → 2.20.0

Plugins

Das Ceph-Plugin für Zabbix Agent 2 ist jetzt ein ladbares Plugin und erfordert zusätzliche Installationsschritte. Weitere Informationen finden Sie in der readme des Ceph-Plugins.

Das Schema `tcp://` ist im MongoDB-Plugin veraltet, wird jedoch aus Gründen der Abwärtskompatibilität mit bestehenden Konfigurationen beibehalten.

Erforderliche Mindestversion von PHP

Die erforderliche Mindestversion von PHP wurde von 8.0.0 auf 8.2.0 angehoben.

Erweiterte UnsafeUserParameters-Liste

Das Zeichen % wurde zur Liste `UnsafeUserParameters` für Zabbix Agent und Zabbix Agent 2 hinzugefügt.

Sonstiges Veraltete Makros entfernt

Die Unterstützung für die folgenden integrierten Makros wurde nun eingestellt:

Entfernt	Stattdessen verwenden
{ACK.DATE}	{EVENT.UPDATE.DATE}
{ACK.MESSAGE}	{EVENT.UPDATE.MESSAGE}
{ACK.TIME}	{EVENT.UPDATE.TIME}
{EVENT.ACK.HISTORY}	{EVENT.UPDATE.HISTORY}
{HOSTNAME<1-9>}	{HOST.HOST}
{IPADDRESS<1-9>}	{HOST.IP}
{PROFILE.*}	{INVENTORY.*}
{TRIGGER.COMMENT}	{TRIGGER.DESCRIPTION}
{TRIGGER.KEY}	{ITEM.KEY}
{STATUS}	{TRIGGER.STATUS}
{USER.ALIAS}	{USER.USERNAME}

JSON- und XML-Validierung aus HTTP-Agent-Datenpunkten entfernt

Die JSON- und XML-Validierung wurde aus dem Feld *Request body* in HTTP-Agent-Datenpunkten entfernt. Das Feld prüft nicht mehr, ob der Wert gültiges XML oder JSON ist, und die Auswahl von *XML data* als *Request body type* erfordert nicht länger die Bibliothek libxml2.

Deaktivierte Links in überwachten Hosts ausgeblendet

Deaktivierte Links *Graphs*, *Dashboards* und *Web* werden in der Liste der überwachten Hosts nicht mehr angezeigt.

JSON-Datentyp

Zabbix unterstützt jetzt JSON als Datentyp für Werte von Datenpunkten. Wenn Sie TimescaleDB verwenden, muss die neue Hypertabelle `history_json` (die zum Speichern von JSON-Werten verwendet wird) manuell konfiguriert werden. Wenn Sie Elasticsearch verwenden, enthält der Standardwert des Zabbix-Server-Konfigurationsparameters `HistoryStorageTypes` jetzt `json`.

Optionsfeld im Formular „Neuer Tag-Filter“ entfernt

Das Optionsfeld, bei dem beim Erstellen eines neuen Tag-Filters zwischen *Alle Tags* und *Tag-Liste* gewählt werden musste, wurde entfernt.

Datenpunkt-Werte als HTML mit iframes angezeigt

Datenpunkt-Daten im Widget Datenpunkt-Verlauf werden jetzt, wenn sie als HTML-formatierter Text angezeigt werden, in iframes isoliert.

Template compatibility with Host Wizard

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

Upgrade-Hinweise für Zabbix 8.0.x

Diese Seite enthält gesammelte Upgrade-Hinweise für Minor-Releases der Hauptversion von Zabbix.

Siehe auch die Upgrade-Hinweise der Hauptversion.

Hinweise zum Upgrade für 8.0.1 Diese Version ist noch nicht veröffentlicht.

Änderungen an Vorlagen

Diese Seite listet alle Änderungen an den Standardvorlagen auf, die mit Zabbix ausgeliefert werden.

Beim Upgrade von Zabbix werden vorhandene Vorlagen nicht automatisch aktualisiert, um das Überschreiben benutzerdefinierter Änderungen zu vermeiden. Informationen zum Upgrade von Vorlagen oder zum Hinzufügen neuer Vorlagen finden Sie unter Template upgrade.

> **Note:**
> Bitte beachten Sie, dass seit Zabbix 6.0 alle Vorlagen einem aktualisierten Format folgen, was sich auf den Import von Vorlagen aus Versionen vor 6.0 auswirken kann. Weitere Informationen finden Sie unter Template changes in 6.0.

Änderungen in 8.0.0

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 Hardwarekonfigurationen

Die Tabelle enthält Beispiele für Hardwarekonfigurationen unter der Annahme einer **Linux/BSD/Unix**-Plattform.

Dies sind Beispiele für Größen- und Hardwarekonfigurationen 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 ihrer Produktionsumgebung bereitstellen.

Installationsgröße	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] 1 Metrik = 1 Datenpunkt + 1 Auslöser + 1 Diagramm
 [2] Beispiel mit allgemeinen Amazon-EC2-Instanzen. Bei Verwendung der ARM64- oder x86_64-Architektur sollte während der Evaluierung und des Testens der Zabbix-Installation vor der Installation in der Produktionsumgebung ein geeigneter Instanztyp wie Compute-/Memory-/Storage-optimiert ausgewählt werden.

Note:
Die tatsächliche Konfiguration hängt sehr stark von der Anzahl aktiver Datenpunkte und Aktualisierungsintervallen ab (siehe Abschnitt Datenbankgröße auf dieser Seite für Details). Für große Installationen wird dringend empfohlen, die Datenbank 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	
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.

Plattform	Server	Agent	Agent 2	Kommentare
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 umgebende Software von Drittanbietern

Falls als obligatorisch angegeben, ist die erforderliche Software/Bibliothek zwingend notwendig. Optionale Komponenten werden zur Unterstützung bestimmter spezifischer Funktionen benötigt.

Software	Pflichtstatus	Unterstützte Versionen	Kommentare
MySQL/Percona	Eine von	8.4.0-9.5.X	Erforderlich, wenn MySQL (oder Percona) als Zabbix-Backend-Datenbank verwendet wird. Die InnoDB-Engine ist erforderlich. Wir empfehlen die Verwendung der Bibliothek C API (libmysqlclient) zum Erstellen von Server/Proxy.
MariaDB		10.11.00-12.0.X	Die InnoDB-Engine ist erforderlich. Die empfohlene Version ist 11.4. Wir empfehlen die Verwendung der Bibliothek MariaDB Connector/C zum Erstellen von Server/Proxy. Siehe auch: Mögliche Deadlocks mit MariaDB und Zugriff auf UI-Elemente mit MariaDB 10.5.1–10.5.9.
PostgreSQL		15.0-18.X	Erforderlich, wenn PostgreSQL als Zabbix-Backend-Datenbank verwendet wird. Abhängig von der Größe der Installation kann es erforderlich sein, die PostgreSQL-Konfigurationseigenschaft *work_mem* zu erhöhen (4 MB ist der Standardwert), damit die von der Datenbank für bestimmte Operationen verwendete Speichermenge ausreichend ist und die Ausführung von Abfragen nicht zu viel Zeit in Anspruch nimmt.
TimescaleDB for PostgreSQL		2.20.X-2.25.X	Erforderlich, wenn TimescaleDB als PostgreSQL-Datenbankerweiterung verwendet wird. Stellen Sie sicher, dass Sie die TimescaleDB Community Edition installieren, die Komprimierung unterstützt. Beachten Sie, dass PostgreSQL 15 seit TimescaleDB 2.10 unterstützt wird. Weitere Informationen zur Versionskompatibilität von PostgreSQL und TimescaleDB finden Sie auch in der TimescaleDB-Dokumentation.

Software	Pflichtstatus	Unterstützte Versionen	Kommentare
SQLite	Optional	3.3.5-3.34.X	SQLite wird nur mit Zabbix-Proxys unterstützt. Erforderlich, wenn SQLite als Zabbix-Proxy-Datenbank verwendet wird.
Elasticsearch		7.X	Elasticsearch wird nur mit Zabbix-Servern unterstützt und ausschließlich zum Speichern von Verlaufsdaten verwendet. Die Unterstützung von Elasticsearch ist derzeit experimentell. Siehe auch die erforderliche Software für Server/Proxy.
smartmontools		7.1 oder höher	Erforderlich für Zabbix Agent 2.
who			Erforderlich für das Plugin zur 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 Teil von 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 die best-mögliche Nutzung Datenbanken, die aus den offiziellen Repositories der Datenbankentwickler installiert wurden.

Frontend

Die minimal unterstützte Bildschirmbreite für das Zabbix Frontend beträgt 1200px.

Wenn etwas als obligatorisch angegeben ist, ist die erforderliche Software/Bibliothek zwingend notwendig. Optionale Komponenten werden zur Unterstützung bestimmter Funktionen benötigt.

Software	Pflichtstatus	Unterstützte Versionen	Kommentare
PHP	Ja	8.2.0 - 8.4.X	
Apache	Eine von	2.4 oder höher	
Nginx		1.20 oder höher	
MySQL	Eine von	Siehe Third-party external surround-ing software	
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 mögliche Textüberlappungen in Diagrammen bei einigen Frontend-Sprachen zu vermeiden.
libxml		2.6.15 oder höher	php-xml (falls vom Distributor als separates Paket bereitgestellt)

Software	Pflichtstatus	Unterstützte Versionen	Kommentare
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 LDAP-Authentifizierung.
openssl			php-openssl; erforderlich für SAML-Authentifizierung.
gettext			php-gettext (*--with-gettext*); erforderlich für Übersetzungen.
cURL		7.19.4 oder höher	php-curl; erforderlich für Duo Universal Prompt MFA und SMTP-Authentifizierung.

Frontend-Bibliotheken von Drittanbietern, die mit Zabbix ausgeliefert werden:

Bibliothek	Pflichtstatus	Mitgelieferte 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 Benutzeroberflächen-Interaktionen, Effekten, Widgets und Themes, die auf jQuery aufbaut.
SAML PHP Toolkit		4.3.1	Ein PHP-Toolkit, das Unterstützung für SAML-2.0-Authentifizierung hinzufügt, damit eine Anmeldung bei Zabbix möglich ist.
Symfony Yaml Component		5.1.0	Fügt Unterstützung für den Export und Import von Zabbix-Konfigurationselementen im YAML-Format hinzu.

Note:
Zabbix kann möglicherweise auch mit früheren Versionen von Apache, MySQL und PostgreSQL funktionieren.

Attention:
Für andere Schriftarten als die standardmäßige DejaVu-Schriftart kann die PHP-Funktion imagerotate erforderlich sein. Falls sie fehlt, werden diese Schriftarten bei der Anzeige eines Diagramms möglicherweise nicht korrekt dargestellt. Diese Funktion ist nur verfügbar, wenn PHP mit gebündeltem GD kompiliert wurde, was bei Debian und anderen Distributionen nicht der Fall ist.

Bibliotheken von Drittanbietern, die zum Schreiben und Debuggen von Zabbix-Frontend-Code verwendet werden:

Bibliothek	Pflichtstatus	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-Framework für Unit-Tests zum Testen des Zabbix Frontend.
SASS		3.4.22	Eine Präprozessor-Skriptsprache, die in Cascading Style Sheets (CSS) interpretiert und 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

Falls als obligatorisch angegeben, ist die erforderliche Software/Bibliothek zwingend notwendig. Optionale Komponenten werden zur Unterstützung bestimmter spezifischer Funktionen benötigt.

Anforderung	Pflichtstatus	Beschreibung
libpcre2	Ja	Die PCRE2-Bibliothek ist für die Unterstützung von Perl Compatible Regular Expression (PCRE) erforderlich. PCRE2 v10.x wird unterstützt.
libevent		Erforderlich für die Interprozesskommunikation. Version 2.0.10 oder höher.
libevent-pthreads		Erforderlich für die Interprozesskommunikation.
libpthread		Erforderlich für die Unterstützung von Mutexen und Lese-/Schreibsperren (kann Teil von libc sein).
libresolv		Erforderlich für die DNS-Auflösung (kann Teil von libc sein).
libiconv		Erforderlich für die Textkodierung/-formatkonvertierung (kann Teil von libc sein). Für Zabbix Server unter Linux obligatorisch.
libz		Erforderlich für die Unterstützung von Komprimierung.
libm		Mathematikbibliothek. Nur für Zabbix Server erforderlich.
libmysqlclient	Eines von	Erforderlich, wenn MySQL verwendet wird.
libmariadb		Erforderlich, wenn MariaDB verwendet wird.
libpq5		Erforderlich, wenn PostgreSQL verwendet wird; die Version von *libpq5* muss der verwendeten PostgreSQL-Datenbankversion entsprechen oder höher sein.
libsqlite3		Erforderlich, wenn SQLite verwendet wird. Nur für Zabbix Proxy erforderlich.
libOpenIPMI	Nein	Erforderlich für die IPMI-Unterstützung. Nur für Zabbix Server erforderlich.
libssh2 oder *libssh*		Erforderlich für SSH-Prüfungen. Version 1.8.0 oder höher (libssh2); 0.9.0 oder höher (libssh).
libcurl		Erforderlich für die folgenden Funktionen: - Web-Überwachung, VMware-Überwachung und [HTTP-Agent]-Datenpunkte(/manual/config/items/itemtypes/http) (für alle: Version 7.19.1 oder höher); - Zabbix Agent-Datenpunkte vom Typ web.page.* (Version 7.19.1 oder höher; siehe auch die Anforderungen für Agent); - SMTP-Authentifizierung (Basic: Version 7.20.0 oder höher; OAuth: Version 7.33 oder höher; siehe auch die Anforderungen für Frontend); - Elasticsearch (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 Zabbix Server/Proxy neu. Für die SMTP-Authentifizierung verwenden Sie zur Laufzeit das Paket `libcurl-full`.
libxml2		Erforderlich für VMware-Überwachung und XML-XPath-Preprocessing.
net-snmp		Erforderlich für die SNMP-Unterstützung. Version 5.3.0 oder höher. Die Unterstützung starker Verschlüsselungsprotokolle (AES192/AES192C, AES256/AES256C) ist ab net-snmp-Bibliothek 5.8 verfügbar; auf Systemen auf Basis von RHEL 8+ wird empfohlen, net-snmp 5.8.15 oder neuer zu verwenden.
libunixodbc		Erforderlich für die Datenbanküberwachung.
libgnutls oder *libopenssl*		Erforderlich bei Verwendung von Verschlüsselung. Mindestversionen: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Erforderlich für die LDAP-Unterstützung.
fping		Erforderlich für ICMP-Ping-Datenpunkte.
c-ares		Erforderlich für die asynchrone DNS-Auflösung, wenn Zabbix mit der Option `--with-ares` konfiguriert ist. Andernfalls wird *libevent* verwendet. Mindestversion: 1.16.0. DNS-Abfrage-Caching (Abfrage-Cache standardmäßig aktiviert) erfordert c-ares 1.26.0 oder neuer.

Agent

Anforderung	Verbindlich	Beschreibung
libpcre2	Ja	Die PCRE2-Bibliothek wird für die Unterstützung von Perl-kompatiblen regulären Ausdrücken (PCRE) benötigt. PCRE2 v10.x wird unterstützt. Erforderlich für die Log-Überwachung. Auch unter Windows erforderlich.
libpthread		Erforderlich für die Unterstützung von Mutexen und Lese-/Schreibsperren (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/-konvertierung nach UTF-8 in Log-Datenpunkten, Dateiinhalten sowie Datenpunkten vom Typ file regex und regmatch (kann Teil von libc sein). Unter Windows nicht erforderlich.
libgnutls or *libopenssl*	Nein	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 eine grundlegende Funktionalität verfügbar (z. B. web.page.get[http://example.com]). Mit libcurl unterstützt der Agent zusätzliche Funktionen, wie HTTP-URLs mit Zugangsdaten (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

Anforderung	Pflichtstatus	Beschreibung
Go	Ja	Erforderlich, um Zabbix Agent 2 und seine Plugins aus dem Quellcode zu erstellen. Go 1.24.10 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 mit `indirect` markierte Bibliotheken sind Abhängigkeiten anderer erforderlicher Bibliotheken): - Zabbix agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Ja	Die PCRE2-Bibliothek ist für die Unterstützung von Perl Compatible Regular Expression (PCRE) erforderlich. PCRE2 v10.x wird unterstützt. Erforderlich für die Log-Überwachung. Auch unter Windows erforderlich.
libopenssl	Nein	Erforderlich bei Verwendung von Verschlüsselung. Auf UNIX-Plattformen ist OpenSSL 1.0.1 oder höher erforderlich. Die OpenSSL-Bibliothek muss mit aktivierter PSK-Unterstützung erstellt worden 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 aus einem Archiv bezogen haben, sind die erforderlichen Abhängigkeiten bereits im Quellbaum enthalten.

Wenn Sie Zabbix aus dem Paket Ihrer Distribution bezogen haben, werden die erforderlichen Abhängigkeiten bereits durch das 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 die Liste der Bibliotheksversionen, mit denen Java gateway nachweislich funktioniert. Zabbix kann möglicherweise auch mit anderen Versionen dieser Bibliotheken funktionieren.

Die folgende Tabelle listet JAR-Dateien auf, die derzeit im Originalcode mit Java gateway gebündelt sind:

Library	Mandatory status	Bundled version	Comments
android-json	Ja	4.3r1	JSON (JavaScript Object Notation) ist ein leichtgewichtiges Format für den Datenaustausch. Dies ist die mit org.json kompatible Android-Implementierung, die aus dem Android SDK extrahiert wurde.
logback-classic		1.5.16	
logback-core		1.5.16	
slf4j-api		2.0.16	

Java gateway kann entweder mit Oracle Java oder mit dem Open-Source-OpenJDK (Version 1.6 oder neuer) erstellt werden. Die von Zabbix bereitgestellten Pakete werden mit OpenJDK kompiliert. Die folgende Tabelle listet die OpenJDK-Pakete auf, die je Distribution zum Erstellen von Zabbix-Paketen verwendet werden:

Distribution	OpenJDK package
AlmaLinux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 9	`java-11-openjdk-devel` (amd64, arm64: 11.0.18.0.10-3)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4.0.1; arm64: 11.0.20.0.8-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

Standard-Portnummern

Die folgende Liste offener Ports pro Komponente gilt für die Standardkonfiguration:

Zabbix-Komponente	Portnummer	Protokoll	Verbindungstyp
Zabbix Agent	10050	TCP	bei Bedarf

Zabbix-Komponente	Portnummer	Protokoll	Verbindungstyp
Zabbix Agent 2	10050	TCP	bei Bedarf
Zabbix Server	10051	TCP	bei Bedarf
Zabbix Proxy	10051	TCP	bei Bedarf
Zabbix Java gateway	10052	TCP	bei Bedarf
Zabbix Web-Service	10053	TCP	bei Bedarf
Zabbix Frontend	80	HTTP	bei Bedarf
	443	HTTPS	bei Bedarf
Zabbix Trapper	10051	TCP	bei Bedarf

Note:
Die Portnummern sollten in der Firewall geöffnet sein, um die Zabbix-Kommunikation zu ermöglichen. Ausgehende TCP-Verbindungen erfordern in der Regel keine expliziten Firewall-Einstellungen.

Datenbankgröße

Zabbix-Konfigurationsdaten benötigen eine feste Menge an Speicherplatz und wachsen nicht wesentlich.

Die Größe der Zabbix-Datenbank hängt hauptsächlich von diesen Variablen ab, die die Menge der gespeicherten Verlaufsdaten 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 einem Aktualisierungsintervall 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 den Verlauf

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 Verlauf aufbewahren möchten und 50 Werte pro Sekunde empfangen, beträgt die Gesamtzahl der Werte ungefähr (**30***24*3600)* **50** = 129.600.000, also etwa 130 Mio. Werte.

Abhängig von der verwendeten Datenbank-Engine und vom Typ der empfangenen Werte (Gleitkommazahlen, Ganzzahlen, Zeichenfolgen, Protokolldateien usw.) kann der Speicherplatz für einen einzelnen Wert zwischen 40 Byte und mehreren hundert Byte variieren. Normalerweise sind es bei numerischen Datenpunkten etwa 90 Byte pro Wert[2]. In unserem Fall bedeutet das, dass 130 Mio. Werte 130 Mio. * 90 Byte = **10,9GB** Speicherplatz benötigen.

Note:
Die Größe von Text-/Log-Datenpunktwerten lässt sich nicht exakt vorhersagen, aber Sie können mit etwa 500 Byte pro Wert rechnen.

- Housekeeper-Einstellung für Trends

Zabbix speichert für jeden Datenpunkt einen 1-Stunden-Satz aus Maximal-/Minimal-/Durchschnitts-/Anzahlwerten in der Tabelle **trends**. Diese Daten werden für Trends und Diagramme über lange Zeiträume verwendet. Der Zeitraum von einer Stunde kann nicht angepasst werden.

Die Zabbix-Datenbank benötigt, abhängig vom Datenbanktyp, etwa 90 Byte pro Gesamtsatz. Angenommen, wir möchten Trenddaten 5 Jahre lang aufbewahren. Werte für 3000 Datenpunkte benötigen 3000*24*365* **90** = **2,2GB** pro Jahr oder **11GB** für 5 Jahre.

- Housekeeper-Einstellungen für Ereignisse

Jedes Zabbix-Ereignis benötigt ungefähr 250 Byte Speicherplatz[1]. Es ist schwer abzuschätzen, wie viele Ereignisse Zabbix täglich erzeugt. Im ungünstigsten Fall können wir annehmen, dass Zabbix ein Ereignis pro Sekunde erzeugt.

Für jedes wiederhergestellte Ereignis wird ein event_recovery-Datensatz erstellt. Normalerweise werden die meisten Ereignisse wiederhergestellt, daher können wir von einem event_recovery-Datensatz pro Ereignis ausgehen. Das bedeutet zusätzliche 80 Byte pro Ereignis.

Optional können Ereignisse Tags haben, wobei jeder Tag-Datensatz ungefähr 100 Byte Speicherplatz benötigt[1]. Die Anzahl der Tags pro Ereignis (#tags) hängt von der Konfiguration ab. Daher benötigt jedes Ereignis zusätzlich #tags * 100 Byte Speicherplatz.

Das bedeutet, dass bei einer Aufbewahrung von Ereignissen über 3 Jahre 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B Speicherplatz erforderlich wären[2].

Note:
[1] Mehr bei nicht-ASCII-Ereignisnamen, Tags und Werten.
 [2] Die Größenabschätzungen 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 erforderlichen Speicherplatz (in Byte)
Zabbix-Konfiguration	Feste Größe. Normalerweise 10MB oder weniger.
Verlauf	days*(items/refresh rate)*24*3600*bytes items : Anzahl der Datenpunkte days : Anzahl der Tage, für die der Verlauf aufbewahrt wird refresh rate : durchschnittliches Aktualisierungsintervall der Datenpunkte bytes : Anzahl der Byte, die zum Speichern eines einzelnen Werts erforderlich sind; hängt von der Datenbank-Engine ab, normalerweise ~90 Byte.
Trends	days*(items/3600)*24*3600*bytes items : Anzahl der Datenpunkte days : Anzahl der Tage, für die der Verlauf aufbewahrt wird bytes : Anzahl der Byte, die zum Speichern eines einzelnen Trends erforderlich sind; hängt von der Datenbank-Engine ab, normalerweise ~90 Byte.
Ereignisse	days*events*24*3600*bytes events : Anzahl der Ereignisse pro Sekunde. Ein (1) Ereignis pro Sekunde im ungünstigsten Fall. days : Anzahl der Tage, für die der Verlauf aufbewahrt wird bytes : Anzahl der Byte, die zum Speichern eines einzelnen Ereignisses erforderlich sind; hängt von der Datenbank-Engine ab, normalerweise ~330 + durchschnittliche Anzahl der Tags pro Ereignis * 100 Byte.

Der insgesamt erforderliche Speicherplatz kann also wie folgt berechnet werden:

Konfiguration + Verlauf + Trends + Ereignisse

Der Speicherplatz wird NICHT sofort nach der Zabbix-Installation belegt. Die Datenbankgröße wächst zunächst und hört dann ab einem bestimmten Punkt 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.

Netzwerkanforderungen

Die folgende Liste offener Ports pro Komponente gilt für die Standardkonfiguration.

Komponenten	Port
Frontend	http auf 80, https auf 443
Server	10051 (zur Verwendung mit aktivem Proxy/Agenten)
Aktiver Proxy	10051
Passiver Proxy	10051
Agent2	10050
Trapper	
JavaGateway	10052
WebService	10053

Note:
Die Portnummern sollten in der Firewall geöffnet werden, um die externe Kommunikation mit Zabbix zu ermöglichen. Ausgehende TCP-Verbindungen erfordern in der Regel keine expliziten Firewall-Einstellungen.

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).

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.

Timescale DB: hoher Speicherverbrauch bei großer Anzahl von Partitionen

PostgreSQL-Versionen 9.6-12 verwenden zu viel Speicher, wenn Tabellen mit einer großen Anzahl von Partitionen aktualisiert werden. Dieses Problem tritt auf, wenn Zabbix Trends auf Systemen mit TimescaleDB aktualisiert, falls Trends in relativ kleine Chunks (z. B. 1 Tag) aufgeteilt sind. Dies führt dazu, dass in den Trend-Tabellen bei den Standard-Housekeeping-Einstellungen Hunderte von Chunks vorhanden sind – ein Zustand, bei dem PostgreSQL wahrscheinlich der Speicher ausgeht.

Das Problem wurde seit Zabbix 5.0.1 für Neuinstallationen mit TimescaleDB behoben, aber wenn TimescaleDB zuvor mit Zabbix eingerichtet wurde, lesen Sie bitte die Migrationshinweise unter ZBX-16347.

Timescale DB 2.5.0: Komprimierungsrichtlinie kann bei Tabellen fehlschlagen, die Integer enthalten

Dieses Problem tritt auf, wenn TimescaleDB 2.5.0/2.5.1 verwendet wird. Es wurde seit TimescaleDB 2.5.2 behoben.

Weitere Informationen finden Sie unter TimescaleDB Issue #3773.

Datenbank-TLS-Verbindung mit MariaDB

Eine Datenbank-TLS-Verbindung wird mit der Option 'verify_ca' für den 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

Es gibt einen Fehler in der OpenIPMI-Bibliothek, die von Zabbix zum Abfragen von IPMI-Daten verwendet wird und der durch speziell präparierte Antworten von einem nicht vertrauenswürdigen Gerät ausgelöst werden kann.
Ein nicht vertrauenswürdiges IPMI-Gerät kann präparierte Daten senden, die die OpenIPMI-Bibliothek zum Absturz bringen, was wiederum dazu führen kann, dass der Zabbix-Serverprozess, der die IPMI-Abfrage 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 der FreeTDS-UnixODBC-Treiber verwendet wird, 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 Datenbankmonitor-Datenpunkt in Zabbix die Informationen nicht abrufen und es erscheint der 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-Gebietsschemas

Es wurde beobachtet, dass Frontend-Gebietsschemas scheinbar ohne erkennbare Logik wechseln, d. h. einige Seiten (oder Teile von Seiten) werden in einer Sprache angezeigt, während andere Seiten (oder Teile von Seiten) in einer anderen Sprache angezeigt werden. Typischerweise kann das Problem auftreten, wenn es mehrere Benutzer gibt, von denen einige ein Gebietsschema verwenden, während andere ein anderes verwenden.

Eine bekannte Behelfslösung besteht darin, Multithreading in PHP und Apache zu deaktivieren.

Das Problem hängt damit zusammen, wie das Setzen des Gebietsschemas in PHP funktioniert: Informationen zum Gebietsschema werden pro Prozess und nicht pro Thread verwaltet. In einer Multithreading-Umgebung ist es daher möglich, dass bei mehreren Projekten, die im selben Apache-Prozess ausgeführt werden, das Gebietsschema in einem anderen Thread geändert wird und dies beeinflusst, wie Daten im Zabbix-Thread verarbeitet werden.

Weitere Informationen finden Sie in den zugehörigen Problemberichten:

- ZBX-10911 (Problem mit wechselnden Frontend-Gebietsschemas)
- ZBX-16297 (Problem bei der Zahlenverarbeitung in Diagrammen bei Verwendung der `bcdiv`-Funktion 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 im Parameter `Server` der Zabbix-Agent-Konfigurationsdatei (zabbix_agentd.conf) können die Antwortzeit des Zabbix-Agenten unter Windows erhöhen. Dies geschieht, weil der Windows-DNS-Caching-Dienst negative Antworten für IPv4-Adressen nicht zwischenspeichert. Für IPv6-Adressen werden negative Antworten jedoch zwischengespeichert, daher besteht eine mögliche Umgehungslösung 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 der Dateierweiterung .yaml kann manchmal nicht importiert werden;
- Nicht in Anführungszeichen gesetzte menschenlesbare Datumsangaben werden automatisch in Unix-Zeitstempel umgewandelt.

Einrichtungsassistent unter SUSE mit NGINX und php-fpm

Der Frontend-Einrichtungsassistent kann die Konfigurationsdatei unter SUSE mit NGINX + php-fpm nicht speichern. Dies wird durch eine Einstellung in der Unit `/usr/lib/systemd/system/php-fpm.service` verursacht, die verhindert, dass Zabbix nach `/etc` schreiben kann. (eingeführt in PHP 7.4).

Es gibt zwei verfügbare Umgehungslösungen:

- Setzen Sie die Option ProtectSystem in der php-fpm-systemd-Unit 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 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 aktualisiert. Obwohl der Wechsel zu PCRE2 viele Vorteile bietet, kann er dazu führen, dass bestimmte vorhandene PCRE-Regexp-Muster ungültig werden oder sich anders verhalten. Dies betrifft insbesondere das Muster *^[\w-\.]*. Um diesen regulären Ausdruck wieder gültig zu machen, ohne die Semantik zu verändern, ändern Sie den Ausdruck in *^[-\w\.]* . Dies liegt daran, dass PCRE2 das Minuszeichen als Trennzeichen behandelt und dadurch einen Bereich innerhalb einer Zeichenklasse erstellt.

Fehler im Geomap-Widget

Die Karten im Geomap-Widget werden möglicherweise nicht korrekt geladen, wenn Sie ein Upgrade von einer älteren Zabbix-Version mit NGINX durchgeführt und während des Upgrades nicht auf die neue NGINX-Konfigurationsdatei umgestellt haben.

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 Zabbix Frontend konfigurieren* beschrieben neu konfigurieren.

Alternativ können Sie eine vorhandene NGINX-Konfigurationsdatei manuell bearbeiten (in der Regel */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.

Zugriff auf UI-Elemente mit MariaDB 10.5.1–10.5.9

Beim Zugriff auf das Zabbix-Web-Frontend mit einer anderen Rolle als Super Admin kann die Meldung „System error occurred. Please contact Zabbix administrator." angezeigt werden. Dieses Problem betrifft Installationen mit MariaDB-Versionen 10.5.1 bis 10.5.9.

Um dieses Problem zu vermeiden, aktualisieren Sie MariaDB auf eine Version neuer als 10.5.9. Weitere Informationen finden Sie unter ZBX-25746.

Profiling übermäßiger Speichernutzung mit tcmalloc

Wenn Sie vermuten, dass Ihre Zabbix-Installation zu viel Speicher verwendet, können Sie die Memory-Profiling-Funktion von tcmalloc verwenden, um den Speicherverbrauch von Zabbix Server/Proxy zu untersuchen.

1. Konfigurieren Sie bei der Installation von Zabbix aus den Quellen 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 das Profiling von tcmalloc beeinträchtigen können.

2. Setzen Sie vor dem Starten des Zabbix Servers 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 Profildump 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 Speicherallokationen verantwortlich.

Siehe auch:

- ZBX-25050 und ZBX-25584 für die zugehörigen Problemberichte.
- GCC Option Summary zu Kompilierungsoptionen (`-std=gnu99`, `-g`, `-O0` usw.).
- Dokumentation zu Gperftools Heap Profiler über Umgebungsvariablen für das tcmalloc-Profiling.

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 Commits 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:

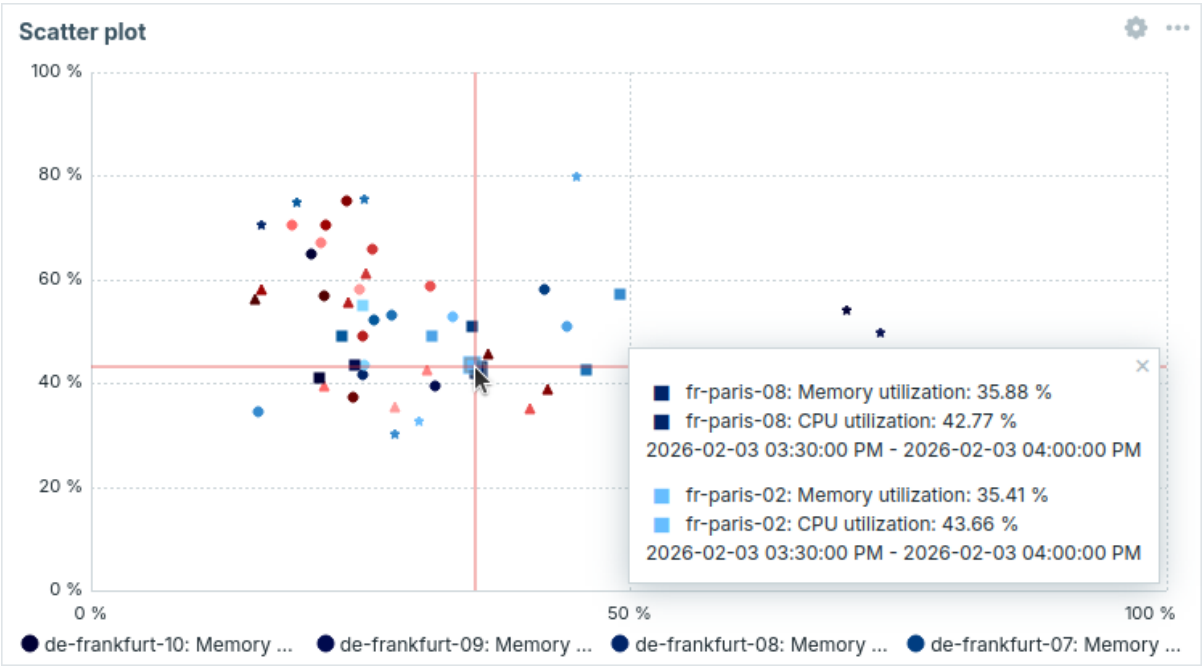
- ZBX-26060 für den zugehörigen Problembericht.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" für das zugrunde liegende Problem.

2 Was ist neu in Zabbix 8.0

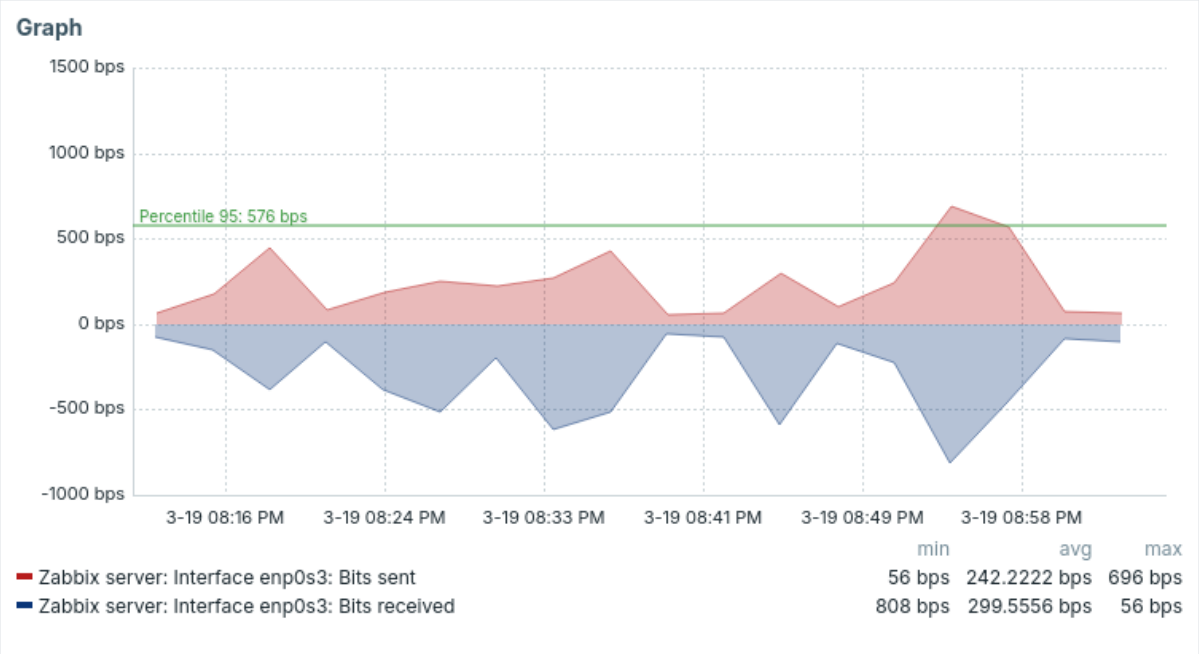
Zabbix 8.0.0 baut auf Zabbix 7.4.0 auf und fügt neue Funktionen und Verbesserungen hinzu.

Siehe Inkompatible Änderungen für diese Version.

Streudiagramm-Widget Das Widget Scatter plot wurde zu den Dashboard-Widgets hinzugefügt. Es zeigt die Beziehung zwischen zwei Metriken, indem einzelne Datenpunkte entlang einer X- und Y-Achse dargestellt werden. Dies hilft dabei, Muster, Cluster, Korrelationen und Ausreißer im Datensatz sichtbar zu machen.



Umkehrung der Y-Achse für das Graph-Widget Graphen im Graph-Widget können jetzt mit umgekehrten Y-Achsenwerten angezeigt werden. Mit der neuen Einstellung *Werte umkehren* können Sie Y-Achsenwerte mit −1 multiplizieren, ohne die ursprünglichen Daten zu verändern.



Siehe auch: Erweiterungen des Graph-Widgets.

Vorlagen

Neue Vorlagen

- Aruba CX 8300s by SNMP, eine Vorlage zur SNMP-basierten Überwachung der Aruba-CX-8300-Switch-Serie.
- Die Vorlagensammlung AWS by HTTP wurde um die Vorlage *AWS Backup Vault by HTTP* ergänzt.
- Die Vorlagensammlung Azure by HTTP wurde um die Vorlage *Azure Sentinel by HTTP* ergänzt.
- Ciena 3906 by SNMP, eine Vorlage zur Überwachung von Ciena-3906-Geräten.
- Cisco Secure Firewall Threat Defense by HTTP, eine Vorlage mit Überwachungsfunktionen für Cisco-Secure-Firewall-Threat-Defense-Geräte über die REST-API.
- Cradlepoint NCM v2 by HTTP, eine Vorlagensammlung zur Überwachung von Cradlepoint NCM v2 und seinen Geräten über HTTP.
- Domain RDAP by HTTP, eine Vorlage zur Überwachung von Domain-Registrierungsdaten über das RDAP-Protokoll.
- IBM Maximo Service Request, eine webhook-basierte Vorlage, die die Verknüpfung von Zabbix mit IBM Maximo ermöglicht.
- MariaDB by ODBC, eine Vorlage zur Überwachung von MariaDB-Datenbanken über ODBC.
- Microsoft Hyper-V Failover Cluster by SSH und Microsoft Hyper-V Standalone by SSH, Vorlagen zur Überwachung von Microsoft-Hyper-V-Clustern und eigenständigen Hosts über SSH.
- OpenAI Platform by HTTP, eine Vorlage zur Überwachung der Entwicklerplattform von OpenAI.
- Percona by ODBC, eine Vorlage zur Überwachung von Percona-Datenbanken über ODBC.
- Ribbon SBC Edge by HTTP, eine Vorlage zur Überwachung von Ribbon-SBC-Edge-Geräten (früher SWe Lite) über HTTP.
- Ribbon SBC SWe Core by HTTP, eine Vorlage zur Überwachung von SBC-SWe-Core-Geräten über HTTP, und Ribbon SBC SWe CE by HTTP, eine Vorlage zur Überwachung von Ribbon-SBC-SWe-Call-Engine-(CE)-Instanzen über HTTP.
- Stormshield SNS by SNMP, eine Vorlage zur Überwachung von Stormshield Network Security (SNS)-Geräten über SNMP.
- VeloCloud SD-WAN Edge by HTTP, eine Vorlage zur Überwachung von VeloCloud-SD-WAN-Edge-Geräten über HTTP.
- Vyatta Virtual Router by SNMP, eine Vorlage zur Überwachung des virtuellen Routers Vyatta 1908e.

Aktualisierte Vorlagen

- Ciena 3906 by SNMP wurde aktualisiert, um Datenpunkte für Dateisysteme und CPU-Auslastung einzuschließen.
- Das Vorlagenset GCP by HTTP wurde um die Vorlage *GCP Cloud Run Service by HTTP* ergänzt.
- GitHub organization by HTTP wurde aktualisiert, um Datenpunkte für die Überwachung von Microsoft Copilot einzuschließen.
- Microsoft 365 reports by HTTP wurde aktualisiert, um Datenpunkte für die Überwachung von Microsoft Copilot einzuschließen.
- MySQL by Zabbix agent, MySQL by Zabbix agent 2, MySQL by Zabbix agent active, MySQL by Zabbix agent 2 active und MySQL by ODBC wurden aktualisiert, um sowohl `SHOW SLAVE STATUS` (alte Syntax) als auch `SHOW REPLICA STATUS` (neue Syntax) zu unterstützen.
- MySQL by ODBC wurde außerdem mit neuen Metriken, Discovery-Regeln für Tabellen und Replikate sowie einem verbesserten Dashboard aktualisiert.
- PostgreSQL by ODBC wurde mit neuen Metriken, Discovery-Regeln und Dashboards sowie allgemeinen Verbesserungen bei Leistung und Beobachtbarkeit aktualisiert.
- Proxmox VE by HTTP wurde mit der verschachtelten LLD-Funktionalität aktualisiert. Zusätzlich wurde das Einheitenformat für Datenpunkte, die Prozentwerte anzeigen, zur besseren Übersichtlichkeit angepasst.

- VeloCloud SD-WAN by HTTP, zuvor *VMWare SD-WAN VeloCloud by HTTP*, wurde umbenannt und aktualisiert, um an die neueste VeloCloud SD-WAN-Plattform angepasst zu sein.
- Vyatta Virtual Router by SNMP wurde mit neuen OID-Datenpunkten und Dashboard-Verbesserungen aktualisiert.

Datenpunkte

JSON-Datentyp Zabbix unterstützt jetzt JSON als Datentyp für Datenpunkt-Werte.

Bisher wurden JSON-Werte von Text-Datenpunkten erfasst und als Zeichenfolgen mit einer Begrenzung von 64 KB gespeichert. Jetzt kann Zabbix JSON-Werte nativ mit einer Begrenzung von 128 MiB speichern und außerdem ungültige JSON-Werte ablehnen (z. B. mit nicht in Anführungszeichen gesetzten Schlüsseln, nachgestellten Kommas oder nicht übereinstimmenden Klammern).

Der JSON-Datentyp wird von allen Datenpunkt-Typen und Datenpunkt-Prototypen unterstützt (außer *Berechnet*) und ist in Echtzeit-Datenexport und Konnektoren verfügbar. JSON-Werte können in allen unterstützten Datenbanken und in Elasticsearch gespeichert werden. Wenn Sie TimescaleDB verwenden, lesen Sie bitte die Hinweise zum Upgrade.

Beachten Sie, dass JSON-Datenpunkte nicht in Auslösern verwendet werden können; Sie können jedoch JSON-Felder mit abhängigen Datenpunkten extrahieren, die keinen JSON-Datentyp haben, und diese in Auslösern verwenden.

Datenpunkte, die eine JSON-Zeichenfolge zurückgeben (`net.if.discovery`, `vfs.file.get` usw.), sind weiterhin Text-Datenpunkte; Sie können sie jedoch bei Bedarf in JSON ändern.

Weitere Details und Informationen zu JSON-Datenlimits finden Sie unter Datenpunkt-Konfiguration.

Typ-Parameter für die S.M.A.R.T.-Datenträgererkennung Der Datenpunkt smart.disk.discovery (S.M.A.R.T.-Plugin von Zabbix Agent 2) akzeptiert jetzt einen optionalen Parameter **type**, um einen Wert anzugeben, nach dem bei den Datenträgern gesucht werden soll.

Plugins

Ceph-Plugin Dieses Plugin arbeitet jetzt in zwei Modi:

- **native** - Dieser Modus verwendet die go-ceph-Bibliothek, um über die native Ceph-API (msgr2-Protokoll) direkt mit dem Ceph-Cluster zu kommunizieren. Dies ist der empfohlene Modus für moderne Ceph-Installationen, wird jedoch nur **unter Linux** und ab Ceph 16 unterstützt.
- **restful** (veraltet) - Dieser Modus verwendet die Ceph-RESTful-API für die Kommunikation. Er ist aus Gründen der Abwärtskompatibilität der Standardmodus, funktioniert jedoch mit Ceph-Version 20 (Tentacle) oder neuer nicht, da das Modul mgr/restful entfernt wurde.

Welcher Modus verwendet wird, wird durch den Wert des Parameters mode (native/restful) bestimmt:

- `Plugins.Ceph.Default.Mode=native` - legt den nativen Modus für das Plugin fest
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - legt den nativen Modus für die benannte Sitzung fest

Beachten Sie, dass sich die verwendeten Benutzerzugangsdaten je nach Modus unterscheiden und nicht miteinander kompatibel sind. Der Parameter `Plugins.Ceph.InsecureSkipVerify` wird im nativen Modus ignoriert, da die Verbindungssicherheit auf der Ceph-Cluster-Seite durch das msgr2-Protokoll definiert wird (standardmäßig sicher).

Beachten Sie, dass das Ceph-Plugin für Zabbix Agent 2 jetzt ein ladbares Plugin ist und zusätzliche Installationsschritte erfordert. Dies ist auf die Anforderung des Pakets librados zurückzuführen (für den nativen Modus). Weitere Informationen finden Sie in der Ceph-Plugin-readme.

MongoDB-Plugin Das Zabbix Agent 2 MongoDB-Plugin bietet mehrere Verbesserungen: - Vollständige Unterstützung für das Parsen herkömmlicher MongoDB-URIs, mit Unterstützung sowohl für die Schemata `mongodb://` als auch `mongodb+srv://`. - Automatische Erkennung von MongoDB-Replikatsatzknoten, wodurch die Fähigkeit zur Überwachung von MongoDB-Clustern verbessert wird. - Unterstützung für x509-Authentifizierung, die sichere Verbindungen zu MongoDB mithilfe von Client-Zertifikaten ermöglicht.

Oracle-Plugin Das Zabbix Agent 2 Oracle-Plugin unterstützt jetzt verschlüsselte Verbindungen zu Oracle-Datenbanken mit dem TCPS-(TLS-)Protokoll.
Dadurch können Oracle-Instanzen über TLS-gesicherte Sockets überwacht werden, was die Sicherheit der Remote-Überwachung verbessert.

Redis-Plugin — TLS-Unterstützung und Validierung beim Start Dem Redis-Plugin für Zabbix Agent 2 wurde TLS-Unterstützung hinzugefügt.

Eine Validierung der TLS-Konfiguration des Plugins beim Start wurde implementiert und die Validierungs-/Fehlermeldungen wurden verbessert. Eine ungültige Konfigurationslogik (zum Beispiel: Verwendung des Verbindungstyps `verify_full` ohne Angabe von `TLSCAFile`) kann verhindern, dass Zabbix Agent 2 startet.

Testausführungsmodus für ladbare Plugins Ladbare Plugins können jetzt im Testmodus mit dem Flag `-t` (`--test`) gestartet werden, wobei ein Datenpunktschlüssel als Argument übergeben wird. In diesem Modus wird das Plugin zu Debugging- und Entwicklungszwecken ausgeführt, und Plugin-Konfigurationsdateien werden ignoriert.

Low-level discovery

Kontrollkästchen „In JSON konvertieren“ in Discovery-Formularen Ein neues Kontrollkästchen *In JSON konvertieren* wurde zum Formular für Discovery-Regeln und zum Formular für Discovery-Prototypen hinzugefügt. Es wird angezeigt, wenn im Dropdown *Typ* „HTTP-Agent“ ausgewählt ist. Mit dieser Option können abgerufene Daten für die automatische Konvertierung in JSON vor der weiteren Verarbeitung markiert werden.

Makrounterstützung für verschachtelte Low-Level-Discovery Low-Level-Discovery-Makros werden jetzt in verschachtelten Low-Level-Discovery-Regeln unterstützt, in:

- JSONPath-Präprozessierungsparametern
- JSONPath-Feld für benutzerdefinierte LLD-Makros

Bearbeitbare Tags für durch Low-Level-Discovery erstellte Auslöser Für Auslöser, die aus Auslöser-Prototypen erstellt wurden, können jetzt manuell Tags hinzugefügt werden. Von Auslöser-Prototypen geerbte Tags werden weiterhin automatisch angewendet. Manuell hinzugefügte Tags können bei erkannten Auslösern geändert werden und werden in Ereignis-Tag-Arrays aufgenommen sowie für tag-fähige Funktionen wie Filter, Dashboard-Widgets und Benachrichtigungsmakros verfügbar sein.

Prozesse

Caching und Wiederverwendung von SNMPv3-EngineIDs Zabbix speichert jetzt Zuordnungen von SNMPv3-EngineID → IP im Cache und versucht, zwischengespeicherte EngineIDs für nachfolgende SNMPv3-Prüfungen wiederzuverwenden. Dadurch wird der Probe-Datenverkehr reduziert und die Leistung des Pollers verbessert.
Wenn eine wiederverwendete EngineID nicht antwortet, greift der Poller auf eine EngineID-Probe zurück und kann veraltete Einträge nach Änderungen an der Schnittstelle oder bei anhaltenden Fehlern entfernen.

Verfeinerte Proxy-Drosselung während der Wiederherstellung des Verlaufs-Caches Die Drosselungslogik des Proxy wurde verfeinert, um die Stabilität des Server während der Wiederherstellung des Verlaufs-Caches zu verbessern. Wenn die Auslastung des Verlaufs-Caches den Drosselungsschwellenwert erreicht, akzeptiert der Server wie bisher weiterhin keine Proxy-Daten. Wenn die Cache-Auslastung auf 60 % fällt, beginnt der Server mit der Verarbeitung der Drosselungsliste, kann jedoch Proxy-Uploads mit sehr großen Stapeln (ungefähr mehr als 10.000 Datensätze) weiterhin ablehnen, bis der Cache-Druck weiter nachlässt. Diese Änderung verringert das Risiko wiederholter Cache-Überlastungen, während sich der Server erholt.

Erhöhter maximaler Timeout für zabbix_get und zabbix_js Der maximale Wert des Parameters `timeout` für die Befehlszeilenprogramme zabbix_get und zabbix_js wurde auf 600 Sekunden erhöht.

Optimierte Bereinigung Housekeeping wurde optimiert, indem die Erstellung von Housekeeper-Aufgaben in Datenbank-Trigger verlagert wurde. Wenn Datenpunkte (einschließlich Low-Level-Discovery-Regeln), Auslöser, Services oder Regeln für die Netzwerkdiscovery gelöscht werden, füllen Datenbank-Trigger (anstelle expliziter Aufrufe auf Anwendungsebene) nun die Tabelle `housekeeper` mit Bereinigungsaufgaben. Weitere Informationen finden Sie im Housekeeping-Verfahren.

Der Housekeeper entfernt jetzt auch Netzwerkdiscovery-Ereignisse, die von entdeckten Hosts oder Services erzeugt wurden, die inzwischen gelöscht wurden, sowie alle Ereignisse, die mit Problemen von Auslösern verknüpft sind, die inzwischen gelöscht wurden (zuvor wurden nur die Probleme selbst entfernt; zugehörige Ereignisse wurden erst entfernt, nachdem die *Speicherperiode für Auslöserdaten* des Housekeepers abgelaufen war).

Manuelles Schließen — von Wiederherstellungsereignissen geerbte Auslöser-Tags Wiederherstellungsereignisse, die nach einem manuellen Schließen erstellt werden, erben zusätzlich zu Datenpunkt- und Host-Tags auch Auslöser-Tags. Diese Tags sind im Array der Ereignis-Tags vorhanden und für Benachrichtigungsmakros wie `{EVENT.RECOVERY.TAGS}` und `{EVENT.RECOVERY.TAGSJSON}` verfügbar.

DNS-Abfrage-Caching für Zabbix-Daemons Zabbix Server, Zabbix Proxy und Zabbix Agent unterstützen die Verwendung des c-ares-Resolvers für alle DNS-Anfragen. Dies bietet DNS-Abfrage-Caching und ein verbessertes Resolver-Failover, wenn Zabbix mit `--with-ares` erstellt wird. Für DNS-Abfrage-Caching ist c-ares 1.26.0 oder höher erforderlich.

Unterstützung für c-ares unter Windows erstellen Der Zabbix Agent kann jetzt unter Microsoft Windows erstellt werden, wobei der c-ares-Resolver verwendet wird. c-ares kann über vcpkg installiert werden, und der Agent-Build unterstützt entweder `ARES=<vcpkg prefix>` oder separate Pfade `ARESINCDIR`/`ARESLIBDIR` für Include- und Bibliotheksverzeichnisse.

Authentifizierung

Importierbare SAML-Zertifikate für Single Sign-on Super-Admin-Benutzer können jetzt Zertifikate und private Schlüssel direkt im Frontend für die Konfiguration von SAML importieren. Drei neue Felder wurden unter *Administration* > *Authentication* > *SAML* hinzugefügt:

- *IdP-Zertifikat* - X.509-Zertifikat, das vom Identitätsanbieter präsentiert wird
- *SP-Zertifikat* - Zertifikat des Dienstanbieters, das für SAML-Austausche verwendet wird
- *SP-Privatschlüssel* - privater Schlüssel, der dem SP-Zertifikat entspricht

Mit diesen Steuerelementen können neue Werte hinzugefügt oder bestehende über den Reiter mit den SAML-Einstellungen geändert werden. Zertifikate und private Schlüssel werden vor dem Speichern im gewählten Storage-Backend validiert; ungültige oder fehlerhaft formatierte Werte werden mit einer erläuternden Fehlermeldung zurückgewiesen.

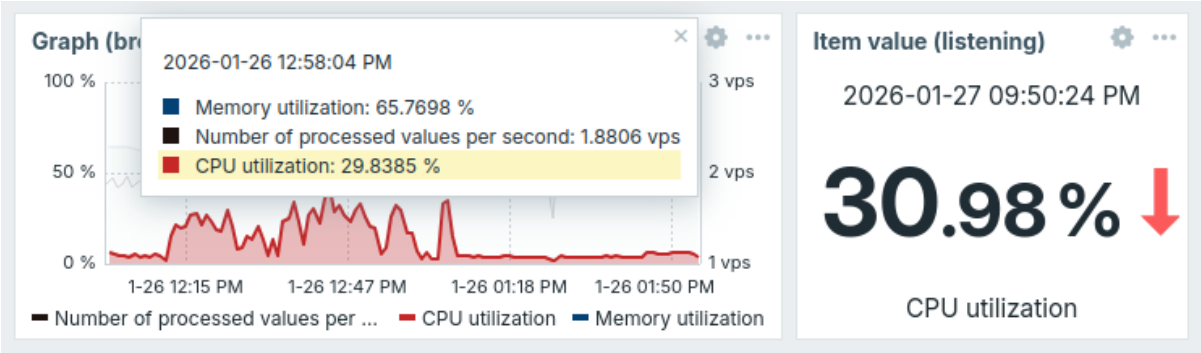
Widgets

Clustering von Geomap-Host-Markierungen Das Widget Geomap unterstützt jetzt den Parameter *Clustering*, um zu steuern, wie nahe beieinander liegende Host-Markierungen zu einer einzelnen Markierung mit Anzahl zusammengefasst werden. Durch Festlegen der Karten-Zoomstufe für das Clustering können Sie große Karten übersichtlich halten und bei vergrößerter Ansicht eine präzise Sichtbarkeit beibehalten.

Aggregierte Spalten in „Top-Datenpunkte“ Das Widget Top items kann jetzt Datenpunkt-Muster in einer einzelnen aggregierten Spalte oder Zeile gruppieren, sodass sich kombinierte Metriken einfach darstellen lassen (zum Beispiel: gesamter eingehender Datenverkehr über alle Netzwerkschnittstellen eines Hosts). Beim Konfigurieren von Datenpunkt-Spalten wurden drei neue Optionen hinzugefügt: *Spalten aggregieren*, *Spaltenaggregationsfunktion* und *Name der kombinierten Spalte*.

Verbesserungen des Graph-Widgets Das Widget Graph bietet mehrere Verbesserungen der Benutzerfreundlichkeit:

- Datenpunkte in der QuickInfo der Diagrammdaten, die beim Überfahren des Diagramms mit der Maus angezeigt wird, werden nach Wert in absteigender Reihenfolge sortiert.
- Wenn Sie in der QuickInfo mit der Maus über einen Datenpunkt fahren, wird dessen Diagramm hervorgehoben, während die anderen abgeblendet werden.
- Wenn Sie einen Datenpunkt in der QuickInfo auswählen, werden seine Daten an andere Widgets übertragen, die das Empfangen dieser Daten unterstützen.
- Die Begrenzung der Anzahl der in der QuickInfo angezeigten Datenpunkte wurde aufgehoben.
- Mit der neuen Einstellung *Host names in labels* können Sie festlegen, ob Host-Namen in der QuickInfo und in der Diagrammlegende angezeigt werden.

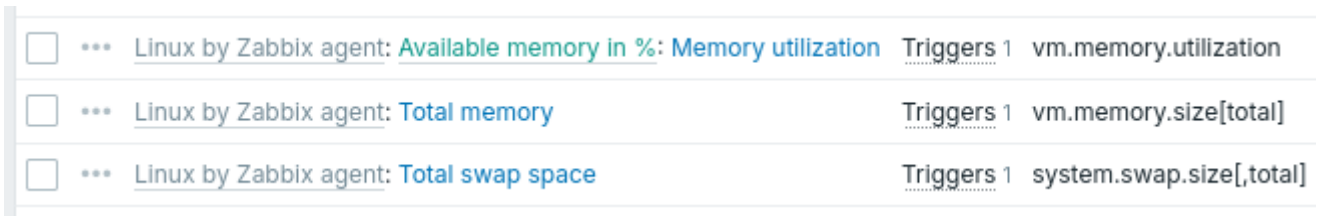


- Mit der neuen Einstellung *Invert values* können Sie ein Diagramm umkehren.

Frontend

Neue und eingebettete Schriftarten Zabbix enthält jetzt neue und eingebettete Schriftarten, die die Lesbarkeit verbessern, schneller geladen werden und zusätzliche Frontend-Sprachen mit minimalen Auswirkungen auf das Layout darstellen. Da sie eingebettet sind, sorgen diese Schriftarten außerdem für ein einheitliches Erscheinungsbild auf allen Systemen.

Neue Schriftarten:



Bisherige Schriftarten:

Die neuen Schriftarten werden in fast allen Themes verwendet. Bei Bedarf stehen die neu hinzugefügten Themes *Blue (classic)* und *Dark (classic)* mit den bisherigen Schriftarten zur Verfügung.

Die Schriftfamilie für Inhalte in Monospace-Schrift und für Diagramme bleibt unverändert.

Inline-Validierung Die folgenden Formulare im Frontend wurden zur Gruppe der Formulare hinzugefügt, die Inline-Validierung unterstützen:

- API-Token
- Authentifizierung
- Autoregistrierung
- Konnektor
- Geografische Karten
- Host-Gruppe
- Host-Prototypen
- Symbolzuordnung
- Bilder
- Wartung
- Medientyp
- Benachrichtigungen
- Profil
- Proxy
- Reguläre Ausdrücke
- Service
- SLA
- Vorlagengruppe
- Problem aktualisieren
- Globale Benutzermakros
- Benutzerrollen
- Benutzer

Eingabefehler werden unmittelbar nach dem Ausfüllen der Felder angezeigt, was die Benutzerfreundlichkeit verbessert und Konfigurationsfehler reduziert.

Modale Formulare Bei der Einrichtung der Low-Level-Discovery wird das Konfigurationsformular für Host-Prototypen jetzt in einem modalen (Pop-up-)Fenster geöffnet.

Vererbte Tags sichtbar in Hosts, Vorlagen, Datenpunkten und Auslösern Vererbte Tags werden jetzt in Vorlagen, Hosts, Datenpunkten, Webszenarien und Auslösern konsistent angezeigt und zurückgegeben. Tags, die aus Vorlagen-/Host-Ketten vererbt werden, sind unter *Monitoring > Latest data* sichtbar. Die Filterung nach vererbten Tags ist überall verfügbar, wo die Filterung nach Tags unterstützt wird — einschließlich der Bereiche *Monitoring > Latest data* und *Data collection* sowie aller Dashboard-Widgets, die eine Filterung nach Host-, Datenpunkt-, Auslöser- oder Webszenario-Tags erlauben — sodass die tagbasierte Auswahl und Unterfilterung unabhängig davon gleich funktioniert, wo ein Tag definiert wurde.

Die Registerkarte *Tags* in den Konfigurationsformularen für Vorlagen, Hosts und Host-Prototypen bietet jetzt ein Optionsfeld zur Auswahl, wie Tags dargestellt werden: Vorlagen zeigen *Vorlagen-Tags / Vererbte und Vorlagen-Tags*, und Hosts sowie Host-Prototypen zeigen *Host-Tags / Vererbte und Host-Tags*. Vererbte Tags werden optisch durch ein neues umrandetes Dokumentsymbol neben der Tag-Bezeichnung hervorgehoben.

Die Widgets Graph und Pie chart enthalten eine neue Einstellung *Datenpunkt-Tags*.

Positionierung von Tooltips Tooltips können jetzt durch Ziehen an eine neue Position verschoben werden. Dies gilt zum Beispiel für den Tooltip des Graphen sowie für Tooltips mit Beschreibungen in den Bereichen Letzte Daten oder Probleme.

Dokumentation

Konsolidierte Dokumentationsseiten für Minor-Releases Die Release-Dokumentation für Minor-Versionen eines größeren Zabbix-Releases wird nun jeweils auf einzelnen Dokumentationsseiten für neue Funktionen und Upgrade-Hinweise zusammengefasst.

Was ist neu in Zabbix 8.0.x

Diese Seite bietet zusammengefasste Informationen zu neuen Funktionen, die in Minor-Releases der Hauptversion von Zabbix enthalten sind.

Siehe auch Was ist neu zur Hauptversion.

Änderungen an bestehenden Vorlagen sowie Informationen zu neuen Vorlagen für verschiedene Geräte, Dienste und Partnerlösungen finden Sie unter Änderungen an Vorlagen.

Was ist neu in Zabbix 8.0.1 Diese Version wurde noch nicht veröffentlicht.

3 Zabbix Appliance

Überblick Die Zabbix Appliance bietet eine Möglichkeit, Zabbix Server und Frontend sofort bereitzustellen, anstatt sie manuell einzurichten oder einen bestehenden Server für Zabbix wiederzuverwenden.

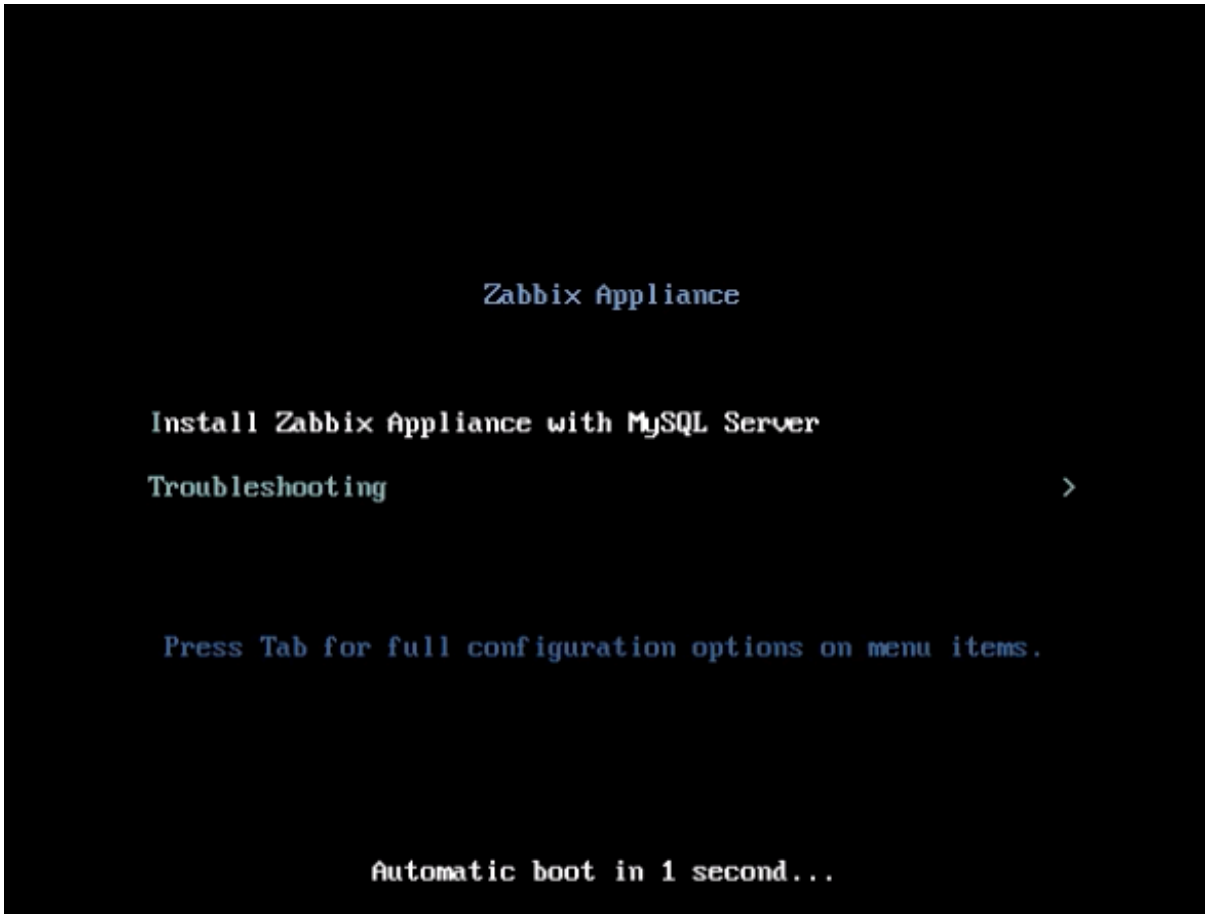
Die Appliance basiert auf AlmaLinux 8 (x86_64) und enthält einen vorkonfigurierten Zabbix Server, der auf MySQL läuft, sowie ein Frontend, das auf dem Nginx-Webserver läuft.

> **Attention:**
> Diese Appliance ist für die Evaluierung von Zabbix vorgesehen. Von der Verwendung in ernsthaften Produktionsumgebungen wird abgeraten.

Appliance-Images stehen in den folgenden Formaten zum Download zur Verfügung:

- 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)

Boot-Menü 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 virtuelle Appliance-Maschine mit dem heruntergeladenen Image.
2. Konfigurieren Sie die Netzwerkeinstellungen der virtuellen Maschine so, dass der Zugriff über einen Browser auf dem Host-Rechner möglich ist. Dies kann durch Aktivieren des *Bridged mode* erreicht werden.
3. Melden Sie sich mit den standardmäßigen System-Zugangsdaten an der virtuellen Maschine an.
4. Führen Sie auf der virtuellen Maschine den folgenden Befehl aus, um die IP-Adresse zu ermitteln:

```
ip addr show
```

5. Öffnen Sie einen Browser auf dem Host-Rechner und rufen Sie die IP-Adresse auf, die die Appliance per DHCP erhalten hat.
6. Melden Sie sich mit den standardmäßigen Zugangsdaten bei Zabbix an und beginnen Sie mit der Überwachung.

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 Benutzerprofileinstellungen ändern oder neue Benutzer erstellen und den Standardbenutzer löschen.

Datenbank

Passwörter für alle Datenbankbenutzer werden während des Installationsprozesses zufällig generiert. Für die Datenbank sind die folgenden Benutzer definiert:

Root:

- Benutzername: root
- Passwort: Das Passwort ist in der Datei `/root/.my.cnf` gespeichert. Für das root-Konto ist keine Passworteingabe erforderlich.

Zabbix Server:

- Benutzername: zabbix_srv
- Passwort: Das Passwort ist in `/etc/zabbix/zabbix\_server.conf` gespeichert.

Zabbix Frontend:

- Benutzername: zabbix_web
- Passwort: Das Passwort ist in `/etc/zabbix/web/zabbix.conf.php` 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://<virtual machine's IP>* zugegriffen werden.

Standardmäßig ist der Zugriff von überall erlaubt. Um den Zugriff einzuschränken, ändern Sie `/etc/nginx/conf.d/zabbix.conf`. Speichern Sie die bearbeitete Datei und starten Sie anschließend Nginx neu, indem Sie sich per SSH als root user anmelden und Folgendes ausführen:

```
systemctl restart nginx
```

Statische IP-Adresse

Standardmäßig verwendet die Appliance DHCP, um die IP-Adresse zu beziehen. So legen Sie eine statische IP-Adresse fest:

- Melden Sie sich als root-Benutzer an.
- Führen Sie die folgenden Befehle aus und ersetzen Sie die Werte durch Ihre eigenen IP-Adressen:

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.10/24    # Appliance IP address/CIDR prefix
nmcli connection modify eth0 ipv4.gateway 192.168.1.1     # Gateway IP address
nmcli connection modify eth0 ipv4.dns 8.8.8.8    # DNS server IP address
nmcli connection modify eth0 ipv4.method manual
systemctl restart network
```

Alternativ können Sie die Datei `/etc/NetworkManager/system-connections/eth0.nmconnection` öffnen und die folgenden Zeilen hinzufügen:

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

Speichern Sie die bearbeitete Datei und führen Sie anschließend den Befehl `systemctl restart network` aus.

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

- Konfigurationsdateien befinden sich in `/etc/zabbix`
- Die Protokolldateien von Zabbix Server, Proxy und Agent befinden sich in `/var/log/zabbix`
- Das Zabbix Frontend befindet sich in `/usr/share/zabbix`
- Das Home-Verzeichnis für den Benutzer *zabbix* ist `/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 (Authentifizierung für den Download erforderlich). Wenn Sie VMware vCenter Converter verwenden, 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 am Frontend anzumelden, die Fehlermeldung *Access denied for user 'replace_user'@'localhost' (using password: YES)* angezeigt wird, kann dies darauf hindeuten, dass die Installation noch nicht abgeschlossen ist.

Wenn der Fehler auch nach einigen Minuten weiterhin auftritt oder Sie ein anderes unerwartetes Verhalten beobachten, bedeutet dies wahrscheinlich, dass der Installationsprozess nicht erfolgreich abgeschlossen wurde. In diesem Fall empfehlen wir, die aktuelle

Appliance zu löschen und sie erneut bereitzustellen, indem Sie dieselben Installationsanweisungen befolgen. Dieser Schritt behebt das Problem in der Regel.

Beachten Sie, dass der Versuch, eine fehlerhafte Installation manuell zu reparieren, nicht empfohlen wird, da dies zu weiteren Komplikationen führen kann.

4 Zabbix-Prozesse

Bitte verwenden Sie die Seitenleiste, um auf Inhalte im Abschnitt zu Zabbix-Prozessen zuzugreifen.

1 Server

Übersicht

Der Zabbix Server ist der zentrale Prozess der Zabbix-Software.

Der Server führt die Abfrage und das Empfangen von Daten durch, berechnet Auslöser und sendet Benachrichtigungen an Benutzer. Er ist die zentrale Komponente, an die Zabbix Agents und Proxys Daten über die Verfügbarkeit und Integrität von Systemen melden. Der Server kann selbst Netzwerkdienste (wie Webserver und Mailserver) mithilfe einfacher Service-Prüfungen aus der Ferne überprüfen.

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 alarmiert, wenn in einem der überwachten Systeme Probleme auftreten.

Die Funktionsweise eines grundlegenden Zabbix Servers ist in drei unterschiedliche Komponenten unterteilt: Zabbix Server, Web-Frontend und Datenbankspeicher.

Alle Konfigurationsinformationen für Zabbix werden in der Datenbank gespeichert, mit der sowohl der Server als auch das Web-Frontend interagieren. Wenn Sie beispielsweise mit dem Web-Frontend (oder der API) einen neuen Datenpunkt erstellen, wird er der Tabelle items in der Datenbank hinzugefügt. Anschließend fragt der Zabbix Server etwa einmal pro Minute die Tabelle items nach einer Liste der aktiven Datenpunkte ab; diese wird dann in einem Cache innerhalb des Zabbix Servers gespeichert. Deshalb kann es bis zu zwei Minuten dauern, bis Änderungen, die im Zabbix Frontend vorgenommen wurden, im Abschnitt „Neueste Daten“ sichtbar werden.

Server ausführen

Wenn als Paket installiert

Der Zabbix Server läuft als Daemon-Prozess. Der Server kann mit folgendem Befehl 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

Falls das oben Genannte nicht funktioniert, müssen Sie ihn manuell starten. Suchen Sie den Pfad zur zabbix_server-Binärdatei und führen Sie Folgendes aus:

```
zabbix_server
```

Sie können die folgenden Befehlszeilenparameter mit dem Zabbix Server verwenden:

```
-c --config <file>              Pfad zur Konfigurationsdatei (Standard ist /usr/local/etc/zabbix_server.co
-f --foreground                 Zabbix Server im Vordergrund ausführen
-R --runtime-control <option>   administrative Funktionen ausführen
-T --test-config                Konfigurationsdatei prüfen und beenden
-h --help                       diese Hilfe anzeigen
-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 zur Laufzeitsteuerung:

Option	Beschreibung	Ziel
config_cache_reload	Konfigurations-Cache neu laden. Wird ignoriert, wenn der Cache derzeit geladen wird.	
history_cache_clear=<**target**>	History-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-Logdatei erfassen.	**historycache** - Statistiken zum History-Cache **valuecache** - Statistiken zum Wert-Cache **preprocessing** - Statistiken zum Präprozessing-Manager **alerting** - Statistiken zum Alert-Manager **lld** - Statistiken zum LLD-Manager **locks** - Liste der Mutexe (ist auf *BSD*-Systemen leer) **connector** - Statistiken für Konnektoren mit der größten Warteschlange
ha_status	Status des High-Availability-(HA)-Clusters protokollieren.	
ha_remove_node=<**target**>	Den High-Availability-(HA)-Knoten entfernen, der durch seinen Namen oder seine ID angegeben ist. Beachten Sie, dass aktive/Standby-Knoten nicht entfernt werden können.	**target** - Name oder ID des Knotens (kann durch Ausführen von ha_status ermittelt werden)
ha_set_failover_delay=<**delay**>	Failover-Verzögerung für High Availability (HA) festlegen. Zeitsuffixe werden unterstützt, z. B. 10s, 1m.	
proxy_config_cache_reload[=<**target**>]	Konfigurations-Cache des Proxy neu laden.	**target** - kommagetrennte Liste von Proxy-Namen Wenn kein target angegeben ist, wird die Konfiguration für alle Proxys neu geladen
secrets_reload	Secrets aus Vault neu laden.	
service_cache_reload	Den Cache des Service-Managers 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 derzeit ausgeführt wird.	
trigger_housekeeper_execute	Die Auslöser-Housekeeping-Prozedur starten. Wird ignoriert, wenn die Housekeeping-Prozedur für Auslöser derzeit ausgeführt wird.	
log_level_increase[=<**target**>]	Log-Ebene erhöhen; wirkt sich auf alle Prozesse aus, wenn kein target 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). Bei größeren Werten geben Sie target als 'process type,N' an.
log_level_decrease[=<**target**>]	Log-Ebene verringern; wirkt sich auf alle Prozesse aus, wenn kein target angegeben ist. Auf *BSD*-Systemen nicht unterstützt.	

Option	Beschreibung	Ziel
prof_enable[=<**target**>]	Profiling aktivieren. Wirkt sich auf alle Prozesse aus, wenn kein target angegeben ist. Aktiviertes Profiling liefert Details zu allen rwlocks/Mutexen nach Funktionsname.	**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, preprocessing worker, 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). Bei größeren Werten geben Sie target als 'process type,N' an. **scope** - `rwlock`, `mutex`, `processing` können mit Prozesstyp und Nummer verwendet werden (z. B. history syncer,1,processing) oder für alle Prozesse eines Typs (z. B. history syncer,rwlock)
prof_disable[=<**target**>]	Profiling deaktivieren. Wirkt sich auf alle Prozesse aus, wenn kein target 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). Bei größeren Werten geben Sie target als 'process type,N' an.

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des Konfigurations-Cache 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 Proxys 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 Verlaufs-Cache 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 zum Verlaufs-Cache in der Server-Logdatei sammeln:
zabbix_server -R diaginfo=historycache
```

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des SNMP-Cache:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Wenn eine SNMPv3-Schnittstelle über die Zabbix-Benutzeroberfläche aktualisiert wird, lädt Zabbix in den meisten Fällen automatisch die neuen SNMPv3-Anmeldedaten für diese Schnittstelle neu; verwenden Sie `-R snmp_cache_reload` nur, wenn die Abfrage nach Änderungen der Anmeldedaten weiterhin fehlschlägt (zum Beispiel aufgrund von Inkonsistenzen bei engineBoots/engineID oder bei nicht RFC-konformen Geräten) oder wenn Sie zur Fehlerbehebung das globale Leeren des SNMP-Cache 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 der Protokollierungsstufe:

```
### Protokollierungsstufe aller Prozesse erhöhen:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Protokollierungsstufe des zweiten Poller-Prozesses erhöhen:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Protokollierungsstufe des Prozesses mit PID 1234 erhöhen:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### Protokollierungsstufe aller http poller-Prozesse verringern:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Beispiel für das 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 als derjenige Nicht-Root-Benutzer ausgeführt, als der er gestartet wird. Daher können Sie den Server problemlos als jeden beliebigen 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 Server-Konfigurationsdatei zugreifen, und jeder Benutzer mit Admin-Rechten in Zabbix kann dann beispielsweise das Datenbankpasswort relativ leicht auslesen.

Konfigurationsdatei

Siehe die Optionen der Konfigurationsdatei für Details zur Konfiguration von zabbix_server.

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` - Alarm-DB-Schreiber
- `alerter` - Prozess zum Senden von Benachrichtigungen
- `availability manager` - Prozess für Aktualisierungen der Host-Verfügbarkeit
- `browser poller` - Poller für Browser-Datenpunkt-Prüfungen
- `configuration syncer` - Prozess zur Verwaltung des In-Memory-Caches von Konfigurationsdaten
- `configuration syncer worker` - Prozess zum Auflösen und Synchronisieren von Benutzermakro-Werten in Datenpunkt-Namen
- `connector manager` - Manager-Prozess für Konnektoren
- `connector worker` - Prozess zur Verarbeitung von Anfragen vom Connector Manager
- `discovery manager` - Manager-Prozess für die Geräteerkennung
- `discovery worker` - Prozess zur Verarbeitung von Erkennungsaufgaben vom Discovery Manager
- `escalator` - Prozess für die Eskalation von Aktionen
- `ha manager` - Prozess zur Verwaltung der Hochverfügbarkeit
- `history poller` - Prozess zur Verarbeitung berechneter Prüfungen, die eine Datenbankverbindung erfordern
- `history syncer` - Verlaufs-DB-Schreiber
- `housekeeper` - Prozess zum Entfernen veralteter Daten (Datenpunkt-Verlauf 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 Web-Monitoring
- `icmp pinger` - Poller für icmpping-Prüfungen
- `internal poller` - Poller für interne Prüfungen

- `ipmi manager` - IPMI-Poller-Manager
- `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 Proxy-Lastverteilung und Hochverfügbarkeit
- `report manager`- Manager für Aufgaben zur geplanten Berichtserstellung
- `report writer` - Prozess zur Erstellung geplanter Berichte
- `self-monitoring` - Prozess zum Sammeln interner Server-Statistiken
- `service manager` - Prozess zur Verwaltung von Services durch Empfang von Informationen über Probleme, Problem-Tags und Problembehebung von 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
- `task manager` - Prozess zur Remote-Ausführung von Aufgaben, die von anderen Komponenten angefordert werden (z. B. Problem schließen, Problem bestätigen, Datenpunkt-Wert jetzt prüfen, Remote-Befehlsfunktionalität)
- `timer` - Timer für die Verarbeitung von Wartungszeiträumen
- `trapper` - Trapper für aktive Prüfungen, Traps, Proxy-Kommunikation
- `trigger housekeeper` - Prozess zum Entfernen von Problemen und Ereignissen, die von Auslösern erzeugt wurden, die inzwischen gelöscht wurden
- `unreachable poller` - Poller für nicht erreichbare Geräte
- `vmware collector` - VMware-Datensammler, der für die Datenerfassung von VMware-Services verantwortlich ist

Die Server-Logdatei kann verwendet werden, um diese Prozesstypen zu beobachten.

Die Server-Logdatei wird mit Lese- und Schreibrechten nur für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen werden verweigert.

Verschiedene Typen von Zabbix-Server-Prozessen können mit dem internen **zabbix[process,<type>,<mode>,<state>]**-Datenpunkt überwacht werden.

Transaktionsstatistiken des History-Syncers

Der Prozesstitel 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
```

In „A+B Auslöser“:

- A - Auslöser, die aufgrund von Verlaufswerten verarbeitet wurden;
- B - Auslöser, die aufgrund von Timern verarbeitet wurden.

Die Zeitangaben in „processed...in N (<timings>) sec“ sind:

- Zeit für das Schreiben von Datenpunkt-Werten in die Datenbank;
- Zeit für die Aktualisierung von Datenpunkt-Daten (Status, Fehler, Host-Inventar usw.);
- Zeit für das Schreiben von Trends in die Datenbank;
- Zeit für die Berechnung von Auslösern;
- Zeit für die Verarbeitung von Ereignissen und Aktionen.

Housekeeping-Verfahren

Der Prozess `housekeeper` entfernt periodisch veraltete Daten (Datenpunkt-Verlauf 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. Daten, die in einem Zyklus nicht entfernt werden, werden in den nächsten übernommen. Automatisches Housekeeping kann unter *Administration* > *Housekeeping* aktiviert und konfiguriert werden.

Zum Entfernen von Daten, die von gelöschten Objekten zurückgelassen wurden, verwendet der Prozess `housekeeper` Aufgaben aus der Tabelle `housekeeper`, die jedes Mal befüllt wird, wenn ein Objekt gelöscht wird. Wenn Sie beispielsweise einen Host löschen, löscht Zabbix auch dessen Datenpunkte, jedoch nicht deren Verlauf, Trends oder Probleme. Stattdessen befüllen Datenbank-Trigger die Tabelle `housekeeper` mit Aufgaben, die aus folgenden Feldern bestehen:

- housekeeperid - Aufgaben-ID
- object - Objekttyp (0 - Datenpunkt; 1 - Auslöser; 2 - Service; 3 - entdeckter Host; 4 - entdeckter Service)
- objectid - Objekt-ID (hilft dem housekeeper, objektbezogene Daten zu finden)

Wenn Sie beispielsweise einen Host mit zwei Datenpunkten und einem Auslöser löschen, wird die Tabelle housekeeper wie folgt befüllt:

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

Datenbank-Trigger befüllen die Tabelle housekeeper, ohne auf objektbezogene Daten zu prüfen; diese Prüfung wird vom Prozess housekeeper durchgeführt.

Jede Aufgabe führt zu einer oder mehreren housekeeper-Operationen, die vom Objekttyp abhängen:

- Für Datenpunkte (einschließlich LLD-Regeln) - entfernt Daten aus allen Verlaufs- und Trendtabellen (history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint), die Werte für diese Datenpunkte enthalten. Außerdem wird die Tabelle problems geprüft und veraltete interne Ereignisse entfernt, die mit diesen Datenpunkten verknüpft sind.
- Für Auslöser - prüft ereignisbezogene Tabellen (problem, event_symptom, event_recovery, events) und entfernt veraltete Ereignisse, die mit diesen Auslösern verknüpft sind, und benachrichtigt außerdem den Prozess service manager über entfernte Ereignisse.

Note:
Ein separater Prozess trigger housekeeper übernimmt eine engere Aufgabe – das Entfernen von Problemen und Ereignissen, die keinen bekannten Quell-Auslöser haben. Seine Ausführungshäufigkeit wird durch ProblemHousekeepingFrequency gesteuert.

 Bis das Housekeeping-Verfahren für Auslöser gestartet wird, können Probleme, die durch inzwischen gelöschte Auslöser verursacht wurden, weiterhin Serviceprobleme erzeugen und diese Services zuweisen. Wenn Ihre Umgebung viele Regeln zur Statusberechnung für Services umfasst, die auf häufig entdeckten/nicht mehr entdeckten Auslösern basieren, sollten Sie erwägen, die Häufigkeit des Housekeeping-Verfahrens durch Anpassen des Server-Konfigurationsparameters ProblemHousekeepingFrequency zu erhöhen.

- Für Services - prüft die Tabelle problems und entfernt veraltete Service-Ereignisse sowie veraltete Serviceprobleme und löst diese damit zum Zeitpunkt des Housekeepings.
- Für die Netzwerkdiscovery - entfernt veraltete Discovery-Ereignisse aus der Tabelle problems.

Der housekeeper entfernt nur Ereignisse, die nicht mit Problemen verknüpft sind. Beispielsweise wird ein veraltetes Problem-/Wiederherstellungsereignis nicht entfernt, wenn es mit einem offenen Problem verknüpft ist. Wenn der housekeeper veraltete Entitäten entfernt, entfernt er zuerst Probleme und dann Ereignisse.

Note:
Tabellen, die den Modus partition verwenden (von TimescaleDB partitionierte Tabellen), werden übersprungen; nur Tabellen, die den Modus regular verwenden, werden verarbeitet.

Unterstützte Plattformen

Aufgrund der Sicherheitsanforderungen und des geschäftskritischen Charakters des Server-Betriebs 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 auch auf anderen Unix-ähnlichen Betriebssystemen funktionieren.

Gebietsschema

Beachten Sie, dass der Server ein UTF-8-Gebietsschema benötigt, damit einige textuelle Datenpunkte korrekt interpretiert werden können. Die meisten modernen Unix-ähnlichen Systeme verwenden standardmäßig ein UTF-8-Gebietsschema, jedoch gibt es einige Systeme, bei denen dies ausdrücklich festgelegt werden muss.

1 Hochverfügbarkeit

Überblick

Hochverfügbarkeit (HA) ist in der Regel in kritischen Infrastrukturen erforderlich, die sich praktisch keine Ausfallzeiten leisten können. Daher muss es für jeden Dienst, der ausfallen kann, eine Failover-Option geben, die übernimmt, wenn der aktuelle Dienst ausfällt.

Zabbix bietet eine **native** Hochverfügbarkeitslösung, die einfach einzurichten ist und keine vorherige HA-Expertise erfordert. Native Zabbix-HA kann als zusätzliche Schutzebene gegen Software-/Hardware-Ausfälle des Zabbix Server nützlich sein oder dazu beitragen, Ausfallzeiten aufgrund von Wartungsarbeiten zu verringern.

Im Zabbix-Hochverfügbarkeitsmodus 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 zu übernehmen.

Der Wechsel zu Zabbix-HA ist unverbindlich. Sie können jederzeit wieder zum Standalone-Betrieb zurückkehren.

Siehe auch: Implementierungsdetails

Aktivieren der Hochverfügbarkeit

Starten des Zabbix-Servers als Cluster-Knoten

Zum Starten eines Zabbix-Servers als Cluster-Knoten sind zwei Parameter in der Konfiguration des Servers erforderlich:

- **HANodeName** muss für jeden Zabbix-Server angegeben werden, der ein HA-Cluster-Knoten sein 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.

- **NodeAddress** muss für jeden Knoten angegeben werden.

Der Parameter NodeAddress (Adresse:Port) wird vom Zabbix-Frontend verwendet, um eine Verbindung zum aktiven Server-Knoten herzustellen. NodeAddress muss mit der IP-Adresse oder dem FQDN-Namen des jeweiligen Zabbix-Servers übereinstimmen.

Starten Sie alle Zabbix-Server nach Änderungen an den Konfigurationsdateien neu. Sie werden dann als Cluster-Knoten gestartet. Den neuen Status der Server können Sie unter *Berichte* → *Systeminformationen* einsehen sowie durch Ausführen von:

```
zabbix_server -R ha_status
```

Dieser Laufzeitbefehl protokolliert den aktuellen Status des HA-Clusters im Zabbix-Server-Log (und auf stdout):

```
Failover delay: 60 seconds
Cluster status:
   #  ID                          Name                 Address                 Status      Last Access
   1. ckzxxqg7u0001lsropenyzh3m zabbix-node-01       64.227.66.193:10051     standby     0s
   2. ckzxyqo1k00013frpq539e1jp zabbix-node-02       64.227.74.25:10051      active      3s
```

Frontend vorbereiten

Stellen Sie sicher, dass die Zabbix Server-Adresse:Port in der Frontend-Konfiguration (**nicht definiert**) ist (zu finden in `conf/zabbix.conf.php` im Dateiverzeichnis des Frontends).

```
// 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 Zabbix Server-Adresse verwendet.

Proxy-Konfiguration

HA-Cluster-Knoten (Server) müssen in der Konfiguration eines passiven oder aktiven Zabbix Proxy aufgeführt werden.

Bei einem passiven Proxy müssen die Knotennamen im Parameter Server des Proxy aufgeführt werden, getrennt durch ein **Komma**.

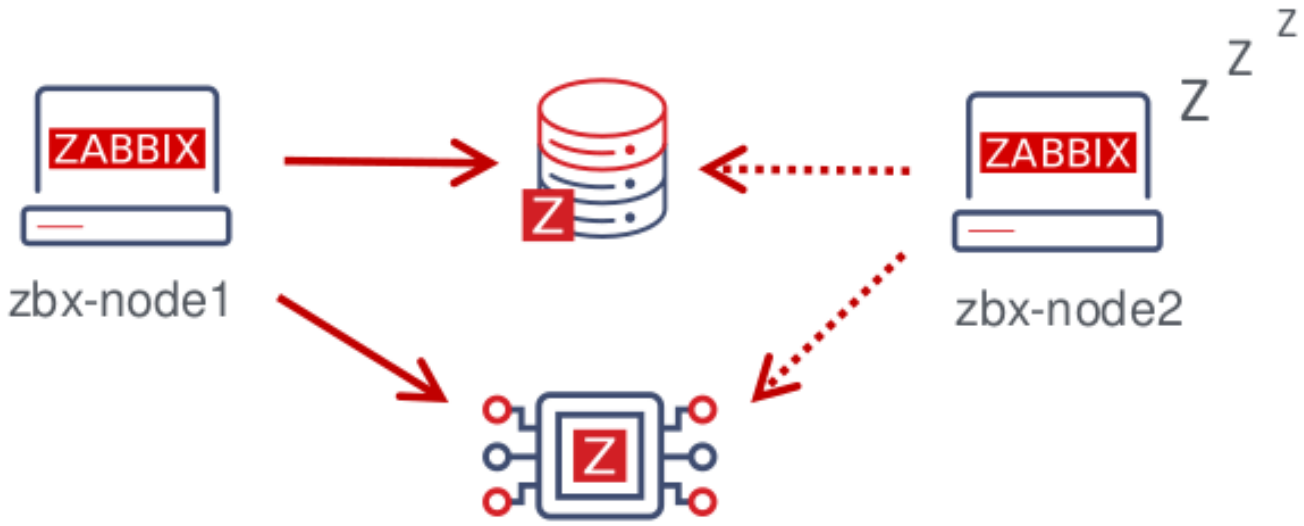
```
Server=zabbix-node-01,zabbix-node-02
```

Bei einem aktiven Proxy müssen die Knotennamen im Parameter Server des Proxy aufgeführt werden, getrennt durch ein **Semikolon**.

```
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 werden.



Um passive Prüfungen zu aktivieren, müssen die Knotennamen im Parameter Server parameter aufgeführt und durch ein **Komma** getrennt werden.

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

Um aktive Prüfungen zu aktivieren, müssen die Knotennamen im Parameter ServerActive parameter aufgeführt werden. Beachten Sie, dass bei aktiven Prüfungen die Knoten durch ein Komma von allen anderen Servern getrennt werden 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 ausfällt. Damit das Failover stattfinden kann, muss sich mindestens ein Knoten im Standby-Status befinden.

Wie schnell erfolgt das Failover? Alle Knoten aktualisieren ihre letzte Zugriffszeit (und ihren Status, falls er geändert wurde) alle 5 Sekunden. Das bedeutet:

- Wenn der aktive Knoten herunterfährt und seinen Status noch 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 delay** + 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
```

Verwalten des HA-Clusters

Der aktuelle Status des HA-Clusters kann mit den dafür vorgesehenen runtime control-Optionen verwaltet werden:

- `ha_status` - HA-Cluster-Status im Zabbix-Server-Log protokollieren (und auf stdout ausgeben)
- `ha_remove_node=target` - einen HA-Knoten entfernen, der durch <target> identifiziert wird - Name oder ID des Knotens (Name/ID kann aus der Ausgabe von ha_status ermittelt werden), z. B.:

```
zabbix_server -R ha_remove_node=zabbix-node-02
```

Beachten Sie, dass aktive/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. 10s, 1m)

Der Knotenstatus kann überwacht werden:

- in *Berichte* → *Systeminformationen*
- im Dashboard-Widget *Systeminformationen*
- mit der `ha_status`-runtime control-Option des Servers (siehe oben).

Der interne Datenpunkt `zabbix[cluster,discovery,nodes]` kann für die Knotenerkennung verwendet werden, da er JSON mit Informationen zu den High-Availability-Knoten zurückgibt.

Deaktivieren des HA-Clusters

So deaktivieren Sie einen Hochverfügbarkeits-Cluster:

- Erstellen Sie Sicherungskopien der Konfigurationsdateien.
- Stoppen Sie die Standby-Knoten.
- Entfernen Sie den Parameter HANodeName vom aktiven primären Server.
- Starten Sie den primären Server neu (er wird im Standalone-Modus gestartet).

Upgrade des HA-Clusters

Um ein Upgrade auf eine Hauptversion für die HA-Knoten durchzuführen:

- stoppen Sie alle Knoten;
- erstellen Sie eine vollständige Datenbanksicherung;
- wenn die Datenbank Replikation verwendet, stellen Sie sicher, dass alle Knoten synchronisiert sind und keine Probleme aufweisen. Führen Sie kein Upgrade durch, wenn die Replikation fehlerhaft ist.
- wählen Sie einen einzelnen Knoten aus, der das Datenbank-Upgrade durchführen wird, ändern Sie seine Konfiguration in den Standalone-Modus, indem Sie HANodeName auskommentieren, und upgraden Sie ihn;
- stellen Sie sicher, dass das Datenbank-Upgrade vollständig abgeschlossen ist (*System information* sollte anzeigen, dass der Zabbix Server läuft);
- starten Sie den Knoten im HA-Modus neu;
- upgraden Sie die übrigen Knoten 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 Upgrade auf eine Nebenversion reicht es aus, den ersten Knoten zu upgraden, sicherzustellen, dass er aktualisiert wurde und läuft, und dann das Upgrade des nächsten Knotens zu starten.

Implementierungsdetails

Der Hochverfügbarkeits-(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 hat keine spezifischen Anforderungen an die Datenbanken, die Zabbix unterstützt. Benutzer können frei entscheiden, ob sie die native Zabbix-HA-Lösung oder eine HA-Lösung eines Drittanbieters verwenden, je nachdem, was den Anforderungen an die Hochverfügbarkeit 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 sein (arbeiten). Ein Standby-Knoten führt nur einen einzigen Prozess aus – den HA-Manager. Ein Standby-Knoten führt keine Datenerfassung, Verarbeitung oder andere reguläre Server-Aktivitäten aus; er lauscht nicht auf Ports und 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 mehr als 'failover delay' Sekunden beträgt, wechselt der Standby-Knoten selbst in den aktiven Zustand und weist dem zuvor aktiven Knoten den Status 'nicht verfügbar' zu.

Der aktive Knoten überwacht seine eigene Datenbankkonnektivität – wenn diese für mehr als `failover delay`−5 Sekunden verloren geht, 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 mehr als 'failover delay' Sekunden beträgt, wird dem Standby-Knoten der Status 'nicht verfügbar' zugewiesen.

Die Knoten sind so ausgelegt, dass sie über kleinere Zabbix-Versionen hinweg kompatibel sind.

2 Agent

Übersicht

Der Zabbix Agent wird auf einem Überwachungsziel eingesetzt, um lokale Ressourcen und Anwendungen aktiv zu überwachen (Festplatten, Arbeitsspeicher, Prozessorstatistiken usw.).

Der Agent erfasst Betriebsinformationen lokal und meldet die Daten zur weiteren Verarbeitung an den Zabbix Server. Im Falle von Ausfällen (z. B. wenn eine Festplatte voll läuft oder ein Dienstprozess abstürzt) kann der Zabbix Server die Administratoren des jeweiligen Rechners, der den Ausfall gemeldet hat, aktiv benachrichtigen.

Zabbix Agents sind dank der Verwendung nativer Systemaufrufe zur Erfassung statistischer Informationen äußerst effizient.

Passive und aktive Prüfungen

Zabbix-Agenten 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 zu überwachenden Datenpunkte vom Server ab (CPU-Auslastung, verfügbarer Speicher usw.), sammelt dann die erforderlichen Daten und sendet regelmäßig neue Werte an ihn 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 (aktiv)".

Unterstützte Plattformen

Informationen zu unterstützten Plattformen finden Sie auf der Seite Requirements.

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 folgen Sie den bereitgestellten Anweisungen.
- Zabbix-Quellcode - laden Sie die Quelldateien herunter und kompilieren Sie den Zabbix Agent, indem Sie den Quellcode konfigurieren und dabei die Option `--enable-agent` verwenden.

Attention:
Im Allgemeinen funktionieren 32-Bit-Zabbix-Agenten auf 64-Bit-Systemen, können jedoch in einigen Fällen fehlschlagen.

Note:
Vorkompilierte Zabbix-Agent-Binärdateien stehen für macOS, IBM AIX, FreeBSD, OpenBSD und Solaris zum Download zur Verfügung. Legacy-Binärdateien, die mit der aktuellen Zabbix Server-/Proxy-Version kompatibel sind, stehen für NetBSD und HP-UX zur Verfügung.

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
```

Manuell starten

Sie können den Zabbix Agent starten, indem Sie die Binärdatei `zabbix_agentd` suchen und direkt ausführen; zum Beispiel:

```
zabbix_agentd
```

Agent auf Windows-Systemen

Der Zabbix Agent wird unter Windows 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 für den Zabbix Agent 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 Erstellen des Zabbix Agent unter Windows.

Weitere Details zur Installation des Zabbix Agent (aus einem ZIP-Archiv) als Windows-Dienst finden Sie auf der Seite Zabbix Agent unter Microsoft Windows.

Optionen

Es ist möglich, mehrere Instanzen des Agent auf einem Host auszuführen. Eine einzelne Instanz kann die Standard-Konfigurationsdatei oder eine in der Befehlszeile angegebene Konfigurationsdatei verwenden. Bei mehreren Instanzen muss jede Agent-Instanz ihre eigene Konfigurationsdatei haben (eine der Instanzen kann die Standard-Konfigurationsdatei verwenden).

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 Konfigurationsdatei angeben, die nicht die Standarddatei ist. Unter UNIX ist der Standardwert /usr/local/etc/zabbix_agentd.conf oder wie durch compile-time Variablen *--sysconfdir* oder *--prefix* festgelegt Unter Windows ist der Standardwert C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Zabbix Agent im Vordergrund ausführen (Standard: true).
-p --print	Bekannte Datenpunkte ausgeben und beenden. *Hinweis*: Um auch Ergebnisse von user parameter zurückzugeben, müssen Sie die Konfigurationsdatei angeben (falls sie sich nicht am Standardspeicherort befindet).
-t --test <item key>	Angegebenen Datenpunkt testen und beenden. *Hinweis*: Um auch Ergebnisse von user parameter zurückzugeben, müssen Sie die Konfigurationsdatei angeben (falls sie sich nicht am Standardspeicherort befindet).
-T --test-config	Konfigurationsdatei validieren und beenden.
-h --help	Hilfeinformationen anzeigen.
-V --version	Versionsnummer anzeigen.
Nur UNIX-Agent	
-R --runtime-control <option>	Administrative Funktionen ausführen. Siehe runtime control.
Nur Windows-Agent	
-m --multiple-agents	Mehrere Agent-Instanzen verwenden (mit den Optionen -i, -d, -s, -x). Zur Unterscheidung der Dienstnamen von Instanzen enthält jeder Dienstname den Wert von Hostname aus der angegebenen Konfigurationsdatei.

Parameter	Beschreibung
-S --startup-type <value>	Den Starttyp des Zabbix-Windows-Agent-Dienstes festlegen. Zulässige Werte: `automatic` - Dienst beim Windows-Start automatisch starten *(Standard)*; `delayed` - Start des Dienstes verzögern, bis die automatisch gestarteten Dienste vollständig gestartet wurden (verfügbar unter Windows Server 2008/Vista und neueren Versionen); `manual` - Dienst manuell starten (durch einen Benutzer oder eine Anwendung); `disabled` - 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	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:

- Ausgabe aller integrierten Agent-Datenpunkte mit Werten
- Testen eines user parameter mit dem Schlüssel "mysql.ping", der in der angegebenen Konfigurationsdatei definiert ist
- Installation eines Dienstes "Zabbix Agent" für Windows unter Verwendung des Standardpfads zur Konfigurationsdatei C:\Program Files\Zabbix Agent\zabbix_agentd.conf
- Installation eines Dienstes "Zabbix Agent [Hostname]" für Windows unter Verwendung der Konfigurationsdatei zabbix_agentd.conf, die sich im selben Ordner wie die Agent-Programmdatei befindet, wobei der Dienstname durch Anhängen des Werts Hostname aus der Konfigurationsdatei eindeutig gemacht wird
- Ändern des Starttyps eines installierten Dienstes "Zabbix Agent" für Windows unter Verwendung der Konfigurationsdatei zabbix_agentd.conf, die sich im selben Ordner wie die Agent-Programmdatei befindet

```
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 Protokollierungsstufe von Agent-Prozessen ändern.

Option	Beschreibung	Ziel
log_level_increase[=<target>]	Protokollierungsstufe 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). Bei größeren Werten geben Sie das Ziel als 'process-type,N' an.
log_level_decrease[=<target>]	Protokollierungsstufe 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:

- Protokollierungsstufe aller Prozesse erhöhen
- Protokollierungsstufe des dritten listener-Prozesses erhöhen
- Protokollierungsstufe des Prozesses mit PID 1234 erhöhen
- Protokollierungsstufe 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

- `active checks` - Prozess zur Durchführung aktiver Prüfungen
- `collector` - Prozess zur Datenerfassung
- `listener` - Prozess zum Lauschen auf passive Prüfungen

Die Agent-Protokolldatei kann verwendet werden, um diese Prozesstypen zu beobachten.

Die Agent-Protokolldatei wird mit Lese- und Schreibberechtigungen nur für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen werden verweigert.

Prozessbenutzer

Der Zabbix Agent unter UNIX ist dafür ausgelegt, als Nicht-Root-Benutzer ausgeführt zu werden. Er wird als derjenige Nicht-Root-Benutzer ausgeführt, als der er gestartet wurde. Daher können Sie den Agent ohne Probleme als jeden beliebigen 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 Agent nur als 'root' ausführen, wenn Sie den Parameter 'AllowRoot' in der Agent-Konfigurationsdatei entsprechend ändern.

Konfigurationsdatei

Ausführliche Informationen zur Konfiguration des Zabbix Agent finden Sie in den Konfigurationsdateioptionen für zabbix_agentd oder den Windows-Agent.

Locale

Beachten Sie, dass der Agent ein UTF-8-Locale benötigt, damit einige textuelle Agent-Datenpunkte den erwarteten Inhalt zurückgeben können. Die meisten modernen Unix-ähnlichen Systeme verwenden standardmäßig ein UTF-8-Locale, jedoch gibt es einige Systeme, bei denen dies ausdrücklich festgelegt werden muss.

Exit-Code

Der Zabbix Agent gibt im Fall eines erfolgreichen Beendens 0 und im Fall eines Fehlers 1 zurück.

3 Agent 2

Überblick

Zabbix Agent 2 ist eine neue Generation des Zabbix Agent, geschrieben in Go (wobei ein Teil des C-Codes vom Zabbix Agent wiederverwendet wurde). Er wurde entwickelt, um:

- die Anzahl der TCP-Verbindungen zu reduzieren.
- eine verbesserte Parallelität von Prüfungen bereitzustellen.
- sich mithilfe von Plugins einfach erweitern zu lassen, die einfache Prüfungen mit minimalem Code ermöglichen und komplexe Prüfungen unterstützen, die aus lang laufenden Skripten und eigenständiger Datenerfassung mit periodischer Berichterstattung bestehen.
- als Ersatz für den Zabbix Agent zu fungieren und dabei alle bisherigen Funktionen zu unterstützen.

Passive und aktive Prüfungen

Zabbix Agent 2 unterstützt passive und aktive Prüfungen, ähnlich wie der Zabbix Agent. Zusätzlich unterstützen aktive Prüfungen von Zabbix Agent 2 flexible/zeitgesteuerte Intervalle und Prüfungsparallelität innerhalb eines aktiven Servers.

Note:
Standardmäßig plant Zabbix Agent 2 nach einem Neustart die erste Datenerfassung 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 Agent auszuführen, setzen Sie den Parameter `ForceActiveChecksOnStart` (auf globaler Ebene) oder `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (wirkt sich nur auf Prüfungen des jeweiligen Plugins aus) in der Konfigurationsdatei. Wenn der Parameter auf Plugin-Ebene gesetzt ist, überschreibt er den globalen Parameter.

Parallelität von Prüfungen

Prüfungen aus verschiedenen Plugins können gleichzeitig ausgeführt werden. Die Anzahl gleichzeitiger Prüfungen innerhalb eines Plugins wird durch die Einstellung für die Plugin-Kapazität begrenzt. Jedes Plugin kann eine fest codierte Kapazitätseinstellung haben (1000 ist der Standardwert), die mit der Einstellung `Plugins.<PluginName>.System.Capacity=N` im *Plugins*-Konfigurationsparameter 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 die Komponente Agent 2 aus (nachdem Sie Ihre Zabbix-Version, die Betriebssystemdistribution und die Betriebssystemversion ausgewählt haben) und folgen Sie den Anweisungen.
- Zabbix-Quellcode - laden Sie die Quelldateien herunter und kompilieren Sie den Agent, indem Sie den Quellcode konfigurieren mit der Option `--enable-agent2`.

Note:
Die Überwachungsfunktionen von Zabbix Agent 2 können mit ladbaren Plugins erweitert werden, die separat verfügbar sind. Weitere Informationen finden Sie unter Ladbare 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
```

Um Zabbix Agent 2 zu stoppen, neu zu starten oder den Status zu prüfen, 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 Binärdatei zabbix_agent2 suchen und direkt ausführen; zum Beispiel:

```
zabbix_agent2
```

Agent 2 auf Windows-Systemen

Zabbix Agent 2 läuft als eigenständiger Prozess; 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 Agent herunter und folgen Sie den Anweisungen auf der Seite Installation des Windows-Agent aus MSI.
- Zabbix-Quellcode - laden Sie die Quelldateien herunter und folgen Sie den Anweisungen auf der Seite Erstellen von Zabbix Agent 2 unter Windows.

Note:
Die Überwachungsfunktionen von Zabbix Agent 2 können mit ladbaren 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 Konfigurationsdatei angeben, die nicht die Standarddatei ist. Unter UNIX ist der Standardwert /usr/local/etc/zabbix_agent2.conf oder wie durch compile-time-Variablen *--sysconfdir* oder *--prefix* festgelegt. Unter Windows ist der Standardwert C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Zabbix Agent im Vordergrund ausführen (Standard: true).
-p --print	Bekannte Datenpunkte ausgeben und beenden. *Hinweis*: Um auch Ergebnisse von user parameter zurückzugeben, müssen Sie die Konfigurationsdatei angeben, falls sie sich nicht am Standardspeicherort befindet.
-t --test <item key>	Angegebenen Datenpunkt testen und beenden. *Hinweis*: Um auch Ergebnisse von user parameter zurückzugeben, müssen Sie die Konfigurationsdatei angeben, falls 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	Agent-Version und Lizenzinformationen ausgeben.
-R --runtime-control <option>	Administrative Funktionen ausführen. Siehe runtime control.
Nur Windows-Agent	
-m --multiple-agents	Mehrere Agent-Instanzen verwenden (mit den Optionen -i, -d, -s, -x). Zur Unterscheidung der Dienstnamen der Instanzen enthält jeder Dienstname den Wert von Hostname aus der angegebenen Konfigurationsdatei.
-S --startup-type <value>	Den Starttyp des Zabbix-Windows-Agent-Dienstes festlegen. Zulässige Werte: `automatic` - Dienst beim Windows-Start automatisch starten; *(Standard)* `delayed` - Start des Dienstes verzögern, bis die automatisch gestarteten Dienste vollständig gestartet wurden; `manual` - Dienst manuell starten (durch einen Benutzer oder eine Anwendung); `disabled` - Dienst deaktivieren, sodass er nicht durch einen Benutzer oder eine Anwendung gestartet werden kann. Sie können diese Option zusammen mit der Option -i verwenden oder separat, 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 anhalten.

Spezifische **Beispiele** für die Verwendung von Befehlszeilenparametern:

- alle integrierten Agent-Datenpunkte mit Werten ausgeben
- einen user parameter mit dem Schlüssel "mysql.ping" testen, der in der angegebenen Konfigurationsdatei definiert ist
- einen Dienst "Zabbix Agent" für Windows unter Verwendung des Standardpfads zur Konfigurationsdatei C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf installieren
- den Starttyp eines installierten Dienstes "Zabbix Agent" für Windows ändern und dabei die Konfigurationsdatei zabbix_agent2.conf verwenden, die sich im selben Ordner wie die Agent-Programmdatei befindet

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

Laufzeitsteuerung

Die Laufzeitsteuerung bietet einige Optionen für die Fernsteuerung.

Option	Beschreibung
log_level_increase	Log-Level erhöhen.
log_level_decrease	Log-Level verringern.
metrics	Verfügbare Metriken auflisten.
version	Agent-Version anzeigen.

Option	Beschreibung
userparameter_reload	Werte der Optionen *UserParameter* und *Include* aus der aktuellen Konfigurationsdatei neu laden.
help	Hilfeinformationen zur Laufzeitsteuerung anzeigen.

Beispiele:

- Log-Level für Agent 2 erhöhen
- Optionen der Laufzeitsteuerung ausgeben

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

Konfigurationsdatei

Die Konfigurationsparameter von Agent 2 sind größtenteils mit dem Zabbix-Agenten kompatibel, mit einigen Ausnahmen.

Neue Parameter	Beschreibung
ControlSocket	Der Pfad des Laufzeit-Control-Sockets. Agent 2 verwendet einen Control-Socket für Laufzeitbefehle.
EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	Diese Parameter werden verwendet, um den persistenten Speicher in Agent 2 für aktive Datenpunkte zu konfigurieren.
ForceActiveChecksOnStart	Legt fest, ob der Agent aktive Prüfungen unmittelbar nach einem Neustart ausführen oder gleichmäßig über die Zeit verteilen soll.
Plugins	Plugins können eigene Parameter haben, im Format `Plugins.<Plugin name>.<Parameter>=<value>`. Ein gemeinsamer Plugin-Parameter ist *System.Capacity*, der das Limit für Prüfungen festlegt, die gleichzeitig ausgeführt werden können.
StatusPort	Der Port, auf dem Agent 2 auf HTTP-Statusanfragen lauscht und eine Liste der konfigurierten Plugins sowie einiger interner Parameter anzeigt
Entfernte Parameter	**Beschreibung**
AllowRoot, *User*	Nicht unterstützt, da Daemonisierung nicht unterstützt wird.
LoadModule, *LoadModulePath*	Ladbare Module werden nicht unterstützt.
StartAgents	Dieser Parameter wurde im Zabbix-Agenten verwendet, um die Parallelität passiver Prüfungen zu erhöhen oder sie zu deaktivieren. In Agent 2 wird die Parallelität auf Plugin-Ebene konfiguriert und kann durch eine Kapazitätseinstellung begrenzt werden.

Weitere Details 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 Mutexe für die Sperrung in OpenSSL bereit. Wenn das Sperren oder Entsperren eines Mutex fehlschlägt, wird eine Fehlermeldung in den Standard- Fehlerausgabestrom (STDERR) geschrieben und Agent 2 wird jeweils mit dem Rückgabecode 2 bzw. 3 beendet.

Agent vs agent 2 comparison

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

Parameter	Zabbix agent	Zabbix agent 2
Programming language	C	Go with some parts in C
Daemonization	yes	by systemd only (yes on Windows)
Supported extensions	Custom loadable modules in C.	Custom plugins in Go.
Requirements		
Supported platforms	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: all desktop and server versions since XP	Linux, Windows: all desktop and server versions, on which a supported Go version can be installed.

Parameter	Zabbix agent	Zabbix agent 2
Supported crypto libraries	GnuTLS 3.1.18 and newer OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x LibreSSL - tested with versions 2.7.4, 2.8.2 (certain limitations apply, see the Encryption page for details).	Linux: OpenSSL 1.0.1 and later. MS Windows: OpenSSL 1.1.1 or later. The OpenSSL library must have PSK support enabled. LibreSSL is not supported.
Monitoring processes		
Processes	A separate active check process for each server/proxy record.	Single process with automatically created threads. The maximum number of threads is determined by the GOMAXPROCS environment variable.
Metrics	**UNIX**: see a list of supported items. **Windows**: see a list of additional Windows-specific items.	**UNIX**: All metrics supported by Zabbix agent. Additionally, the agent 2 provides Zabbix-native monitoring solution for: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd, and other monitoring targets - see a full list of agent 2 specific items. **Windows**: All metrics supported by Zabbix agent, and also net.tcp.service* checks of HTTPS, LDAP. Additionally, the agent 2 provides Zabbix-native monitoring solution for: PostgreSQL, Redis.
Concurrency	Active checks for single server are executed sequentially.	Checks from different plugins or multiple checks within one plugin can be executed concurrently.
Third-party traps	no	yes
Additional features		
Persistent storage	no	yes
Persistent files for log*[] metrics	yes (only on Unix)	no
Log data upload	Can be performed during log gathering to free the buffer.	Log gathering is stopped when the buffer is full, therefore the BufferSize parameter must be at least MaxLinesPerSecond x 2.
Changes user at runtime	yes (Unix-like systems only)	no (controlled by systemd)
User-configurable ciphersuites	yes	no

See also:

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

Built-in plugins

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

Plugin name	Description	Supported item keys	Comments
Agent	Metrics of the Zabbix agent being used.	agent.hostname, agent.ping, agent.version	Supported keys have the same parameters as Zabbix agent keys.
CPU	System CPU monitoring (number of CPUs/CPU cores, discovered CPUs, utilization percentage).	system.cpu.discovery, system.cpu.num, system.cpu.util	Supported keys have the same parameters as Zabbix agent keys.

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

Plugin name	Description	Supported item keys	Comments
Oracle	Oracle Database monitoring.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Install the Oracle Instant Client before using the plugin.
Proc	Process CPU utilization percentage.	proc.cpu.util	Supported key has the same parameters as Zabbix agent key.
Redis	Redis server monitoring.	redis.config, redis.info, redis.ping, redis.slowlog.count	To configure an encrypted connection to Redis, specify TLS parameters in the agent configuration file as a named session or default parameters. TLS parameters cannot be passed as item key parameters. Note that an incorrect or otherwise invalid TLS configuration can prevent Zabbix agent 2 from starting, so verify certificate files, permissions and paths before enabling TLS.

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

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

Plugin name	Description	Supported item keys	Comments
Systemd	Monitoring of systemd services.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP connection availability check.	net.tcp.port	Supported key has the same parameters as Zabbix agent key.
UDP	Monitoring of the UDP services availability and performance.	net.udp.service, net.udp.service.perf	Supported keys have the same parameters as Zabbix agent keys.
Uname	Retrieval of information about the system.	system.hostname, system.sw.arch, system.uname	Supported keys have the same parameters as Zabbix agent keys.
Uptime	System uptime metrics collection.	system.uptime	Supported key has the same parameters as Zabbix agent key.
VFSDev	VFS metrics collection.	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	Supported keys have the same parameters as Zabbix agent keys.
WebCertificate	Monitoring of TLS/SSL website certificates.	web.certificate.get	
WebPage	Web page monitoring.	web.page.get, web.page.perf, web.page.regexp	Supported keys have the same parameters as Zabbix agent keys.
ZabbixAsync	Asynchronous metrics collection.	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	Supported keys have the same parameters as Zabbix agent keys.
ZabbixStats	Zabbix server/proxy internal metrics or number of delayed items in a queue.	zabbix.stats	Supported keys have the same parameters as Zabbix agent keys.
ZabbixSync	Synchronous metrics collection.	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	Supported keys have the same parameters as Zabbix agent keys.

Loadable plugins

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

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

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

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

Plugin name	Description	Supported item keys	Comments
Ceph	Ceph monitoring.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	See also Ceph plugin configuration parameters.
Ember+	Monitoring of Ember+.	ember.get	See also Ember+ plugin configuration parameters.
MongoDB	Monitoring of MongoDB servers and clusters (document-based, distributed database).	mongodb.cfg.discovery, mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session parameters. Currently TLS parameters cannot be passed as item key parameters. See also MongoDB plugin configuration parameters.
MSSQL	Monitoring of MSSQL database.	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	To configure an encrypted connection to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also MSSQL plugin configuration parameters.

Plugin name	Description	Supported item keys	Comments
NVIDIA GPU	Monitoring of NVIDIA GPU.	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	See also NVIDIA GPU plugin configuration parameters.
PostgreSQL	Monitoring of PostgreSQL and its forks.	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also PostgreSQL plugin configuration parameters.

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

4 Proxy

Übersicht

Der Zabbix Proxy ist ein Prozess, der Monitoring-Daten von einem oder mehreren überwachten Geräten erfassen und die Informationen an den Zabbix Server senden kann und dabei im Wesentlichen im Auftrag des Servers arbeitet. Alle erfassten Daten werden lokal zwischengespeichert und anschließend an den Zabbix Server übertragen, zu dem der Proxy gehört.

Die Bereitstellung eines Proxy 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 zentrales Monitoring von entfernten Standorten, Niederlassungen und Netzwerken ohne lokale Administratoren.

Ein Zabbix Proxy benötigt eine separate Datenbank.

> **Attention:**
> Beachten Sie, dass mit dem Zabbix Proxy unterstützte Datenbanken SQLite, MySQL und PostgreSQL sind.

Siehe auch: Verwendung von Proxys in einer verteilten Umgebung

Proxy ausführen

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 Genannte nicht funktioniert, müssen Sie ihn manuell starten. Suchen Sie den Pfad zur Binärdatei `zabbix_proxy` und führen Sie Folgendes aus:

```
zabbix_proxy
```

Sie können die folgenden Befehlszeilenparameter mit Zabbix Proxy verwenden:

```
-c --config <file>              Pfad zur Konfigurationsdatei
-f --foreground                 Zabbix Proxy im Vordergrund ausführen
-R --runtime-control <option>   administrative Funktionen ausführen
-T --test-config                Konfigurationsdatei prüfen und beenden
-h --help                       diese Hilfe anzeigen
-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	Konfigurations-Cache neu laden. Wird ignoriert, wenn der Cache gerade geladen wird. Ein aktiver Zabbix Proxy verbindet sich mit dem Zabbix Server und fordert Konfigurationsdaten an. Ein passiver Zabbix Proxy fordert Konfigurationsdaten vom Zabbix Server an, wenn sich der Server das nächste Mal mit dem Proxy verbindet.	
history_cache_clear=<**target**>	History-Cache für den durch seine ID angegebenen Datenpunkt leeren. Betrifft alle Werte des Datenpunkts außer dem ersten und letzten Wert.	**target** - ID des Datenpunkts
diaginfo[=<**section**>]	Diagnoseinformationen in der Proxy-Logdatei sammeln.	**historycache** - Statistiken zum History-Cache **preprocessing** - Statistiken des Präprozessierungsmanagers **locks** - Liste der Mutexe (ist auf *BSD*-Systemen leer)
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.	
log_level_increase[=<**target**>]	Log-Level erhöhen; betrifft alle Prozesse, wenn kein Ziel angegeben ist. Auf *BSD*-Systemen nicht unterstützt.	**Prozesstyp** - Alle Prozesse des angegebenen Typs (z. B. poller) Siehe alle Proxy-Prozesstypen. **Prozesstyp,N** - Prozesstyp und Nummer (z. B. poller,3) **pid** - Prozesskennung (1 bis 65535). Geben Sie bei größeren Werten target als 'Prozesstyp,N' an.
log_level_decrease[=<**target**>]	Log-Level 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/Mutexen nach Funktionsname.	**Prozesstyp** - Alle Prozesse des angegebenen Typs (z. B. history syncer) Siehe alle Proxy-Prozesstypen. **Prozesstyp,N** - Prozesstyp und Nummer (z. B. history syncer,1) **pid** - Prozesskennung (1 bis 65535). Geben Sie bei größeren Werten target als 'Prozesstyp,N' an. **scope** - `rwlock`, `mutex`, `processing` können mit Prozesstyp und Nummer verwendet werden (z. B. history syncer,1,processing) oder mit allen Prozessen eines Typs (z. B. history syncer,rwlock)
prof_disable[=<**target**>]	Profiling deaktivieren. Betrifft alle Prozesse, wenn kein Ziel angegeben ist.	**Prozesstyp** - Alle Prozesse des angegebenen Typs (z. B. history syncer) Siehe alle Proxy-Prozesstypen. **Prozesstyp,N** - Prozesstyp und Nummer (z. B. history syncer,1) **pid** - Prozesskennung (1 bis 65535). Geben Sie bei größeren Werten target als 'Prozesstyp,N' an.

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des Proxy-Konfigurations-Caches:

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

Beispiel für die Verwendung der Laufzeitsteuerung zum Leeren des History-Caches 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 Proxy-Logdatei sammeln:
zabbix_proxy -R diaginfo

### Statistiken zum History-Cache in der Proxy-Logdatei 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-Benutzeroberfläche aktualisiert wird, lädt Zabbix in den meisten Fällen die neuen SNMPv3-Zugangsdaten für diese Schnittstelle automatisch neu; verwenden Sie `-R snmp_cache_reload` nur dann, wenn die Abfrage nach Änderungen der Zugangsdaten weiterhin fehlschlägt (zum Beispiel aufgrund von Inkonsistenzen bei engineBoots/engineID oder nicht RFC-konformen Geräten) oder wenn Sie zur Fehlerbehebung ein globales Leeren des SNMP-Caches erzwingen müssen.

Beispiel für die Verwendung der Laufzeitsteuerung zum Auslösen der Ausführung des Housekeepers:

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

Beispiele für die Verwendung der Laufzeitsteuerung zum Ändern des Log-Levels:

```
### Log-Level aller Prozesse erhöhen:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Log-Level des zweiten poller-Prozesses erhöhen:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Log-Level des Prozesses mit PID 1234 erhöhen:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Log-Level aller http poller-Prozesse verringern:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Prozessbenutzer

Der Zabbix Proxy ist dafür ausgelegt, als Nicht-Root-Benutzer ausgeführt zu werden. Er wird als derjenige Nicht-Root-Benutzer ausgeführt, als der er gestartet wird. Daher können Sie den Proxy problemlos als jeden 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 Proxy nur als „root“ ausführen, wenn Sie den Parameter „AllowRoot“ in der Proxy-Konfigurationsdatei entsprechend anpassen.

Konfigurationsdatei

Siehe die Optionen der Konfigurationsdatei für Details zur Konfiguration von zabbix_proxy.

Proxy-Prozesstypen und Threads

- `agent poller` - asynchroner Poller-Prozess für passive Prüfungen mit einem Worker-Thread
- `availability manager` - Prozess für Host-Verfügbarkeitsaktualisierungen
- `browser poller` - Poller für Browser-Datenpunkt-Prüfungen
- `configuration syncer` - Prozess zur Verwaltung des In-Memory-Caches von Konfigurationsdaten
- `data sender` - Proxy-Datensender
- `discovery manager` - Manager-Prozess für die Erkennung von Geräten
- `discovery worker` - Prozess zur Verarbeitung von Erkennungsaufgaben vom Discovery Manager
- `history syncer` - History-DB-Schreiber
- `housekeeper` - Prozess zum Entfernen veralteter Datenpunkt-Historie
- `http agent poller` - asynchroner Poller-Prozess für HTTP-Prüfungen mit einem Worker-Thread
- `http poller` - Poller für Web-Monitoring
- `icmp pinger` - Poller für icmpping-Prüfungen
- `internal poller` - Poller für interne Prüfungen
- `ipmi manager` - IPMI-Poller-Manager
- `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 Worker-Threads für die Vorverarbeitung
- `preprocessing worker` - Thread für die Datenvorverarbeitung
- `self-monitoring` - Prozess zum Sammeln interner Server-Statistiken
- `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
- `task manager` - Prozess zur Remote-Ausführung von Aufgaben, die von anderen Komponenten angefordert werden (z. B. Problem schließen, Problem quittieren, Datenpunkt-Wert jetzt prüfen, Remote-Befehlsfunktionalität)
- `trapper` - Trapper für aktive Prüfungen, Traps, Proxy-Kommunikation
- `unreachable poller` - Poller für nicht erreichbare Geräte
- `vmware collector` - VMware-Datensammler, der für die Datenerfassung von VMware-Diensten verantwortlich ist

Die Proxy-Logdatei kann verwendet werden, um diese Prozesstypen zu beobachten.

Die Proxy-Logdatei wird mit Lese- und Schreibrechten nur für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen werden verweigert.

Verschiedene Typen von Zabbix-Proxy-Prozessen können mit dem internen **zabbix[process,<type>,<mode>,<state>]**-Datenpunkt überwacht werden.

Transaktionsstatistiken des History-Syncers

Der Prozesstitel 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 zum Schreiben von Datenpunkt-Werten in die Datenbank;
- Zeit zum Aktualisieren von Datenpunkt-Daten (Status, Fehler).

Housekeeping-Verfahren

Der Zabbix Proxy verfügt über den Prozess `housekeeper`, der veraltete Datenpunkt-Verläufe und Trends entfernt. Er wird in Zyklen ausgeführt, wobei die Häufigkeit durch `HousekeepingFrequency` und das Löschlimit pro Zyklus durch `ProxyLocalBuffer` sowie `ProxyOfflineBuffer` bestimmt wird. Anders als beim Housekeeping-Verfahren des Zabbix Server verwendet der Proxy-Prozess `housekeeper` nicht die Tabelle `housekeeper` – stattdessen löscht er alle veralteten Daten einmal pro Housekeeping-Zyklus.

Unterstützte Plattformen

Zabbix Proxy läuft auf derselben Liste von unterstützten Plattformen wie der Zabbix Server.

Speicherpuffer

Der Speicherpuffer ermöglicht es, neue Daten (Datenpunkt-Werte, Netzwerkdiscovery, automatische Host-Registrierung) im Puffer zu speichern und an den Zabbix Server hochzuladen, ohne auf die Datenbank zuzugreifen. Der Speicherpuffer wurde seit Zabbix 7.0 für den Proxy eingeführt.

In Installationen vor Zabbix 7.0 wurden die erfassten Daten vor dem Hochladen an den Zabbix Server in der Datenbank gespeichert. Für diese Installationen bleibt dies auch nach dem Upgrade auf Zabbix 7.0 das Standardverhalten.

Für eine optimierte Leistung wird empfohlen, die Verwendung des Speicherpuffers auf dem Proxy zu konfigurieren. Dies ist möglich, indem der Wert von ProxyBufferMode von ”disk” (fest kodierter Standardwert für bestehende Installationen) auf ”hybrid” (empfohlen) oder ”memory” geändert wird. Außerdem muss die Größe des Speicherpuffers festgelegt werden (Parameter ProxyMemoryBufferSize).

Im Hybridmodus wird der Puffer vor Datenverlust geschützt, indem nicht gesendete Daten in die Datenbank geschrieben werden, wenn der Proxy gestoppt wird, der Puffer voll ist oder die Daten zu alt sind. Sobald alle Werte in die Datenbank geschrieben wurden, verwendet der Proxy wieder den Speicherpuffer.

Im Speichermodus wird der Speicherpuffer verwendet, es gibt jedoch keinen Schutz vor Datenverlust. Wenn der Proxy gestoppt wird oder der Speicher überfüllt wird, 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 startet, 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 ”0” ist.

Gebietsschema

Beachten Sie, dass der Proxy ein UTF-8-Gebietsschema benötigt, damit einige textuelle Datenpunkte korrekt interpretiert werden können. Die meisten modernen Unix-ähnlichen Systeme verwenden standardmäßig ein UTF-8-Gebietsschema; bei einigen Systemen muss dies jedoch möglicherweise ausdrücklich festgelegt werden.

Berechnung von Warteschlangen während der Wartung

> **Attention:**
> Der Zabbix Proxy kennt keine Wartungszeiträume; siehe Berechnung von Warteschlangen während der Wartung für Details.

5 Java gateway

Überblick

Das Zabbix Java gateway kann aus dem Quellcode oder aus Paketen installiert werden.

Es gibt native Unterstützung für die Überwachung von JMX-Anwendungen in Form eines Zabbix-Daemons namens „Zabbix Java gateway“. Das 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 ist keine zusätzliche Softwareinstallation erforderlich; sie muss lediglich mit der Option `-Dcom.sun.management.jmxremote` in der Befehlszeile 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 aus verwendet werden (Zabbix-Proxys können nicht verkettet werden). Der Zugriff auf jedes Java gateway wird direkt in der Konfigurationsdatei des Zabbix Servers oder Proxys eingerichtet, 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 hat, werden nur die Datenpunkte vom Typ **JMX agent** zur Abfrage an das Java gateway weitergeleitet.

Wenn ein Datenpunkt über den Java gateway aktualisiert werden muss, verbindet sich der Zabbix Server oder Proxy mit dem Java gateway und fordert den Wert an, den der Java gateway wiederum abruft und an den Server oder Proxy zurückgibt. Daher speichert der Java gateway keine Werte im Cache.

Zabbix Server oder Proxy verfügt über einen speziellen Prozesstyp, der sich mit dem Java gateway verbindet und durch die Option **StartJavaPollers** gesteuert wird. Intern startet der Java gateway mehrere Threads, die durch die Option **START_POLLERS** gesteuert werden. Auf der Server-Seite wird eine Verbindung beendet, wenn sie länger als **Timeout** Sekunden dauert, der Java gateway kann jedoch weiterhin damit beschäftigt sein, den Wert vom JMX-Zähler abzurufen. Um dieses Problem zu lösen, gibt es im Java gateway die Option **TIMEOUT**, mit der sich ein Timeout für JMX-Netzwerkoperationen festlegen lässt.

Der Zabbix Server oder Proxy versucht, Anfragen an ein einzelnes JMX-Ziel so weit wie möglich zu bündeln (von den Datenpunkt-Intervallen beeinflusst) und sie zur besseren Leistung über eine einzige Verbindung an das Java gateway zu senden.

Es wird empfohlen, **StartJavaPollers** kleiner oder gleich **START_POLLERS** zu halten, da es andernfalls Situationen geben kann, in denen im Java gateway keine Threads verfügbar sind, um eingehende Anfragen zu verarbeiten; in einem solchen Fall verwendet das Java gateway ThreadPoolExecutor.CallerRunsPolicy, was bedeutet, dass der Haupt-Thread die eingehende Anfrage verarbeitet und vorübergehend keine neuen Anfragen annimmt.

Wenn Sie versuchen, Wildfly-basierte Java-Anwendungen mit dem Zabbix Java gateway zu überwachen, installieren Sie bitte die neueste jboss-client.jar, die auf der Wildfly download page verfügbar ist.

2 Einrichtung aus RHEL-Paketen

Überblick

Wenn die Installation aus Paketen erfolgt ist, helfen Ihnen die folgenden Informationen bei der Einrichtung des Zabbix Java gateway.

Konfiguration und Ausführung des Java gateway

Die Konfigurationsparameter des Zabbix Java gateway können in der Datei angepasst werden:

`/etc/zabbix/zabbix_java_gateway.conf`

Weitere Details finden Sie in den Zabbix Java gateway-Konfigurations- parametern.

So starten Sie das Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

So starten Sie das Zabbix Java gateway beim Booten automatisch:

RHEL 7 und neuer:

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

RHEL vor Version 7:

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

Konfigurieren des Servers für die Verwendung mit Java gateway

Wenn Java gateway eingerichtet ist und läuft, müssen Sie dem Zabbix Server mitteilen, wo Zabbix Java gateway zu finden ist. Dies geschieht durch Angabe der Parameter JavaGateway und JavaGatewayPort in der Server-Konfigurationsdatei. Wenn der Host, auf dem die JMX-Anwendung läuft, 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 für die JMX-Überwachung. Wenn Sie diese jedoch verwenden möchten, müssen Sie die Anzahl der vorab gestarteten Instanzen von Java-Pollern angeben. Dies erfolgt auf dieselbe Weise wie bei regulären Pollern und Trappern.

```
StartJavaPollers=5
```

Vergessen Sie nicht, den Server oder Proxy neu zu starten, nachdem Sie deren Konfiguration abgeschlossen haben.

Debugging des Java gateway

Die Protokolldatei des Zabbix Java gateway 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 `level="info"` in ”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

Siehe die Seite JMX monitoring für weitere Details.

3 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.

Konfiguration und Ausführung 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 Zabbix-Java-gateway-Konfigurations- parametern.

So starten Sie Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

So starten Sie Zabbix Java gateway beim Systemstart automatisch:

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

Konfigurieren des Servers für die Verwendung mit Java gateway

Wenn Java gateway betriebsbereit ist, müssen Sie dem Zabbix Server mitteilen, wo Zabbix Java gateway zu finden ist. Dies geschieht durch Angabe der Parameter JavaGateway und JavaGatewayPort in der Server-Konfigurationsdatei. Wenn der Host, auf dem die JMX-Anwendung läuft, 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 für die JMX-Überwachung. Wenn Sie diese jedoch verwenden möchten, müssen Sie die Anzahl der vorab gestarteten Instanzen von Java-Pollern angeben. Dies erfolgt auf dieselbe Weise wie bei regulären Pollern und Trappern.

```
StartJavaPollers=5
```

Vergessen Sie nicht, den Server oder Proxy neu zu starten, sobald Sie die Konfiguration abgeschlossen haben.

Debugging des Java gateway

Die Protokolldatei des Zabbix Java gateway 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 `level="info"` in ”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

Siehe die Seite JMX monitoring für weitere Details.

Einrichtung aus den Quellen

Übersicht

Wenn Zabbix Java gateway aus den Quellen installiert wurde, helfen Ihnen die folgenden Informationen bei der Einrichtung.

Übersicht der Dateien

Wenn Sie Java gateway aus den Quellen bezogen haben, sollten Sie unter $PREFIX/sbin/zabbix_java eine Sammlung von Shell-Skripten, JAR- und Konfigurationsdateien vorfinden. Die Funktion dieser Dateien wird nachstehend zusammengefasst.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Die JAR-Datei von Java gateway selbst.

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

Abhängigkeiten von 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 von Java gateway.

```
settings.sh
```

Konfigurationsdatei, die von den oben genannten Start- und Stoppskripten eingelesen wird.

Konfiguration und Ausführung von Java gateway

Standardmäßig lauscht Java gateway auf Port 10052. Wenn Sie planen, Java gateway auf einem anderen Port auszuführen, können Sie dies im Skript settings.sh angeben. In der Beschreibung der Java gateway-Konfigurationsdatei finden Sie Informationen dazu, wie Sie diese und andere Optionen festlegen.

> **Warning:**
> Port 10052 ist nicht bei IANA registriert.

Sobald Sie mit den Einstellungen zufrieden sind, können Sie Java gateway durch Ausführen des Startskripts starten:

```
./startup.sh
```

Ebenso können Sie, wenn Sie Java gateway nicht mehr benötigen, das Shutdown-Skript ausführen, um ihn 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 läuft, müssen Sie dem Zabbix Server mitteilen, wo Zabbix Java gateway zu finden ist. Dies geschieht durch Angabe der Parameter JavaGateway und JavaGatewayPort in der Server-Konfigurationsdatei. Wenn der Host, auf dem die JMX-Anwendung läuft, 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 für die JMX-Überwachung. Wenn Sie diese jedoch verwenden möchten, müssen Sie die Anzahl der vorab gestarteten Instanzen von Java-Pollern angeben. Dies erfolgt auf dieselbe Weise wie bei der Angabe regulärer Poller und Trapper.

```
StartJavaPollers=5
```

Vergessen Sie nicht, den Server oder Proxy neu zu starten, sobald Sie die Konfiguration abgeschlossen haben.

Debugging des Java gateway

Falls es Probleme mit dem Java gateway gibt oder eine Fehlermeldung, die Sie zu einem Datenpunkt im Frontend sehen, nicht aussagekräftig genug ist, sollten Sie einen Blick in die Protokolldatei des Java gateway werfen.

Standardmäßig protokolliert das Java gateway seine Aktivitäten in die Datei /tmp/zabbix_java.log mit der Protokollierungsstufe ”info”. Manchmal reichen diese Informationen nicht aus und es werden Informationen auf der Protokollierungsstufe ”debug” benötigt. Um die Protokollierungsstufe zu erhöhen, ändern Sie die Datei lib/logback.xml und setzen Sie das Attribut level des Tags <root> auf ”debug”:

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Beachten Sie, dass im Gegensatz zu Zabbix Server oder Zabbix Proxy kein Neustart des 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 die Protokollierungsstufe 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 Ihren Anforderungen entsprechend an. Weitere Details finden Sie im Logback Manual.

Manchmal ist es zu Debugging-Zwecken nützlich, das Java gateway als Konsolenanwendung statt als Daemon zu starten. Kommentieren Sie dazu die Variable PID_FILE in settings.sh aus. Wenn PID_FILE weggelassen wird, startet das Skript startup.sh das Java gateway als Konsolenanwendung und veranlasst Logback, stattdessen die Datei lib/logback-console.xml zu verwenden, die nicht nur auf der Konsole protokolliert, sondern auch die Protokollierungsstufe ”debug” aktiviert hat.

Beachten Sie abschließend, dass das Java gateway für die Protokollierung SLF4J verwendet. Daher können Sie Logback durch ein Framework Ihrer Wahl ersetzen, indem Sie eine entsprechende JAR-Datei im Verzeichnis lib ablegen. Weitere Details finden Sie im SLF4J Manual.

JMX-Überwachung

Siehe die Seite JMX monitoring für weitere Details.

6 Sender

Überblick

Zabbix sender ist ein Befehlszeilenprogramm, das verwendet werden kann, um Performancedaten zur Verarbeitung an den Zabbix Server zu senden.

Das Programm wird üblicherweise in lang laufenden Benutzerskripten für das regelmäßige Senden von Verfügbarkeits- und Performancedaten verwendet.

Damit Ergebnisse direkt an den Zabbix Server oder Proxy gesendet werden können, muss ein Trapper-Datenpunkt konfiguriert sein.

Siehe auch die Python-Bibliothek für Zabbix, die integrierte Funktionalität bietet, um wie Zabbix sender zu arbeiten.

Zabbix sender ausführen

Ein Beispiel für das Ausführen von Zabbix UNIX sender:

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

wobei:

- z - Host des Zabbix-Servers (es kann auch eine IP-Adresse 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 gesetzt werden.

Zabbix sender kann verwendet werden, um mehrere Werte aus einer Eingabedatei zu senden. Weitere Informationen finden Sie in der Zabbix sender manpage.

Wenn eine Konfigurationsdatei angegeben ist, verwendet Zabbix sender alle Adressen, die im Konfigurationsparameter ServerActive des Agent definiert sind, zum Senden von Daten. Wenn das Senden an eine Adresse fehlschlägt, versucht der Sender, die Daten an die anderen Adressen zu senden. Wenn das Senden von Batch-Daten an eine Adresse fehlschlägt, werden die folgenden Batches nicht an diese Adresse gesendet.

Zabbix sender akzeptiert Zeichenfolgen in UTF-8-Kodierung (sowohl für UNIX-ähnliche Systeme als auch für Windows) ohne Byte Order Mark (BOM) am Anfang der Datei.

Zabbix sender unter Windows kann auf ähnliche Weise ausgeführt werden:

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

Die Echtzeit-Sendeszenarien von zabbix_sender sammeln mehrere an ihn übergebene Werte, die in kurzem zeitlichen Abstand eintreffen, und senden sie in einer einzigen Verbindung an den Server. Ein Wert, der nicht mehr als 0,2 Sekunden vom vorherigen Wert entfernt ist, kann in denselben Stapel aufgenommen werden, die maximale Abfragezeit beträgt jedoch weiterhin 1 Sekunde.

Note:
Zabbix sender wird beendet, wenn in der angegebenen Konfigurationsdatei ein ungültiger Parametereintrag vorhanden ist (der nicht der Notation *parameter=value* folgt).

Ausführen von Zabbix sender mit Low-Level-Discovery

Ein Beispiel für das Ausführen 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 Datenpunkttyp Zabbix trapper haben (in diesem Beispiel mit dem Schlüssel `trapper.discovery.item`).

7 Abrufen

Überblick

Zabbix get ist ein Befehlszeilenprogramm, das zur Kommunikation mit dem Zabbix Agent verwendet werden kann und die benötigten Informationen vom Agent abruft.

Das Programm wird in der Regel zur Fehlerbehebung von Zabbix Agents verwendet.

Siehe auch die Python-Bibliothek für Zabbix, die eine integrierte Funktionalität bietet, um wie Zabbix get zu arbeiten.

Ausführen von Zabbix get

Ein Beispiel für das Ausführen 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 das Ausführen von Zabbix get zum Erfassen einer Zeichenkette 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 Datenpunktschlüssel hier ein Leerzeichen enthält, daher werden Anführungszeichen verwendet, um den Datenpunktschlüssel für die Shell zu kennzeichnen. Die Anführungszeichen sind nicht Teil des Datenpunktschlüssels; sie werden von der Shell entfernt und nicht an den Zabbix Agent übergeben.

Wenn ein Datenpunktschlüssel nicht unterstützt wird, gibt Zabbix get den Exit-Code 1 zurück.

Zabbix get akzeptiert die folgenden Befehlszeilenparameter:

```
-s --host <Host-Name oder IP>             Host-Namen oder IP-Adresse eines Hosts angeben
-p --port <Portnummer>                    Portnummer des auf dem Host laufenden Agent angeben (Standard: 1
-I --source-address <IP-Adresse>          Quell-IP-Adresse angeben
-t --timeout <Sekunden>                   Timeout angeben. Gültiger Bereich: 1-600 Sekunden (Standard: 30
-k --key <Datenpunktschlüssel>            Schlüssel des Datenpunkts angeben, für den der Wert abgerufen we
-P --protocol <Wert>                      Für die Kommunikation mit dem Agent verwendetes Protokoll. Werte
                                            auto - Verbindung über JSON-Protokoll herstellen, bei Bedarf a
                                            json - Verbindung über JSON-Protokoll herstellen
                                            plaintext - Verbindung über Klartextprotokoll herstellen, bei
-h --help                                 Diese Hilfemeldung anzeigen
-V --version                              Versionsnummer anzeigen

--tls-connect <Wert>                      Art der Verbindung zum Agent. Werte:
                                            unencrypted - Verbindung ohne Verschlüsselung herstellen (Stan
                                            psk - Verbindung über TLS und einen Pre-Shared Key herstellen
                                            cert - Verbindung über TLS und ein Zertifikat herstellen
--tls-ca-file <CA-Datei>                  Vollständiger Pfadname zu einer Datei mit den Zertifikaten der c
--tls-crl-file <CRL-Datei>                Vollständiger Pfadname zu einer Datei mit gesperrten Zertifikate
--tls-agent-cert-issuer <Zertifikatsaussteller>   Zulässiger Aussteller des Agent-Zertifikats
--tls-agent-cert-subject <Zertifikatssubjekt> Zulässiges Subjekt des Agent-Zertifikats
--tls-cert-file <Zertifikatsdatei>        Vollständiger Pfadname zu einer Datei mit dem Zertifikat oder de
--tls-key-file <Schlüsseldatei>           Vollständiger Pfadname zu einer Datei mit dem privaten Schlüssel
--tls-psk-identity <PSK-Identität>        Eindeutige, groß-/kleinschreibungssensitive Zeichenkette zur Ide
--tls-psk-file <PSK-Datei>                Vollständiger Pfadname zu einer Datei mit dem Pre-Shared Key
--tls-cipher13 <cipher-string>            Cipher-String für OpenSSL 1.1.1 oder neuer für TLS 1.3. Überschr
--tls-cipher <cipher-string>              GnuTLS-Prioritätsstring (für TLS 1.2 und höher) oder OpenSSL-Cip
```

Siehe auch die Zabbix get manpage für weitere Informationen.

Zabbix get unter Windows kann auf ähnliche Weise ausgeführt werden:

```
zabbix_get.exe [options]
```

8 JS

Übersicht

zabbix_js ist ein Befehlszeilenprogramm, das zum Testen eingebetteter Skripte verwendet werden kann.

Dieses Programm führt ein Benutzerskript mit einem String-Parameter aus und gibt das Ergebnis aus. Skripte werden mit der eingebetteten Zabbix-Scripting-Engine ausgeführt.

Bei Kompilierungs- oder Ausführungsfehlern gibt zabbix_js den Fehler auf stderr aus und wird mit dem Code 1 beendet.

Verwendung

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

zabbix_js akzeptiert die folgenden Befehlszeilenparameter:

```
-s, --script script-file          Geben Sie den Dateinamen des auszuführenden Skripts an. Wenn "-" als Dat
-i, --input input-file            Geben Sie den Dateinamen der Eingabeinformationen an. Der Inhalt wird al
-p, --param input-param           Geben Sie den Eingabeparameter an: die Variable, die als Wert an das Skr
-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 (
-h, --help                        Hilfeinformationen anzeigen.
-V, --version                     Die Versionsnummer anzeigen.
-w <webdriver url>                Aktiviert Browser-Monitoring.
```

Beispiele:

Beispiel 1: Ausführen eines Skripts mit einer Variablen, 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`):

```
Example of input information from the file
```

Verwendung:

```
zabbix_js -s script-file.js -i example.txt
```

Ausgabe: `Example of input information from the file`

Beispiel 3: Ausführen eines Skripts mit aus stdin gelesener Eingabe

Die Datei mit dem Skript (`script-file.js`):

```
return value;
```

Verwendung:

```
zabbix_js -s script-file.js -i -
```

Eingabe (stdin):

```
Example of input from stdin
```

Ausgabe: `Example of input from stdin`

9 Web-Service

Übersicht

Der Zabbix-Webservice ist ein Prozess, der für die Kommunikation mit externen Webservices verwendet wird. Derzeit wird der Zabbix-Webservice zum Erstellen und Senden von geplanten Berichten verwendet; zusätzliche Funktionen sollen künftig hinzugefügt werden.

Der Zabbix-Server verbindet sich über HTTP(S) mit dem Webservice. Der Zabbix-Webservice erfordert, dass Google Chrome auf demselben Host installiert ist; bei einigen Distributionen kann der Dienst auch mit Chromium funktionieren (siehe bekannte Probleme).

Installation

Das offizielle Paket zabbix-web-service ist im Zabbix-Repository verfügbar.

Um den Zabbix-Web-Service aus den Quellen zu kompilieren, geben Sie die Configure-Option `--enable-webservice` an.

Um den Zabbix-Web-Service zu konfigurieren, aktualisieren Sie die Parameter der Konfigurationsdatei `zabbix_web_service.conf`.

Attention:
Es wird dringend empfohlen, die Verschlüsselung zwischen Zabbix Server und Zabbix-Web-Service mithilfe von Zertifikaten einzurichten. Standardmäßig werden die zwischen Zabbix Server und Zabbix-Web-Service übertragenen Daten nicht verschlüsselt, was zu unbefugtem Zugriff führen kann.

5 Konfiguration

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte im Abschnitt „Konfiguration“ zuzugreifen.

1 Konfigurieren einer Vorlage

Übersicht

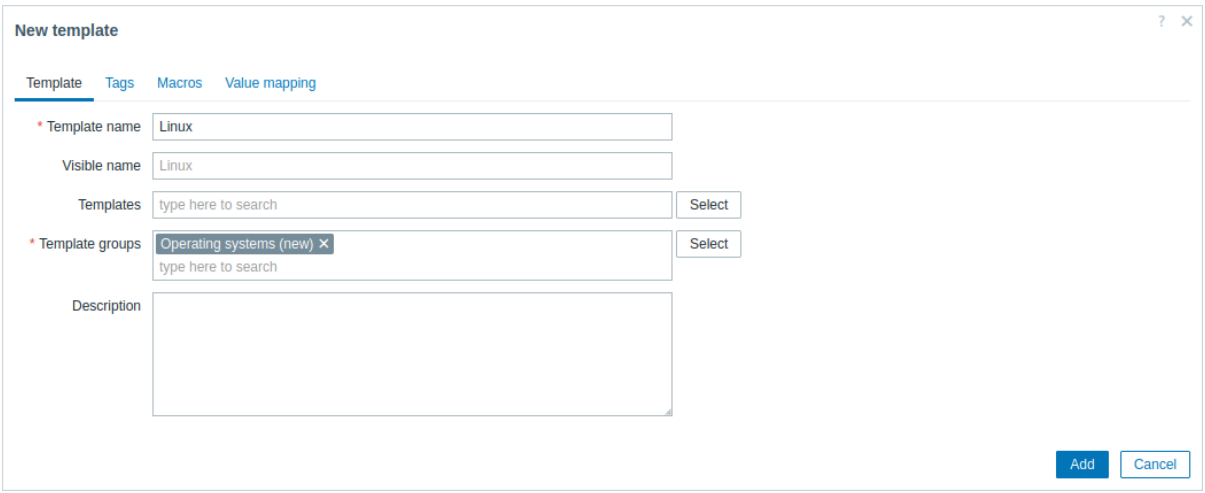
Zum Konfigurieren einer Vorlage müssen Sie zunächst eine Vorlage erstellen, indem Sie ihre allgemeinen Parameter definieren, und ihr anschließend Entitäten (Datenpunkte, Auslöser, Graphen usw.) hinzufügen.

Erstellen einer Vorlage

Gehen Sie wie folgt vor, um eine Vorlage zu erstellen:

1. Gehen Sie zu *Datensammlung* > *Vorlagen*.
2. Klicken Sie auf *Vorlage erstellen*.
3. Bearbeiten Sie die Vorlagenattribute.

Die Registerkarte **Vorlage** enthält allgemeine Vorlagenattribute.



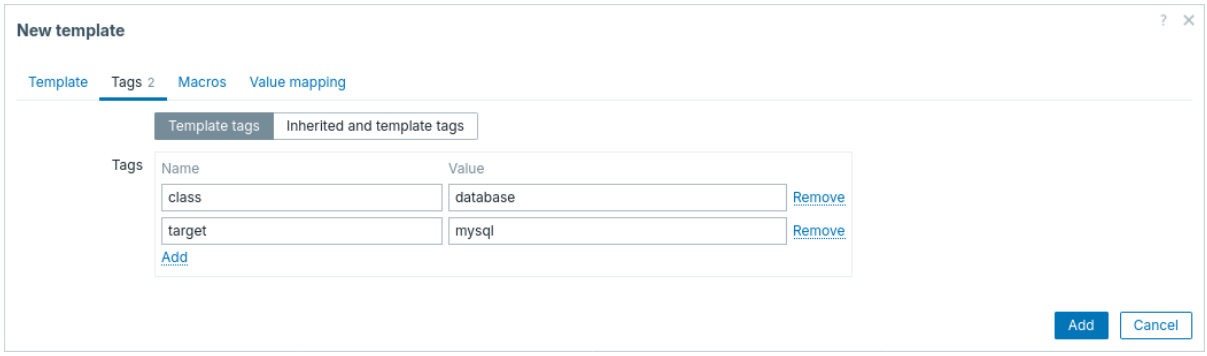
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Vorlagenattribute:

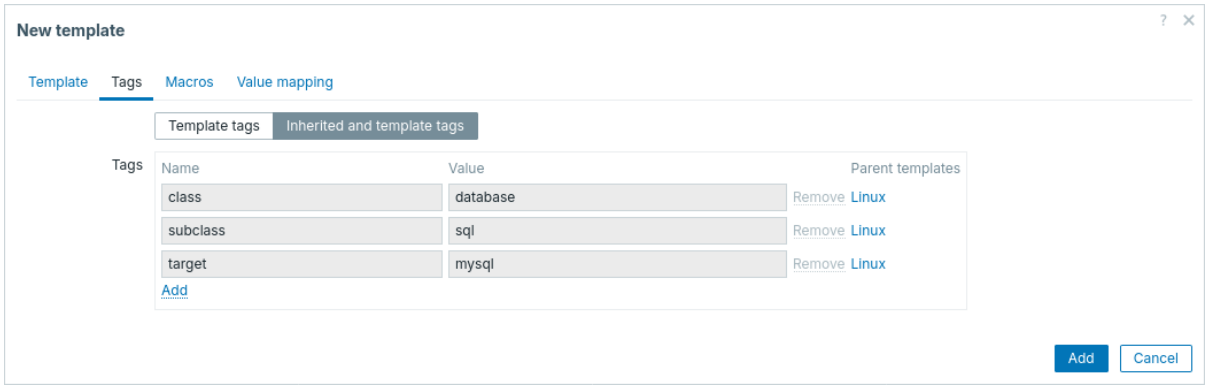
Parameter	Beschreibung
Vorlagenname	Eindeutiger Vorlagenname. Alphanumerische Zeichen, Leerzeichen, Punkte, Bindestriche und Unterstriche sind zulässig. Führende und nachgestellte Leerzeichen sind nicht zulässig.
Sichtbarer Name	Wenn Sie diesen Namen festlegen, wird er in Listen, Karten usw. angezeigt.

Parameter	Beschreibung
Vorlagen	Verknüpfen Sie eine oder mehrere Vorlagen mit dieser Vorlage. Alle Entitäten (Datenpunkte, Auslöser usw.) werden von den verknüpften Vorlagen geerbt. Um eine neue Vorlage zu verknüpfen, geben Sie ihren Namen in das Feld *Vorlagen* ein — dadurch wird eine Dropdown-Liste mit passenden Vorlagen angezeigt. Alternativ klicken Sie auf *Auswählen*, um das Pop-up *Vorlagen* zu öffnen, das zunächst keine Vorlagen anzeigt. Um die Liste zu füllen, geben Sie entweder einen Vorlagengruppennamen in das Feld *Vorlagengruppe* ein (und wählen einen aus der Vorschlagsliste aus) oder drücken die Schaltfläche *Auswählen* neben dem Feld *Vorlagengruppe*, um ein Pop-up *Vorlagengruppen* zu öffnen. Sobald Sie eine Vorlagengruppe auswählen, wird das Pop-up *Vorlagen* aktualisiert und zeigt die Vorlagen an, die zu dieser Gruppe gehören. Die ausgewählten Vorlagen werden verknüpft, wenn die Konfiguration gespeichert oder aktualisiert wird. Um die Verknüpfung einer Vorlage aufzuheben, verwenden Sie eine der beiden Optionen im Block *Vorlagen*: *Verknüpfung aufheben* - hebt die Verknüpfung der Vorlage auf, behält jedoch ihre Entitäten (Datenpunkte, Auslöser usw.) bei; *Verknüpfung aufheben und löschen* - hebt die Verknüpfung der Vorlage auf und entfernt alle ihre Entitäten (Datenpunkte, Auslöser usw.).
Vorlagengruppen	Vorlagengruppen, zu denen die Vorlage gehört.
Beschreibung	Beschreibung der Vorlage.
Anbieter und Version	Anbieter und Version der Vorlage; werden nur beim Aktualisieren vorhandener Vorlagen angezeigt (mitgelieferte Vorlagen, die von Zabbix bereitgestellt werden, importierte Vorlagen oder Vorlagen, die über die Template API geändert wurden), wenn die Vorlagenkonfiguration solche Informationen enthält. Kann im Zabbix Frontend nicht geändert werden. Bei mitgelieferten Vorlagen wird die Version wie folgt angezeigt: Hauptversion von Zabbix, Trennzeichen ("-"), Revisionsnummer (wird mit jeder neuen Version der Vorlage erhöht und bei jeder Hauptversion von Zabbix zurückgesetzt). Zum Beispiel 7.0-0, 7.0-5, 8.0-0, 8.0-3.

Die Registerkarte **Tags** ermöglicht es Ihnen, Vorlagen-Tags zu definieren. Alle Probleme von Hosts, die mit dieser Vorlage verknüpft sind, werden mit den hier eingegebenen Werten getaggt.



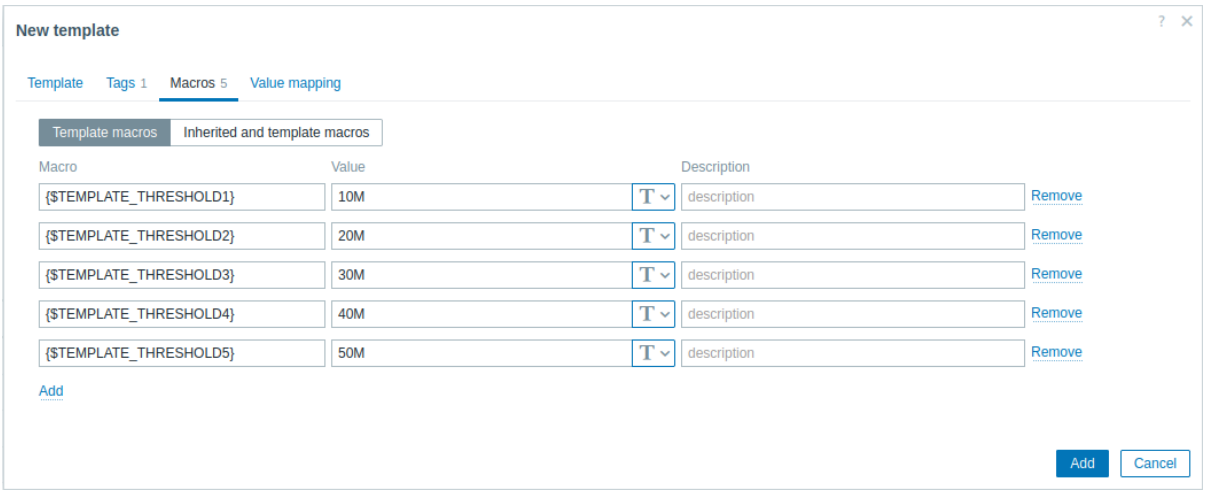
Wenn Sie die Option *Geerbte und Vorlagen-Tags* auswählen, können Sie auch Tags aus verknüpften Vorlagen anzeigen, die von der Vorlage geerbt werden.



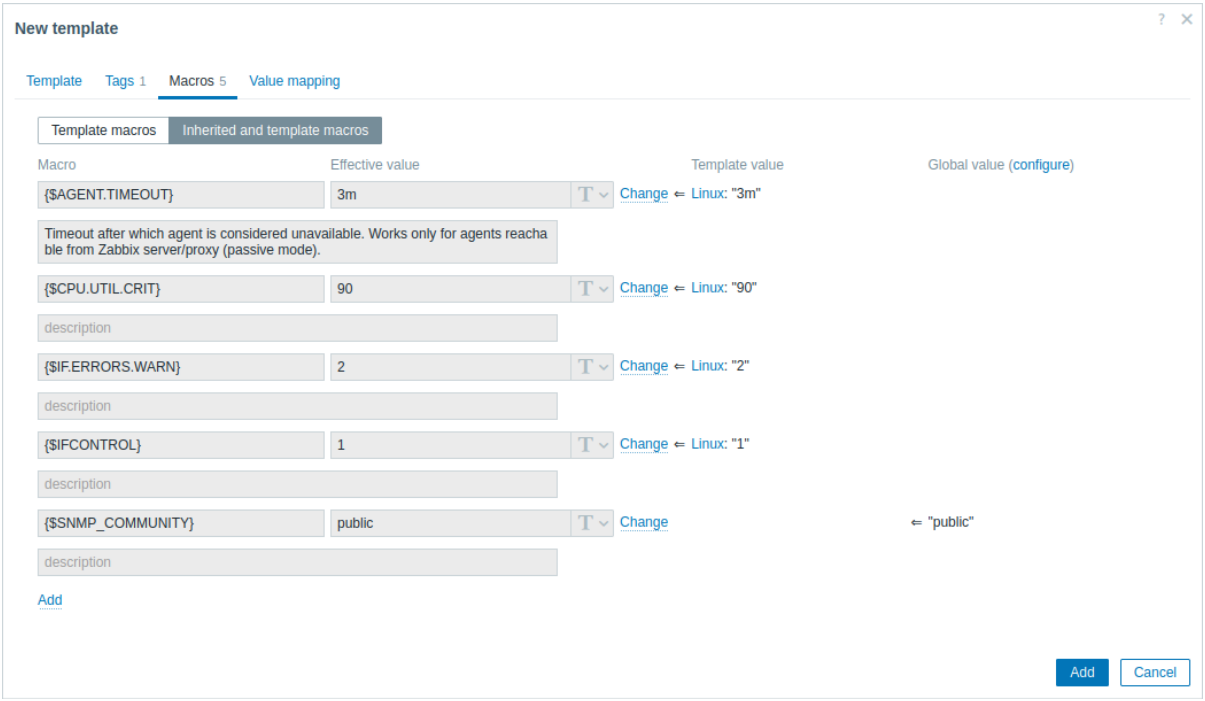
Der Einfachheit halber werden Links zu den jeweiligen Vorlagen bereitgestellt.

Benutzermakros, {INVENTORY.*}-Makros, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID}-Makros werden in Tags unterstützt.

Die Registerkarte **Makros** ermöglicht es Ihnen, Vorlagen-Benutzermakros als Name-Wert-Paare zu 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 der Vorlage geerbt werden, sowie die Werte, zu denen die Makros aufgelöst werden.



Der Einfachheit halber 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, wodurch effektiv eine Kopie des Makros auf der Vorlage erstellt wird.

Die Registerkarte **Wertzuordnung** ermöglicht die Konfiguration einer benutzerfreundlichen Darstellung von Datenpunktdaten in Wertzuordnungen.

Schaltflächen:

Add	Fügt die Vorlage hinzu. Die hinzugefügte Vorlage sollte in der Liste erscheinen.
Update	Aktualisiert die Eigenschaften einer vorhandenen Vorlage.
Clone	Erstellt eine weitere Vorlage auf Grundlage der Eigenschaften der aktuellen Vorlage. Dies **schließt** die Entitäten (Datenpunkte, Auslöser usw.) ein, die sowohl von verknüpften Vorlagen geerbt als auch direkt an die aktuelle Vorlage angehängt sind, **schließt jedoch** Anbieter und Version der aktuellen Vorlage für die geklonte Vorlage aus, damit diese von der ursprünglichen unterschieden werden kann.

Delete	Löscht die Vorlage; Entitäten der Vorlage (Datenpunkte, Auslöser usw.) bleiben bei den verknüpften Hosts erhalten.
Delete and clear	Löscht die Vorlage und alle ihre Entitäten von verknüpften Hosts.
Cancel	Bricht die Bearbeitung der Vorlageneigenschaften ab.

Hinzufügen von Datenpunkten, Auslösern und Diagrammen

Attention:
Datenpunkte müssen zuerst zu einer Vorlage hinzugefügt werden. Auslöser und Diagramme können nicht ohne den entsprechenden Datenpunkt hinzugefügt werden.

Es gibt zwei Möglichkeiten, Datenpunkte zur Vorlage hinzuzufügen:

1. Um neue Datenpunkte zu erstellen, folgen Sie den Anweisungen unter Erstellen eines Datenpunkts.

2. Um vorhandene Datenpunkte zur Vorlage hinzuzufügen:

- Gehen Sie zu *Datenerfassung* > *Hosts* (oder *Vorlagen*).
- Klicken Sie in der Zeile des gewünschten Hosts/der gewünschten Vorlage auf *Datenpunkte*.
- Markieren Sie die Kontrollkästchen der Datenpunkte, die Sie zur Vorlage hinzufügen möchten.
- Klicken Sie unterhalb der Datenpunktliste auf *Kopieren*.
- Wählen Sie die Vorlage (oder 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). Beachten Sie dabei erneut, dass diese nur hinzugefügt werden können, wenn die erforderlichen Datenpunkte zuvor hinzugefügt wurden.

Hinzufügen von Dashboards

Um Dashboards zu einer Vorlage unter *Datensammlung* > *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 unter Konfigurieren von Dashboards.

Attention:
Beim Konfigurieren von Widgets auf einem Vorlagen-Dashboard (anstelle eines globalen Dashboards) sind die Host-bezogenen Parameter nicht verfügbar, und einige Parameter haben eine andere Bezeichnung. Dies liegt daran, dass Vorlagen-Dashboards Daten nur 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* wird in *Show data in maintenance* umbenannt usw. Weitere Informationen zur Verfügbarkeit von Parametern in Widgets von Vorlagen-Dashboards finden Sie bei den jeweiligen Parametern für jedes Dashboard-Widget.

Note:
Einzelheiten zum Zugriff auf Host-Dashboards, die aus Vorlagen-Dashboards erstellt wurden, finden Sie im Abschnitt Host-Dashboards.

Konfigurieren von Low-Level-Discovery-Regeln

Siehe den Abschnitt Low-Level-Discovery im Handbuch.

Webszenarien hinzufügen

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.

2 Konfigurieren einer Vorlagengruppe

Overview

Template groups are used for the logical grouping of templates and assigning user permissions to them.

Each template must have at least one template group assigned. A template may belong to multiple template groups, and each template group may contain multiple templates.

Note that in Zabbix, all permissions are based on groups: user groups, host groups, and template groups. So, even if a single user needs access to a single template, it is granted by adding the user to a user group that has permission to access the template group containing that template.

Konfiguration

> **Attention:**
> Nur Benutzer mit der Rolle Super admin können Vorlagengruppen erstellen.

Es gibt zwei Möglichkeiten, eine Vorlagengruppe im Zabbix Frontend zu erstellen.

Option eins:

- Gehen Sie zu: *Datenerfassung* → *Vorlagengruppen*
- Klicken Sie oben rechts auf dem Bildschirm auf *Vorlagengruppe erstellen*
- Geben Sie den Gruppennamen in das Formular ein

Option zwei: Geben Sie beim Konfigurieren einer Vorlage einen noch nicht vorhandenen Gruppennamen in das Eingabefeld *Vorlagengruppen* ein.

Sobald die Vorlagengruppe erstellt wurde, können Sie in der Liste unter *Datenerfassung* → *Vorlagengruppen* auf den Vorlagennamen klicken, um den Gruppennamen zu bearbeiten, die Gruppe zu klonen oder die Gruppe zu löschen.

Beim Löschen einer Vorlagengruppe wird nur die logische Gruppe gelöscht, nicht die Vorlagen in der Gruppe. Es ist nicht möglich, eine Vorlagengruppe zu löschen, wenn sie die einzige Gruppe für eine vorhandene Vorlage 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.

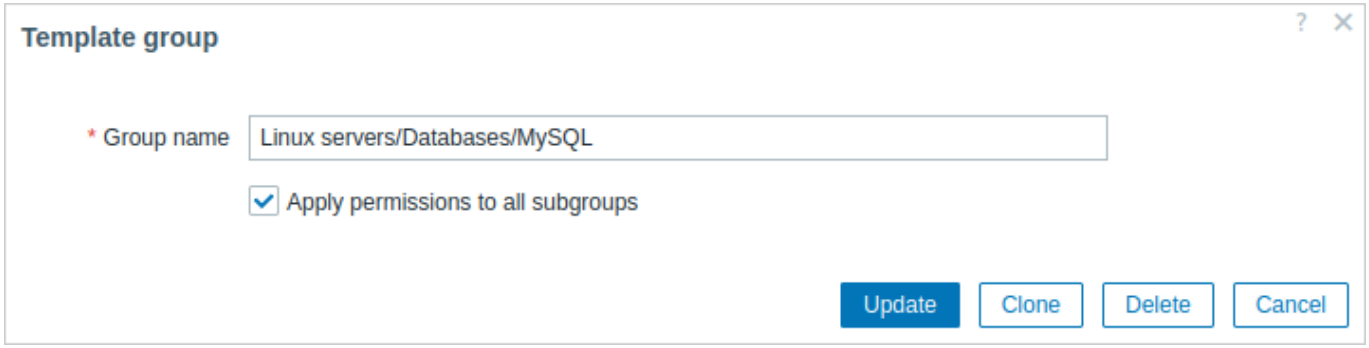
Es ist nicht erforderlich, vor dem Erstellen einer Untergruppe übergeordnete Vorlagengruppen zu erstellen. Sie können wählen, ob Sie mit dem Erstellen einer Untergruppe beginnen (zum Beispiel `Linux servers/Databases`) oder mit einer beliebigen übergeordneten Vorlagengruppe (in unserem Beispiel `Linux servers`). Wenn Sie mit dem Erstellen einer Untergruppe beginnen, werden übergeordnete Vorlagengruppen **nicht** automatisch erstellt.

Berechtigungen für Vorlagengruppen

- Beim Erstellen einer Untergruppe zu einer bestehenden übergeordneten Vorlagengruppe (zum Beispiel beim Erstellen von `Linux servers/Databases`, wenn `Linux servers` bereits existiert) werden die Berechtigungen der Benutzergruppe für die Untergruppe von der übergeordneten Gruppe geerbt.

- Beim Erstellen einer übergeordneten Vorlagengruppe zu einer bestehenden Untergruppe (zum Beispiel beim Erstellen von `Linux servers`, wenn `Linux servers/Databases` bereits existiert) werden für die übergeordnete Gruppe keine Berechtigungen 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 dieselbe Berechtigungsstufe auf alle aktuellen und zukünftigen Untergruppen der bearbeiteten Vorlagengruppe angewendet.

Wenn also Benutzergruppen unterschiedliche Berechtigungen für die Untergruppen der bearbeiteten Vorlagengruppe erhalten haben, werden durch das Aktivieren dieses Kontrollkästchens allen aktuellen und zukünftigen Untergruppen dieselben Benutzerberechtigungen wie der bearbeiteten Gruppe gewährt.

Beachten Sie, dass diese Option nicht in der Datenbank gespeichert wird und bestehende Berechtigungen überschreibt. Alle über diese Option vorgenommenen Änderungen können nur manuell rückgängig gemacht werden.

3 Verknüpfen/Trennen

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

Um eine Vorlage mit dem Host zu verknüpfen, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* → *Hosts*.
2. Klicken Sie auf den gewünschten Host.
3. Beginnen Sie im Feld *Vorlagen* mit der Eingabe des Vorlagennamens. Eine Liste passender Vorlagen wird angezeigt; scrollen Sie nach unten, um eine auszuwählen.
   Alternativ können Sie neben dem Feld auf *Auswählen* klicken und in einem Popup-Fenster eine oder mehrere Vorlagen aus der Liste auswählen.
4. Klicken Sie im Formular für die Host-Attribute auf *Hinzufügen/Aktualisieren*.

Der Host verfügt nun über alle Entitäten der Vorlage.
Dazu gehören Datenpunkte, Auslöser, Diagramme, Low-Level-Discovery-Regeln, Webszenarien sowie Dashboards.

Attention:
Das Verknüpfen mehrerer Vorlagen mit demselben Host schlägt fehl, wenn diese Vorlagen Datenpunkte mit demselben Datenpunktschlüssel enthalten. Da Auslöser und Diagramme Datenpunkte verwenden, können auch sie 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:

- bereits vorhandene identische Entitäten auf dem Host werden als Entitäten der Vorlage aktualisiert, und **alle vorhandenen Anpassungen der Entität auf Host-Ebene gehen verloren**;
- Entitäten aus der Vorlage werden hinzugefügt;
- alle direkt verknüpften Entitäten, die vor der Verknüpfung der Vorlage nur auf dem Host vorhanden waren, bleiben unverändert.

In den Listen sind jetzt alle Entitäten aus der Vorlage mit dem Vorlagennamen vorangestellt, was darauf hinweist, dass sie zu dieser bestimmten Vorlage gehören. Der Vorlagenname selbst (in grauer Schrift) ist ein Link, über den Sie auf die Liste dieser Entitäten auf Vorlagenebene zugreifen können.

Note:
Bei einigen Datenpunkten, wie externen Prüfungen, HTTP-Agent-Prüfungen, einfachen Prüfungen, SSH-Prüfungen und Telnet-Prüfungen, ist eine Host-Schnittstelle optional. Wenn zum Zeitpunkt der Verknüpfung einer Vorlage auf dem Host keine Schnittstelle definiert ist, werden diese Datenpunkte ohne Schnittstelle hinzugefügt. Wenn Sie später eine Host-Schnittstelle hinzufügen, wird sie bereits vorhandenen Datenpunkten nicht automatisch zugewiesen. Um die neu hinzugefügte Host-Schnittstelle allen Datenpunkten der Vorlage auf einmal zuzuweisen, heben Sie die Verknüpfung der Vorlage mit dem Host auf und verknüpfen Sie sie dann erneut. Um die Datenpunkthistorie beizubehalten, verwenden Sie die Option *Verknüpfung aufheben* und nicht *Verknüpfung aufheben und löschen*.

Wenn einer Entität nicht der Vorlagenname vorangestellt ist, bedeutet dies, dass sie bereits zuvor auf dem Host vorhanden war und nicht durch die Vorlage hinzugefügt wurde.

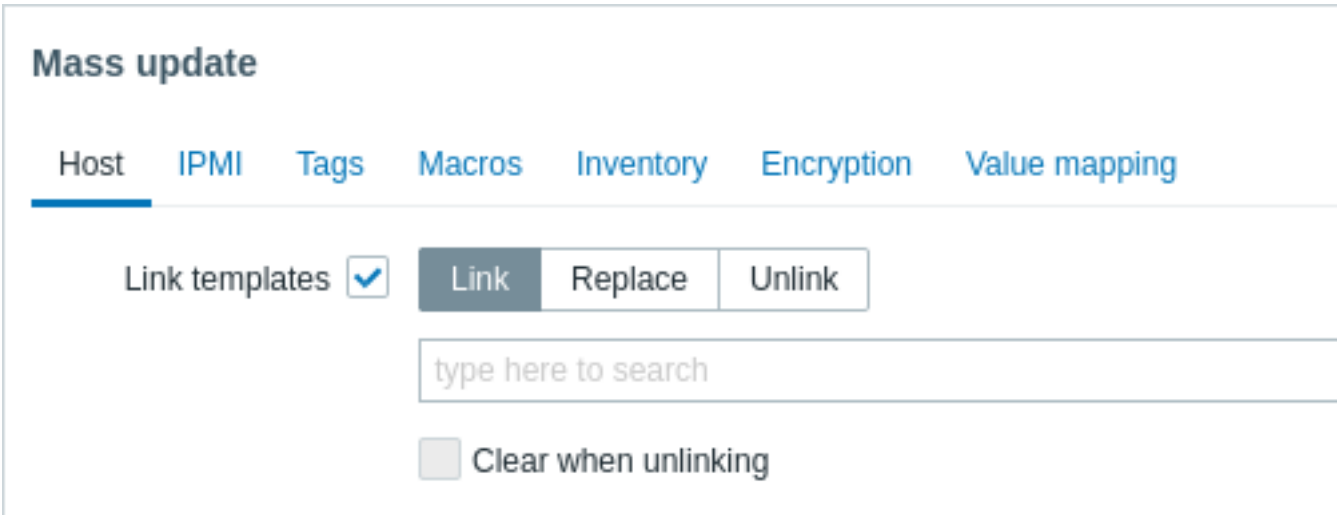
Eindeutigkeitskriterien für Entitäten

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 vieler Hosts zu aktualisieren, wählen Sie unter *Datenerfassung* → *Hosts* einige Hosts aus, indem Sie ihre Kontrollkästchen markieren. Klicken Sie dann unterhalb der Liste auf **Massenaktualisierung** und wählen Sie anschließend *Vorlagen verknüpfen*:



Um zusätzliche Vorlagen zu verknüpfen, beginnen Sie mit der Eingabe des Vorlagennamens in das Autovervollständigungsfeld, bis eine Dropdown-Liste mit passenden Vorlagen erscheint. Scrollen Sie einfach nach unten, um die zu verknüpfende Vorlage auszuwählen.

Mit der Option *Ersetzen* können Sie eine neue Vorlage verknüpfen und gleichzeitig jede Vorlage aufheben, die zuvor mit den Hosts verknüpft war. 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 entfernen (Datenpunkte, Auslöser usw.).

Note:
Zabbix bietet eine umfangreiche Auswahl vordefinierter Vorlagen. Sie können diese als Referenz verwenden, sollten sie jedoch nicht unverändert in der Produktion einsetzen, 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.

Verknüpfung einer Vorlage aufheben

Gehen Sie wie folgt vor, um die Verknüpfung einer Vorlage mit einem Host aufzuheben:

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 *Verknüpfung aufheben* oder *Verknüpfung aufheben und löschen*, um die Verknüpfung der Vorlage aufzuheben.
4. Klicken Sie im Formular der Host-Attribute auf *Aktualisieren*.

Wenn Sie die Option *Verknüpfung aufheben* wählen, wird lediglich die Zuordnung zur Vorlage entfernt, während alle zugehörigen Entitäten beim Host verbleiben. Dazu gehören Datenpunkte, Auslöser, Diagramme, Low-Level-Discovery-Regeln und Webszenarien, jedoch keine Dashboards. Beachten Sie, dass auch Wertezuordnungen und Tags entfernt werden, die von verknüpften Vorlagen geerbt wurden.

Wenn Sie die Option *Verknüpfung aufheben und löschen* wählen, werden sowohl die Zuordnung zur Vorlage als auch alle ihre Entitäten (Datenpunkte, Auslöser usw.) entfernt.

4 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 Vorlagen-Konfigurationsformular.
2. Suchen Sie das Feld *Vorlagen*.
3. Klicken Sie auf *Auswählen*, um das Pop-up-Fenster *Vorlagen* zu öffnen.
4. Wählen Sie im Pop-up-Fenster die erforderlichen Vorlagen aus und klicken Sie dann auf *Auswählen*, um die Vorlagen zur Liste hinzuzufügen.
5. Klicken Sie im Vorlagen-Konfigurationsformular auf *Hinzufügen* oder *Aktualisieren*.

Dadurch werden alle Entitäten der konfigurierten Vorlage sowie alle Entitäten der verknüpften Vorlagen nun in der Vorlagenkonfiguration angezeigt. Dazu gehören Datenpunkte, Auslöser, Diagramme, Low-Level-Discovery-Regeln und Webszenarien, Dashboards sind jedoch ausgeschlossen. Die Dashboards verknüpfter Vorlagen werden jedoch trotzdem von Hosts geerbt.

Um die Verknüpfung einer der verknüpften Vorlagen aufzuheben, klicken Sie im Vorlagen-Konfigurationsformular auf *Verknüpfung aufheben* oder *Verknüpfung aufheben und löschen* und dann auf *Aktualisieren*.

Die Option *Verknüpfung aufheben* entfernt lediglich die Zuordnung zur verknüpften Vorlage, ohne deren Entitäten (Datenpunkte, Auslöser usw.) zu entfernen.

Die Option *Verknüpfung aufheben und löschen* entfernt sowohl die Zuordnung zur verknüpften Vorlage als auch alle ihre Entitäten (Datenpunkte, Auslöser usw.).

5 Massenaktualisierung

Übersicht

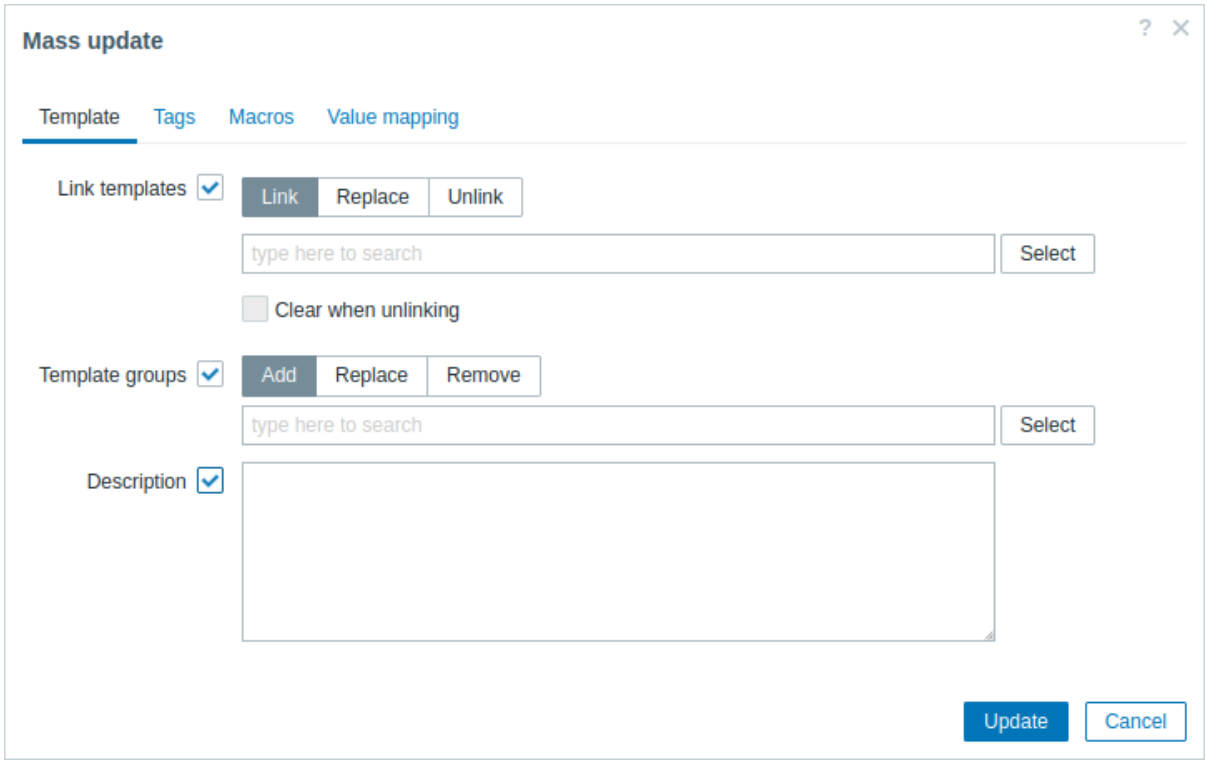
Manchmal möchten Sie ein bestimmtes Attribut für mehrere Vorlagen gleichzeitig ändern. Anstatt jede einzelne Vorlage zur Bearbeitung zu öffnen, können Sie dafür die Funktion zur Massenaktualisierung verwenden.

Massenaktualisierung verwenden

Um mehrere Vorlagen per Massenaktualisierung zu aktualisieren, gehen Sie wie folgt vor:

1. Aktivieren 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. Aktivieren Sie die Kontrollkästchen der Attribute, die aktualisiert werden sollen, und geben Sie dafür einen neuen Wert ein.

Die Registerkarte **Vorlage** enthält allgemeine Optionen für die Massenaktualisierung von Vorlagen.



Für die Aktualisierung *Vorlagen verknüpfen* stehen bei Auswahl der entsprechenden Schaltfläche folgende Optionen zur Verfügung:

- *Verknüpfen* - gibt an, welche zusätzlichen Vorlagen verknüpft werden sollen;
- *Ersetzen* - gibt an, welche Vorlagen verknüpft werden sollen, während gleichzeitig alle zuvor verknüpften Vorlagen getrennt werden;
- *Trennen* - gibt an, welche Vorlagen getrennt werden sollen.

Um die zu verknüpfenden/zu trennenden Vorlagen anzugeben, beginnen Sie im Autovervollständigungsfeld mit der Eingabe des Vorlagennamens, bis eine Dropdown-Liste mit den passenden Vorlagen erscheint. Scrollen Sie einfach nach unten, um die gewünschten Vorlagen auszuwählen.

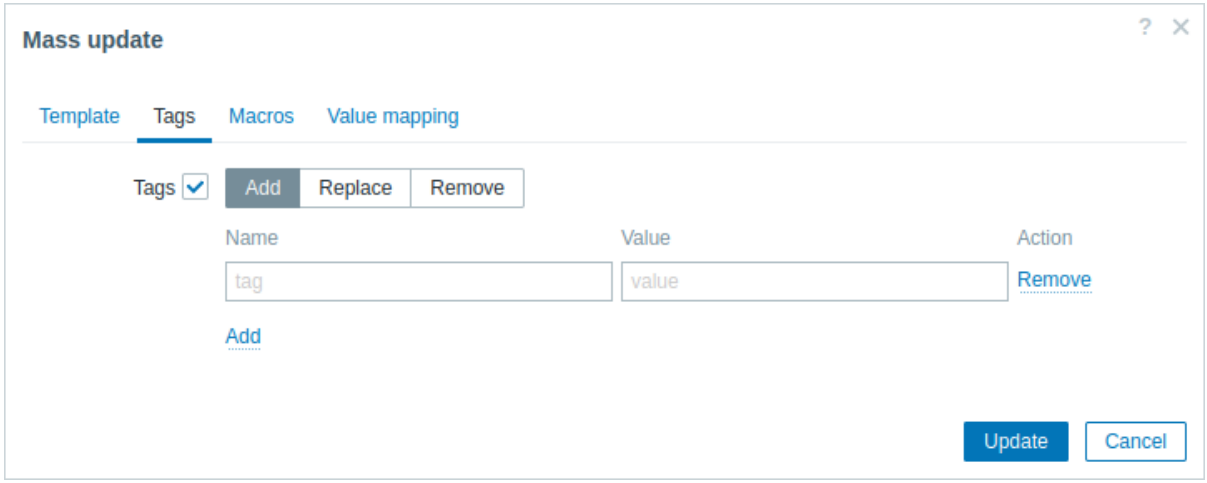
Mit der Option *Beim Trennen löschen* können zuvor verknüpfte Vorlagen getrennt und gleichzeitig alle von ihnen geerbten Elemente entfernt werden (Datenpunkte, Auslöser, Graphen usw.).

Für die Aktualisierung *Vorlagengruppen* stehen bei Auswahl der entsprechenden Schaltfläche folgende Optionen zur Verfügung:

- *Hinzufügen* - ermöglicht es, zusätzliche Vorlagengruppen aus den vorhandenen auszuwählen oder für die Vorlagen vollständig neue Vorlagengruppen einzugeben;
- *Ersetzen* - entfernt die Vorlage aus allen vorhandenen Vorlagengruppen und ersetzt diese durch die in diesem Feld angegebenen Gruppen (vorhandene oder neue Vorlagengruppen);
- *Entfernen* - entfernt bestimmte Vorlagengruppen aus Vorlagen.

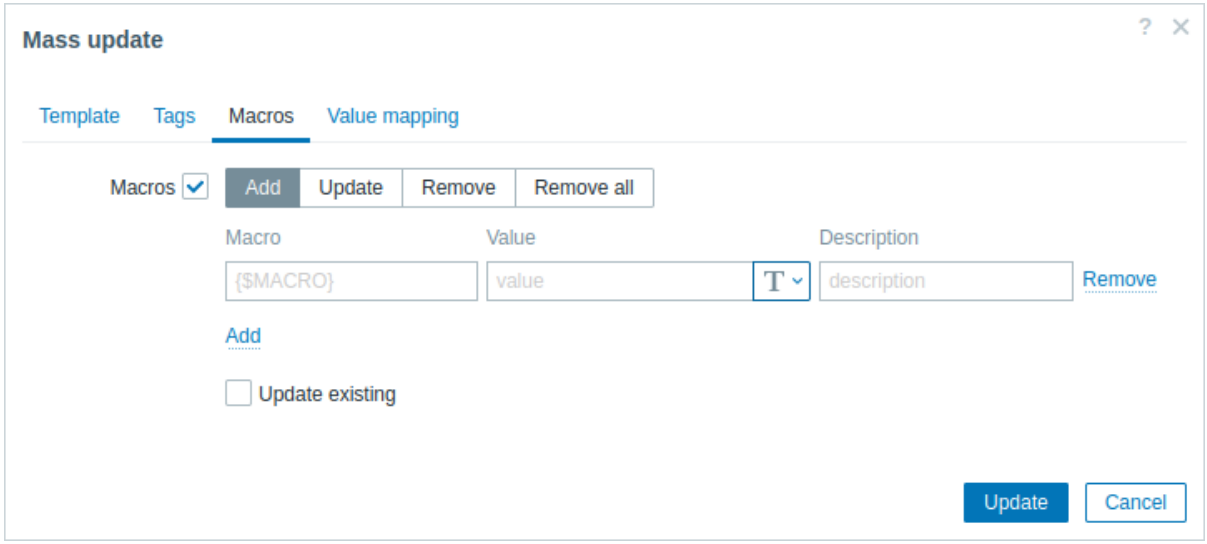
Diese Felder unterstützen die Autovervollständigung - sobald Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit passenden Vorlagengruppen angeboten. Wenn die Vorlagengruppe neu ist, erscheint sie ebenfalls in der Dropdown-Liste und wird durch *(new)* hinter der Zeichenfolge gekennzeichnet. Scrollen Sie einfach nach unten, um sie auszuwählen.

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.

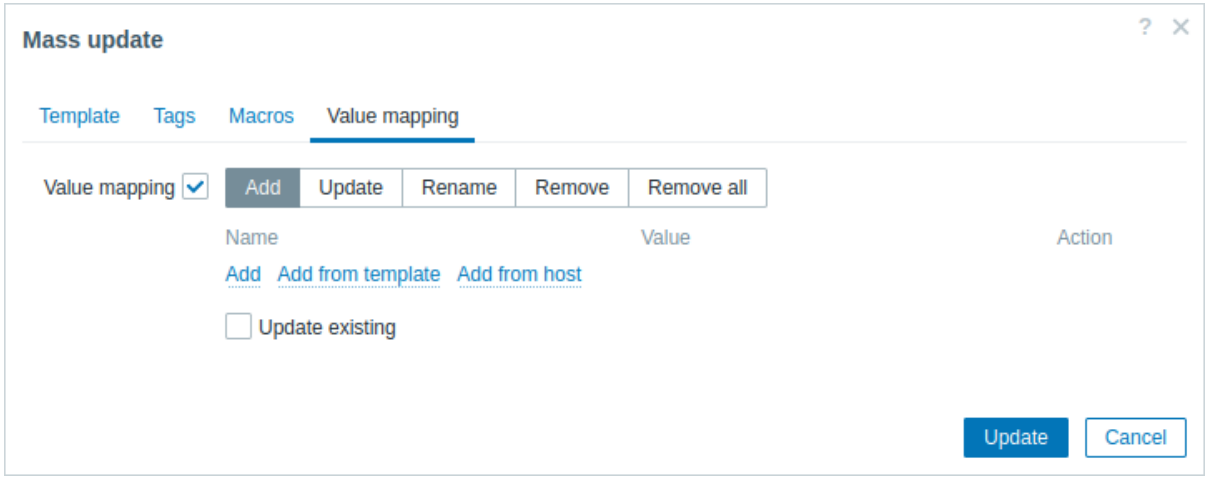
Die Registerkarte **Makros** ermöglicht die Massenaktualisierung von Makros auf Vorlagenebene.



Die folgenden Optionen sind verfügbar, wenn Sie die entsprechende Schaltfläche für die Makro-Aktualisierung auswählen:

- *Hinzufügen* - ermöglicht die Angabe zusätzlicher Benutzermakros für die Vorlagen. Wenn das Kontrollkästchen *Vorhandene aktualisieren* aktiviert ist, werden Wert, Typ und Beschreibung für den angegebenen Makronamen aktualisiert. Ist es nicht aktiviert und ein Makro mit diesem Namen existiert bereits in der/den Vorlage(n), wird es nicht aktualisiert.
- *Aktualisieren* - ersetzt Werte, Typen und Beschreibungen der in dieser Liste angegebenen Makros. Wenn das Kontrollkästchen *Fehlende hinzufügen* aktiviert ist, wird ein Makro, das zuvor in einer Vorlage nicht existierte, als neues Makro hinzugefügt. Ist es nicht aktiviert, werden nur Makros aktualisiert, die bereits in einer Vorlage existieren.
- *Entfernen* - entfernt die angegebenen Makros aus den Vorlagen. Wenn das Kontrollkästchen *Außer ausgewählte* aktiviert ist, werden alle Makros außer den in der Liste angegebenen entfernt. Ist es nicht aktiviert, werden nur die in der Liste angegebenen Makros entfernt.
- *Alle entfernen* - entfernt alle Benutzermakros aus den Vorlagen. Wenn das Kontrollkästchen *Ich bestätige, alle Makros zu entfernen* nicht aktiviert ist, wird ein neues Popup-Fenster geöffnet, in dem die Entfernung aller Makros bestätigt werden muss.

Die Registerkarte **Wertzuordnung** ermöglicht die Massenaktualisierung von Wertzuordnungen.



Für die Aktualisierung von Wertzuordnungen sind Schaltflächen mit den folgenden Optionen verfügbar:

- *Hinzufügen* - Wertzuordnungen zu den Vorlagen hinzufügen. Wenn Sie *Bestehende aktualisieren* markieren, werden alle Eigenschaften der Wertzuordnung mit diesem Namen aktualisiert. Andernfalls wird eine Wertzuordnung mit diesem Namen, falls sie bereits existiert, nicht aktualisiert.
- *Aktualisieren* - bestehende Wertzuordnungen aktualisieren. Wenn Sie *Fehlende hinzufügen* markieren, wird eine Wertzuordnung, die zuvor in einer Vorlage nicht existierte, als neue Wertzuordnung hinzugefügt. Andernfalls werden nur die Wertzuordnungen aktualisiert, die in einer Vorlage bereits existieren.
- *Umbenennen* - einer bestehenden Wertzuordnung einen neuen Namen geben.
- *Entfernen* - die angegebenen Wertzuordnungen aus den Vorlagen entfernen. Wenn Sie *Außer ausgewählte* markieren, werden alle Wertzuordnungen **außer** den angegebenen entfernt.
- *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 die Entfernung bestätigt werden muss.

Die Optionen *Aus Vorlage hinzufügen* und *Von Host hinzufügen* sind für Vorgänge zum Hinzufügen/Aktualisieren von Wertzuordnungen verfügbar.
Sie ermöglichen die Auswahl von Wertzuordnungen aus einer Vorlage bzw. einem Host.

Wenn Sie alle erforderlichen Änderungen vorgenommen haben, klicken Sie auf *Aktualisieren*. Die Attribute werden entsprechend für alle ausgewählten Vorlagen aktualisiert.

1 Hosts und Host-Gruppen

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.

1 Host-Assistent

Übersicht

Der Host Wizard ist eine geführte, schrittweise Benutzeroberfläche zum Einrichten Ihres Überwachungsziels (Gerät, Anwendung, Dienst usw.) in Zabbix. Er führt Sie durch die folgenden Schritte:

- Auswählen einer Vorlage
- Erstellen oder Auswählen eines Hosts
- 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:

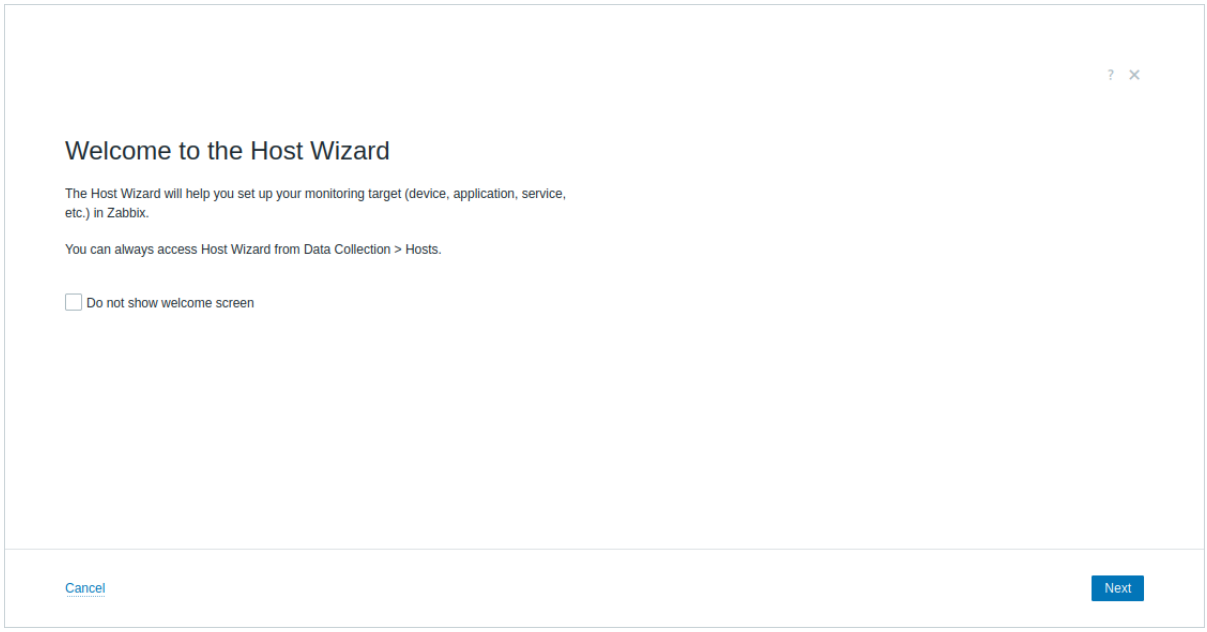
- Gehen Sie zu: *Datenerfassung* > *Hosts*
- Klicken Sie oben rechts auf dem Bildschirm auf *Host Wizard*
- Folgen Sie den Anweisungen im Wizard

Sie können auch einen vorhandenen Host bearbeiten, indem Sie in verschiedenen Frontend-Bereichen auf seinen Namen (oder das Symbol mit den drei Punkten daneben) klicken, um über das Host-Menü auf den Host Wizard zuzugreifen. Beachten Sie, dass beim Bearbeiten eines vorhandenen Hosts einige Schritte übersprungen werden, wenn dieser bereits die von der ausgewählten Vorlage erforderliche Konfiguration enthält.

Alternativ können Sie den klassischen Ansatz zum Erstellen und Konfigurieren eines Hosts verwenden.

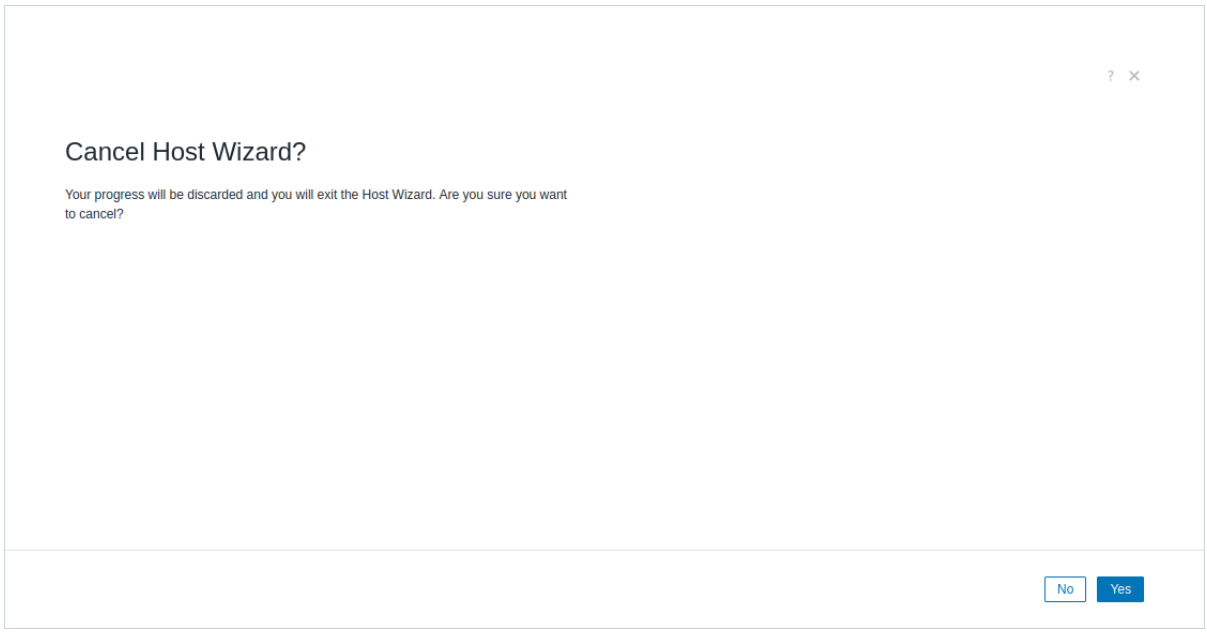
Willkommen beim Host-Assistenten

Wenn Sie den Host-Assistenten starten, wird ein Begrüßungsbildschirm angezeigt:



Um diesen Bildschirm in zukünftigen Sitzungen zu überspringen, können Sie das Kontrollkästchen *Do not show welcome screen* aktivieren und auf *Next* klicken.

Wenn Sie in den nächsten Schritten des Host-Assistenten mit der Konfiguration beginnen und versuchen, den Assistenten durch Klicken auf das Schließen-Symbol oder durch Drücken von ESC zu schließen, wird ein Bestätigungsbildschirm angezeigt:



Durch Klicken auf *Yes* wird der Host-Assistent beendet, ohne dass Ihr Fortschritt gespeichert wird.
Durch Klicken auf *No*, ESC oder das Schließen-Symbol kehren Sie zum letzten Schritt zurück.

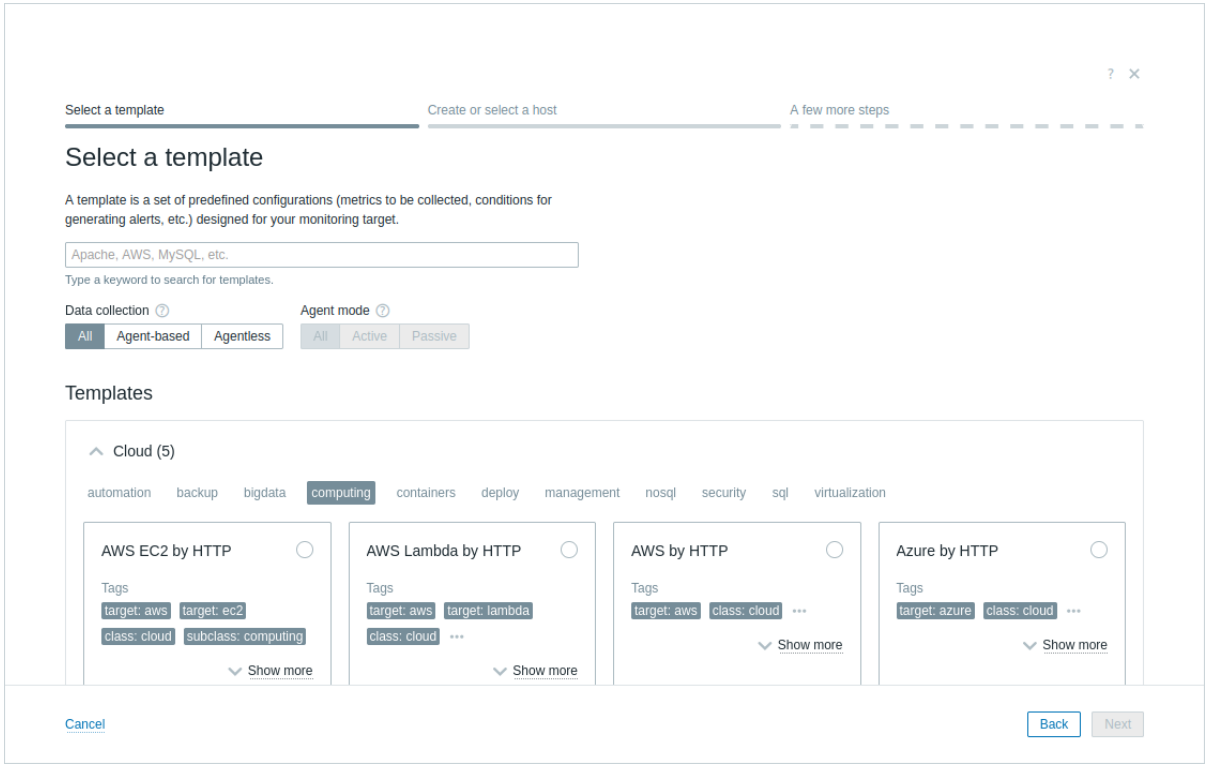
Eine Vorlage auswählen

Der erste Schritt beim Einrichten Ihres Überwachungsziels besteht darin, eine Vorlage auszuwählen – eine Reihe vordefinierter Konfigurationen (zu erfassende Metriken, Bedingungen zum Erzeugen von Warnmeldungen usw.), die für Ihr Überwachungsziel vorgesehen sind.

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 Vorlage basieren.
- Vorlagen anhand von Schlüsselwörtern im Vorlagennamen oder in Namen und Werten von Vorlagen-Tags suchen.
- Vorlagen nach der Methode der Datenerfassung filtern (Agent-basiert oder Agent-los; Agent-basierte Vorlagen enthalten mindestens einen Zabbix Agent-Datenpunkt).
- Vorlagen nach dem Agent-Modus filtern (aktiv oder passiv; aktive Vorlagen enthalten mindestens einen Zabbix-Agent-(aktiv)-Datenpunkt, während passive Vorlagen mindestens einen passiven Datenpunkt enthalten; Einzelheiten finden Sie unter Passive and active agent checks).
- Wenn Sie den Host-Assistenten zum Konfigurieren eines bestehenden Hosts verwenden, können Sie außerdem Vorlagen filtern, die bereits mit dem Host verknüpft sind.



Attention:
Beim Auswählen einer Vorlage wird möglicherweise die folgende Meldung angezeigt: *Einige Vorlagen (n) sind nicht mit dem Host-Assistenten kompatibel. Informationen zum Aktualisieren finden Sie hier. Benutzerdefinierte Vorlagen werden nicht unterstützt.* Dies weist darauf hin, dass einige out-of-the-box-Vorlagen noch nicht mit dem Host-Assistenten kompatibel sind und aktualisiert werden müssen; siehe die Anweisungen zur Aktualisierung von Vorlagen.

Die von Ihnen ausgewählte Vorlage bestimmt die nächsten Schritte im Host-Assistenten. 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 Agenten.

Erstellen oder Auswählen eines Hosts

Eine Vorlage muss mit einem Host verknüpft sein – einer Entität in Zabbix, die Ihr Überwachungsziel repräsentiert. Wenn die Verknüpfung hergestellt ist, erhält der Host alle Vorlagen-Entitäten, wie z. B. Datenpunkte (zu erfassende Metriken) und Auslöser (Bedingungen zum Generieren von Warnmeldungen), die für die Überwachung erforderlich sind.

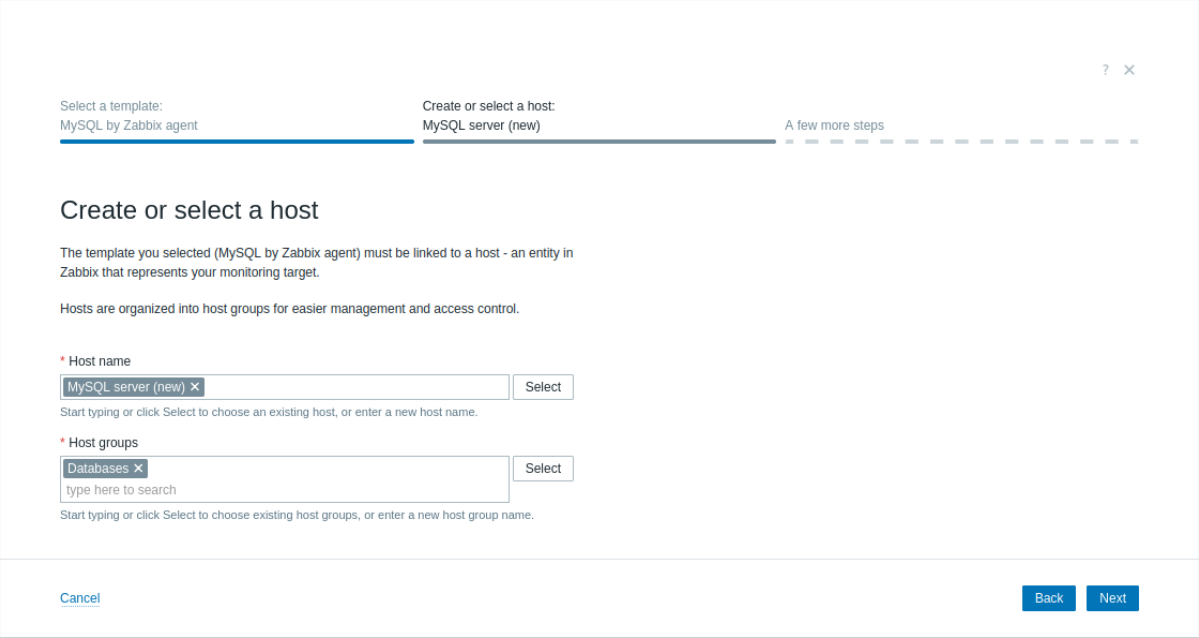
Jeder Host muss außerdem zu mindestens einer Hostgruppe gehören, die zur Organisation von Hosts und zur Zuweisung von Benutzerberechtigungen verwendet wird.

In diesem Schritt können Sie:

- Einen neuen Host und eine neue 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 auch zusätzlichen Hostgruppen zuweisen, ohne ihn aus bestehenden Gruppen zu entfernen.

Note:
Durch das Auswählen eines vorhandenen Hosts wird seine aktuelle Konfiguration nicht entfernt oder überschrieben, sofern dies in späteren Schritten nicht ausdrücklich angegeben ist. Erkannte Hosts können nicht ausgewählt werden.

Sie könnten beispielsweise einen neuen Host mit dem Namen *MySQL server* erstellen, um den lokal installierten MySQL-Server zu repräsentieren, 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 erstellen, z. B. *MySQL servers*), um den Host zusammen mit anderen datenbankbezogenen Hosts zu organisieren und die Berechtigungsverwaltung 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 von der Vorlage erforderlich).

In diesem Schritt:

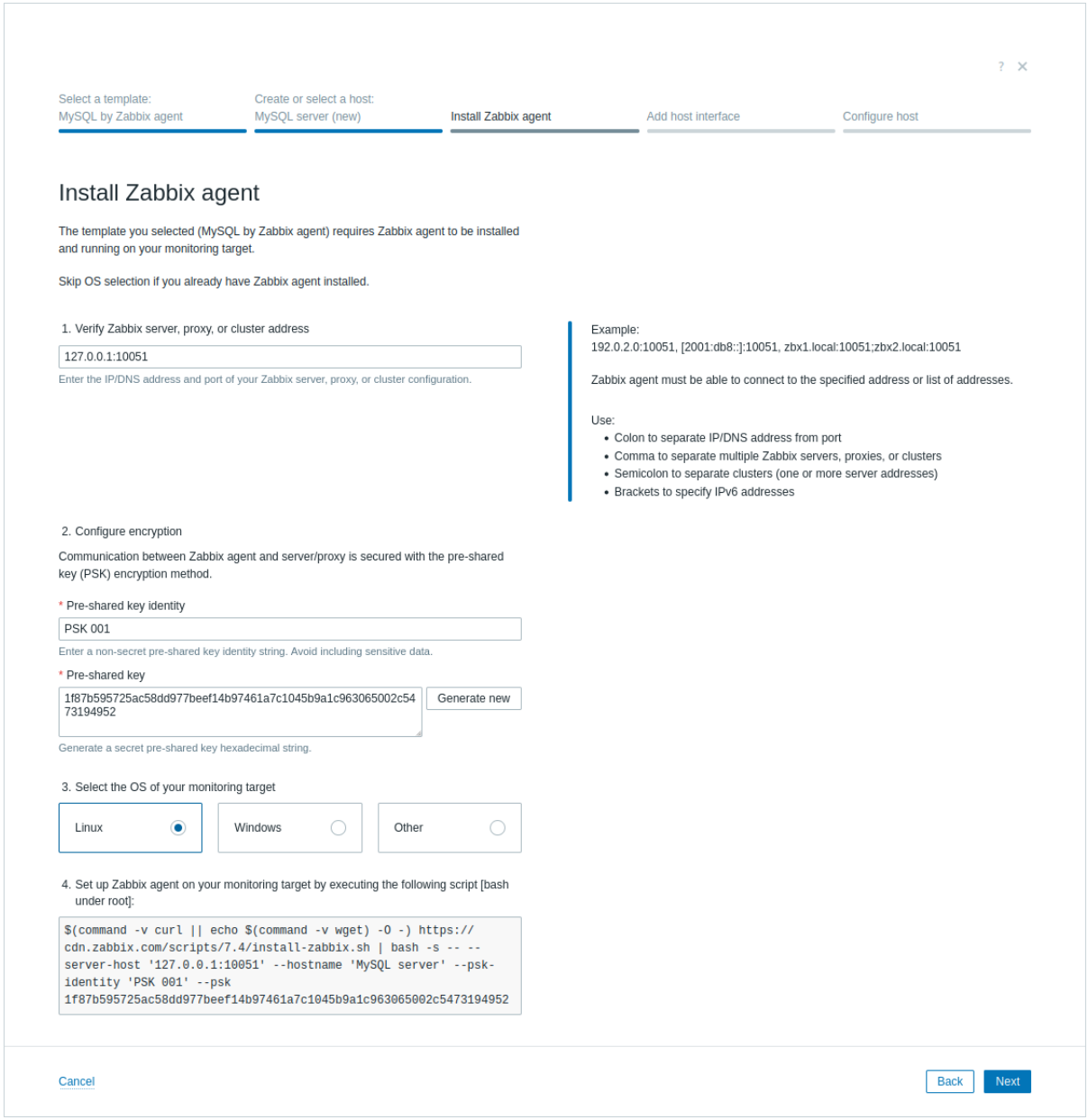
- Überprüfen Sie die Adresse des Zabbix Server, Proxy oder Clusters (z. B. 192.0.2.0:10051). Dieser Wert wird verwendet, um die Parameter Server und ServerActive für den Zabbix Agent zu konfigurieren.
- Geben Sie eine neue, eindeutige und nicht geheime Identität für den vorinstallierten Schlüssel ein (z. B. *PSK 001* oder *mysql-agent-psk1*), und generieren Sie den vorinstallierten Schlüssel. Diese Werte werden verwendet, um die Verschlüsselung mit 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 Rechners aus, auf dem sich Ihr Überwachungsziel befindet.
- Installieren Sie den Zabbix Agent auf diesem System, indem Sie das bereitgestellte Skript darauf ausführen oder den Installationsanweisungen folgen.

Sie könnten zum Beispiel *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-Assistenten zurück.

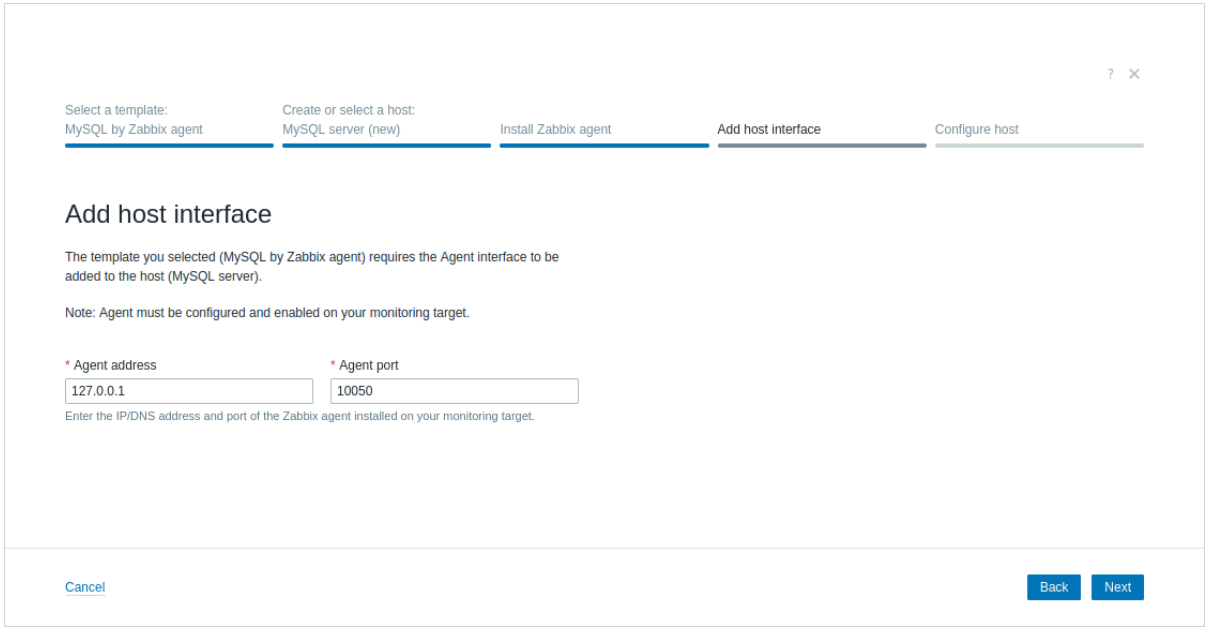


Host-Schnittstelle hinzufügen

Die Host-Schnittstelle definiert, wie der Zabbix Server über das Netzwerk eine Verbindung zu Ihrem Überwachungsziel herstellt. Sie legt Verbindungsparameter wie IP-Adresse, DNS-Namen, Port und Schnittstellentyp (Agent, SNMP, JMX oder IPMI) fest, abhängig von der für die ausgewählte Vorlage erforderlichen Methode zur Datenerfassung.

Geben Sie in diesem Schritt die von der ausgewählten Vorlage benötigten Details der Host-Schnittstelle ein.

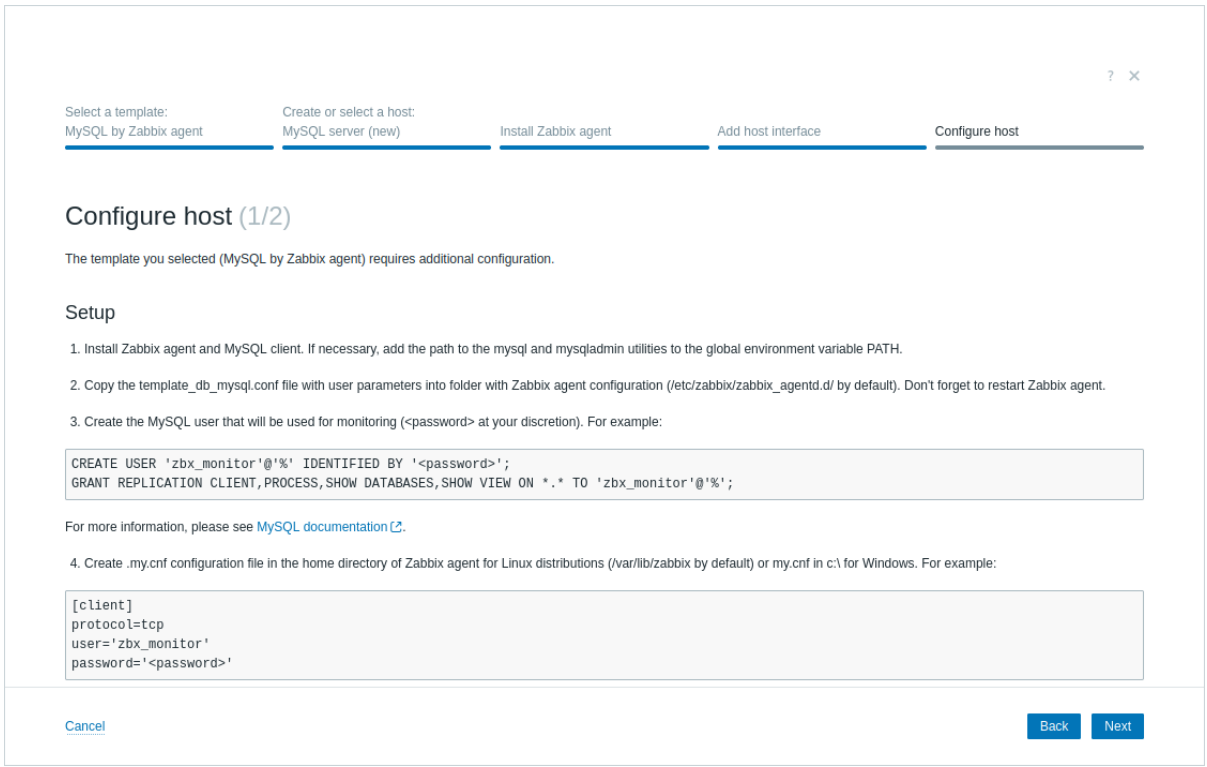
Sie können beispielsweise die Standardadresse (127.0.0.1) und den Standardport (10050) des Agent verwenden, wenn Zabbix Server, Zabbix Agent und der MySQL-Server lokal auf demselben Rechner ausgeführt werden.



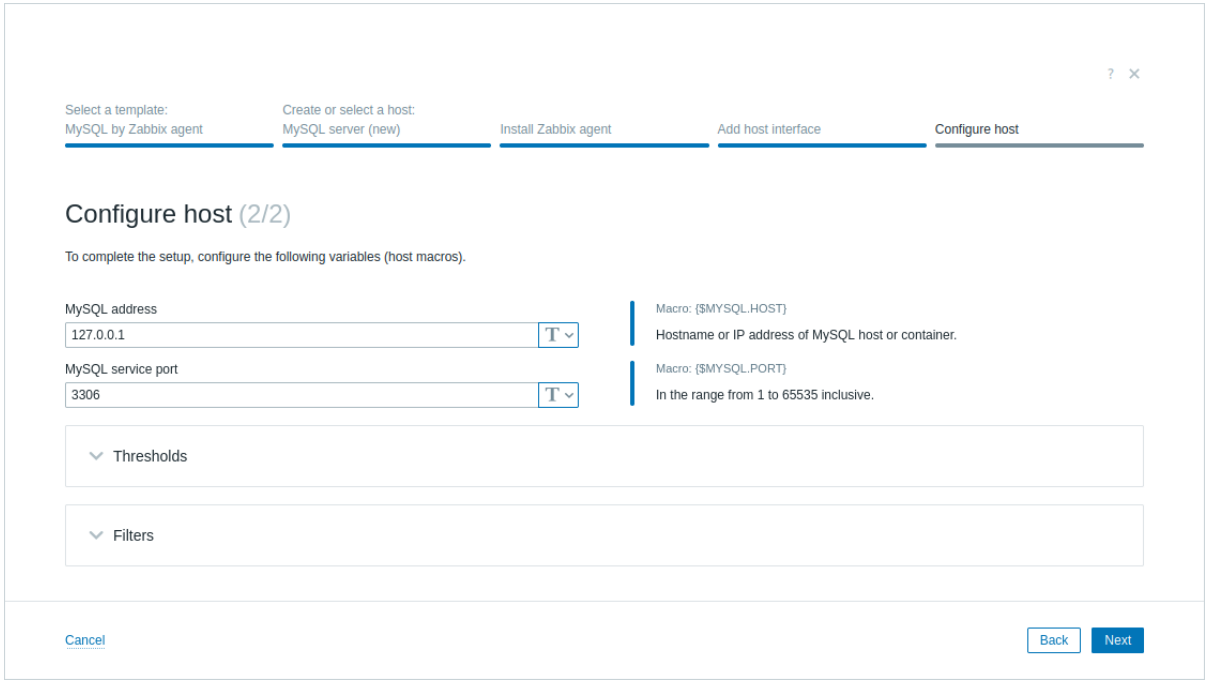
Host konfigurieren

Einige Vorlagen erfordern zusätzliche Konfiguration, bevor der Host erstellt werden kann. Dazu können gehören:

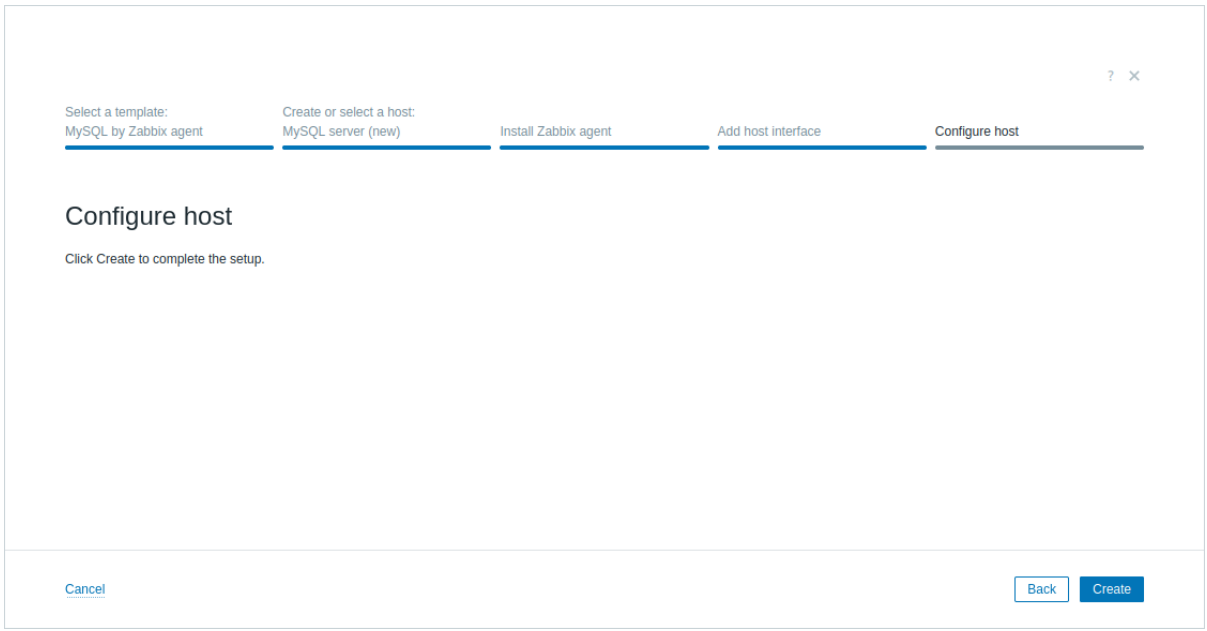
- Manuelle Konfiguration Ihres Überwachungsziels (z. B. Aktivieren bestimmter Dienste, Erstellen von Dienstbenutzern oder Erteilen von Berechtigungen):



- Definieren von Host-Makros – Variablen, die das Verhalten von Datenpunkten, Verbindungseinstellungen, Anmeldedaten für die Authentifizierung und andere Parameter steuern:



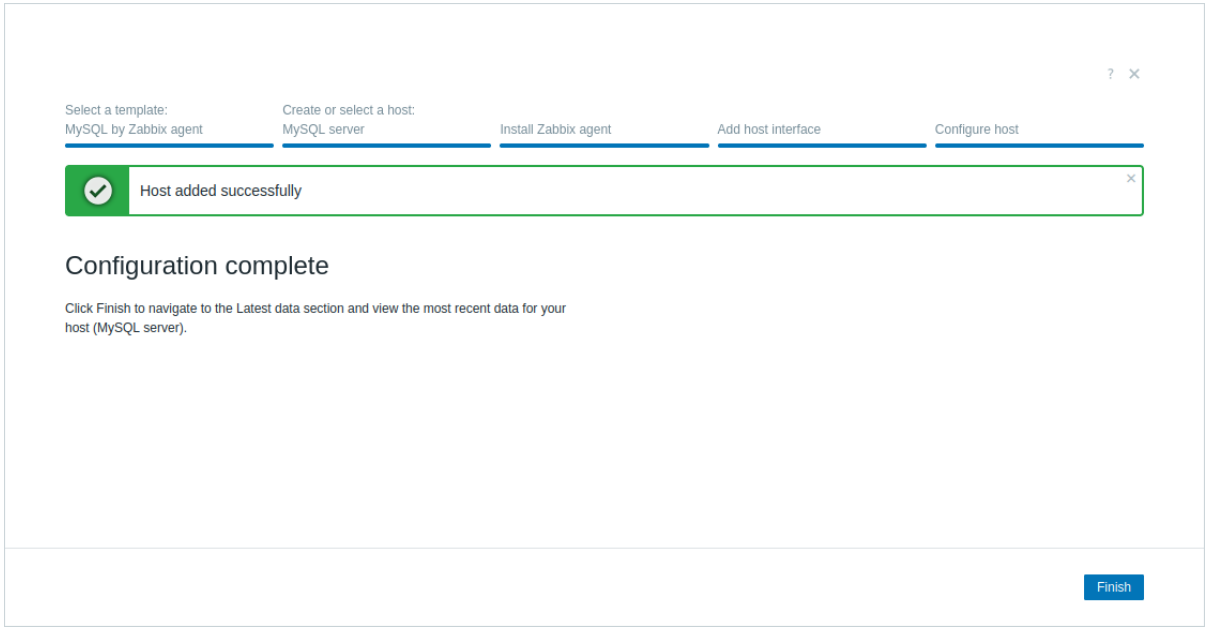
Nachdem Sie alle erforderlichen Konfigurationsschritte abgeschlossen haben, klicken Sie auf *Create*, um den Host zu Zabbix hinzuzufügen (oder auf *Update*, um die Konfiguration eines vorhandenen Hosts zu aktualisieren):



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.



2 Konfigurieren eines Hosts

Übersicht

Um einen Host im Zabbix Frontend zu konfigurieren, gehen Sie wie folgt vor:

- Gehen Sie zu: *Datenerfassung* > *Hosts* oder *Monitoring* > *Hosts*
- Klicken Sie oben rechts auf dem Bildschirm auf *Host erstellen* (oder auf den Hostnamen, um einen bestehenden Host zu bearbeiten)
- Geben Sie die Parameter des Hosts in das Formular ein

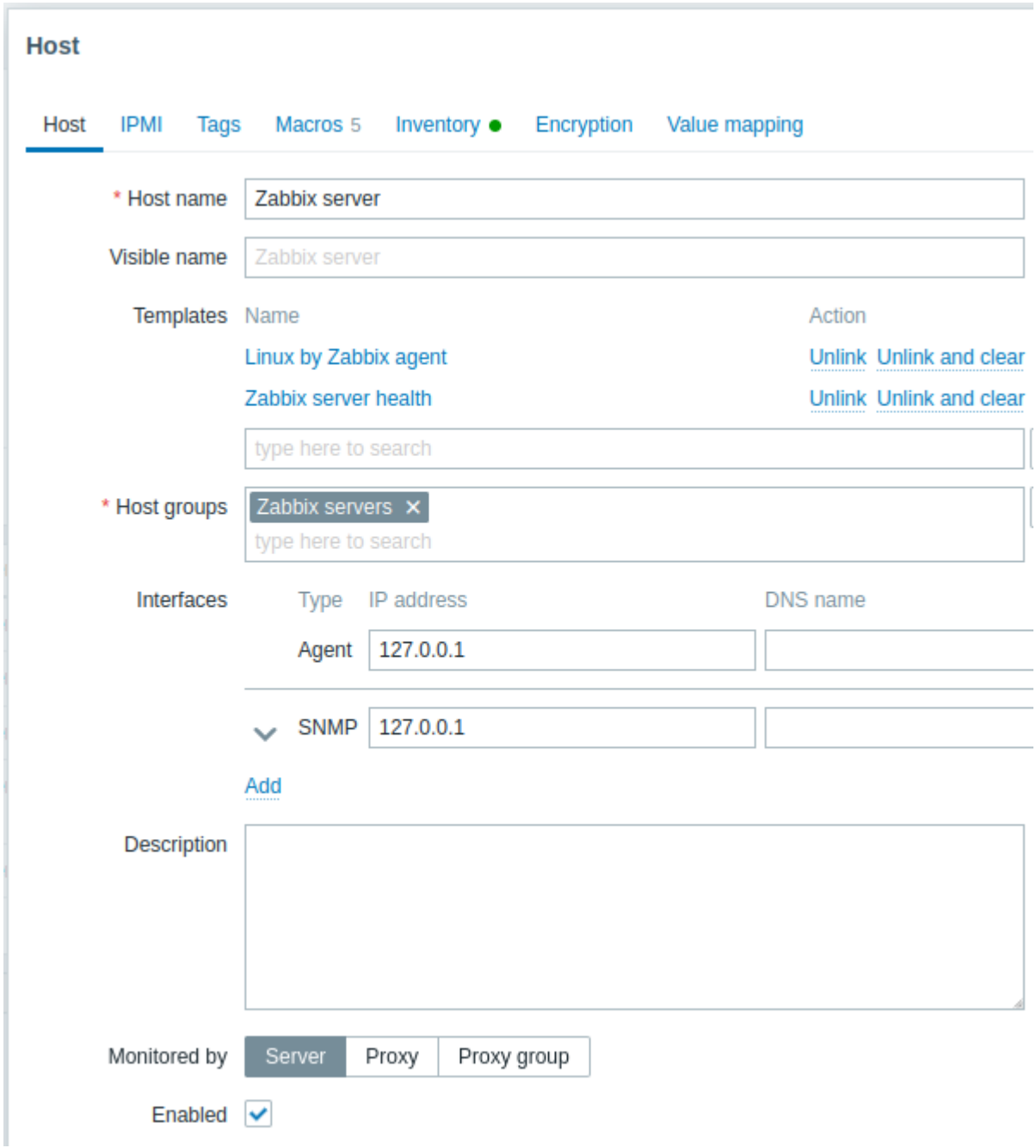
Sie können auch die Schaltfläche *Klonen* im Konfigurationsformular eines bestehenden Hosts verwenden, um einen neuen Host zu erstellen. Dieser Host übernimmt alle Eigenschaften des bestehenden Hosts, einschließlich verknüpfter Vorlagen, Entitäten (Datenpunkte, Auslöser usw.) aus diesen Vorlagen sowie der Entitäten, die direkt mit dem bestehenden Host verknüpft sind.

Beachten Sie, dass ein geklonter Host alle Vorlagen-Entitäten in ihrem ursprünglichen Zustand aus der Vorlage beibehält. Änderungen an diesen Entitäten, die auf Ebene des bestehenden Hosts vorgenommen wurden (z. B. ein geändertes Datenpunkt-Intervall, ein modifizierter regulärer Ausdruck oder hinzugefügte Prototypen zur Low-Level-Discovery-Regel), werden nicht auf den neuen Host geklont; stattdessen entsprechen sie dem Zustand in der Vorlage.

Alternativ können Sie den Host-Assistenten verwenden, um einen Host über eine geführte Schritt-für-Schritt-Oberfläche zu konfigurieren.

Konfiguration

Die Registerkarte **Host** enthält allgemeine Host-Attribute:



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Host name	Geben Sie einen eindeutigen Host-Namen ein. Alphanumerische Zeichen, Leerzeichen, Punkte, Bindestriche und Unterstriche sind zulässig. Führende und nachgestellte Leerzeichen sind jedoch nicht erlaubt. *Hinweis:* Wenn auf dem Host, den Sie konfigurieren, ein Zabbix-Agent ausgeführt wird, muss der Parameter *Hostname* in der Agent-Konfigurationsdatei denselben Wert haben wie der hier eingegebene Host-Name. Der Name im Parameter wird bei der Verarbeitung von aktiven 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 Host-Namens angezeigt. Dieses Attribut unterstützt UTF-8.

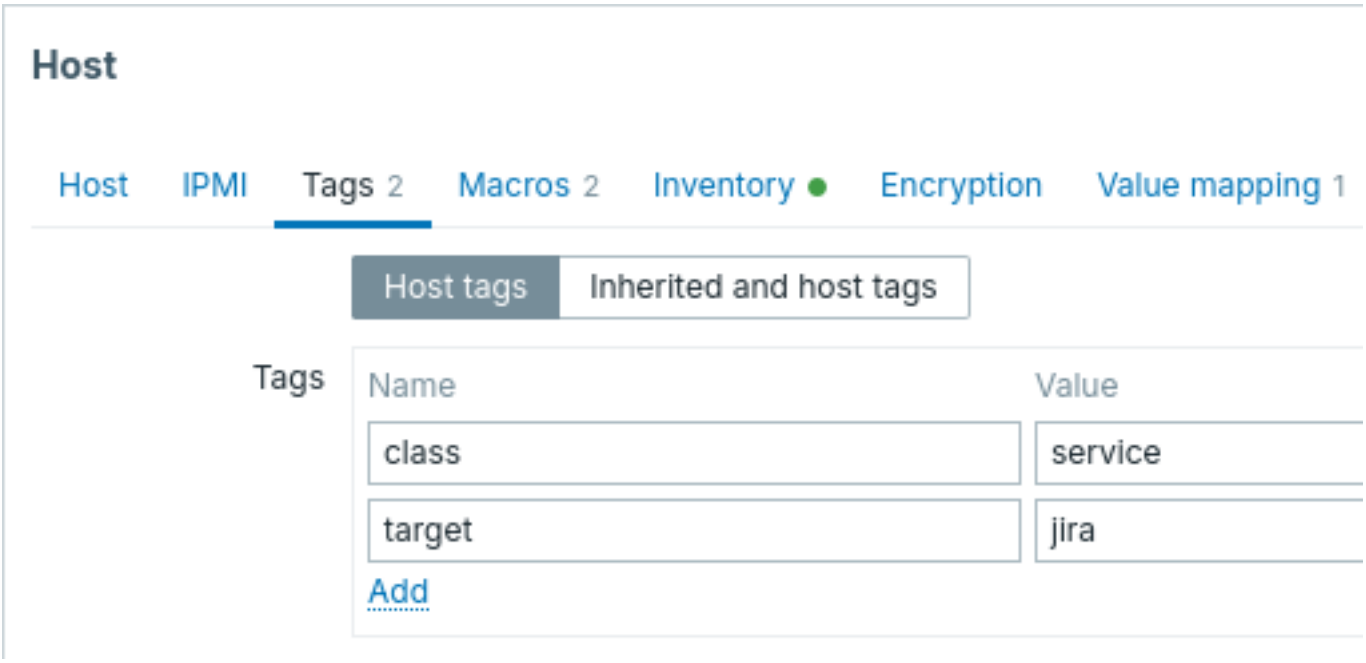
Parameter	Beschreibung
Templates	Verknüpfen Sie Vorlagen mit dem Host. Alle Entitäten (Datenpunkte, Auslöser, usw.) werden von der Vorlage geerbt. Um eine neue Vorlage zu verknüpfen, beginnen Sie mit der Eingabe des Vorlagennamens in das Texteingabefeld. Eine Liste passender Vorlagen wird angezeigt; scrollen Sie nach unten, um eine 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, wenn das Host-Konfigurationsformular gespeichert oder aktualisiert wird. Um die Verknüpfung einer Vorlage aufzuheben, verwenden Sie eine der beiden Optionen im Block *Linked templates*: *Unlink* - hebt die Verknüpfung der Vorlage auf, behält jedoch ihre Entitäten (Datenpunkte, Auslöser, usw.) bei; *Unlink and clear* - hebt die Verknüpfung der Vorlage auf und entfernt alle ihre Entitäten (Datenpunkte, Auslöser, usw.). Die aufgeführten Vorlagennamen sind anklickbare Links, die zum Vorlagen-Konfigurationsformular führen.
Host groups	Wählen Sie die Host-Gruppen aus, zu denen der Host gehört. Ein Host muss zu mindestens einer Host-Gruppe gehören. Eine neue Gruppe kann erstellt und mit dem Host verknüpft werden, indem ein noch nicht vorhandener Gruppenname hinzugefügt wird.
Interfaces	Für einen Host werden mehrere Host-Schnittstellentypen unterstützt: *Agent*, *SNMP*, *JMX* und *IPMI*. Standardmäßig sind keine Schnittstellen definiert. Um eine neue Schnittstelle hinzuzufügen, klicken Sie im Block *Interfaces* auf *Add*, wählen den Schnittstellentyp aus und geben Informationen zu *IP/DNS*, *Connect to* und *Port* ein. *Hinweis:* Schnittstellen, die in irgendwelchen Datenpunkten verwendet werden, können nicht entfernt werden; der Link *Remove* ist für sie ausgegraut. Die „IP“ oder der „DNS“-Name einer SNMP-Schnittstelle wird auch für SNMP-Traps verwendet. Beim Abgleich wird nur die in der Host-Schnittstelle ausgewählte „IP“ oder der ausgewählte „DNS“-Name verwendet. Weitere Details zur Konfiguration einer SNMP-Schnittstelle (v1, v2 und v3) finden Sie unter Configuring SNMP monitoring.
IP address	IP-Adresse des Hosts (optional).
DNS name	DNS-Name des Hosts (optional).
Connect to	Durch Klicken auf die entsprechende Schaltfläche teilen Sie dem Zabbix-Server mit, was zum Abrufen von Daten von Agenten verwendet werden 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 Zabbix-Agent, 161 für 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 Proxy-Gruppe überwacht.
Proxy	Der Name des zugewiesenen Proxy wird angezeigt (nur wenn der Zabbix-Server einen aus der ausgewählten Proxy-Gruppe zugewiesen hat). Dieses Feld wird nur angezeigt, wenn der Host von einer Proxy-Gruppe überwacht wird.

Parameter	Beschreibung
Enabled	Wenn das Kontrollkästchen aktiviert ist, ist der Host aktiviert – bereit für die Überwachung. Wenn das Kontrollkästchen deaktiviert ist, ist der Host deaktiviert – wird nicht überwacht: Bei passiven Datenabfragen, die vom Zabbix-Server/Proxy initiiert werden (zum Beispiel Zabbix-Agent, SNMP-Agent, einfache Prüfungen), wird die Überwachung nach der Konfigurationssynchronisierung deaktiviert. Mit dem Host verknüpfte Auslöser und Aktionen werden ebenfalls erst deaktiviert, nachdem die Konfiguration neu geladen wurde. Bei aktiven Prüfungen des Zabbix-Agenten wird die Überwachung innerhalb des Zeitraums (ca. 5 Sekunden) beendet, in dem der Zabbix-Agent Informationen darüber erhält, dass der Host deaktiviert wurde. Während dieses kurzen Intervalls sammelt der Host weiterhin lokal Daten für die aktiven Prüfungen und versucht, sie an den Server/Proxy zu senden; da der Host jedoch als *Disabled* markiert ist, lehnt der Server/Proxy die Daten ab. 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 Attribute für die IPMI-Verwaltung.

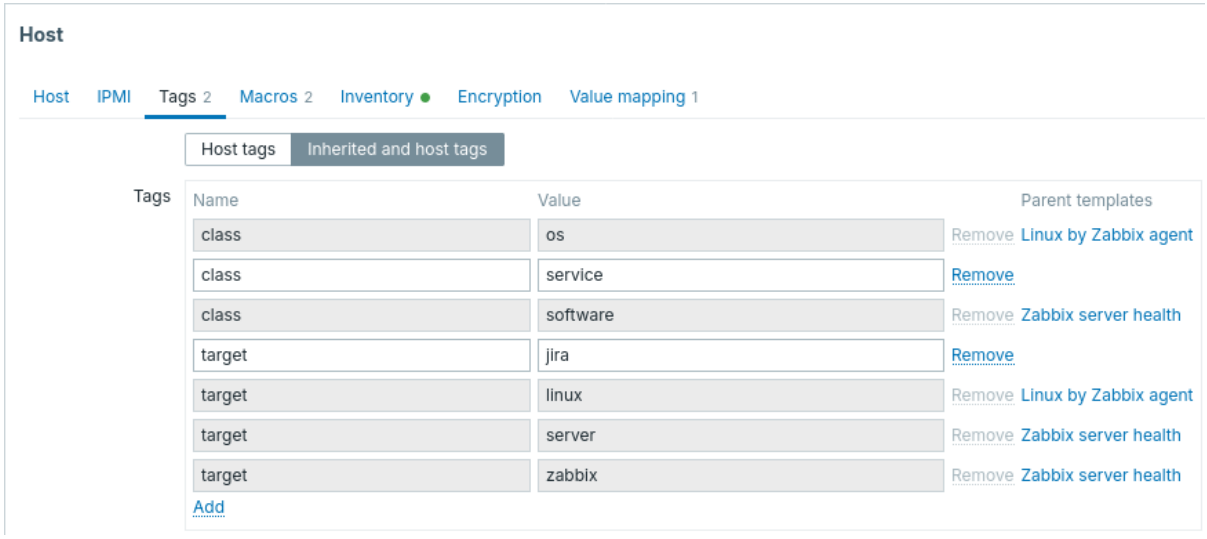
Parameter	Beschreibung
Authentication algorithm	Wählen Sie den Authentifizierungsalgorithmus aus.
Privilege level	Wählen Sie die Berechtigungsstufe aus.
Username	Benutzername für die Authentifizierung. Benutzermakros können verwendet werden.
Password	Passwort für die Authentifizierung. Benutzermakros können verwendet werden.

Die Registerkarte **Tags** ermöglicht es Ihnen, Host-Ebene-Tags zu definieren. Alle Probleme dieses Hosts werden mit den hier eingegebenen Werten getaggt.



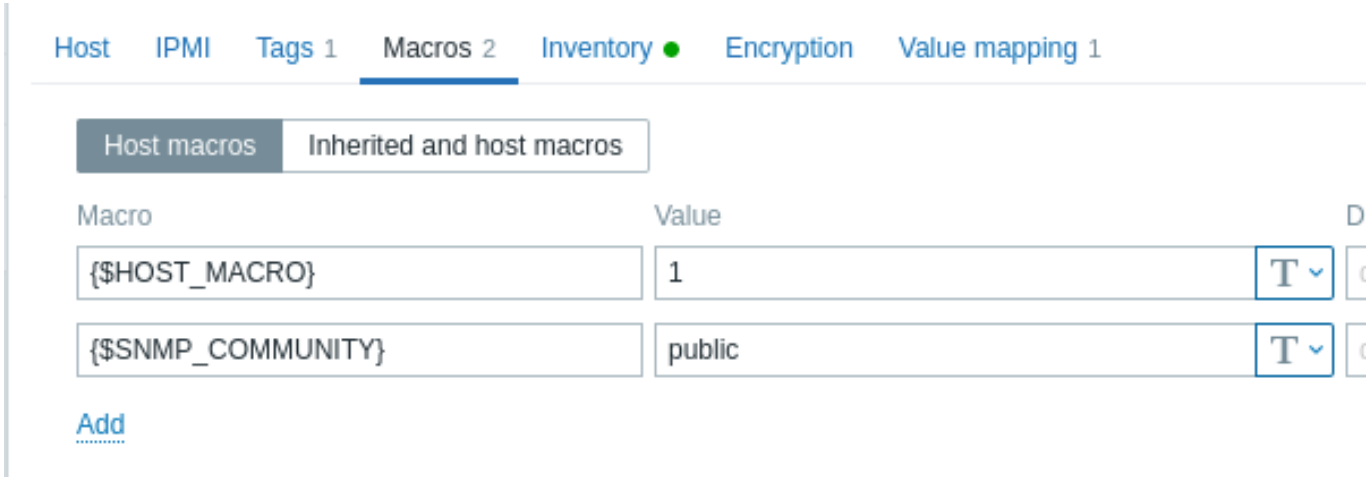
Benutzermakros, {INVENTORY.*}-Makros, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID}-Makros werden in Tags unterstützt.

Sie können hier auch Tags auf Vorlagenebene anzeigen, wenn Sie die Option *Inherited and host tags* auswählen. Dort werden alle für den Host definierten Benutzertags sowie deren Herkunft angezeigt.

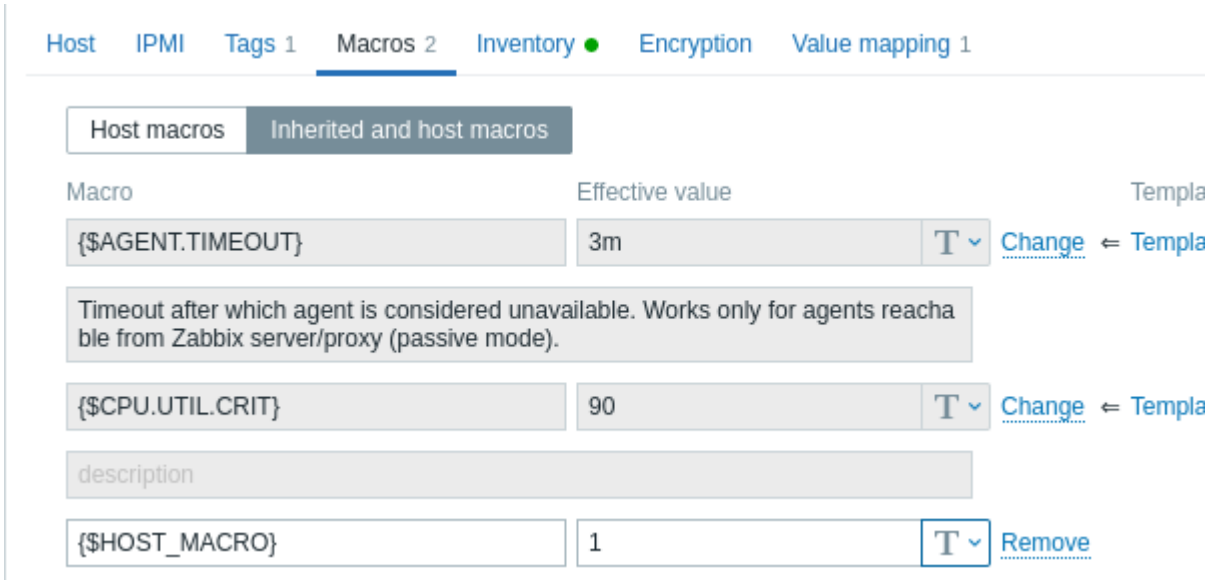


Der Einfachheit halber werden Links zu den jeweiligen Vorlagen bereitgestellt.

Die Registerkarte **Macros** ermöglicht es Ihnen, Host-Ebene-Benutzermakros als Name-Wert-Paare zu definieren. Beachten Sie, dass Makrowerte als Klartext, geheimer Text oder Vault-Geheimnis gespeichert werden können. Das Hinzufügen einer Beschreibung wird ebenfalls unterstützt.

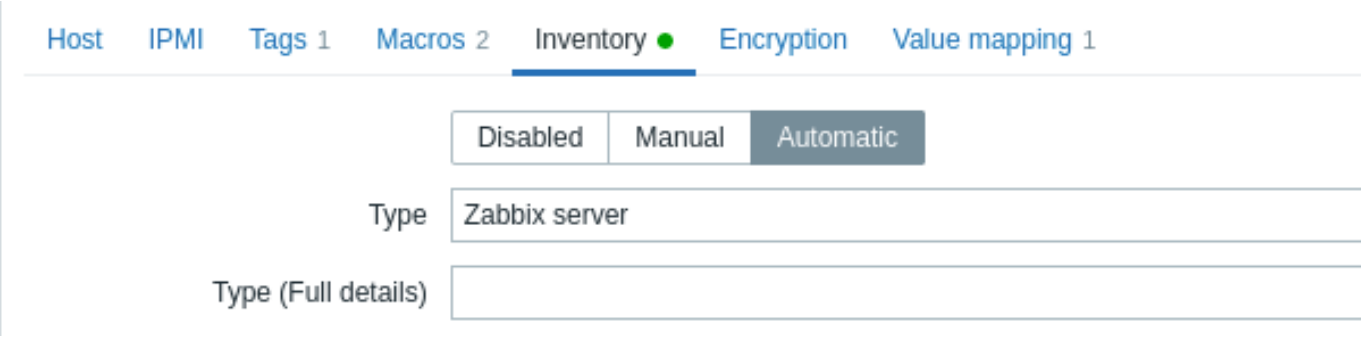


Sie können hier auch Benutzermakros auf Vorlagenebene und globale Benutzermakros anzeigen, wenn Sie die Option *Inherited and host macros* auswählen. Dort werden alle für den Host definierten Benutzermakros mit dem Wert, zu dem sie aufgelöst werden, sowie ihrer Herkunft angezeigt.



Der Einfachheit halber werden Links zu den jeweiligen Vorlagen und zur Konfiguration globaler Makros bereitgestellt. Es ist auch möglich, ein Vorlagen-/globales Makro auf Host-Ebene zu bearbeiten, wodurch effektiv eine Kopie des Makros auf dem Host erstellt wird.

Die Registerkarte **Inventory** ermöglicht es Ihnen, Inventar-Informationen für den Host manuell einzugeben. Sie können auch auswählen, die *Automatic*-Befüllung des Inventars zu aktivieren oder die Inventarbefüllung für diesen Host zu deaktivieren.



Wenn das Inventar aktiviert ist (manuell oder automatisch), wird zusammen mit 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: keine Verschlüsselung (Standard), mit PSK (Pre-Shared Key) oder Zertifikat.
Verbindungen vom Host	Wählen Sie aus, welche Verbindungstypen vom Host aus erlaubt sind (d. h. vom Zabbix Agent und Zabbix sender). Mehrere Verbindungstypen können gleichzeitig ausgewählt werden (nützlich zum Testen und zum Wechsel auf einen anderen Verbindungstyp). Standard ist „Keine Verschlüsselung“.
Issuer	Zulässiger Aussteller des Zertifikats. Das Zertifikat wird zunächst mit der CA (Certificate Authority) validiert. Ist es gültig und von der CA signiert, kann das Feld *Issuer* verwendet werden, um die zulässige CA weiter einzuschränken. Dieses Feld ist für den Einsatz vorgesehen, wenn Ihre Zabbix-Installation Zertifikate von mehreren CAs verwendet. Wenn dieses Feld leer ist, wird jede CA akzeptiert.
Subject	Zulässiger Betreff des Zertifikats. Das Zertifikat wird zunächst mit der CA validiert. Ist es gültig und von der CA signiert, 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 Key. Geben Sie keine sensiblen Informationen in der PSK-Identität an, da sie unverschlüsselt über das Netzwerk übertragen wird, um dem Empfänger mitzuteilen, welcher PSK verwendet werden soll.
PSK	Pre-Shared Key (Hex-Zeichenfolge). Maximale Länge: 512 Hex-Ziffern (256-Byte-PSK), wenn Zabbix die Bibliothek GnuTLS oder OpenSSL verwendet, 64 Hex-Ziffern (32-Byte-PSK), wenn Zabbix die Bibliothek mbed TLS (PolarSSL) verwendet. Beispiel: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Wertezuordnung

Die Registerkarte **Wertezuordnung** ermöglicht die Konfiguration einer benutzerfreundlichen Darstellung von Datenpunkt-Daten in Wertezuordnungen.

3 Konfigurieren einer Hostgruppe

Übersicht

Host-Gruppen werden für die logische Gruppierung von Hosts und die Zuweisung von Benutzerberechtigungen verwendet.

Jedem Host muss mindestens eine Host-Gruppe zugewiesen sein. Ein Host kann zu mehreren Host-Gruppen gehören, 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 also ein einzelner Benutzer Zugriff auf einen einzelnen Host benötigt, wird dieser gewährt, indem der Benutzer zu einer Benutzergruppe hinzugefügt wird, die die Berechtigung hat, auf die Host-Gruppe zuzugreifen, die diesen Host enthält.

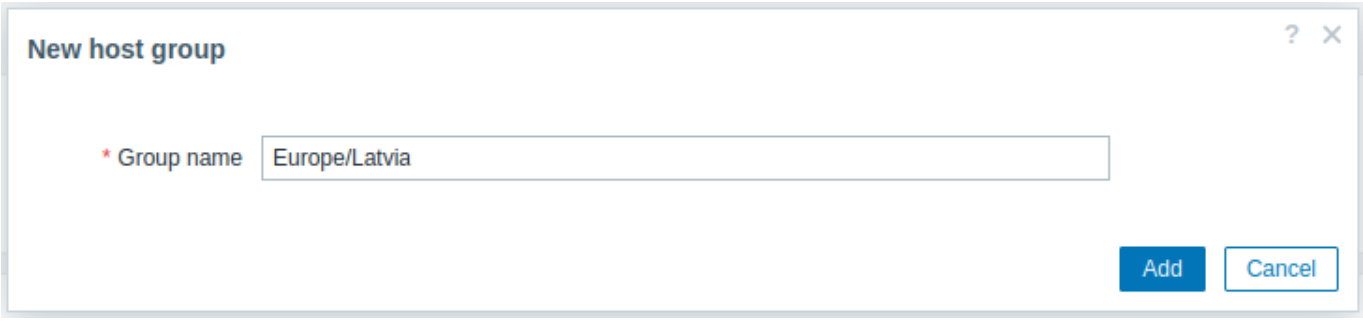
Konfiguration

Attention:
Nur Benutzer mit der Rolle Super admin können Host-Gruppen erstellen.

Es gibt zwei Möglichkeiten, eine Host-Gruppe im Zabbix Frontend zu erstellen.

Option eins:

- Gehen Sie zu: *Datensammlung* → *Host-Gruppen*
- Klicken Sie oben rechts auf dem Bildschirm auf *Host-Gruppe erstellen*
- Geben Sie den Gruppennamen in das Formular ein



Option zwei: Geben Sie beim Konfigurieren eines Hosts einen noch nicht vorhandenen Gruppennamen in das Eingabefeld *Host-Gruppen* ein.

Sobald die Host-Gruppe erstellt wurde, können Sie in der Liste unter *Datensammlung* → *Host-Gruppen* auf den Gruppennamen klicken, um den Gruppennamen zu bearbeiten, die Gruppe zu klonen oder die Gruppe zu löschen.

Beim Löschen einer Host-Gruppe wird nur die logische Gruppe gelöscht, nicht die Hosts in der Gruppe. Es ist nicht möglich, eine Host-Gruppe zu löschen, wenn sie für einen vorhandenen Host die einzige Gruppe 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`.

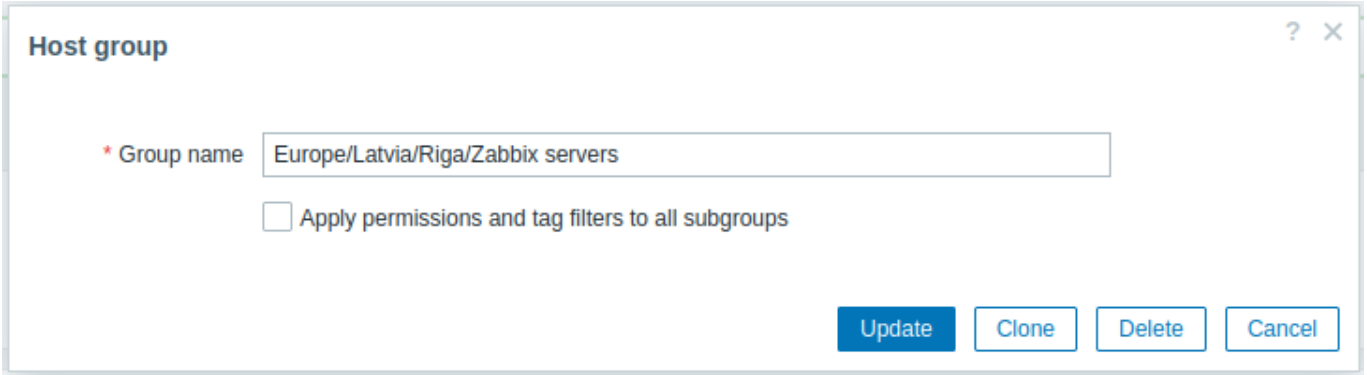
Beim Erstellen einer Untergruppe ist die Verwendung von führenden oder nachgestellten Schrägstrichen sowie mehrerer Schrägstriche hintereinander nicht zulässig. Das Escaping von '/' wird nicht unterstützt.

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 bestehenden übergeordneten Host-Gruppe (zum Beispiel beim Erstellen von `Europe/Latvia`, wenn `Europe` bereits existiert) werden die Berechtigungen der Benutzergruppe für die Untergruppe von der übergeordneten Gruppe geerbt.
- Beim Erstellen einer übergeordneten Host-Gruppe zu einer bestehenden Untergruppe (zum Beispiel beim Erstellen von `Europe`, wenn `Europe/Latvia` bereits existiert) werden für die übergeordnete Gruppe keine Berechtigungen gesetzt.

Beim Bearbeiten einer beliebigen Host-Gruppe können Sie außerdem eine zusätzliche Option festlegen: *Berechtigungen und Tag-Filter auf alle Untergruppen anwenden*.



Wenn Sie dieses Kontrollkästchen aktivieren und auf *Aktualisieren* klicken, werden dieselbe Berechtigungsstufe und dieselben Tag-Filter auf alle aktuellen und zukünftigen Untergruppen der bearbeiteten Host-Gruppe angewendet.

Wenn also einer oder mehreren Benutzergruppen unterschiedliche Berechtigungen für die Untergruppen der bearbeiteten Host-Gruppe zugewiesen wurden, führt das Aktivieren des Kontrollkästchens dazu, dass alle aktuellen und zukünftigen Untergruppen dieselben Benutzerberechtigungen und tagbasierten Berechtigungen wie die bearbeitete Gruppe erhalten.

Beachten Sie, dass diese Option nicht in der Datenbank gespeichert wird und bestehende Berechtigungen überschreibt. Alle über diese Option vorgenommenen Änderungen können nur manuell rückgängig gemacht werden.

4 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* Details wie den Gerätetyp, die Seriennummer, den Standort, die verantwortliche Person, URLs usw. eingeben – Daten, die die Inventarinformationen füllen.

Wenn in den Inventarinformationen des Hosts eine URL enthalten ist und sie mit „http“ oder „https“ beginnt, wird sie im Abschnitt *Inventar* als anklickbarer Link angezeigt.

Automatischer Modus

Das Host-Inventar kann auch automatisch befüllt werden. Damit dies funktioniert, muss beim Konfigurieren eines Hosts der Inventarmodus auf der Registerkarte *Inventar* auf *Automatisch* gesetzt werden.

Anschließend können Sie Host-Datenpunkte konfigurieren, um jedes beliebige Feld des Host-Inventars mit ihrem Wert zu befüllen, wobei das Zielfeld mit dem entsprechenden Attribut (genannt *Datenpunkt wird Host-Inventarfeld befüllen*) in der Datenpunkt-Konfiguration angegeben wird.

Datenpunkte, die sich besonders für die automatisierte Erfassung von Inventardaten eignen:

- system.hw.chassis[full|type|vendor|model|serial] - Standard ist [full], Root-Berechtigungen erforderlich
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - Standard ist [all,full]
- system.hw.devices[pci|usb] - Standard ist [pci]
- system.hw.macaddr[interface,short|full] - Standard ist [all,full], interface ist ein regulärer Ausdruck
- system.sw.arch
- system.sw.os[name|short|full] - Standard ist [name]
- system.sw.packages[regexp,manager,short|full] - Standard ist [all,all,full]

Auswahl des Inventarmodus

Der Inventarmodus kann im Konfigurationsformular des Hosts ausgewählt werden.

Der standardmäßige Inventarmodus für neue Hosts wird anhand der Einstellung *Standard-Inventarmodus für Hosts* unter *Verwaltung → Allgemein →* *Sonstiges* ausgewählt.

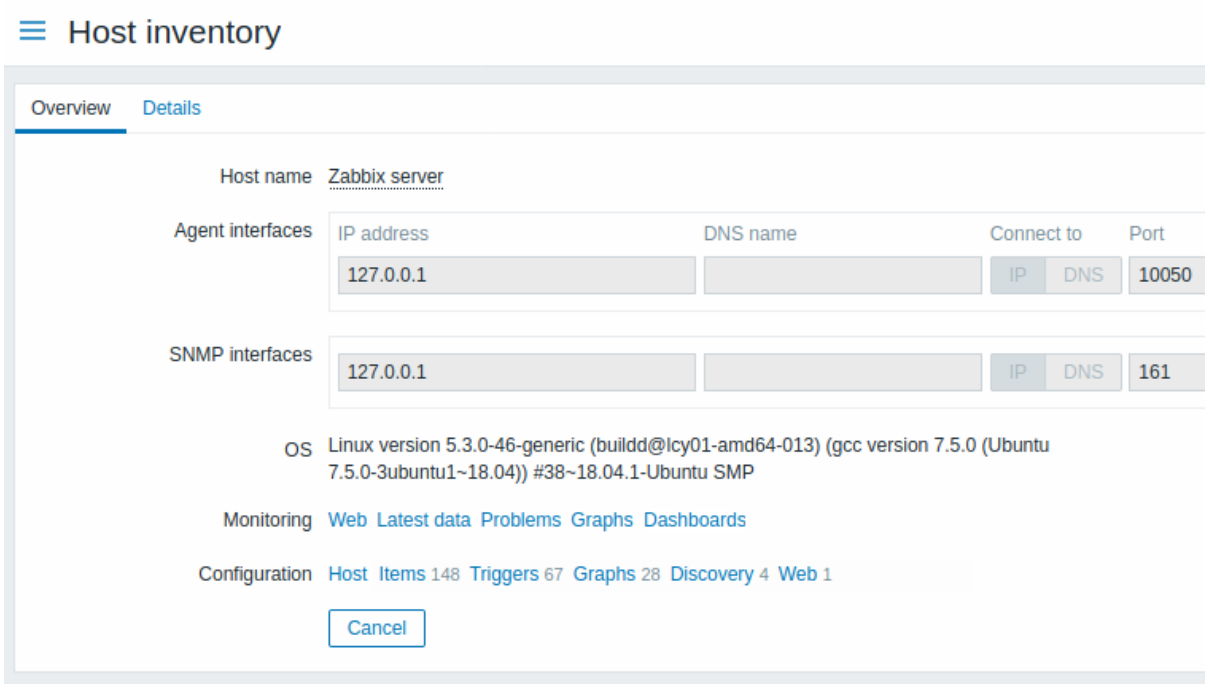
Für Hosts, die durch Netzwerkdiscovery oder Autoregistrierungsaktionen hinzugefügt wurden, ist es möglich, eine Operation *Host-Inventarmodus setzen* zu definieren und dabei den manuellen oder automatischen Modus auszuwählen. Diese Operation überschreibt die Einstellung *Standard-Inventarmodus für Hosts*.

Inventarübersicht

Die Details aller vorhandenen Inventardaten sind im Menü *Inventar* verfügbar.

Unter *Inventar* → *Übersicht* können Sie die Anzahl der Hosts nach verschiedenen Feldern des Inventars anzeigen lassen.

Unter *Inventar* → *Hosts* können Sie alle Hosts sehen, die über Inventarinformationen verfügen. Wenn Sie auf den Hostnamen klicken, werden die Inventardetails in einem Formular angezeigt.



Die Registerkarte **Übersicht** zeigt:

Parameter	Beschreibung
Host name	Name des Hosts. Ein Klick auf den Namen öffnet ein Menü mit den für den Host definierten Skripten. Der Hostname wird mit einem orangefarbenen Symbol angezeigt, wenn sich der Host in Wartung befindet.
Visible name	Sichtbarer Name des Hosts (falls definiert).
Host (Agent, SNMP, JMX, IPMI) interfaces	Dieser Block enthält Details zu den für den Host konfigurierten Schnittstellen.
OS	Inventarfeld des Betriebssystems des Hosts (falls definiert).
Hardware	Inventarfeld zur Hardware des Hosts (falls definiert).
Software	Inventarfeld zur Software des Hosts (falls definiert).
Description	Beschreibung des Hosts.
Monitoring	Links zu Überwachungsbereichen mit Daten für diesen Host: *Web*, *Neueste Daten*, *Probleme*, *Diagramme*, *Dashboards*.
Configuration	Links zu Konfigurationsbereichen für diesen Host: *Host*, *Datenpunkte*, *Auslöser*, *Diagramme*, *Discovery*, *Web*. Die Anzahl der konfigurierten Entitäten wird nach jedem Link angezeigt.

Die Registerkarte **Details** zeigt alle ausgefüllten Inventarfelder an (die nicht leer sind).

Inventar-Makros

Es stehen Host-Inventar-Makros {INVENTORY.*} zur Verwendung in Benachrichtigungen zur Verfügung, zum Beispiel:

„Server in {INVENTORY.LOCATION1} hat ein Problem, verantwortliche Person ist {INVENTORY.CONTACT1}, Telefonnummer {INVENTORY.POC.PRIMARY.PHONE.A1}."

Weitere Details finden Sie auf der Seite Unterstützte Makros.

5 Massenaktualisierung

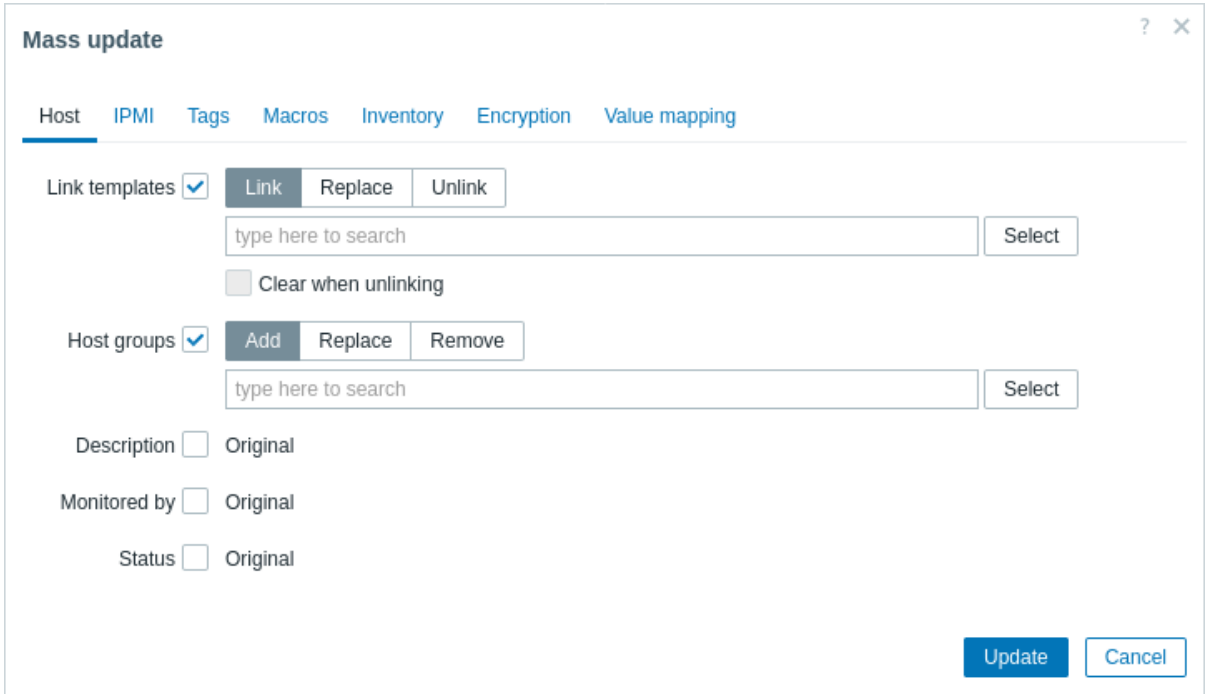
Übersicht

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

Um mehrere Hosts per Massenaktualisierung zu aktualisieren, gehen Sie wie folgt vor:

- Markieren Sie die Kontrollkästchen vor den Hosts, die Sie aktualisieren möchten, in der Host-Liste
- Klicken Sie unterhalb der Liste auf *Massenaktualisierung*
- Wechseln Sie zur Registerkarte mit den erforderlichen Attributen (*Host*, *IPMI*, *Tags*, *Makros*, *Inventar*, *Verschlüsselung* oder *Wertzuordnung*)
- Markieren Sie die Kontrollkästchen der Attribute, die aktualisiert werden sollen, und geben Sie dafür neue Werte ein



Die folgenden Optionen sind verfügbar, wenn Sie die entsprechende Schaltfläche für die Aktualisierung der **Vorlagen**-Verknüpfung auswählen:

- *Verknüpfen* - geben Sie an, welche zusätzlichen Vorlagen verknüpft werden sollen
- *Ersetzen* - geben Sie an, welche Vorlagen verknüpft werden sollen, wobei alle Vorlagen getrennt werden, die zuvor mit den Hosts verknüpft waren
- *Trennen* - geben Sie an, welche Vorlagen getrennt werden sollen

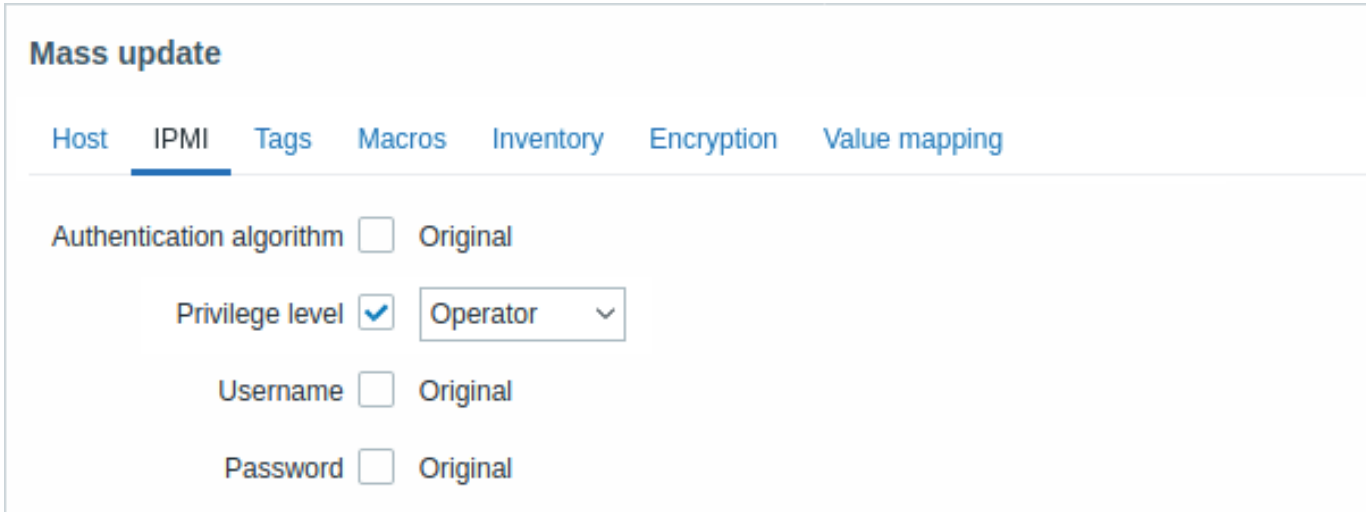
Um die zu verknüpfenden/zu trennenden Vorlagen anzugeben, beginnen Sie mit der Eingabe des Vorlagennamens in das Auto-Complete-Feld, bis eine Dropdown-Liste mit passenden Vorlagen erscheint. Scrollen Sie einfach nach unten, um die gewünschte Vorlage auszuwählen.

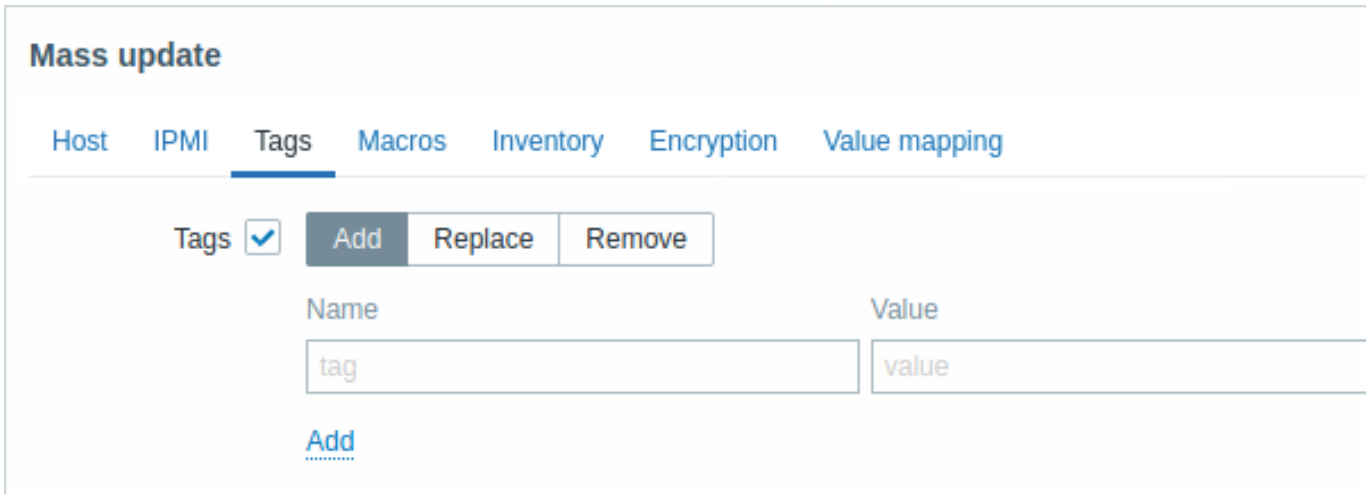
Mit der Option *Beim Trennen löschen* können Sie nicht nur zuvor verknüpfte Vorlagen trennen, sondern auch alle von ihnen geerbten Elemente entfernen (Datenpunkte, Auslöser usw.).

Die folgenden Optionen sind verfügbar, wenn Sie die entsprechende Schaltfläche für die Aktualisierung der **Hostgruppe** auswählen:

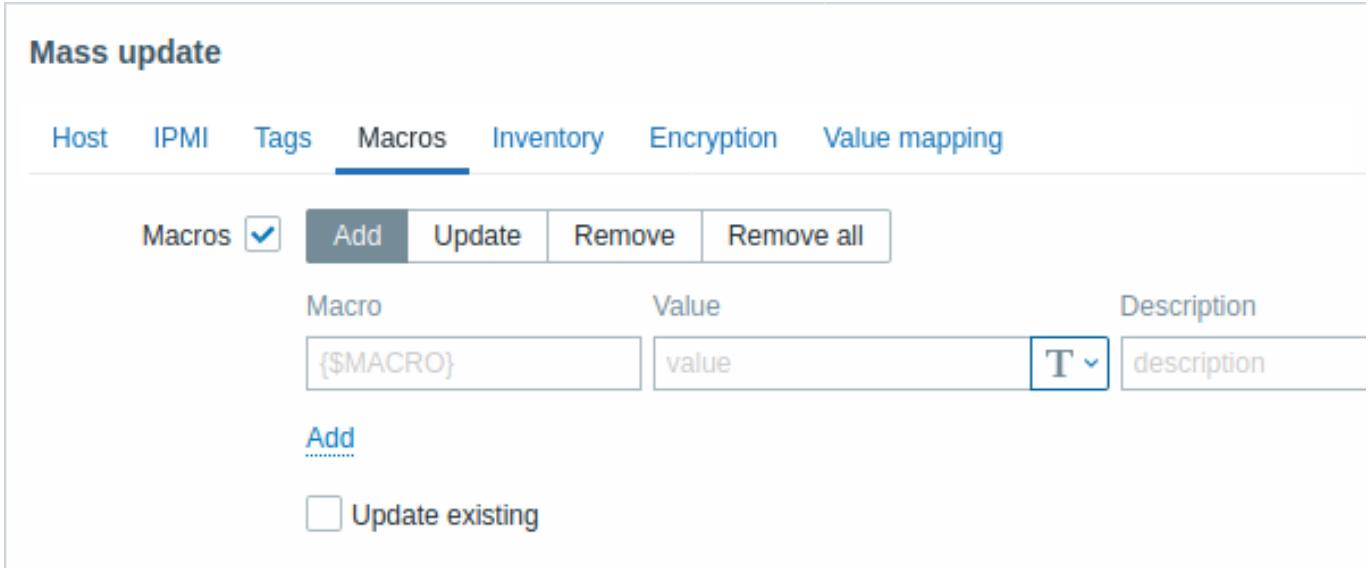
- *Hinzufügen* - ermöglicht es, zusätzliche Hostgruppen aus den vorhandenen auszuwählen oder vollständig neue Hostgruppen für die Hosts einzugeben
- *Ersetzen* - entfernt den Host aus allen vorhandenen Hostgruppen und ersetzt diese durch die in diesem Feld angegebenen (vorhandenen oder neuen) Hostgruppen
- *Entfernen* - entfernt bestimmte Hostgruppen von Hosts

Diese Felder unterstützen Auto-Complete - wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit passenden Hostgruppen angeboten. Wenn die Hostgruppe neu ist, erscheint sie ebenfalls in der Dropdown-Liste und wird durch *(neu)* hinter der Zeichenfolge gekennzeichnet. Scrollen Sie einfach nach unten, um sie auszuwählen.





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 demselben Host hinzugefügt werden können.

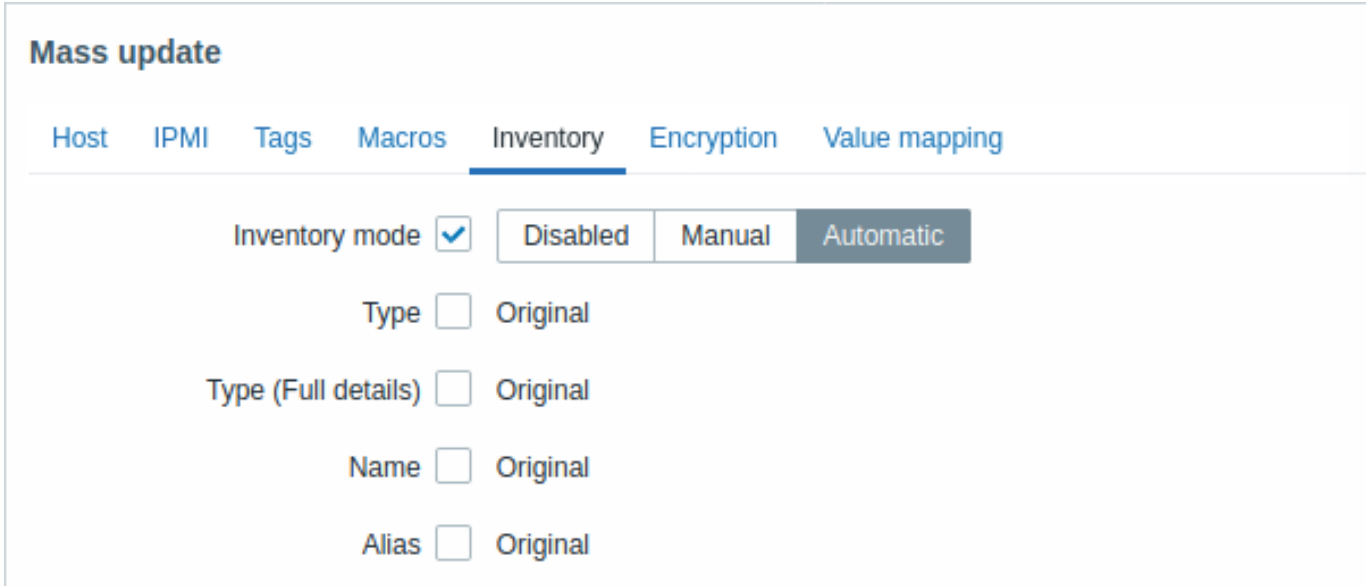


Die folgenden Optionen sind verfügbar, wenn Sie die entsprechende Schaltfläche für die Makro-Aktualisierung auswählen:

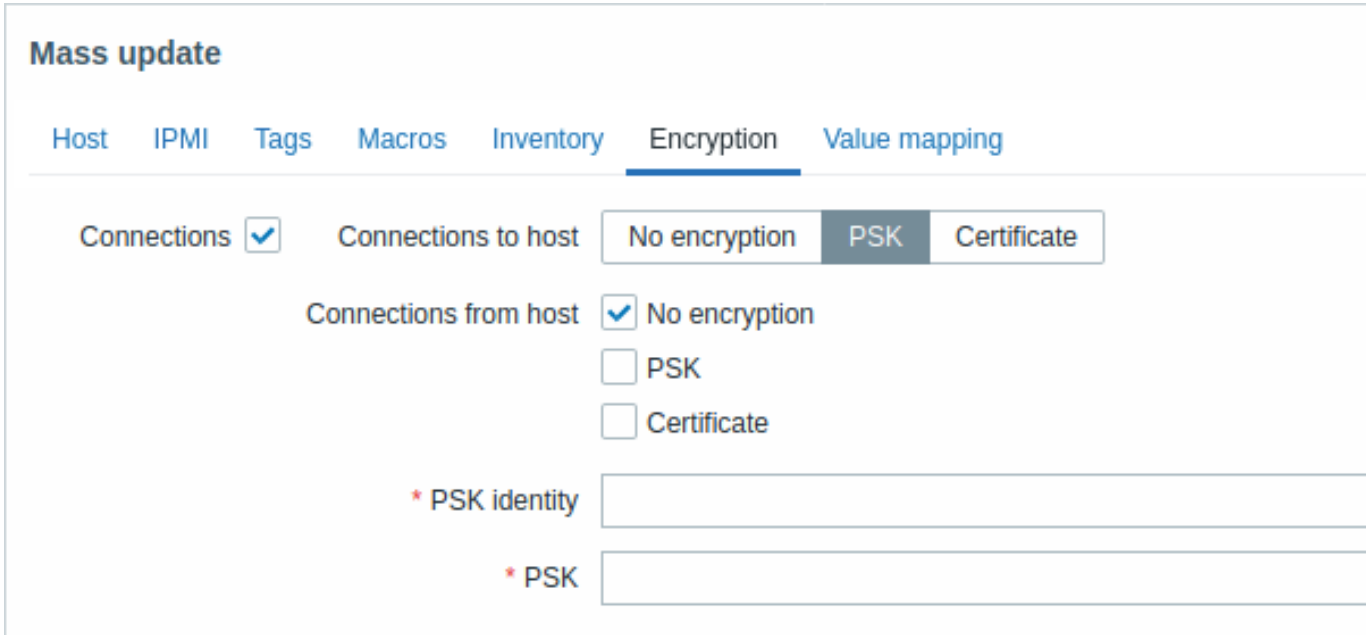
- *Hinzufügen* - ermöglicht es, zusätzliche Benutzermakros für die Hosts anzugeben. Wenn das Kontrollkästchen *Vorhandene aktualisieren* aktiviert ist, werden Wert, Typ und Beschreibung für den angegebenen Makronamen aktualisiert. Ist es nicht aktiviert und ein Makro mit diesem Namen existiert bereits auf dem/den Host(s), wird es nicht aktualisiert.
- *Aktualisieren* - ersetzt 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 existierte, als neues Makro hinzugefügt. Ist es nicht aktiviert, werden nur Makros aktualisiert, die bereits auf einem Host existieren.
- *Entfernen* - entfernt die angegebenen Makros von Hosts. Wenn das Kontrollkästchen *Außer ausgewählte* aktiviert ist, werden

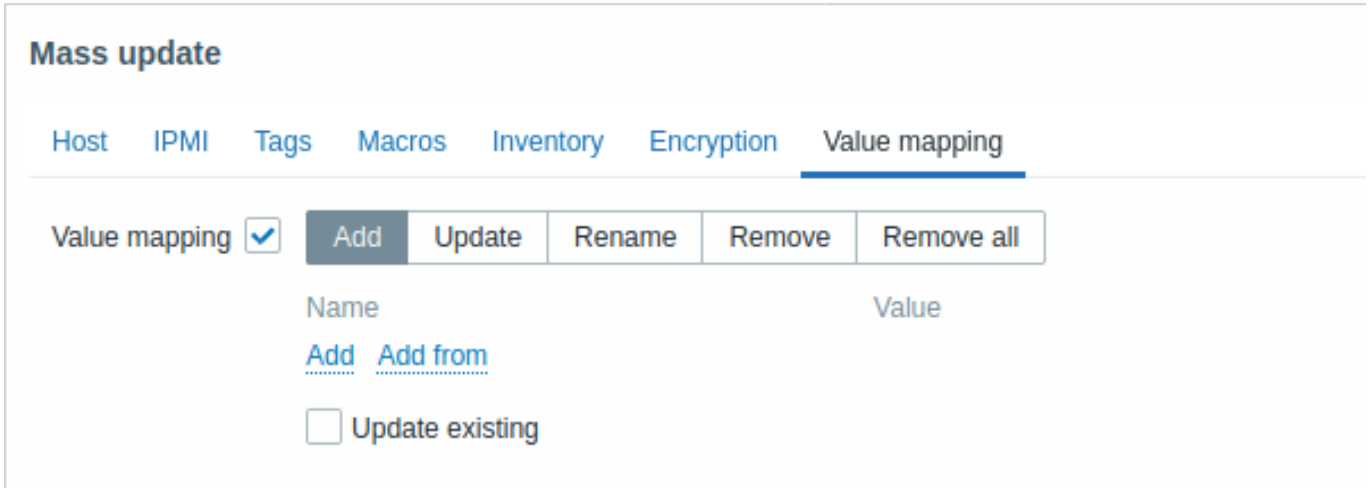
alle Makros außer den in der Liste angegebenen entfernt. Ist es nicht aktiviert, werden nur die in der Liste angegebenen Makros entfernt.

- *Alle entfernen* - entfernt alle Benutzermakros von Hosts. Wenn das Kontrollkästchen *Ich bestätige, alle Makros zu entfernen* nicht aktiviert ist, wird ein neues Popup-Fenster geöffnet, in dem das Entfernen aller Makros bestätigt werden muss.



Um Inventarfelder per Massenaktualisierung aktualisieren zu können, sollte der *Inventarmodus* auf „Manuell“ oder „Automatisch“ gesetzt sein.





Schaltflächen mit den folgenden Optionen sind für die Aktualisierung der Wertzuordnung verfügbar:

- *Hinzufügen* - fügt den Hosts Wertzuordnungen hinzu. Wenn Sie *Vorhandene aktualisieren* markieren, werden alle Eigenschaften der Wertzuordnung mit diesem Namen aktualisiert. Andernfalls wird eine Wertzuordnung mit diesem Namen, falls sie bereits existiert, nicht aktualisiert.
- *Aktualisieren* - aktualisiert vorhandene Wertzuordnungen. Wenn Sie *Fehlende hinzufügen* markieren, wird eine Wertzuordnung, die zuvor auf einem Host nicht existierte, als neue Wertzuordnung hinzugefügt. Andernfalls werden nur die Wertzuordnungen aktualisiert, die bereits auf einem Host existieren.
- *Umbenennen* - vergibt einen neuen Namen für eine vorhandene Wertzuordnung
- *Entfernen* - entfernt die angegebenen Wertzuordnungen von den Hosts. Wenn Sie *Außer ausgewählte* markieren, werden **alle** Wertzuordnungen entfernt, **außer** den angegebenen.
- *Alle entfernen* - entfernt alle Wertzuordnungen von den Hosts. Wenn das Kontrollkästchen *Ich bestätige, alle Wertzuordnungen zu entfernen* nicht markiert ist, wird ein neues Popup-Fenster geöffnet, in dem das Entfernen bestätigt werden muss.

Wenn Sie alle erforderlichen Änderungen vorgenommen haben, klicken Sie auf *Aktualisieren*. Die Attribute werden dann entsprechend für alle ausgewählten Hosts aktualisiert.

2 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.

1 Erstellen eines Datenpunkts

Übersicht

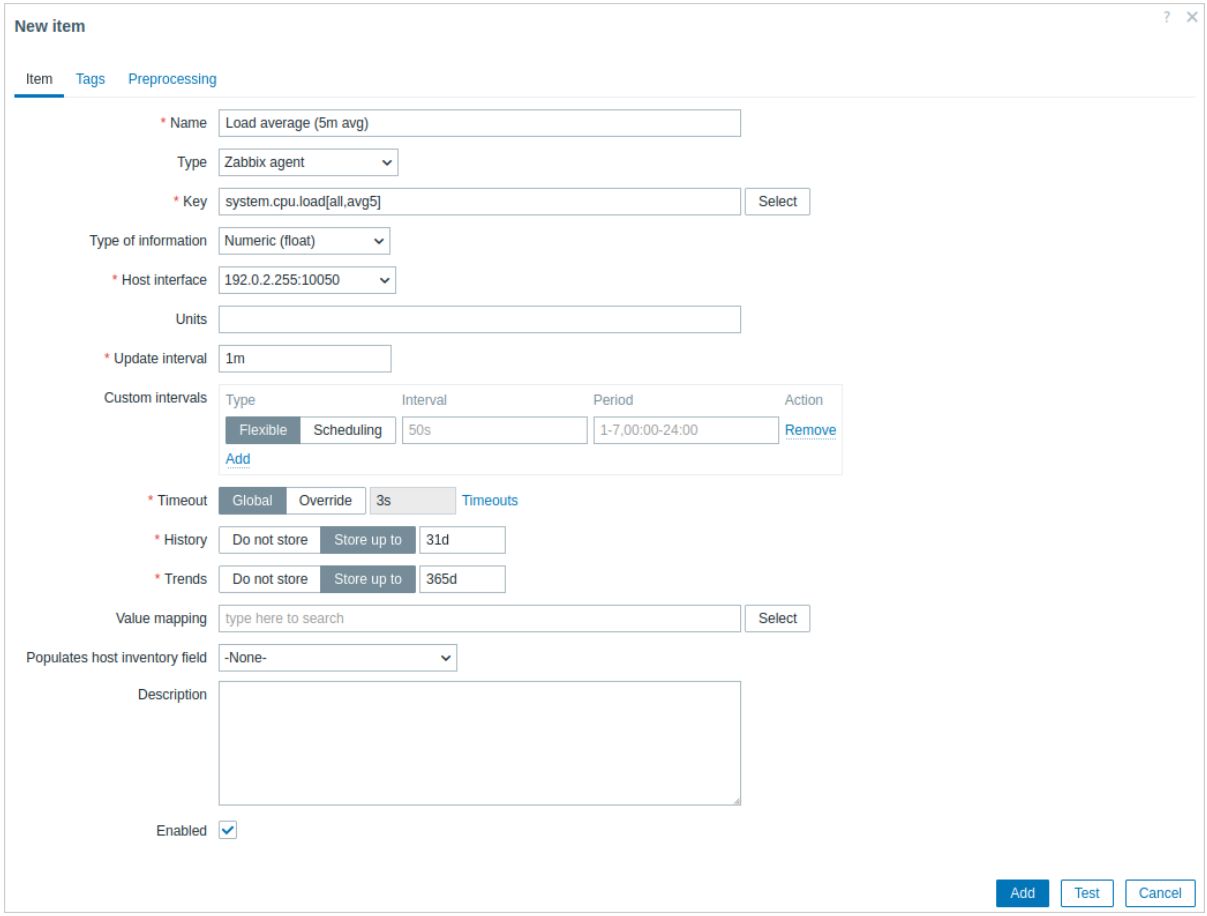
Um einen Datenpunkt im Zabbix Frontend zu erstellen, gehen Sie wie folgt vor:

- Gehen Sie zu: *Datenerfassung* > *Hosts*
- Klicken Sie in der Zeile des Hosts auf *Datenpunkte*
- Klicken Sie oben rechts auf dem Bildschirm auf *Datenpunkt erstellen*
- Geben Sie die Parameter des Datenpunkts in das Formular ein

Sie können einen Datenpunkt auch erstellen, indem Sie einen vorhandenen öffnen, auf die Schaltfläche *Klonen* klicken und ihn dann unter einem anderen Namen speichern.

Konfiguration

Die Registerkarte **Datenpunkt** enthält allgemeine Attribute des Datenpunkts.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Name des Datenpunkts. Benutzermakros werden unterstützt.
Type	Typ des Datenpunkts. Siehe die einzelnen Abschnitte zu Datenpunkttypen.
Key	Schlüssel des Datenpunkts (bis zu 2048 Zeichen). Die unterstützten Datenpunktschlüssel finden Sie in den einzelnen Abschnitten zu den Datenpunkttypen. Der Schlüssel muss innerhalb eines einzelnen Hosts eindeutig sein. Wenn der Schlüsseltyp „Zabbix agent“, „Zabbix agent (active)“ oder „Simple check“ ist, muss der Schlüsselwert von Zabbix agent oder Zabbix server unterstützt werden. Siehe auch: das korrekte Schlüsselformat.
Type of information	Der Datentyp, der verwendet wird, um den Wert des Datenpunkts zu validieren und ihn in der Datenbank zu speichern, jeweils nach einer eventuellen Konvertierung: **Numeric (unsigned)** - 64-Bit-Integer ohne Vorzeichen; **Numeric (float)** - 64-Bit-Gleitkommazahl; **Character** - kurzer Text; **Log** - langer Text mit oder ohne Log-Eigenschaften (Zeitstempel, Quelle, Schweregrad, Log-Ereignis-ID); **Text** - langer Text; **Binary** - Binärzahl (nur für abhängige Datenpunkte unterstützt); **JSON** - strukturierte JSON-Daten, nativ in der Datenbank gespeichert (nicht für berechnete Datenpunkte unterstützt). Datenpunkte, die Werte mit dem Datentyp Binary oder JSON zurückgeben, werden in Formeln berechneter Datenpunkte oder in Ausdrücken von Auslösern nicht unterstützt. Siehe Grenzwerte für Datenpunktdaten für Speichergrenzen pro Datentyp und Datenbank-Backend. Für Datenpunkte, die Daten in einem einzigen spezifischen Format zurückgeben, wird automatisch ein passender Datentyp ausgewählt – zum Beispiel geben `system.cpu.load` und `system.cpu.util` beide eine 64-Bit-Gleitkommazahl zurück, daher wird *Numeric (float)* automatisch ausgewählt.

Parameter	Beschreibung
Host interface	Wählen Sie die Host-Schnittstelle aus. Dieses Feld ist verfügbar, wenn ein Datenpunkt auf Host-Ebene bearbeitet wird.
Units	Wenn ein Einheitensymbol festgelegt ist, wendet Zabbix eine Nachbearbeitung auf den empfangenen Datenpunktwert an und zeigt ihn mit dem angegebenen Einheiten-Suffix an. Unterstützte Einheitensymbole mit spezieller Formatierung (und Beispiele für empfangenen Datenpunktwert → angezeigten Wert): **B** - Bytes (1024 → 1 KB) **Bps** - Bytes pro Sekunde (1024 → 1 KBps) **s** - Sekunden, angezeigt mit bis zu den drei größten Zeitangaben ungleich null (881764 → 10d 4h 56m) **uptime** - verstrichene Zeit im Format hh:mm:ss oder N days, 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 die korrekte Formatierung muss der empfangene Datenpunktwert *Numeric (unsigned)* sein. Bei anderen Einheiten (wie 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 ! vorangestellt wird, werden Einheitenumrechnung und Präfixbildung deaktiviert (1024 !B → 1024 B, 61 !s → 61 s). Weitere Beispiele und Details zu Einheitensymbolen und Einheitenumrechnung finden Sie unter Suffixe von Datenpunktwerten.
Update interval	Rufen Sie 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 mit Text gemischte Makros werden nicht unterstützt. *Hinweis*: Das Aktualisierungsintervall kann nur dann 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, nachdem der Datenpunkt aktiv geworden ist oder nachdem das Aktualisierungsintervall geändert wurde, früher als der konfigurierte Wert erfolgen kann. Neue Datenpunkte werden innerhalb von 60 Sekunden nach ihrer Erstellung geprüft, es sei denn, sie haben ein geplantes oder flexibles Aktualisierungsintervall und das *Update interval* ist auf 0 gesetzt. Ein vorhandener passiver Datenpunkt kann sofort auf einen Wert abgefragt werden, indem die Schaltfläche *Execute now* gedrückt wird.
Custom intervals	Sie können benutzerdefinierte Regeln für die Prüfung des Datenpunkts erstellen: **Flexible** - erstellt eine Ausnahme zum *Update interval* (Intervall mit anderer Häufigkeit). **Scheduling** - erstellt einen benutzerdefinierten Abfragezeitplan. Detaillierte Informationen finden Sie unter Benutzerdefinierte Intervalle. Zeitsuffixe werden im Feld *Interval* unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzermakros werden unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen. Mehrere Makros in einem Feld oder mit Text gemischte Makros werden nicht unterstützt.
Timeout	Legen Sie das Zeitlimit für die Prüfung des Datenpunkts fest (verfügbar für unterstützte Datenpunkttypen). Wählen Sie die Timeout-Option: **Global** - es wird das Proxy-/globale Zeitlimit verwendet (angezeigt im ausgegrauten Feld *Timeout*). **Override** - es wird ein benutzerdefiniertes Zeitlimit verwendet (im Feld *Timeout* festgelegt; zulässiger Bereich: 1 - 600s). Zeitsuffixe, z. B. 30s, 1m, und Benutzermakros werden unterstützt. Durch Klicken auf den Link *Timeouts* können Sie Proxy-Zeitlimits oder globale Zeitlimits konfigurieren (wenn kein Proxy verwendet wird). Beachten Sie, dass der Link *Timeouts* nur für Benutzer vom Typ *Super admin* sichtbar ist, die Berechtigungen für die Frontend-Abschnitte *Administration* > *General* oder *Administration* > *Proxies* haben.

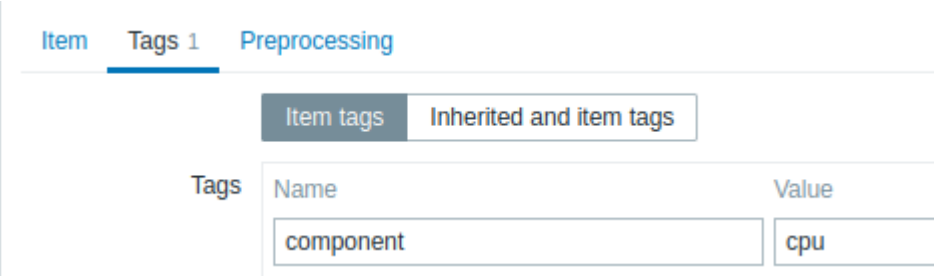
Parameter	Beschreibung
History	Wählen Sie entweder: **Do not store** - die Historie des Datenpunkts wird nicht gespeichert. Nützlich für Master-Datenpunkte, wenn nur abhängige Datenpunkte eine Historie behalten müssen. Diese Einstellung kann nicht durch globale Einstellungen des Housekeepers überschrieben werden. **Store up to** - geben Sie die Dauer an, für die die detaillierte Historie in der Datenbank aufbewahrt wird (1 Stunde bis 25 Jahre). Ältere Daten werden vom Housekeeper entfernt. Speicherung in Sekunden. Zeitsuffixe werden unterstützt, z. B. 2h, 1d. Benutzermakros werden unterstützt. Der Wert *Store up to* kann global unter *Administration* > *Housekeeping* überschrieben werden. Wenn eine globale überschreibende Einstellung vorhanden ist, wird ein oranges Info-Symbol angezeigt. Wenn Sie den Mauszeiger darauf positionieren, wird eine Warnmeldung angezeigt, z. B. *Overridden by global housekeeper settings (1d)*. Es wird empfohlen, die aufgezeichneten Werte so kurz wie möglich aufzubewahren, um die Größe der Werthistorie in der Datenbank zu reduzieren. Anstatt eine lange Historie von Werten zu speichern, können Sie Trenddaten länger aufbewahren. Siehe auch Historie und Trends.
Trends	Wählen Sie entweder: **Do not store** - Trends werden nicht gespeichert. Diese Einstellung kann nicht durch globale Einstellungen des Housekeepers überschrieben werden. **Store up to** - geben Sie die Dauer an, für die aggregierte Historie (stündliches Minimum, Maximum, Durchschnitt, Anzahl) in der Datenbank aufbewahrt wird (1 Tag bis 25 Jahre). Ältere Daten werden vom Housekeeper entfernt. Speicherung in Sekunden. Zeitsuffixe werden unterstützt, z. B. 24h, 1d. Benutzermakros werden unterstützt. Der Wert *Store up to* kann global unter *Administration* > *Housekeeping* überschrieben werden. Wenn eine globale überschreibende Einstellung vorhanden ist, wird ein oranges Info-Symbol angezeigt. Wenn Sie den Mauszeiger darauf positionieren, wird eine Warnmeldung angezeigt, z. B. *Overridden by global housekeeper settings (7d)*. *Hinweis:* Das Speichern von Trends ist für nicht numerische Daten – Character, Log und Text – nicht verfügbar. Siehe auch Historie und Trends.
Value mapping	Wenden Sie eine Wertezuordnung auf diesen Datenpunkt an. Die Wertezuordnung ändert empfangene Werte nicht, sie dient nur zur Anzeige von Daten. Sie funktioniert mit Datenpunkten vom Typ *Numeric(unsigned)*, *Numeric(float)* und *Character*. Zum Beispiel „Windows service states“.
Log time format	Nur für Datenpunkte vom Typ **Log** 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 in Unix-Zeit auf 0 gesetzt, was dem 1. Januar 1970 entspricht. Betrachten Sie zum Beispiel die folgende Zeile aus der Zabbix-agent-Logdatei: ” 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 Log-Zeitformat für diese Zeile wäre „ppppppp:yyyyMMdd:hhmmss“. Beachten Sie, dass die Zeichen „p“ und „:“ Platzhalter sind und beliebige Zeichen außer „yMdhms“ sein können.
Populates host inventory field	Sie können ein Feld des Host-Inventars auswählen, das mit dem Wert des Datenpunkts befüllt wird. Dies funktioniert, wenn die automatische Inventarisierung für den Host aktiviert ist, und mit Datenpunkten vom Typ *Numeric(unsigned)*, *Numeric(float)*, *Character* und *Text*.
Description	Geben Sie eine Beschreibung des Datenpunkts ein. Benutzermakros werden unterstützt.
Enabled	Aktivieren Sie das Kontrollkästchen, um den Datenpunkt zu aktivieren, damit er verarbeitet wird. Wenn Sie den Datenpunkt deaktivieren, wird er sofort aus dem Verlaufscache entfernt (mit Ausnahme seines letzten Werts, der für Logs beibehalten wird).

Parameter	Beschreibung
Latest data	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:
Datenpunkttyp-spezifische Felder werden auf entsprechenden Seiten beschrieben.

Note:
Wenn Sie einen vorhandenen Datenpunkt auf Vorlage-Ebene auf Host-Ebene bearbeiten, sind einige Felder schreibgeschützt. Sie können den Link in der Formularüberschrift verwenden, zur Vorlagenebene wechseln und sie dort bearbeiten. Beachten Sie dabei, dass Änderungen auf Vorlagenebene den Datenpunkt für alle Hosts ändern, mit denen die Vorlage verknüpft ist.

Die Registerkarte **Tags** ermöglicht das Definieren von Datenpunkt-Tags.



Vorverarbeitung von Datenpunkt-Werten

Die Registerkarte **Vorverarbeitung** ermöglicht es, Transformationsregeln für die empfangenen Werte zu definieren.

Datenpunkt-Test

Attention:
Um den Datenpunkt-Test durchzuführen, stellen Sie sicher, dass die Systemzeit auf dem Server und dem Proxy synchronisiert ist. Falls die Serverzeit nachgeht, kann der Datenpunkt-Test die Fehlermeldung „The task has been expired." zurückgeben. Unterschiedliche Zeitzonen auf dem Server und dem Proxy wirken sich jedoch nicht auf das Testergebnis aus.

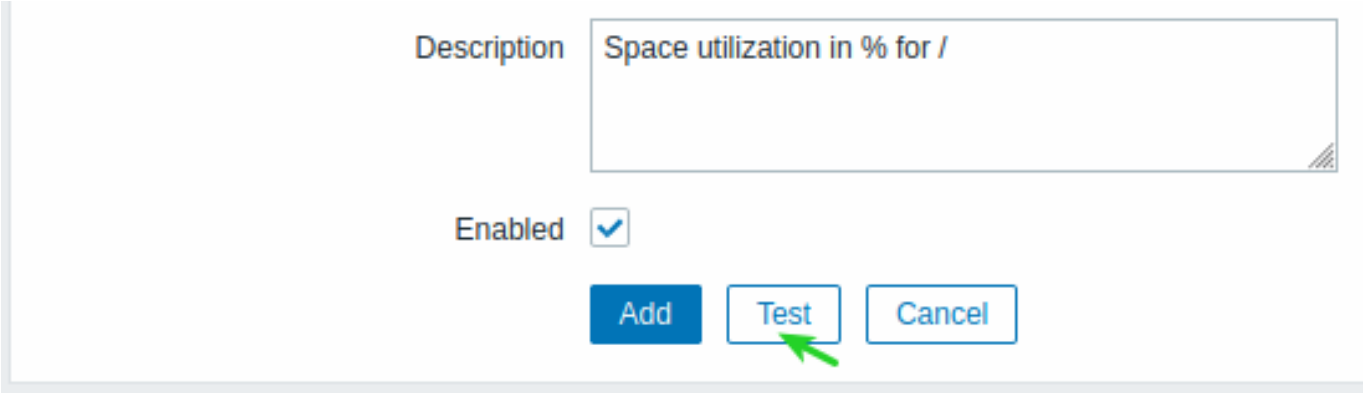
Es ist möglich, einen Datenpunkt zu testen und bei korrekter Konfiguration einen echten Wert zurückzuerhalten. Der Test kann auch erfolgen, bevor ein Datenpunkt gespeichert wird.

Der Test ist für Host- und Vorlagen-Datenpunkte, Datenpunkt-Prototypen und Low-Level-Discovery-Regeln verfügbar. Für aktive Datenpunkte ist der Test nicht verfügbar.

Der Datenpunkt-Test ist für die folgenden passiven Datenpunkt-Typen verfügbar:

- Zabbix-Agent
- SNMP-Agent (v1, v2, v3)
- IPMI-Agent
- SSH-Prüfungen
- Telnet-Prüfungen
- JMX-Agent
- Einfache Prüfungen (außer `icmpping*`- und `vmware.*`-Datenpunkten)
- Zabbix-intern
- Berechnete Datenpunkte
- Externe Prüfungen
- Datenbankmonitor
- HTTP-Agent
- Skript
- Browser

Um einen Datenpunkt zu testen, klicken Sie unten im Datenpunkt-Konfigurationsformular auf die Schaltfläche *Test*. Beachten Sie, dass die Schaltfläche *Test* für Datenpunkte deaktiviert ist, die nicht getestet werden können (z. B. aktive Prüfungen, ausgeschlossene einfache Prüfungen).

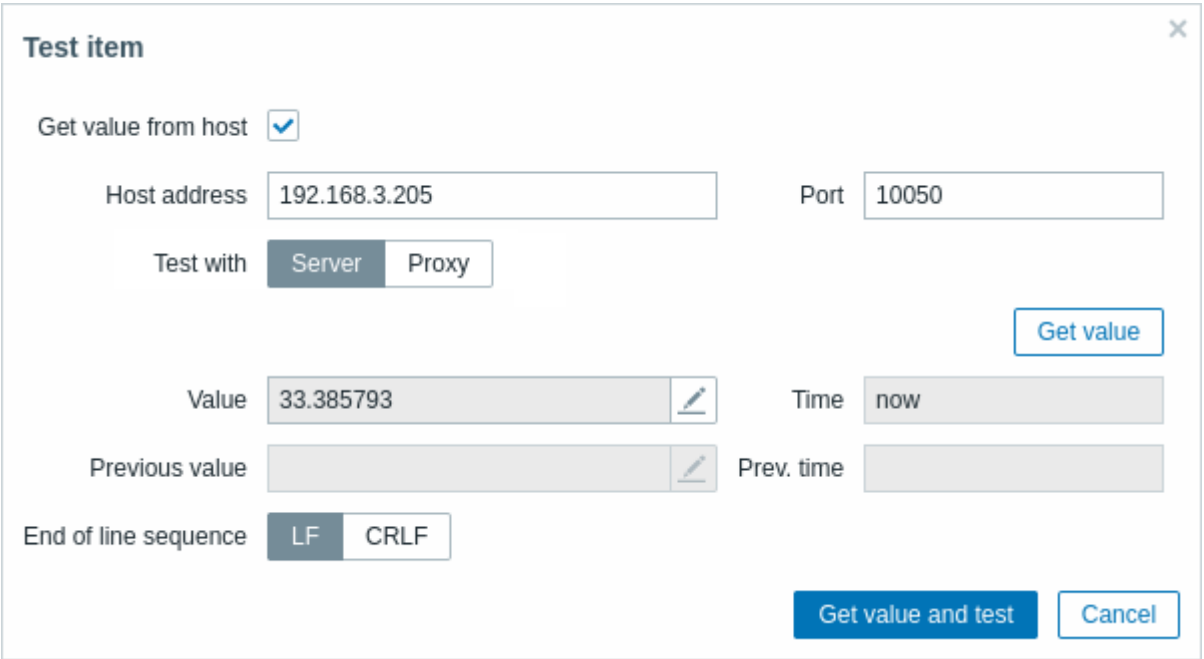


Das Formular für den Datenpunkt-Test enthält Felder für die erforderlichen Host-Parameter (Host-Adresse, Port, Test mit Server/Proxy (Proxy-Name)) und datenpunktspezifische Details (wie SNMPv2-Community oder SNMPv3-Sicherheitsanmeldedaten). Diese Felder sind kontextabhängig:

- Die Werte werden nach Möglichkeit vorausgefüllt, d. h. bei Datenpunkten, die einen Agent erfordern, werden die Informationen von der ausgewählten Agent-Schnittstelle des Hosts übernommen.
- Bei Vorlagen-Datenpunkten müssen die Werte manuell ausgefüllt werden.
- Makrowerte im Klartext werden aufgelöst.
- Felder, bei denen der Wert (oder ein Teil des Werts) ein Secret- oder Vault-Makro ist, sind leer und müssen manuell eingegeben werden. Wenn ein Datenpunkt-Parameter einen Secret-Makrowert 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 Datenpunkt-Typs nicht benötigt werden (z. B. sind das Feld für die Host-Adresse und das Proxy-Feld bei berechneten Datenpunkten deaktiviert)

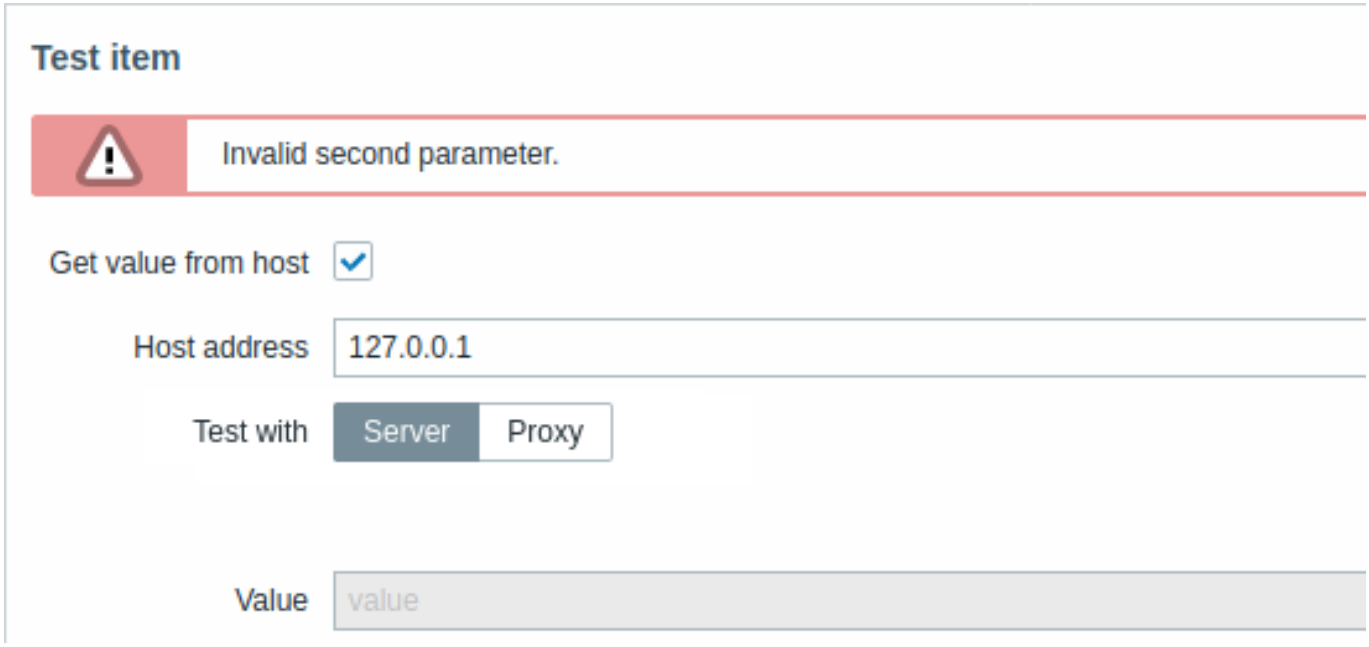
Um den Datenpunkt zu testen, klicken Sie auf *Get value*. Wenn der Wert erfolgreich abgerufen wird, wird er in das Feld *Value* eingetragen, wobei der aktuelle Wert (falls vorhanden) in das Feld *Previous value* verschoben wird. Außerdem wird das Feld *Prev. time* berechnet, d. h. die Zeitdifferenz zwischen den beiden Werten (Klicks), und es wird versucht, eine EOL-Sequenz zu erkennen und zu CRLF zu wechseln, 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 beim Senden an das Frontend auf eine maximale Größe von 512 KB gekürzt. Wenn ein Testergebnis gekürzt wird, wird ein Warnsymbol angezeigt; fahren Sie mit der Maus darüber, um Details anzuzeigen. Wenn ein Wert gekürzt wird, kann die Validierung des Datentyps bei nachfolgenden Tests fehlschlagen, in denen große Werte verglichen werden (z. B. JSON); in solchen Fällen wird eine Fehlermeldung angezeigt. Beachten Sie, dass Daten größer als 512 KB von Zabbix Server weiterhin vollständig verarbeitet werden.

Wenn die Konfiguration nicht korrekt 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.

Formularschaltflächen

Mit den Schaltflächen am unteren Rand des Formulars können mehrere Operationen ausgeführt werden.

Add	Einen Datenpunkt hinzufügen. Diese Schaltfläche ist nur für neue Datenpunkte verfügbar.
Update	Die Eigenschaften eines Datenpunkts aktualisieren.
Clone	Einen weiteren Datenpunkt auf Grundlage der Eigenschaften des aktuellen Datenpunkts erstellen.
Execute now	Eine Prüfung auf einen neuen Datenpunktwert sofort ausführen. Nur für **passive** Prüfungen unterstützt (siehe weitere Details). *Beachten Sie*, dass beim sofortigen Prüfen auf einen Wert der Konfigurations-Cache nicht aktualisiert wird; daher spiegelt der Wert keine sehr aktuellen Änderungen an der Datenpunktkonfiguration wider.
Test	Testen, ob die Datenpunktkonfiguration korrekt ist, indem ein Wert abgerufen wird.
Clear history and trends	Den Datenpunktverlauf und die Trends 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 bei 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 gekürzt, damit sie dem Grenzwert des Datenbank-Wertetyps entsprechen:

Datenbank	Informationstyp		
	Zeichen	**Log**	**Text**
MySQL	255 Zeichen	65536 Bytes	65536 Bytes
PostgreSQL	255 Zeichen	65536 Zeichen	65536 Zeichen
SQLite (nur Zabbix Proxy)	255 Zeichen	65536 Zeichen	65536 Zeichen

Binärwerte sind auf 16MiB (16777216 Bytes) begrenzt. Werte, die diesen Grenzwert überschreiten, werden verworfen, und ein entsprechender Fehler wird im Frontend angezeigt.

Werte mit dem Datentyp JSON sind auf 128MiB (134217728 Bytes) begrenzt. Werte, die diesen Grenzwert überschreiten, werden verworfen, und ein entsprechender Fehler wird im Frontend angezeigt. Für MySQL/MariaDB sollten Sie die Einrichtung täglicher Partitionen für die Tabelle `history_json` (zum Speichern von JSON-Werten verwendet) in Betracht ziehen, da sie schnell sehr groß werden und den housekeeper für längere Zeit blockieren kann. Wenn Ihr Anwendungsfall große JSON-Werte (1MiB oder mehr) umfasst, lesen Sie, wie Sie Ihr System für die Unterstützung großer JSON-Werte konfigurieren.

Wenn Zabbix beliebige Datenpunkt-Daten in die Datenbank schreibt, verwendet es INSERT-Abfragen und protokolliert diese (wenn `DebugLevel` auf 4 oder 5 gesetzt ist). Große Log-Einträge werden auf 64KB 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

Timeout für Datenpunkte

Der Timeout für Datenpunkte gibt an, wie lange Zabbix warten soll, bevor die Prüfung als fehlgeschlagen abgebrochen wird.

Wenn der Timeout erreicht ist, wird die Prüfung abgebrochen, auch wenn der Datenabruf noch nicht abgeschlossen ist. Wenn Daten nur teilweise empfangen werden, wird der Datenpunkt nicht unterstützt und eine Fehlermeldung wird protokolliert (zum Beispiel, wenn bei einer SNMP-Prüfung Daten erfolgreich nur für eine von mehreren OIDs erfasst werden).

Für viele Datenpunkt-Typen können Sie **flexible** Timeouts für Datenpunkte festlegen:

- pro einzelner Datenpunkt
- pro Datenpunkt-Typ (auf Proxy-Ebene)
- pro Datenpunkt-Typ (auf globaler Ebene)

Ein benutzerdefinierter Timeout *pro einzelner Datenpunkt* ist nützlich, wenn Sie für einen bestimmten Datenpunkt einen längeren Timeout festlegen möchten, während die Timeouts für andere Datenpunkte niedrig bleiben.

Reihenfolge der Timeout-Priorität

1. Der Timeout eines einzelnen Datenpunkts überschreibt alle anderen Timeouts.
2. Timeouts auf Proxy-Ebene überschreiben globale Timeouts.

Der Timeout aus der Konfiguration von Zabbix Server oder Proxy spielt **keine** Rolle bei Prüfungen, für die flexible Timeouts festgelegt sind.

Unterstützung für flexible Timeouts

Flexible Datenpunkt-Timeouts werden für diese Datenpunkt-Typen unterstützt:

- Zabbix Agent (sowohl passive als auch aktive Prüfungen)
- SNMP-Agent (außer Legacy-SNMP-Prüfungen[1])
- Einfacher Check (außer `icmpping*` und VMware-Datenpunkte[2])
- SSH-Agent
- Telnet-Agent
- Externer Check
- Datenbankmonitor
- Skript
- HTTP-Agent
- Browser

[1] Für Legacy-SNMP-Prüfungen (Abfrage einer einzelnen OID) gelten die Timeout-Einstellungen aus der Server- oder Proxy-Konfiguration.
 [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 Parameter VMwareTimeout aus der Server- oder Proxy-Konfiguration.

Nicht unterstützte Datenpunkte

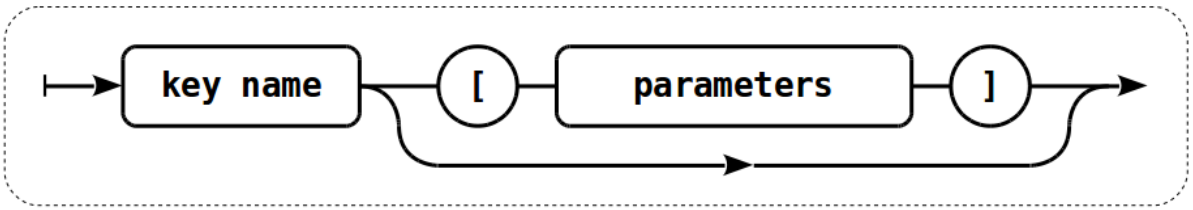
Nicht unterstützte Datenpunkte werden mit dem Status *Nicht unterstützt* gemeldet und weiterhin in ihrem normalen *Aktualisierungsintervall* erneut geprüft.

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).

1 Format des Datenpunktschlüssels

Das Format des Datenpunktschlüssels, einschließlich der Schlüsselparameter, muss den Syntaxregeln entsprechen. Die folgenden Abbildungen zeigen die unterstützte Syntax. Zulässige Elemente und Zeichen an jeder Stelle können durch Verfolgen der Pfeile bestimmt werden – wenn ein Block über die Linie erreicht werden kann, ist er zulässig, andernfalls ist er nicht zulässig.



Um einen gültigen Datenpunktschlüssel zu erstellen, beginnt man mit der Angabe des Schlüsselnamens. Anschließend kann gewählt werden, ob Parameter verwendet werden oder nicht – wie durch die zwei Linien dargestellt, denen gefolgt werden kann.

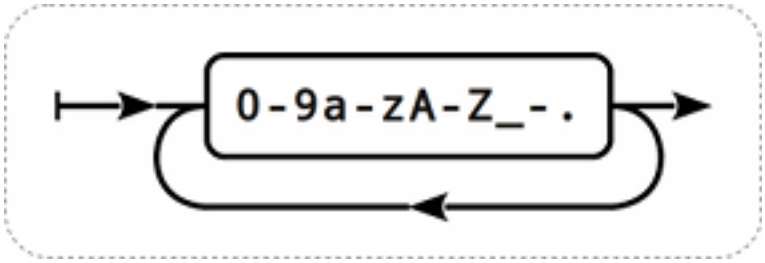
Schlüsselname

Der Schlüsselname selbst hat einen begrenzten Bereich zulässiger Zeichen, die einfach aufeinander folgen. Zulässige Zeichen sind:

`0-9a-zA-Z_-.`

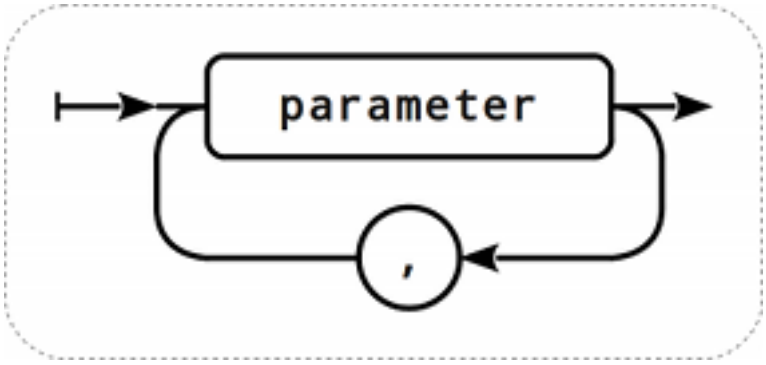
Das bedeutet:

- alle Zahlen;
- alle Kleinbuchstaben;
- alle Großbuchstaben;
- Unterstrich;
- Bindestrich;
- Punkt.

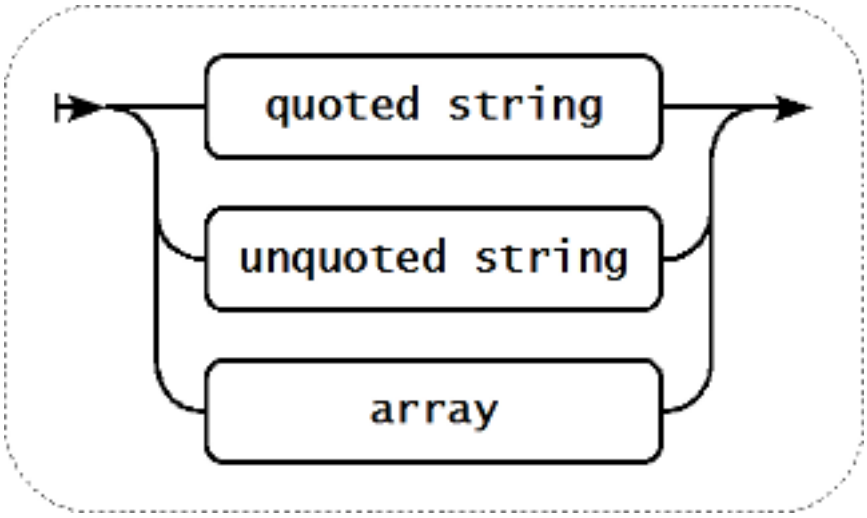


Schlüsselparameter

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.

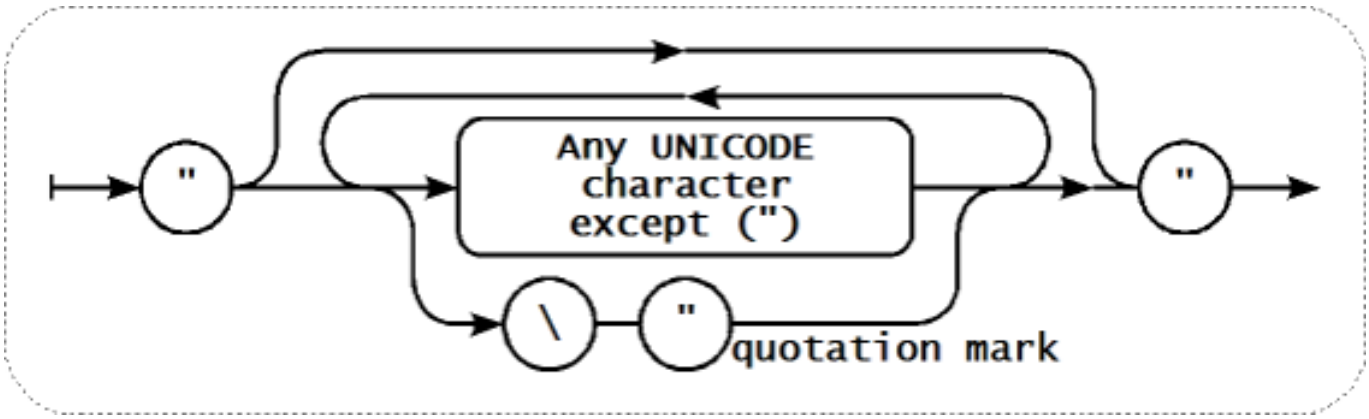


Der Parameter kann auch leer gelassen werden, sodass der Standardwert verwendet wird. In diesem Fall muss die entsprechende Anzahl von Kommas hinzugefügt werden, wenn weitere Parameter angegeben werden. Zum Beispiel würde der Datenpunktschlüssel **icmpping[„200„500]** festlegen, dass das Intervall zwischen 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.

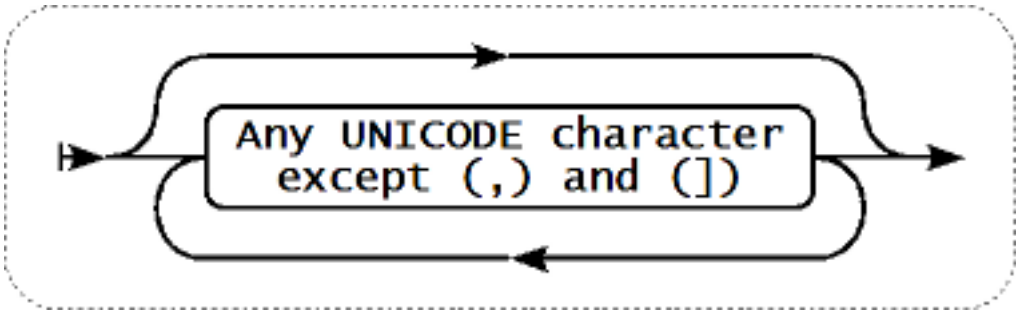


Warning:
Verwenden Sie zum Setzen von Anführungszeichen bei Datenpunkt-Schlüsselparametern ausschließlich doppelte Anführungszeichen. Einfache Anführungszeichen werden nicht unterstützt.

Attention:
Mehrstufige Parameter-Arrays, z. B. `[a,[b,[c,d]],e]`, sind nicht zulässig.

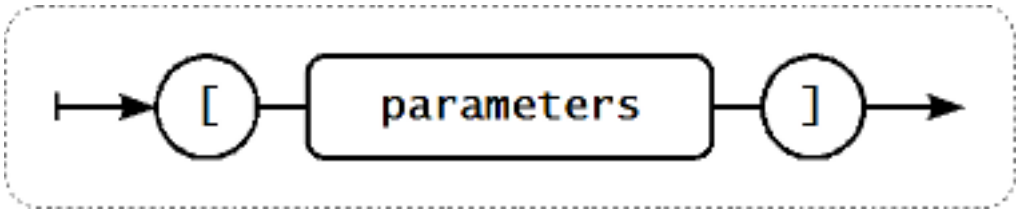
Parameter - nicht in Anführungszeichen gesetzte Zeichenkette

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.



2 Benutzerdefinierte Intervalle

Übersicht

Es ist möglich, benutzerdefinierte Regeln für die Zeiten zu erstellen, zu denen ein Datenpunkt geprüft wird. Dafür gibt es zwei Methoden: *Flexible Intervalle*, mit denen das Standard-Aktualisierungsintervall neu definiert werden kann, und *Zeitplanung*, bei der eine Datenpunkt-Prüfung zu einer bestimmten Zeit oder zu einer Folge von Zeiten ausgeführt werden kann.

Flexible Intervalle

Flexible Intervalle ermöglichen es, das Standard-Aktualisierungsintervall für bestimmte Zeiträume neu zu definieren. Ein flexibles Intervall wird mit *Intervall* und *Zeitraum* definiert, wobei gilt:

- *Intervall* – das Aktualisierungsintervall für den angegebenen Zeitraum. Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d.
- *Zeitraum* – der Zeitraum, in dem das flexible Intervall aktiv ist (siehe Zeiträume für eine detaillierte Beschreibung des Formats von *Zeitraum*)

Wenn sich mehrere flexible Intervalle überschneiden, wird für den überschneidenden Zeitraum der kleinste *Intervall*-Wert verwendet. Beachten Sie, dass bei einem kleinsten Wert von '0' in sich überschneidenden flexiblen Intervallen keine Abfrage erfolgt. Außerhalb der flexiblen Intervalle wird das Standard-Aktualisierungsintervall verwendet.

Beachten Sie, dass der Datenpunkt genau einmal geprüft wird, wenn das flexible Intervall der Länge des Zeitraums entspricht. Wenn das flexible Intervall größer als der Zeitraum ist, wird der Datenpunkt möglicherweise einmal geprüft oder überhaupt nicht geprüft (daher wird eine solche Konfiguration nicht empfohlen). Wenn das flexible Intervall kleiner 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 die Abfrage wird nach Ablauf des Zeitraums gemäß dem Standard-*Aktualisierungsintervall* fortgesetzt. Beispiele:

Intervall	Zeitraum	Beschreibung
10	1-5,09:00-18:00	Datenpunkt wird während der Arbeitszeit alle 10 Sekunden geprüft.
0	1-7,00:00-7:00	Datenpunkt wird nachts nicht geprüft.
0	7-7,00:00-24:00	Datenpunkt wird sonntags nicht geprüft.
60	1-7,12:00-12:01	Datenpunkt wird jeden Tag um 12:00 geprüft. Beachten Sie, dass dies als Behelfslösung für geplante Prüfungen verwendet wurde; für solche Prüfungen wird die Verwendung von Planungsintervallen empfohlen.

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 auch `<from>` weggelassen wird, dann entspricht der Filter allen möglichen Werten.
- `<step>` definiert die Sprünge des Zahlenwerts 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 niedrigerer Ebene definiert ist, oder andernfalls allen möglichen Werten. Wenn zum Beispiel der Stundenfilter weggelassen wird, dann entspricht nur die Stunde '0', vorausgesetzt, Minuten- und Sekundenfilter werden ebenfalls weggelassen; andernfalls entspricht ein leerer Stundenfilter allen Stundenwerten.

Gültige `<from>`- und `<to>`-Werte für ihr jeweiliges 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, aber `md01-031` und `wd01-07` 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 durch mehrere Intervalle 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

Intervall	Wird ausgeführt
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"
```

2 Vorverarbeitung von Datenpunktwerten

Übersicht

Die Vorverarbeitung ermöglicht es Ihnen, Transformationen auf die empfangenen Datenpunkt-Werte anzuwenden, bevor sie in der Datenbank gespeichert werden. Diese Transformationen/Vorverarbeitungsschritte werden vom Zabbix Server oder Proxy ausgeführt (wenn Datenpunkte vom 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;
- Anwenden regulärer Ausdrücke, um Werte zu extrahieren oder zu ändern;
- Ausführen 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 *Custom on fail* 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.

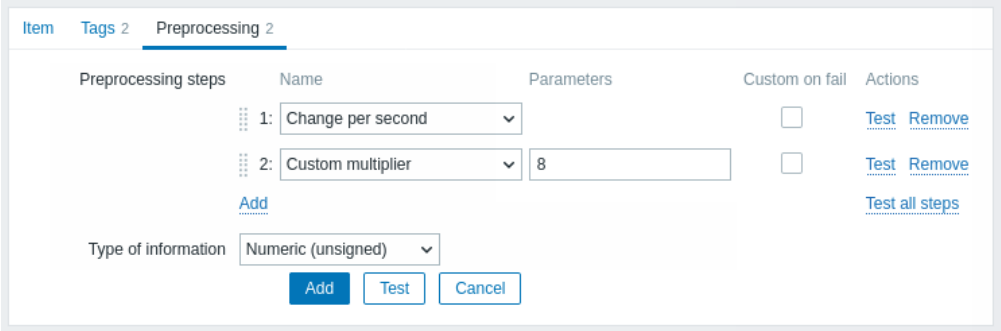
Alle an die Vorverarbeitung übergebenen Werte werden zunächst als Zeichenfolgen behandelt. Die Konvertierung in den gewünschten Werttyp (wie in der Datenpunkt-Konfiguration definiert) erfolgt am Ende der Vorverarbeitungspipeline. Bestimmte Vorverarbeitungsschritte können jedoch bei Bedarf frühere Konvertierungen auslösen. Detaillierte 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 im Reiter **Preprocessing** des Konfigurationsformulars für den Datenpunkt definiert.



Klicken Sie auf *Add*, um eine unterstützte Transformation auszuwählen.

Das Feld *Type of information* wird am unteren Rand des Reiters angezeigt, wenn mindestens ein Vorverarbeitungsschritt definiert ist. Falls erforderlich, kann der Informationstyp geändert werden, ohne den Reiter *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 der Transformation, um vollständige Details dazu anzuzeigen.

Name	Beschreibung	Typ
Regulärer Ausdruck	Vergleichen Sie den Wert mit dem regulären Ausdruck und ersetzen Sie ihn durch die erforderliche Ausgabe.	Text
Ersetzen	Suchen Sie die Suchzeichenfolge und ersetzen Sie sie durch eine andere (oder nichts).	
Trimmen	Entfernen Sie angegebene Zeichen vom Anfang und Ende des Werts.	
Rechts trimmen	Entfernen Sie angegebene Zeichen vom Ende des Werts.	
Links trimmen	Entfernen Sie angegebene Zeichen vom Anfang des Werts.	
XML XPath	Extrahieren Sie einen Wert oder ein Fragment aus XML-Daten mithilfe der XPath-Funktionalität.	Strukturierte Daten
JSON Path	Extrahieren Sie einen Wert oder ein Fragment aus JSON-Daten mithilfe der JSONPath-Funktionalität.	
CSV zu JSON	Konvertieren Sie CSV-Dateidaten in das JSON-Format.	
XML zu JSON	Konvertieren Sie Daten im XML-Format in JSON.	
SNMP-Walk-Wert	Extrahieren Sie einen Wert anhand des angegebenen OID-/MIB-Namens und wenden Sie Formatierungsoptionen an.	SNMP
SNMP-Walk zu JSON	Konvertieren Sie SNMP-Werte in JSON.	
SNMP-Get-Wert	Wenden Sie Formatierungsoptionen auf den SNMP-Get-Wert an.	
Benutzerdefinierter Multiplikator	Multiplizieren Sie den Wert mit dem angegebenen Ganzzahl- oder Gleitkommawert.	Arithmetik
Einfache Änderung	Berechnen Sie die Differenz zwischen dem aktuellen und dem vorherigen Wert.	Änderung
Änderung pro Sekunde	Berechnen Sie die Geschwindigkeit der Wertänderung (Differenz zwischen dem aktuellen und dem vorherigen Wert) pro Sekunde.	
Boolean zu Dezimal	Konvertieren Sie den Wert vom booleschen Format in das Dezimalformat.	Zahlensysteme

Name	Beschreibung	Typ
Oktal zu Dezimal	Konvertieren Sie den Wert vom oktalen Format in das Dezimalformat.	
Hexadezimal zu Dezimal	Konvertieren Sie den Wert vom hexadezimalen Format in das Dezimalformat.	
JavaScript	Geben Sie JavaScript-Code ein.	Benutzerdefinierte Skripte
Im Bereich	Definieren Sie einen Bereich, in dem sich ein Wert befinden soll.	Validierung
Entspricht regulärem Ausdruck	Geben Sie einen regulären Ausdruck an, dem ein Wert entsprechen muss.	
Entspricht nicht regulärem Ausdruck	Geben Sie einen regulären Ausdruck an, dem ein Wert nicht entsprechen darf.	
Auf Fehler in JSON prüfen	Prüfen Sie auf eine Fehlermeldung auf Anwendungsebene, die sich an JSONPath befindet.	
Auf Fehler in XML prüfen	Prüfen Sie auf eine Fehlermeldung auf Anwendungsebene, die sich an XPath befindet.	
Mit regulärem Ausdruck auf Fehler prüfen	Prüfen Sie mithilfe eines regulären Ausdrucks auf eine Fehlermeldung auf Anwendungsebene.	
Auf nicht unterstützten Wert prüfen	Prüfen Sie, ob kein Datenpunkt-Wert abgerufen werden konnte.	
Unveränderte Werte verwerfen	Verwerfen Sie einen Wert, wenn er sich nicht geändert hat.	Drosselung
Unveränderte Werte mit Heartbeat verwerfen	Verwerfen Sie einen Wert, wenn er sich innerhalb des definierten Zeitraums nicht geändert hat.	
Prometheus-Muster	Verwenden Sie die folgende Abfrage, um die erforderlichen Daten aus Prometheus-Metriken zu extrahieren.	Prometheus
Prometheus zu JSON	Konvertieren Sie die erforderlichen Prometheus-Metriken in JSON.	

Beachten Sie, dass Zabbix für die Vorverarbeitungsschritte *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 verwaltet. Wenn der Zabbix Server oder Proxy neu gestartet wird oder Änderungen an den Vorverarbeitungsschritten vorgenommen werden, wird der letzte Wert des entsprechenden Datenpunkts zurückgesetzt, was zu Folgendem führt:

- bei den Schritten *Einfache Änderung* und *Änderung pro Sekunde* wird der nächste Wert ignoriert, da kein vorheriger Wert vorhanden ist, aus dem 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 hätte verworfen werden sollen.

Regulärer Ausdruck

Den Wert mit dem regulären Ausdruck abgleichen und durch die erforderliche Ausgabe ersetzen.

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:

- Wenn der Eingabewert nicht übereinstimmt, wird der Datenpunkt nicht unterstützt;
- Der reguläre Ausdruck unterstützt die Extraktion von maximal 10 erfassten Gruppen mit der Sequenz \N;
- Wenn Sie das Kontrollkästchen *Custom on fail* markieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht nicht unterstützt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.
- Einige vorhandene Beispiele finden Sie im Abschnitt regular expressions.

Ersetzen

Suchen Sie die Suchzeichenfolge und ersetzen Sie sie durch eine andere (oder nichts).

Parameter:

- **Suchzeichenfolge** - die zu findende und zu ersetzende Zeichenfolge, Groß-/Kleinschreibung wird beachtet (erforderlich);

- **Ersetzung** - die Zeichenfolge, durch die die Suchzeichenfolge ersetzt wird. Die Ersetzungszeichenfolge kann auch leer sein, wodurch die Suchzeichenfolge beim Auffinden effektiv gelöscht werden kann.

Kommentare:

- Alle Vorkommen der Suchzeichenfolge werden ersetzt;
- Es ist möglich, Escape-Sequenzen zu verwenden, um nach Zeilenumbrüchen, Wagenrückläufen, Tabulatoren und Leerzeichen ”\n \r \t \s” zu suchen oder diese zu ersetzen; der Backslash kann als ”\\” maskiert werden und Escape-Sequenzen können als ”\\n” maskiert werden;
- Das Maskieren von Zeilenumbrüchen, Wagenrückläufen und Tabulatoren erfolgt bei der Low-Level-Discovery automatisch.

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 libxml-Unterstützung kompiliert sein;
- Namespaces werden nicht unterstützt;
- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Wenn ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

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 Path

Extrahieren Sie einen Wert oder ein Fragment aus JSON-Daten mithilfe der JSONPath-Funktionalität.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht als nicht unterstützt markiert, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

CSV in JSON

Konvertiert CSV-Dateidaten in das JSON-Format.

Weitere Informationen finden Sie unter: CSV-in-JSON-Präprozessierung.

XML in JSON umwandeln

Daten im XML-Format in JSON umwandeln.

Weitere Informationen finden Sie unter: Serialisierungsregeln.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht auf nicht unterstützt gesetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

SNMP-Walk-Wert

Wert anhand des angegebenen OID-/MIB-Namens extrahieren und Formatierungsoptionen anwenden:

- **Unverändert** - Hex-String als nicht maskierten Hex-String zurückgeben (*beachten Sie*, dass Anzeigehinweise weiterhin angewendet werden);
- **UTF-8 aus Hex-STRING** - Hex-String in eine UTF-8-Zeichenkette umwandeln;

- **MAC aus Hex-STRING** - Hex-String als MAC-Adresse validieren und eine korrekte MAC-Adresszeichenkette zurückgeben (wobei ' ' durch ':' ersetzt werden);
- **Integer aus BITS** - die ersten 8 Bytes einer Bit-Zeichenkette, dargestellt als Folge von Hex-Zeichen (z. B. „1A 2B 3C 4D"), in eine vorzeichenlose 64-Bit-Ganzzahl umwandeln. Bei Bit-Zeichenketten mit mehr als 8 Bytes werden nachfolgende Bytes ignoriert.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht auf „nicht unterstützt" gesetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

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 Werten aus dem angegebenen SNMP-OID-Pfad befüllt.

Kommentare:

- Ähnliche Optionen zur Wertformatierung wie im Schritt *SNMP-Walk-Wert* sind verfügbar;
- Sie können diesen Vorverarbeitungsschritt für die SNMP-OID-Erkennung verwenden;
- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht in den Status „nicht unterstützt" versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

SNMP-Get-Wert

Wenden Sie Formatierungsoptionen auf den SNMP-Get-Wert an:

- **UTF-8 aus Hex-STRING** - Hex-String in eine UTF-8-Zeichenkette umwandeln;
- **MAC aus Hex-STRING** - Hex-String als MAC-Adresse validieren und eine korrekte MAC-Adresszeichenkette zurückgeben (wobei ' ' durch ':' ersetzt werden);
- **Integer aus BITS** - die ersten 8 Bytes einer Bit-Zeichenkette, dargestellt als Folge von Hex-Zeichen (z. B. „1A 2B 3C 4D"), in eine vorzeichenlose 64-Bit-Ganzzahl umwandeln. Bei Bit-Zeichenketten mit mehr als 8 Bytes werden die nachfolgenden Bytes ignoriert.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung angeben: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Falle eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt" versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Benutzerdefinierter Multiplikator

Multiplizieren Sie den Wert mit dem angegebenen Ganzzahl- oder Gleitkommawert.

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 setzen.
- Beachten Sie, dass, wenn der Informations-Typ des Datenpunkts *Numeric (unsigned)* ist, eingehende Werte mit einem Nachkommateil vor Anwendung des benutzerdefinierten Multiplikators abgeschnitten werden (d. h. „0.9" wird zu „0");
- Wenn Sie einen benutzerdefinierten Multiplikator verwenden oder den Wert als *Change per second* für Datenpunkte speichern, deren Informations-Typ auf *Numeric (unsigned)* gesetzt ist, und der 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 Notation, zum Beispiel 1e+70; Benutzermakros und LLD-Makros; Zeichenfolgen, die Makros enthalten, zum Beispiel `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Die Makros müssen zu einer Ganzzahl oder einer Gleitkommazahl aufgelöst werden.
- Wenn Sie das Kontrollkästchen *Custom on fail* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht auf „nicht unterstützt" gesetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Einfache Änderung

Berechnet die Differenz zwischen dem aktuellen und dem vorherigen Wert.

Kommentare:

- Dieser Schritt kann nützlich sein, um einen ständig wachsenden Wert zu messen;

- Ausgewertet als **Wert**-**prev_value**, wobei *Wert* der aktuelle Wert ist; *prev_value* der zuvor empfangene Wert;
- Pro Datenpunkt ist nur eine Änderungsoperation („Einfache Änderung“ oder „Änderung pro Sekunde“) zulässig;
- Wenn der aktuelle Wert kleiner ist als der vorherige Wert, verwirft Zabbix diese Differenz (speichert nichts) und wartet auf einen weiteren Wert;
- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

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 ständig wachsenden Werts zu berechnen;
- Da diese Berechnung Gleitkommazahlen erzeugen kann, wird empfohlen, den „Typ der Information“ auf *Numerisch (Gleitkommazahl)* zu setzen, auch wenn die eingehenden Rohwerte Ganzzahlen sind. Dies ist besonders bei kleinen Zahlen relevant, bei denen der Dezimalteil wichtig ist. 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 damit nur den Dezimalteil abzuschneiden.
- Ausgewertet als (**value**-**prev_value**)/(**time**-**prev_time**), wobei *value* der aktuelle Wert ist, *prev_value* der zuvor empfangene Wert, *time* der aktuelle Zeitstempel und *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 ist als der vorherige Wert, verwirft Zabbix diese Differenz (speichert nichts) und wartet auf einen weiteren Wert. Dies hilft dabei, beispielsweise mit einem Umlauf (Overflow) von 32-Bit-SNMP-Zählern korrekt zu arbeiten.
- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht auf „nicht unterstützt“ gesetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Boolesch in Dezimal umwandeln

Wandelt den Wert vom booleschen Format in ein Dezimalformat um.

Kommentare:

- Die textuelle Darstellung wird entweder in 0 oder 1 umgewandelt. Somit wird „TRUE“ als 1 gespeichert und „FALSE“ als 0. Alle Werte werden ohne Beachtung 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. Zusätzlich wird jeder numerische Wert ungleich null als TRUE betrachtet, und null wird als FALSE betrachtet.
- Wenn Sie das Kontrollkästchen *Custom on fail* markieren, können benutzerdefinierte Optionen für die Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht auf „nicht unterstützt“ gesetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Oktal in Dezimal umwandeln

Wandelt den Wert vom Oktalformat in das Dezimalformat um.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können Sie benutzerdefinierte Optionen für die Fehlerbehandlung festlegen: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Hexadezimal in Dezimal

Konvertiert den Wert vom Hexadezimalformat in das Dezimalformat.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können Sie benutzerdefinierte Optionen für die Fehlerbehandlung angeben: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

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 Mindest-/Höchstwerte (einschließlich) angeben.

Kommentare:

- Numerische Werte werden akzeptiert (einschließlich einer beliebigen Anzahl von Ziffern, optionalem Dezimalteil und optionalem Exponentialteil, negativen Werten);
- Der Mindestwert muss kleiner als der Höchstwert sein;
- Mindestens ein Wert muss vorhanden sein;
- Benutzermakros und Low-Level-Discovery-Makros können verwendet werden;
- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen für die Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Falle eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Entspricht regulärem Ausdruck

Geben Sie einen regulären Ausdruck an, dem ein Wert entsprechen muss.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Falle eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Entspricht nicht dem regulären Ausdruck

Geben Sie einen regulären Ausdruck an, dem ein Wert nicht entsprechen darf.

Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: Entweder wird der Wert verworfen, ein angegebener Wert gesetzt oder eine angegebene Fehlermeldung gesetzt. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht auf „nicht unterstützt“ gesetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Auf Fehler in JSON prüfen

Prüft auf eine Fehlermeldung auf Anwendungsebene, die sich unter dem JSONPath befindet. Die Verarbeitung wird beendet, wenn dies 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;
- Bei einem Fehlschlag beim Parsen von ungültigem JSON wird kein Fehler gemeldet;
- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Falle eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Auf Fehler in XML prüfen

Prüft auf eine Fehlermeldung auf Anwendungsebene, die sich am XPath befindet. Die Verarbeitung wird beendet, wenn sie 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;
- Bei einem Fehlschlag beim Parsen von ungültigem XML wird kein Fehler gemeldet;

- Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* markieren, können benutzerdefinierte Optionen für die Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

Auf Fehler mit einem regulären Ausdruck prüfen

Prüfen Sie mit einem regulären Ausdruck auf eine Fehlermeldung auf Anwendungsebene. Beenden Sie die Verarbeitung bei Erfolg, wenn die Meldung nicht leer ist; andernfalls setzen Sie die Verarbeitung mit dem Wert fort, 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;
- Wenn Sie das Kontrollkästchen *Custom on fail* markieren, können benutzerdefinierte Optionen für die Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Im Fall eines fehlgeschlagenen Vorverarbeitungsschritts wird der Datenpunkt nicht in den Status „nicht unterstützt“ versetzt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.

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 Geltungsbereich der Fehlerverarbeitung aus:
beliebiger Fehler - jeder Fehler;
Fehler entspricht - nur der Fehler, der dem in *pattern* angegebenen regulären Ausdruck entspricht;
Fehler entspricht nicht - nur der Fehler, der nicht dem in *pattern* angegebenen regulären Ausdruck entspricht
- **pattern** - der reguläre Ausdruck, mit dem der Fehler abgeglichen wird. Wenn im Parameter scope *beliebiger Fehler* ausgewählt ist, wird dieses Feld nicht angezeigt. Falls angezeigt, ist dieses Feld erforderlich.

Kommentare:

- Normalerweise würde das Fehlen bzw. der Fehler beim Abrufen eines Werts dazu führen, dass der Datenpunkt nicht unterstützt wird. Mit diesem Vorverarbeitungsschritt können Sie dieses Verhalten ändern. Wenn Sie das Kontrollkästchen *Custom on fail* markieren (bei diesem Vorverarbeitungsschritt immer markiert und ausgegraut), können benutzerdefinierte Optionen für die Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen. Falls ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht nicht unterstützt, wenn die Option zum Verwerfen des Werts oder zum Setzen eines angegebenen Werts ausgewählt ist.
- Dieser Vorverarbeitungsschritt prüft nur, ob kein Datenpunktwert abgerufen werden konnte. Er prüft beispielsweise nicht, ob der Typ des abgerufenen Werts (z. B. Zeichenkette) mit dem Informationstyp des Datenpunkts (z. B. numerisch) übereinstimmt; siehe dazu Beispiele zur Vorverarbeitung. Wenn ein Typkonflikt vorliegt, kann der Datenpunkt nach Ausführung aller Vorverarbeitungsschritte dennoch nicht unterstützt werden. Um auf einen Typkonflikt zu prüfen, können Sie beispielsweise den Vorverarbeitungsschritt *Custom multiplier* verwenden; siehe Beispiele zur Vorverarbeitung.
- Das Erfassen von Gruppen in regulären Ausdrücken 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 über allen anderen platziert;
- Mehrere Schritte *Auf nicht unterstützten Wert prüfen* werden in der angegebenen Reihenfolge unterstützt. Ein Schritt für *beliebiger Fehler* wird in dieser Gruppe automatisch als letzter Schritt 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

Verwirft einen Wert, wenn er sich innerhalb des definierten Zeitraums (in Sekunden) nicht geändert hat.

Kommentare:

- Positive Ganzzahlwerte werden zur Angabe der Sekunden unterstützt (Minimum: 1 Sekunde);
- Zeitsuffixe können verwendet werden (z. B. 30s, 1m, 2h, 1d);
- Benutzermakros und Low-Level-Discovery-Makros können verwendet werden;
- 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, daher werden keine Probleme für zugehörige Auslöser erstellt/gelöst. Funktionen arbeiten nur auf Basis von Daten, die tatsächlich in der Datenbank gespeichert sind. Da Trends auf Basis der Daten in der Datenbank erstellt werden, gibt es auch keine Trenddaten für diese Stunde, wenn eine Stunde lang kein Wert gespeichert wird.
- 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 in JSON umwandeln

Wandeln Sie die erforderlichen Prometheus-Metriken in JSON um.

Weitere Informationen 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.

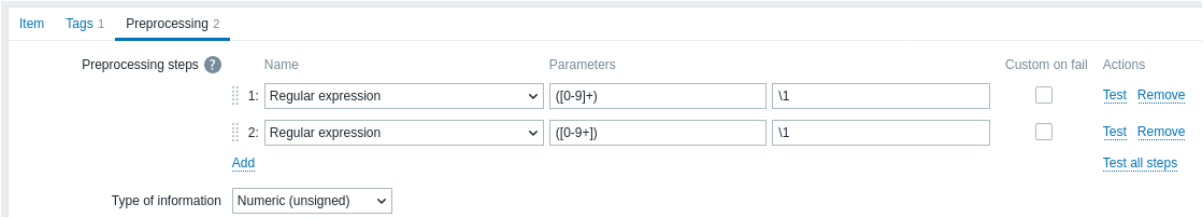
Testen

Siehe Testen der Vorverarbeitung.

1 Testen der Vorverarbeitung

Testen

Das Testen von Vorverarbeitungsschritten ist nützlich, um sicherzustellen, dass komplexe Vorverarbeitungs-Pipelines die von ihnen erwarteten Ergebnisse liefern, ohne darauf warten zu müssen, dass der Datenpunkt-Wert empfangen und vorverarbeitet wird.

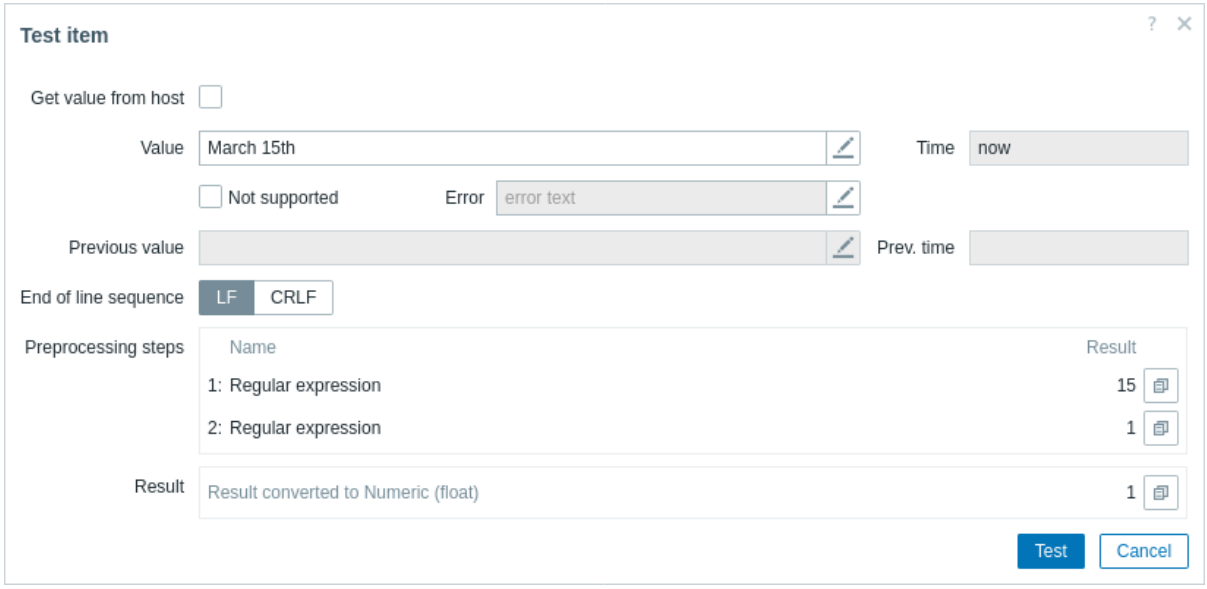


Es ist möglich, zu testen:

- mit einem hypothetischen Wert
- mit einem realen Wert von einem Host

Jeder Vorverarbeitungsschritt kann einzeln getestet werden, ebenso können alle Schritte zusammen getestet werden. Wenn Sie im Block *Aktionen* auf die Schaltfläche *Test* bzw. *Alle Schritte testen* klicken, wird ein Testfenster geöffnet.

Testen eines hypothetischen Werts



Parameter	Beschreibung
Wert vom Host abrufen	Wenn Sie einen hypothetischen Wert testen möchten, lassen Sie dieses Kontrollkästchen deaktiviert. Siehe auch: Testen eines echten Werts.
Wert	Geben Sie den Eingabewert zum Testen ein. Ein Klick in das Parameterfeld oder auf die Schaltfläche zum Anzeigen/Bearbeiten öffnet ein Textbereichsfenster zur Eingabe des Werts oder eines Codeblocks.
Nicht unterstützt	Aktivieren Sie dieses Kontrollkästchen, um einen nicht unterstützten Wert zu testen. Diese Option ist nützlich, um den Vorverarbeitungsschritt *Prüfung auf nicht unterstützten Wert* zu testen.
Fehler	Geben Sie den Fehlertext ein. Dieses Feld ist aktiviert, wenn *Wert vom Host abrufen* deaktiviert ist, aber *Nicht unterstützt* aktiviert ist. Wenn *Wert vom Host abrufen* aktiviert ist, wird dieses Feld mit der tatsächlichen Fehlermeldung (schreibgeschützt) vom Host gefüllt.
Zeit	Die Zeit des Eingabewerts wird angezeigt: `now` (schreibgeschützt).
Vorheriger Wert	Geben Sie einen vorherigen Eingabewert zum Vergleichen ein. Nur für die Vorverarbeitungsschritte *Änderung* und *Drosselung*.
Vorherige Zeit	Geben Sie die Zeit des vorherigen Eingabewerts zum Vergleichen ein. Nur für die Vorverarbeitungsschritte *Änderung* und *Drosselung*. Der Standardwert basiert auf dem Feldwert des Datenpunkts für „Aktualisierungsintervall“ (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`.
Makros	Wenn Makros verwendet werden, werden sie zusammen mit ihren Werten aufgelistet. Die Werte können zu Testzwecken bearbeitet werden, die Änderungen werden jedoch nur im Testkontext gespeichert.
Zeilenendesequenz	Wählen Sie die Zeilenendesequenz für mehrzeilige Eingabewerte aus: **LF** - LF-Sequenz (line feed) **CRLF** - CRLF-Sequenz (carriage-return line-feed).
Vorverarbeitungsschritte	Die Vorverarbeitungsschritte werden aufgelistet; das Testergebnis wird für jeden Schritt angezeigt, nachdem auf die Schaltfläche *Test* geklickt wurde. Testergebnisse werden beim Senden an das Frontend auf eine maximale Größe von 512 KB gekürzt. Testergebnisse können kopiert werden (nicht mehr als die gekürzten 512 KB). Wenn ein Ergebnis gekürzt wird, wird ein Warnsymbol angezeigt. Die Beschreibung der Warnung wird beim Überfahren mit der Maus angezeigt. Beachten Sie, dass Daten größer als 512 KB von Zabbix Server dennoch vollständig verarbeitet werden. Wenn der Schritt beim Testen fehlschlägt, wird ein Fehlersymbol angezeigt. Die Fehlerbeschreibung wird beim Überfahren mit der Maus angezeigt. Falls für den Schritt „Benutzerdefiniert bei Fehler“ angegeben ist und diese Aktion ausgeführt wird, erscheint direkt nach der Zeile des Vorverarbeitungstestschritts eine neue Zeile, die anzeigt, welche Aktion ausgeführt wurde und welches Ergebnis sie erzeugt hat (Fehler oder Wert).

Parameter	Beschreibung
Ergebnis	Das Endergebnis des Testens der Vorverarbeitungsschritte wird in allen Fällen angezeigt, wenn alle Schritte zusammen getestet werden (wenn Sie auf die Schaltfläche *Alle Schritte testen* klicken). Auch die Art der Konvertierung in den Werttyp des Datenpunkts wird angezeigt, zum Beispiel `Ergebnis konvertiert zu Numerisch (vorzeichenlos)`. Testergebnisse werden beim Senden an das Frontend auf eine maximale Größe von 512 KB gekürzt. Das Endergebnis kann kopiert werden (nicht mehr als die gekürzten 512 KB). Wenn ein Ergebnis gekürzt wird, wird ein Warnsymbol angezeigt. Die Beschreibung der Warnung wird beim Überfahren mit der Maus angezeigt. Beachten Sie, dass Daten größer als 512 KB von Zabbix Server dennoch 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 Konfiguration des Datenpunkts ändern und dann zum Testfenster zurückkehren kann, ohne die Informationen erneut eingeben zu müssen. Bei einer Seitenaktualisierung gehen die Werte jedoch verloren.

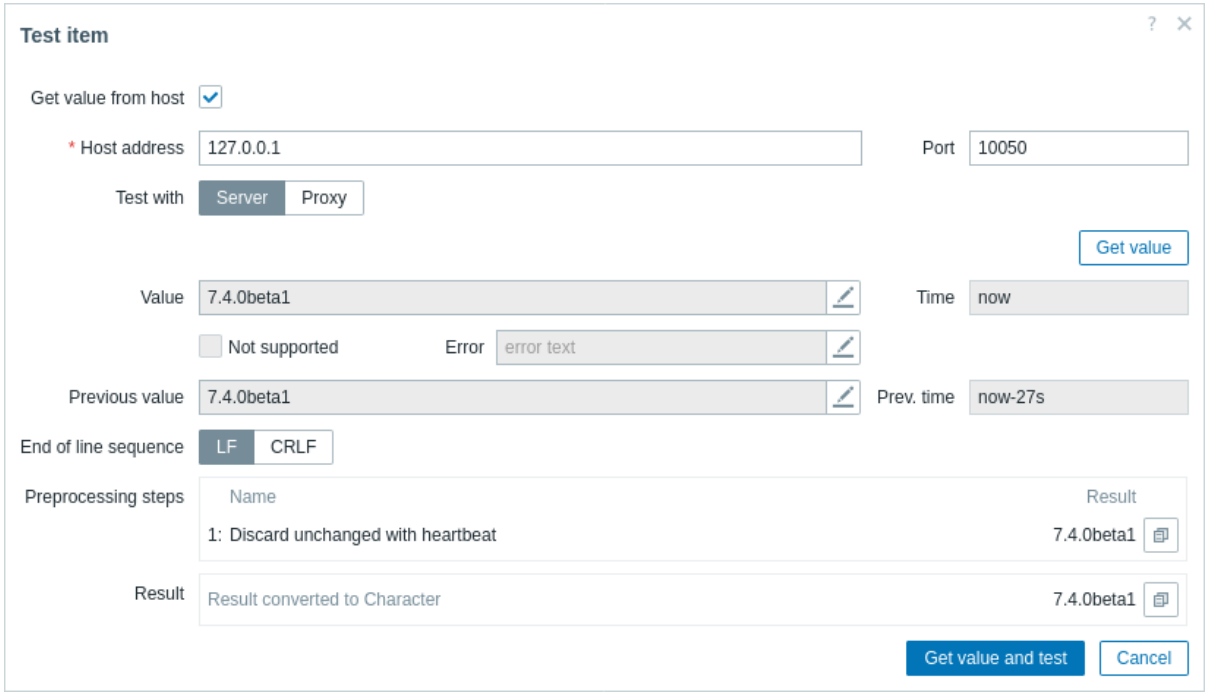
Das Testen wird von Zabbix Server durchgeführt. Das Frontend sendet eine entsprechende Anfrage an den Server und wartet auf das Ergebnis. Die Anfrage enthält den Eingabewert und die Vorverarbeitungsschritte (mit erweiterten Benutzermakros). Für die Schritte *Änderung* und *Drosselung* können optional ein vorheriger Wert und eine vorherige Zeit angegeben werden. Der Server antwortet mit Ergebnissen für jeden Vorverarbeitungsschritt.

Alle technischen Fehler oder Fehler bei der Eingabevalidierung werden im Fehlerfeld oben im Testfenster angezeigt.

Testen eines echten Werts

So testen Sie die Vorverarbeitung mit einem echten Wert:

- Aktivieren Sie das Kontrollkästchen *Wert vom Host abrufen*
- Geben Sie die Host-Parameter ein oder prüfen Sie sie (Host-Adresse, Port, Proxy-Name/kein Proxy) sowie Datenpunkt-spezifische Details (wie SNMPv2-Community oder SNMPv3- Sicherheitsanmeldedaten). Diese Felder sind kontextabhängig:
  - Die Werte werden nach Möglichkeit vorausgefüllt, d. h. bei Datenpunkten, die einen Agent erfordern, indem die Informationen aus der ausgewählten Agent-Schnittstelle des Hosts übernommen werden
  - Die Werte müssen bei Vorlagen-Datenpunkten manuell ausgefüllt werden
  - Makrowerte im Klartext werden aufgelöst
  - Wenn der Feldwert (oder ein Teil des Werts) ein geheimes Makro oder ein Vault-Makro ist, ist das Feld leer und muss manuell ausgefüllt werden. Wenn ein Datenpunkt-Parameter einen geheimen Makrowert enthält, wird die folgende Warnmeldung angezeigt: ”Datenpunkt enthält benutzerdefinierte Makros mit geheimen Werten. Werte dieser Makros sollten 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 im Konfigurationsformular des Datenpunkts eine Wertezuordnung angegeben haben (Feld „Wert anzeigen“), zeigt der Datenpunkt-Testdialog nach dem Endergebnis eine weitere Zeile mit dem Namen „Ergebnis mit angewendeter Wertezuordnung“ an.

Parameter, die speziell für das Abrufen eines echten Werts von einem Host gelten:

Parameter	Beschreibung
Wert vom Host abrufen	Aktivieren Sie dieses Kontrollkästchen, um einen echten 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.)	Siehe SNMP-Monitoring konfigurieren für weitere Details zur Konfiguration einer SNMP-Schnittstelle (v1, v2 und v3). Diese Felder werden automatisch aus der Host-Schnittstelle des Datenpunkts ausgefüllt.
Proxy	Geben Sie den Proxy an, wenn der Host über einen Proxy überwacht wird. Dieses Feld wird automatisch mit dem Proxy des Hosts ausgefüllt (falls vorhanden).
Wert	Vom Host abgerufener Wert. Ein Klick in das Parameterfeld oder auf die Anzeigen-/Bearbeiten-Schaltfläche öffnet ein Textbereichsfenster mit dem Wert oder Codeblock. Werte werden nur im Frontend auf eine maximale Größe von 512 KB gekürzt. Wenn ein Ergebnis gekürzt wird, wird ein Warnsymbol angezeigt. Die Warnungsbeschreibung wird beim Überfahren mit der Maus angezeigt. Beachten Sie, dass Daten größer als 512 KB von Zabbix Server dennoch vollständig verarbeitet werden.

Zu den übrigen Parametern siehe Testen eines hypothetischen Werts oben.

2 Details zur Vorverarbeitung

Übersicht

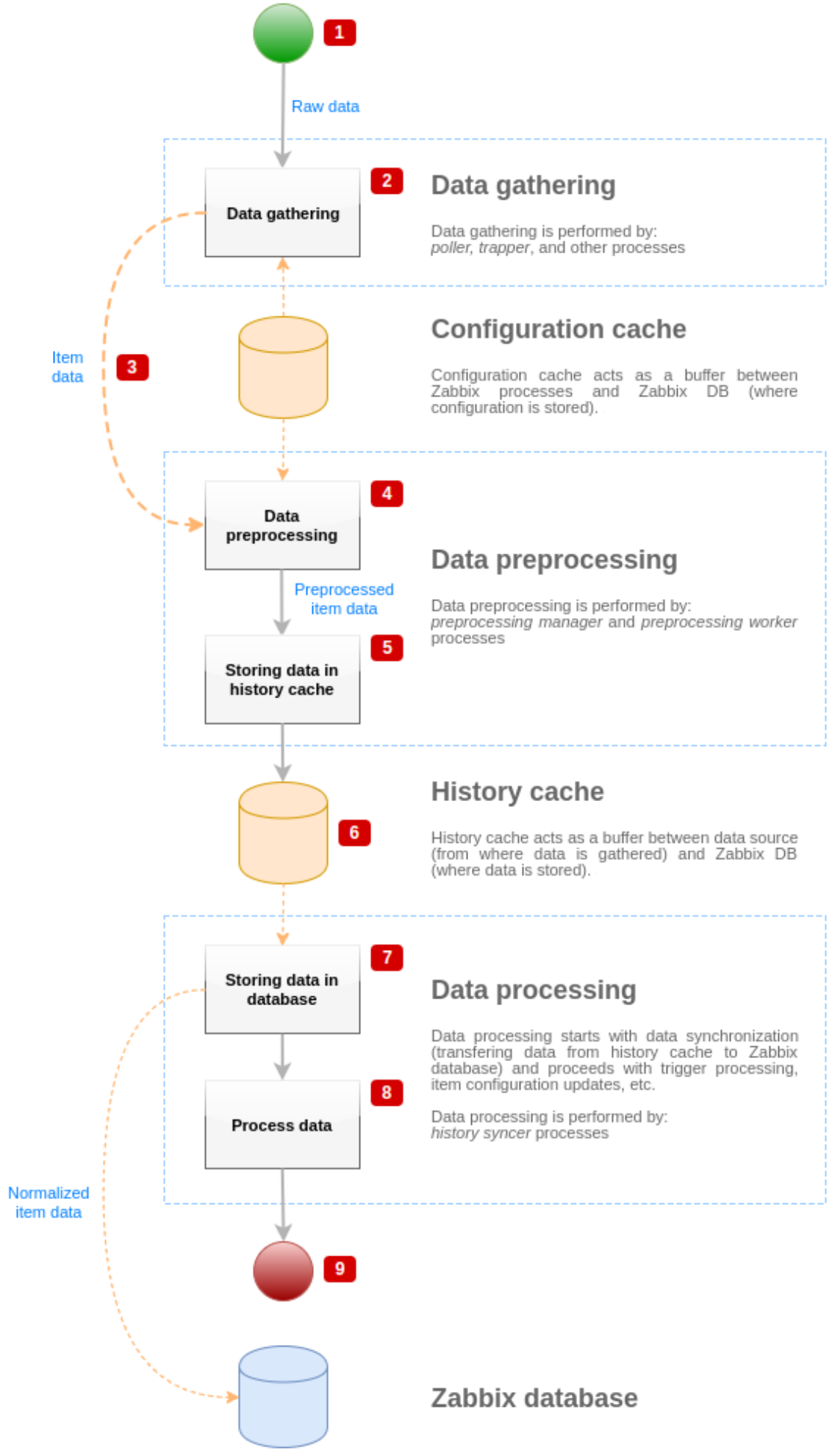
Dieser Abschnitt enthält Details zur Vorverarbeitung von Datenpunktwerten. Die Vorverarbeitung von Datenpunktwerten ermöglicht es, Transformationsregeln für die empfangenen Datenpunktwerte zu definieren und auszuführen.

Die Vorverarbeitung wird vom Prozess „preprocessing manager“ zusammen mit den „preprocessing workers“ verwaltet, 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. 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.

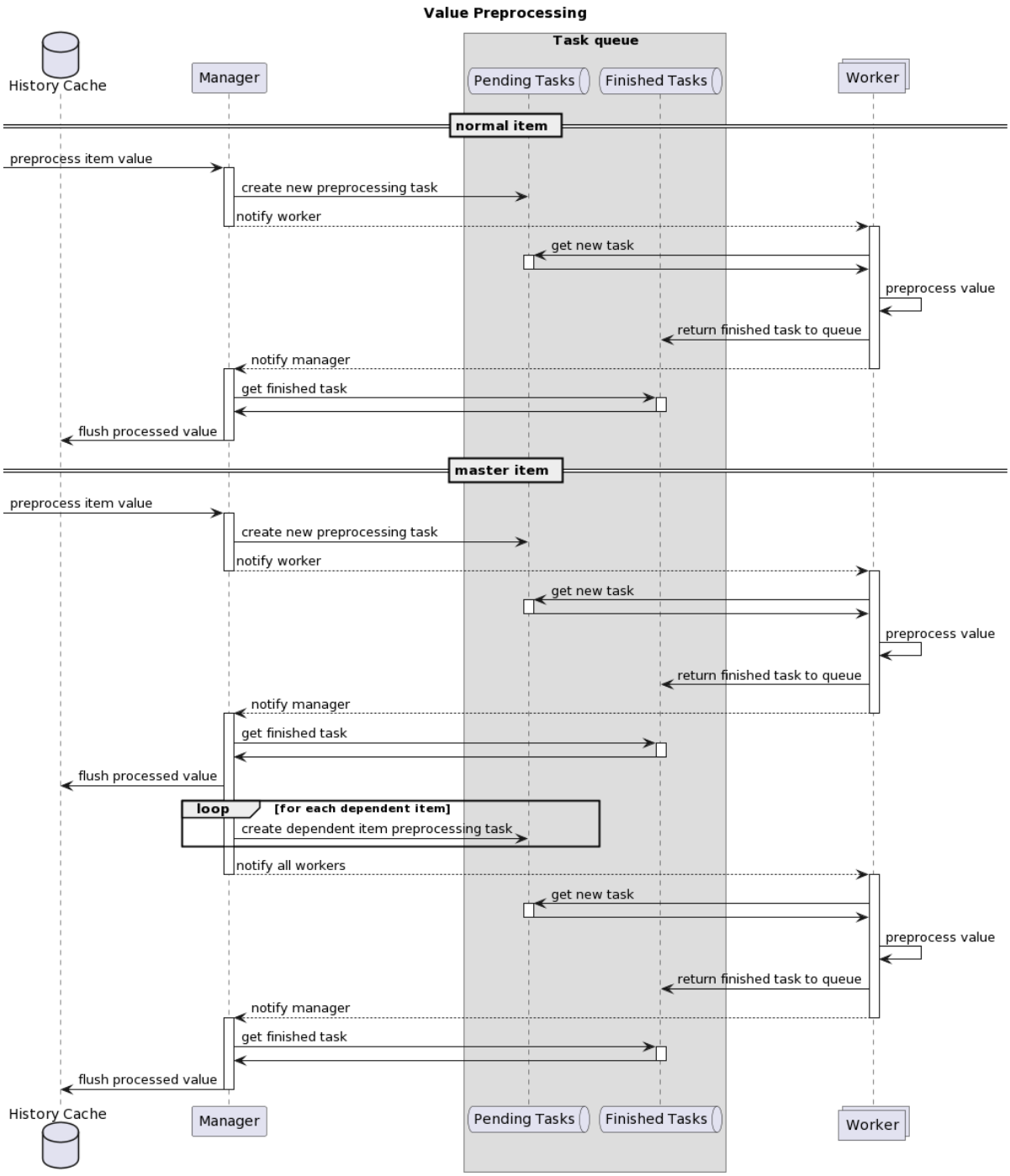
Note:
Ein Datenpunkt kann in den Zustand NOT SUPPORTED wechseln, wenn die Datennormalisierung fehlschlägt (zum Beispiel wenn ein Textwert nicht in eine Zahl konvertiert werden kann).

- 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-Server 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; diese sollte mindestens der Anzahl der verfügbaren CPU-Kerne entsprechen.

Wenn Preprocessing-Aufgaben nicht CPU-gebunden sind und häufige Netzwerkanfragen beinhalten, wird empfohlen, zusätzliche Worker zu konfigurieren. Die optimale Anzahl von Preprocessing-Workern kann von vielen Faktoren abhängen, darunter die Anzahl der „vorverarbeitbaren“ Datenpunkte (Datenpunkte, für die Preprocessing-Schritte ausgeführt werden müssen), die Anzahl der Prozesse zur Datenerfassung, die durchschnittliche Anzahl der Schritte bei der Datenpunkt-Vorverarbeitung usw. Eine unzureichende Anzahl an Workern kann zu hoher Speicherauslastung führen. Informationen zur Fehlerbehebung bei übermäßiger Speichernutzung in Ihrer Zabbix-Installation finden Sie unter Profiling excessive memory usage with tcmalloc.

Unter der Annahme, dass keine aufwendigen Preprocessing-Operationen wie das Parsen großer XML-/JSON-Blöcke stattfinden, kann die Anzahl der Preprocessing-Worker jedoch der Gesamtzahl der Datensammler entsprechen. Auf diese Weise steht für erfasste Daten in den meisten Fällen (außer wenn Daten vom Sammler gesammelt gebündelt eintreffen) mindestens ein freier Preprocessing-Worker zur Verfügung.

Warning:
Zu viele Prozesse zur Datenerfassung (Poller, Poller für nicht erreichbare Hosts, ODBC-Poller, HTTP-Poller, Java-Poller, Pinger, Trapper, Proxy-Poller) zusammen mit IPMI-Manager, SNMP-Trapper und Preprocessing-Workern können das Dateideskriptor-Limit pro Prozess für den Preprocessing-Manager ausschöpfen.

 Wenn das Dateideskriptor-Limit pro Prozess ausgeschöpft wird, führt dies dazu, dass der Zabbix-Server stoppt, typischerweise kurz nach dem Start, manchmal aber auch erst später. Um solche Probleme zu vermeiden, prüfen Sie die Zabbix-Server-Konfigurationsdatei, um die Anzahl gleichzeitiger Prüfungen und Prozesse zu optimieren. Stellen Sie außerdem bei Bedarf sicher, dass das Dateideskriptor-Limit 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.

3 Beispiele für die Vorverarbeitung

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.

4 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 Infixnotation.

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 $>`	Wert, auf den über den JSONPath vom Wurzelknoten des Eingabedokuments verwiesen wird; nur eindeutige Pfade werden unterstützt. Beispiel: `$.object.name`
`<jsonpath starting with @>`	Wert, auf den über den JSONPath vom aktuellen Objekt/Element verwiesen wird; nur eindeutige Pfade werden unterstützt. Beispiel: `@.name`

Unterstützte Operatoren:

Operator	Typ	Beschreibung	Ergebnis
–	Binär	Subtraktion	Zahl
+	Binär	Addition	Zahl
/	Binär	Division	Zahl

Operator	Typ	Beschreibung	Ergebnis
*	Binär	Multiplikation	Zahl
==	Binär	Gleichheit	Boolesch (1/0)
!=	Binär	Ungleichheit	Boolesch (1/0)
	Binär	Kleiner als	Boolesch (1/0)
<=	Binär	Kleiner oder gleich	Boolesch (1/0)
>	Binär	Größer als	Boolesch (1/0)
>=	Binär	Größer oder gleich	Boolesch (1/0)
=~	Binär	Entspricht regulärem Ausdruck	Boolesch (1/0)
!	Unär	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": "Referenz",
      "author": "Nigel Rees",
      "title": "Sprüche des Jahrhunderts",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "Belletristik",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "Belletristik",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "Belletristik",
      "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 Stadt",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Drucken und Zusammenstellen 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": "Belletristik",
    "no filters": "keine \"Filter\""
  },
  "closed message": "Geschäft ist geschlossen",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	Belletristik
`$.filters['no filters']`	definite	keine ”Filter”
`$.filters`	definite	{ ”price”: 10, ”category”: ”Belletristik”, ”no filters”: ”keine \”Filter\”” }
`$.books[1].title`	definite	Sword of Honour
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sprüche des Jahrhunderts”]
`$.books[1]['author', "title"]`	indefinite	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] Hinweis: Diese Abfrage zeigt, dass arithmetische Operationen in Abfragen verwendet werden können; sie kann zu `$.books[?(@.id == 2)].title` vereinfacht werden.
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”Der Herr der Ringe”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sprüche des Jahrhunderts”, ”Moby Dick”, ”Der Herr der Ringe”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sprüche des Jahrhunderts”, ”Moby Dick”, ”Der Herr der Ringe”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sprüche des Jahrhunderts”, ”Sword of Honour”, ”Der Herr der Ringe”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”Der Herr der Ringe”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sprüche des Jahrhunderts”, ”Der Herr der Ringe”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”Der Herr der Ringe”]

JSONPath	Type	Result
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”Der Herr der Ringe”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”Referenz”,
		”author”: ”Nigel Rees”,
		”title”: ”Sprüche des Jahrhunderts”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”Belletristik”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”Belletristik”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”Belletristik”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”Der Herr der Ringe”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Drucken und Zusammenstellen eines Buches im A5-Format”, ”Neubinden eines zerrissenen Buches”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour
`$..tags.first().length()`	definite	5 Hinweis: `$..tags` ist ein unbestimmter Pfad, daher gibt er ein Array der übereinstimmenden Elemente zurück, d. h. `[["a", "b", "c", "d", "e" ]]`; `first()` gibt das erste Element zurück, d. h. `["a", "b", "c", "d", "e"]`; `length()` berechnet die Länge des Elements, d. h. 5.
`$.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")].price`	indefinite	[1001,1000] Hinweis: Bei Vergleichen mit booleschen Werten müssen Textkonstanten verwendet werden.

JSONPath	Type	Result
`$.services[?(@.active=="false")].name`	Undefined	[1002]
		Hinweis: Bei Vergleichen mit booleschen Werten müssen Textkonstanten verwendet werden.
`$.services[?(@.servicegroup=="demo")].name`	Text	[1002]

1 Escaping von Sonderzeichen aus LLD-Makrowerten in JSONPath

Wenn Low-Level-Discovery-Makros in der JSONPath-Vorverarbeitung verwendet werden und ihre Werte aufgelöst werden, gelten die folgenden Regeln für das Escaping von Sonderzeichen:

- nur Backslash-Zeichen (\) und doppelte Anführungszeichen (”) werden beim Escaping berücksichtigt;
- wenn der aufgelöste Makrowert diese Zeichen enthält, wird jedes von ihnen mit einem Backslash escaped;
- wenn sie bereits mit einem Backslash escaped sind, wird dies nicht als Escaping betrachtet und sowohl der Backslash als auch die folgenden Sonderzeichen werden erneut escaped.

Zum Beispiel:

JSONPath	LLD-Makrowert	Nach der Ersetzung
$.[?(@.value == ”{#MACRO}”)]	special ”value”	$.[?(@.value == ”special \”value\””)]
	c:\temp	$.[?(@.value == ”c:\\temp”)]
	a\\b	$.[?(@.value == ”a\\\\b”)]

Bei Verwendung im Ausdruck sollte das Makro, das Sonderzeichen enthalten kann, in doppelte Anführungszeichen eingeschlossen werden:

JSONPath	LLD-Makrowert	Nach der Ersetzung	Ergebnis
$.[?(@.value == ”{#MACRO}”)]	special ”value”	$.[?(@.value == ”special \”value\””)]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \”value\”)]	**Ungültiger JSONPath-Ausdruck**

Bei Verwendung im Pfad sollte das Makro, das Sonderzeichen enthalten kann, in eckige Klammern **und** doppelte Anführungszeichen eingeschlossen werden:

JSONPath	LLD-Makrowert	Nach der Ersetzung	Ergebnis
$.[”{#MACRO}”].value	c:\temp	$.[”c:\\temp”].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Ungültiger JSONPath-Ausdruck**

5 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-Preprocessing

Die JavaScript-Preprocessing wird durchgeführt, indem eine JavaScript-Funktion mit einem einzelnen Parameter ‘value’ und einem vom Benutzer bereitgestellten Funktionsrumpf aufgerufen wird. Das Ergebnis des Preprocessing-Schritts 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 Parametern der JavaScript-Preprocessing; dies wird vom Server in eine JavaScript-Funktion eingebettet:

```
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 führt dazu, dass der Eingabewert verworfen wird, ähnlich wie beim Preprocessing „Wert verwerfen" in der Aktion „Benutzerdefiniert bei Fehler".

Fehler können durch das Auslösen von Werten/Objekten zurückgegeben werden (normalerweise entweder Zeichenfolgen oder Error-Objekte).

Zum Beispiel:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Jedes Skript hat ein Ausführungs-Timeout von 10 Sekunden (abhängig vom 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-Preprocessing-Schritts wird zwischengespeichert und bei der nächsten Anwendung des Schritts wiederverwendet. Änderungen an den Preprocessing-Schritten des Datenpunkts führen dazu, dass das zwischengespeicherte Skript zurückgesetzt und später neu kompiliert wird.

Aufeinanderfolgende Laufzeitfehler (3 hintereinander) führen dazu, dass die Engine neu initialisiert wird, um die Möglichkeit zu verringern, dass ein Skript die Ausführungsumgebung für die nachfolgenden Skripte beeinträchtigt (diese Aktion wird mit DebugLevel 4 und höher protokolliert).

Die JavaScript-Preprocessing ist mit der JavaScript-Engine Duktape implementiert.

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 umwandeln

Wandeln Sie den von einem Zabbix-Agent-Datenpunkt vom Typ web.page.get empfangenen Wert (z. B. web.page.get[http://127.0.0.1:80/serve
status?auto]) in ein JSON-Objekt um.

```
// Apache-Status in JSON umwandeln

// 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;

// 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
    }
}
```


1 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, mit dem sich einfache HTTP-Anfragen ausführen lassen. Fehler werden als Ausnahmen ausgelöst.

Attention:
Die Initialisierung mehrerer `HttpRequest`-Objekte ist auf 10 pro Skriptausführung begrenzt.

Method	Description
`addHeader(value)`	Fügt ein HTTP-Header-Feld hinzu. Dieses Feld wird für alle folgenden Anfragen verwendet, bis es mit der Methode `clearHeader()` gelöscht wird. Die Gesamtlänge der Header-Felder, die zu einem einzelnen `HttpRequest`-Objekt hinzugefügt werden können, ist auf 128 KByte begrenzt (einschließlich Sonderzeichen und Header-Namen).
`clearHeader()`	Löscht den HTTP-Header. Wenn keine Header-Felder gesetzt sind, setzt `HttpRequest` Content-Type auf application/json, wenn die gesendeten 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 an die URL mit optionaler *data*-Nutzlast und gibt die Antwort zurück.
`delete(url, data)`	Sendet eine HTTP-DELETE-Anfrage an die URL mit optionaler *data*-Nutzlast und gibt die Antwort zurück.
`getHeaders(<asArray>)`	Gibt das Objekt der empfangenen HTTP-Header-Felder zurück. Der Parameter `asArray` kann auf "true" (z. B. `getHeaders(true)`), "false" gesetzt oder undefiniert sein. Wenn er auf "true" gesetzt ist, werden die Werte der empfangenen HTTP-Header-Felder als Arrays zurückgegeben; dies sollte verwendet werden, um die Feldwerte mehrerer Header mit demselben Namen abzurufen. Wenn er nicht gesetzt ist oder auf "false" gesetzt ist, werden die Werte der empfangenen HTTP-Header-Felder als Zeichenfolgen zurückgegeben.
`get(url, data)`	Sendet eine HTTP-GET-Anfrage an die URL mit optionaler *data*-Nutzlast 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 an die URL mit optionaler *data*-Nutzlast und gibt die Antwort zurück.
`put(url, data)`	Sendet eine HTTP-PUT-Anfrage an die URL mit optionaler *data*-Nutzlast und gibt die Antwort zurück.
`post(url, data)`	Sendet eine HTTP-POST-Anfrage an die URL mit optionaler *data*-Nutzlast 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)`	Setzt die aktivierten HTTP-Authentifizierungsmethoden (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) im Parameter 'bitmask'.
	Das Flag HTTPAUTH_NONE ermöglicht das Deaktivieren der HTTP-Authentifizierung.
	Beispiele:
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	Sendet eine HTTP-TRACE-Anfrage an die URL mit optionaler *data*-Nutzlast 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
<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
<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
<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
<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').

2 JavaScript-Objekte für Browser-Datenpunkte

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.
`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).

Method	Description
`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).
`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 auf der Webseite und stellt Methoden zur Interaktion damit bereit.

Die folgenden Methoden werden vom `Element`-Objekt unterstützt.

Method	Description
`getAttribute(name)`	Gibt einen Attributwert des Elementattributs als Zeichenfolge zurück (oder `null`, wenn das angegebene Attribut nicht gefunden wurde). Parameter: `name` - (string) Attributname.
`getProperty(name)`	Gibt einen Eigenschaftswert der Elementeigenschaft als Zeichenfolge zurück (oder `null`, wenn die angegebene Eigenschaft nicht gefunden wurde). Parameter: `name` - (string) Eigenschaftsname.
`getText()`	Gibt einen Textwert des Elementtexts als Zeichenfolge zurück.
`click()`	Klickt auf ein Element.
`clear()`	Löscht den Inhalt eines bearbeitbaren Elements.
`sendKeys(keys)`	Sendet Tastenanschläge. Parameter: `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.
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.

Eigenschaft		Typ	Beschreibung
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.
	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.

6 CSV-zu-JSON-Vorverarbeitung

Übersicht

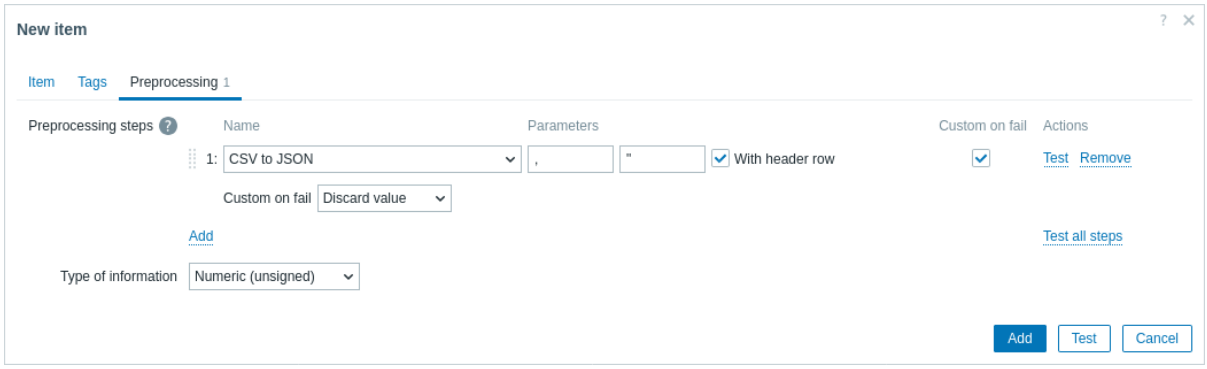
In diesem Vorverarbeitungsschritt ist es möglich, CSV-Dateidaten in das JSON-Format zu konvertieren. Dies wird unterstützt in:

- Datenpunkten (Datenpunkt-Prototypen)
- Low-Level-Discovery-Regeln

Konfiguration

So konfigurieren Sie einen Vorverarbeitungsschritt für CSV zu JSON:

- Gehen Sie zur Registerkarte Vorverarbeitung in der Konfiguration von Datenpunkt/Discovery-Regel
- Klicken Sie auf *Hinzufügen*
- Wählen Sie die Option *CSV zu JSON*



Mit dem ersten Parameter kann ein benutzerdefiniertes Trennzeichen festgelegt werden. Beachten Sie, dass, wenn die erste Zeile der CSV-Eingabe mit „Sep=“ beginnt und darauf ein einzelnes UTF-8-Zeichen folgt, dieses Zeichen als Trennzeichen verwendet wird, falls der erste Parameter nicht gesetzt ist. Wenn der erste Parameter nicht gesetzt ist und kein Trennzeichen aus der Zeile „Sep=“ ermittelt wird, wird ein Komma als Trennzeichen verwendet.

Mit dem zweiten optionalen Parameter kann ein Anführungszeichen festgelegt werden.

Wenn das Kontrollkästchen *Mit Kopfzeile* aktiviert ist, werden die Werte der Kopfzeile als Spaltennamen interpretiert (weitere Informationen finden Sie unter Verarbeitung der Kopfzeile).

Wenn das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktiviert ist, wird der Datenpunkt im Falle eines fehlgeschlagenen Vorverarbeitungsschritts nicht in den Status „nicht unterstützt“ versetzt. Zusätzlich können benutzerdefinierte Optionen für die Fehlerbehandlung festgelegt werden: den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.

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"
   }
]
```


3 Datenpunkttypen

Übersicht

Datenpunkt-Typen umfassen verschiedene Methoden zur Erfassung von Daten aus Ihrem System. Jeder Datenpunkt-Typ verfügt über einen eigenen Satz unterstützter Datenpunkt-Schlüssel und erforderlicher Parameter.

Die folgenden Datenpunkt-Typen werden derzeit von Zabbix angeboten:

- Zabbix-Agent-Prüfungen
- SNMP-Agent-Prüfungen
- SNMP-Traps
- IPMI-Prüfungen
- Einfache Prüfungen
  - VMware-Überwachung
- Logdatei-Überwachung
- Berechnete Datenpunkte
  - Aggregatberechnungen
- Interne Zabbix-Prüfungen
- SSH-Prüfungen
- Telnet-Prüfungen
- Externe Prüfungen
- Trapper-Datenpunkte
- JMX-Überwachung
- ODBC-Prüfungen
- Abhängige Datenpunkte
- HTTP-Prüfungen
- Prometheus-Prüfungen
- Skript-Datenpunkte
- Browser-Datenpunkte

Details zu allen Datenpunkt-Typen finden Sie auf den Unterseiten dieses Abschnitts. Auch wenn Datenpunkt-Typen viele Möglichkeiten zur Datenerfassung bieten, gibt es weitere Optionen über Benutzerparameter oder ladbare Module.

Einige Prüfungen werden ausschließlich vom Zabbix Server ausgeführt (als agentenloses Monitoring), während andere den Zabbix Agent oder sogar das Zabbix Java gateway erfordern (bei der JMX-Überwachung).

Attention:
Wenn ein bestimmter Datenpunkt-Typ eine bestimmte Schnittstelle erfordert (z. B. benötigt eine IPMI-Prüfung eine IPMI-Schnittstelle auf dem Host), muss diese Schnittstelle in der Host-Definition vorhanden sein.

In der Host-Definition 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 Host-Schnittstellen (in der Reihenfolge: Agent→SNMP→JMX→IPMI) nach der ersten geeigneten, mit der er verknüpft werden kann.

Alle Datenpunkte, die Text zurückgeben (Zeichen-, Log- und Text-Informationstypen), können auch ausschließlich Leerraum zurückgeben (wo zutreffend), wobei der Rückgabewert auf eine leere Zeichenfolge gesetzt wird (unterstützt seit 2.0).

1 Zabbix Agent

Übersicht

Dieser Abschnitt enthält Details zu den Datenpunktschlüsseln, die die Kommunikation mit dem Zabbix Agent zur Datenerfassung verwenden.

Es gibt passive und aktive Agent-Prüfungen. Bei der Konfiguration eines Datenpunkts können Sie den erforderlichen Typ auswählen:

- *Zabbix Agent* - für passive Prüfungen
- *Zabbix Agent (active)* - für aktive Prüfungen

Beachten Sie, dass alle vom Zabbix Agent unterstützten Datenpunktschlüssel auch vom Zabbix Agent 2 der neuen Generation unterstützt werden. Siehe die zusätzlichen Datenpunktschlüssel, 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 sind ohne Parameter und zusätzliche Informationen aufgeführt. Klicken Sie auf einen 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 offenen Dateideskriptoren.	
log	Die Überwachung einer Protokolldatei.	Protokollüberwachung
log.count	Die Anzahl übereinstimmender Zeilen in einer überwachten Protokolldatei.	
logrt	Die Überwachung einer rotierten Protokolldatei.	
logrt.count	Die Anzahl übereinstimmender Zeilen in einer überwachten rotierten Protokolldatei.	
modbus.get	Liest Modbus-Daten.	Modbus
net.dns	Prüft den Status eines DNS-Dienstes.	Netzwerk
net.dns.perf	Prüft die Leistung eines DNS-Dienstes.	
net.dns.record	Führt eine DNS-Abfrage aus.	
net.if.collisions	Die Anzahl der Kollisionen außerhalb des Fensters.	
net.if.discovery	Die Liste der Netzwerkschnittstellen.	
net.if.in	Die Statistiken des eingehenden Datenverkehrs auf einer Netzwerkschnittstelle.	
net.if.out	Die Statistiken des ausgehenden Datenverkehrs auf einer Netzwerkschnittstelle.	
net.if.total	Die Summe der Statistiken des eingehenden und ausgehenden Datenverkehrs auf einer Netzwerkschnittstelle.	
net.tcp.listen	Prüft, ob sich dieser TCP-Port im Status LISTEN befindet.	
net.tcp.port	Prüft, ob es möglich ist, eine TCP-Verbindung zum angegebenen Port herzustellen.	
net.tcp.service	Prüft, ob ein Dienst läuft und TCP-Verbindungen akzeptiert.	
net.tcp.service.perf	Prüft die Leistung eines TCP-Dienstes.	
net.tcp.socket.count	Gibt die Anzahl der TCP-Sockets zurück, die den Parametern entsprechen.	
net.udp.listen	Prüft, ob sich dieser UDP-Port im Status LISTEN befindet.	
net.udp.service	Prüft, ob ein Dienst läuft 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.	
proc.cpu.util	Der prozentuale CPU-Auslastungswert 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	Auslesen von Hardware-Sensoren.	Sensoren
system.boottime	Die Systemstartzeit.	System
system.cpu.discovery	Die Liste der erkannten CPUs/CPU-Kerne.	
system.cpu.intr	Die Geräte-Interrupts.	
system.cpu.load	Die CPU-Last.	
system.cpu.num	Die Anzahl der CPUs.	
system.cpu.switches	Die Anzahl der Kontextwechsel.	
system.cpu.util	Der prozentuale CPU-Auslastungswert.	
system.hostname	Der System-Hostname.	
system.hw.chassis	Die Chassis-Informationen.	
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 Swap-in-Statistiken (vom Gerät in den Speicher).	
system.swap.out	Die Swap-out-Statistiken (aus dem Speicher auf das Gerät).	
system.swap.size	Die Größe des Swap-Speichers in Byte oder als Prozentsatz der Gesamtkapazität.	
system.uname	Identifikation des Systems.	
system.uptime	Die Systemlaufzeit in Sekunden.	
system.users.num	Die Anzahl der angemeldeten Benutzer.	
vfs.dev.discovery	Die Liste der Blockgeräte und ihres Typs.	Virtuelle Dateisysteme

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
vfs.dev.read	Die Statistiken zu Lesevorgängen auf Datenträgern.	
vfs.dev.write	Die Statistiken zu Schreibvorgängen auf Datenträgern.	
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-Algorithmus cksum.	
vfs.file.contents	Abrufen des Inhalts einer Datei.	
vfs.file.exists	Prüft, ob die Datei existiert.	
vfs.file.get	Gibt Informationen über eine Datei zurück.	
vfs.file.md5sum	Die MD5-Prüfsumme der Datei.	
vfs.file.owner	Ruft den Eigentümer einer Datei ab.	
vfs.file.permissions	Gibt eine 4-stellige Zeichenfolge zurück, die die Oktalzahl mit den UNIX-Berechtigungen enthält.	
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 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, verfügbarem Speicherplatz, Inode-Statistiken und Einhängeoptionen.	
vfs.fs.inode	Die Anzahl oder der Prozentsatz der Inodes.	
vfs.fs.size	Der Speicherplatz in Byte oder als Prozentsatz der Gesamtkapazität.	
vm.memory.size	Die Speichergröße in Byte oder als Prozentsatz der Gesamtkapazität.	Virtueller Speicher
web.page.get	Ruft den Inhalt einer Webseite ab.	Webüberwachung
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 Host-Name 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 des Zabbix Server oder Proxy zurück. Wenn `ip` und `port` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen, andernfalls von der lokalen Instanz.	
zabbix.stats	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.	

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 Kennung der Codepage;
- **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) oder *skip* - Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus);
- **output** - eine optionale Vorlage zur 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 sie endet), während eine Escape-Sequenz **\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 neuesten Zeilen innerhalb von "maxdelay" Sekunden zu analysieren. Lesen Sie vor der Verwendung die Hinweise zu maxdelay!
- **options** - zusätzliche Optionen:
mtime-noreread - nicht eindeutige Einträge nur dann erneut lesen, wenn sich die Dateigröße ändert (Änderungen 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 aktive Prüfung konfiguriert sein.
- Wenn die Datei fehlt oder Berechtigungen keinen Zugriff erlauben, wird der Datenpunkt nicht unterstützt.
- Wenn `output` leer bleibt, wird die gesamte Zeile zurückgegeben, die den übereinstimmenden Text enthält. Beachten Sie, dass alle globalen Typen regulärer Ausdrücke außer 'Result is TRUE' immer die gesamte übereinstimmende Zeile zurückgeben und der Parameter `output` ignoriert wird.
- Die Inhaltsextraktion mit dem Parameter `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`, um eine Zahl aus einem Protokolleintrag zu extrahieren:

```
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`, um einen Protokolleintrag vor dem Senden an den Server umzuschreiben:

```
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 Name 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) oder *skip* - Verarbeitung älterer Daten überspringen (betrifft nur neu erstellte Datenpunkte).
- **maxdelay** - die maximale Verzögerung in Sekunden. Typ: float. Werte: 0 - (Standard) Zeilen der Protokolldatei niemals ignorieren; > 0.0 - ältere Zeilen ignorieren, damit die neuesten Zeilen innerhalb von "maxdelay" Sekunden analysiert werden. 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 (Änderungen 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 Pfadname des Verzeichnisses, in dem persistente Dateien gespeichert werden. Siehe auch die zusätzlichen 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 keinen Zugriff erlauben, wird der Datenpunkt nicht unterstützt.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Die Überwachung einer rotierten Protokolldatei.
 Rückgabewert: *Log*.
 Siehe unterstützte Plattformen.

Parameter:

- **file regexp** - der absolute Pfad zur Datei, wobei der Dateiname mit einem regulären Ausdruck angegeben wird. Beachten Sie, dass der reguläre Ausdruck nur auf den Dateinamen angewendet wird und nicht mit dem gesamten Namen übereinstimmen muss (z. B. passt /path/to/agent auf zabbix_agentd.log).
- **regexp** - ein regulärer Ausdruck, der das erforderliche Inhaltsmuster beschreibt.
- **encoding** - die Codepage-Kennung.
- **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) oder *skip* - Verarbeitung älterer Daten überspringen (betrifft nur neu erstellte Datenpunkte).
- **output** - eine optionale Vorlage zur 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 sie endet), während eine Escape-Sequenz **\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) Protokolldateizeilen niemals ignorieren; > 0.0 - ältere Zeilen ignorieren, um die neuesten Zeilen innerhalb von "maxdelay" Sekunden zu analysieren. Lesen Sie vor der Verwendung die Hinweise zu maxdelay!
- **options** - der Typ der Protokolldateirotation 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 ändern (wird standardmäßig verwendet, wenn keine `mtime-*`-Option explizit gesetzt ist),
mtime-noreread - nicht eindeutige Datensätze, nur erneut lesen, 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 Verzeichnispfad, in dem persistente Dateien gespeichert werden. Siehe auch zusätzliche Hinweise zu persistenten Dateien.

Kommentare:

- Der Datenpunkt muss als aktive Prüfung konfiguriert sein.
- Die Protokolldateirotation basiert auf der letzten Änderungszeit der Dateien.
- Beachten Sie, dass logrt für die Arbeit mit einer aktuell aktiven Protokolldatei ausgelegt ist, während mehrere andere passende inaktive Dateien rotiert sind. Wenn ein Verzeichnis beispielsweise viele aktive Protokolldateien enthält, sollte für jede eine separate logrt-Datenpunkt erstellt werden. Andernfalls kann es, wenn ein logrt-Datenpunkt zu viele Dateien erfasst, zu Speicherausschöpfung 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 Typen regulärer Ausdrücke außer 'Result is TRUE' immer die gesamte übereinstimmende Zeile zurückgeben und der Parameter `output` ignoriert wird.
- Die Inhaltsextraktion mit dem Parameter `output` erfolgt auf dem Agent.
- Im Parameter `file regexp` müssen der Pfad des Protokollverzeichnisses 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 Positionen toleriert werden, außer an der Stelle, die den Pfad des Protokollverzeichnisses 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 sammelt Daten aus Protoko
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #dieser Datenpunkt sammelt
```

Beispiele für andere Systeme:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #dieser Datenpunkt passt auf eine Datei wie "logfile1
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #dieser Datenpunkt sammelt Date
```

Beispiel für die Verwendung des Parameters `output`, um eine Zahl aus einem Protokolldatensatz zu extrahieren:

```
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 Protokolldatensatz 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 Logdatei mit Rotation.
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Parameter:

- **file regexp** - der absolute Pfad zur Datei, wobei der Dateiname mit einem regulären Ausdruck angegeben wird. Beachten Sie, dass der reguläre Ausdruck nur auf den Dateinamen angewendet wird und nicht mit dem gesamten Namen übereinstimmen muss (z. B. passt /path/to/agent auf zabbix_agentd.log).
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt.
- **encoding** - die Kennung der Codepage.
- **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) oder *skip* - 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 Logdatei niemals ignorieren; > 0.0 - ältere Zeilen ignorieren, damit die neuesten Zeilen innerhalb von "maxdelay" Sekunden analysiert werden. Lesen Sie vor der Verwendung die Hinweise zu maxdelay!
- **options** - der Typ der Logdatei-Rotation 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 ändern (Standard),
mtime-noreread - nicht eindeutige Datensätze, nur erneut lesen, 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 aktive Prüfung konfiguriert sein.
- Übereinstimmende Zeilen werden in den neuen Zeilen seit der letzten Logprüfung durch den Agent gezählt und hängen daher vom Aktualisierungsintervall des Datenpunkts ab.
- Die Logrotation basiert auf der letzten Änderungszeit der Dateien.
- Im Parameter `file regexp` müssen der Pfad des Logverzeichnisses und der reguläre Ausdruck für die Logdatei durch das korrekte Verzeichnistrennzeichen getrennt werden:
  - Unter Windows muss das Trennzeichen ein Backslash (\) sein; Schrägstriche (/) werden an anderen Positionen möglicherweise toleriert, außer an der Stelle, die den Pfad des Logverzeichnisses und den regulären Ausdruck für die Logdatei 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"] #this item will count the number of ma
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will count
```

Beispiele für andere Systeme:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #this item will count the number of matches
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #this item will count the
```

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, der ersten Coil oder des ersten Eingangs;
- **count** - die Anzahl der zu lesenden Datensätze;
- **type** - der Datentyp;
- **endianness** - die Endianness-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, außer bei Verwendung von Zabbix Agent 2) - 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, außer bei Verwendung von Zabbix Agent 2) - das Timeout für die Anfrage in Sekunden (Standard ist 1 Sekunde);
- **count** (unter Windows ignoriert, außer bei Verwendung von Zabbix Agent 2) - die Anzahl der Versuche für die Anfrage (Standard ist 2);
- **protocol** - das für DNS-Abfragen verwendete Protokoll: *udp* (Standard) oder *tcp*.

Kommentare:

- Die möglichen 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 Anforderung eines PTR-Records der DNS-Name tatsächlich eine IP-Adresse ist.
- Internationalisierte Domainnamen werden nicht unterstützt; verwenden Sie stattdessen bitte 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 Performance eines DNS-Dienstes.
 Rückgabewert: *Float* (0 - Dienst ist nicht verfügbar; 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:

- Die möglichen 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 beim Anfordern eines PTR-Records der DNS-Name tatsächlich eine IP-Adresse ist.
- Internationalisierte Domainnamen werden nicht unterstützt; verwenden Sie stattdessen bitte IDNA-kodierte Namen.
- Der Datenpunkt gibt eine Antwortzeit anstelle von 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 aus.

Zabbix Agent 2 bietet außerdem net.dns.get, das zusätzliche Funktionen wie weitere Record-Typen und eine bessere Steuerung der DNS-Überwachung 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:

- Die möglichen 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 im umgekehrten oder nicht umgekehrten Format angeben (siehe Beispiele unten). Beachten Sie, dass beim Anfordern eines PTR-Records der DNS-Name tatsächlich eine IP-Adresse ist.
- Internationalisierte Domainnamen werden nicht unterstützt; 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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.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, die 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 zusammen mit dem Vorverarbeitungsschritt *Änderung pro Sekunde* verwenden, um die Statistik in Bytes pro Sekunde 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, die 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 zusammen mit dem Präprozessierungsschritt *Änderung pro Sekunde* verwenden, um die Statistik in 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 Statistiken des eingehenden und 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, die 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 vom Gerätetreiber übertragenen oder empfangenen komprimierten Pakete

Kommentare:

- Sie können diesen Schlüssel zusammen mit dem Vorverarbeitungsschritt *Änderung pro Sekunde* verwenden, um die Statistik in Bytes pro Sekunde zu erhalten.
- Der Modus *dropped* wird nur unter Linux und HP-UX unterstützt. Verworfene Pakete werden nur unterstützt, wenn auf Ihrer Plattform sowohl `net.if.in` als auch `net.if.out` 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 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, Windows.

Parameter:

- **port** - TCP-Portnummer

Unter Linux-Kernels 2.6.14 und höher werden Informationen über lauschende TCP-Sockets nach Möglichkeit über die NETLINK-Schnittstelle des Kernels abgerufen. 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 es möglich ist, eine TCP-Verbindung zum angegebenen Port herzustellen.
 Rückgabewerte: 0 - Verbindung nicht 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-Performance-Tests verwenden Sie `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Diese Prüfungen können zu zusätzlichen Meldungen in den Logdateien von System-Daemons führen (SMTP- und SSH-Sitzungen werden üblicherweise protokolliert).

Beispiel:

```
net.tcp.port[,80] #dieser Datenpunkt kann verwendet werden, um die Verfügbarkeit des Web-Servers auf Port
```

net.tcp.service[service,<ip>,<port>]

 Prüft, ob ein Dienst läuft und TCP-Verbindungen akzeptiert.
 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 (Standard ist 127.0.0.1);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Kommentare:

- Diese Prüfungen können zu zusätzlichen Meldungen in den Logdateien der System-Daemons führen (üblicherweise werden SMTP- und SSH-Sitzungen protokolliert).
- Die Prüfung verschlüsselter Protokolle (wie IMAP auf Port 993 oder POP auf Port 995) wird derzeit nicht unterstützt. Als Umgehungslösung 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 Login-Aufforderung (':' 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 Performance eines TCP-Dienstes.
 Rückgabewerte: *Float* (0 - 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 (Standard ist 127.0.0.1);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Kommentare:

- Die Prüfung verschlüsselter Protokolle (wie IMAP auf Port 993 oder POP auf Port 995) wird derzeit nicht unterstützt. Als Umgehungslösung 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 läuft und auf UDP-Anfragen antwortet.
 Rückgabewerte: 0 - Dienst ist nicht verfügbar; 1 - Dienst läuft.
 Siehe unterstützte Plattformen.

Parameter:

- **service** - *ntp* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (Standard ist 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 - Dienst ist nicht verfügbar; 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 (Standard ist 127.0.0.1);
- **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
```

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>]

 Die prozentuale 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** - nach Befehlszeile filtern (regulärer Ausdruck);
- **mode** - der Modus für die 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 der prozentualen Auslastung eines einzelnen CPU-Kerns. Beispielsweise beträgt die CPU-Auslastung eines Prozesses, der zwei Kerne vollständig nutzt, 200 %.
- Die Daten zur CPU-Auslastung des Prozesses werden von einem Kollektor 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 Kollektor entfernt.
- Wenn der Parameter `zone` auf *current* (oder den Standardwert) gesetzt ist und der Agent auf einem Solaris ohne Zonenunterstützung kompiliert wurde, aber auf einem neueren Solaris mit Zonenunterstützung läuft, gibt der Agent NOTSUPPORTED zurück (der Agent kann die Ergebnisse nicht nur auf die aktuelle Zone beschränken). *all* wird in diesem Fall jedoch unterstützt.

Beispiele:

```
proc.cpu.util[,root] #CPU-Auslastung aller Prozesse, die unter dem Benutzer "root" laufen
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 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** - nach Befehlszeile filtern (dies ist ein regulärer 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* (für NetBSD nicht unterstützt), *summary*. Siehe die Liste der Prozessparameter, die für jeden Modus und jedes Betriebssystem zurückgegeben werden.

Kommentare:

- Wenn ein Wert nicht abgerufen werden kann, zum Beispiel aufgrund eines Fehlers (Prozess bereits beendet, fehlende Berechtigungen, Fehler beim Systemaufruf), wird −1 zurückgegeben.
- Siehe die Hinweise zur Auswahl von Prozessen mit den Parametern `name` und `cmdline` (Linux-spezifisch).

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] #Kombinierte Daten für Prozesse jedes Typs, die unter dem Benutzer zabbix laufe
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 Der vom Prozess verwendete Speicher in Byte.
 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 (dies ist ein regulärer Ausdruck);
- **memtype** - der vom Prozess verwendete Speichertyp

Kommentare:

- Der Parameter `memtype` wird nur unter Linux, FreeBSD, Solaris[6] und AIX unterstützt.
- Wenn mehrere Prozesse gemeinsam genutzten Speicher verwenden, kann die Summe des von den Prozessen verwendeten Speichers zu großen, unrealistischen Werten führen.

Siehe die Hinweise zur Auswahl von Prozessen mit den Parametern `name` und `cmdline` (Linux-spezifisch).

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 Prozessen verwendete Speicher, die unter dem Benutzer "root" laufen
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 - laufend,
sleep - unterbrechbarer Schlaf,
trace - angehalten,
zomb - Zombie;
- **cmdline** - nach Befehlszeile filtern (dies ist ein regulärer Ausdruck);
- **zone** - die Zielzone: *current* (Standard) oder *all*. Dieser Parameter wird nur unter Solaris unterstützt.

Kommentare:

- Die Zustandsparameter *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` (oder den Standardwert) gesetzt ist und der Agent auf einem Solaris ohne Zonenunterstützung kompiliert wurde, aber auf einem neueren Solaris läuft, auf dem Zonen unterstützt werden, gibt der Agent NOTSUPPORTED zurück (der Agent kann die Ergebnisse nicht nur auf die aktuelle Zone beschränken). *all* wird in diesem Fall jedoch unterstützt.
- Siehe die Hinweise zur Auswahl von Prozessen mit den Parametern `name` und `cmdline` (Linux-spezifisch).

Beispiele:

```
proc.num[,mysql] #die Anzahl der Prozesse, die unter dem Benutzer mysql laufen
proc.num[apache2,www-data] #die Anzahl der apache2-Prozesse, die unter dem Benutzer www-data laufen
proc.num[,oracle,sleep,oracleZABBIX] #die Anzahl der Prozesse im Zustand sleep, die unter Oracle laufen un
```

sensor[device,sensor,<mode>]

 Messwert des Hardware-Sensors.
 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 Datenpunkts sensor unter Linux.
- Liest die MIB *hw.sensors* unter OpenBSD.
- Unter Linux 2.4 werden device und sensor als reguläre Ausdrücke behandelt, wenn mode angegeben ist.
- 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] #the temperature of one CPU
sensor["cpu[0-2]$",temp,avg] #the average temperature of the first three 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 die vollständige Zeichenfolge), *fqdn* (gibt den Fully Qualified Domain Name zurück);
- **transform** – mögliche Werte: *none* (Standard) oder *lower* (in Kleinbuchstaben umwandeln).

Der Wert wird ermittelt, indem `nodename` aus der Ausgabe der System-API 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 mit dem angegebenen regulären Ausdruck `interface` übereinstimmt (*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* - mit `type` als *utc*; *String* - mit `type` als *local*.
 Siehe unterstützte Plattformen.

Parameter:

- **type** - mögliche Werte: *utc* - (Standard) die Zeit seit der Epoch (00:00:00 UTC, 1. Januar 1970), gemessen in Sekunden, oder *local* - die Zeit im Format 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'

Muss als passiver Check für den Zabbix Agent verwendet werden; in Zabbix Agent 2 kann er als aktiver Check konfiguriert werden.

Beispiel:

```
system.localtime[local] #Erstellen Sie einen Datenpunkt mit diesem Schlüssel und verwenden Sie ihn dann, u
```

system.run[command,<mode>]

 Führt den angegebenen Befehl auf dem Host aus.
 Rückgabewert: *Text*-Ergebnis des Befehls oder 1 - mit `mode` als *nowait* (unabhängig vom Ergebnis des Befehls).
 Siehe unterstützte Plattformen.

Parameter:

- **command** - auszuführender Befehl;
- **mode** - mögliche Werte: *wait* - auf das Ende der Ausführung warten (Standard) oder *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 durch den Befehl erzeugt wird. Eine Prüfung des Exit-Codes wird nicht durchgeführt.
- Damit der Rückgabewert korrekt verarbeitet werden kann, muss er vom Datentyp `text` sein. Ein leeres Ergebnis ist ebenfalls zulässig.
- Der Rückgabewert ist auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden); Datenbankgrenzen gelten ebenfalls.
- Siehe auch: Befehlsausführung.

Beispiel:

```
system.run[ls -l /] #gibt eine detaillierte Dateiliste des Root-Verzeichnisses zurück
```

system.stat[resource,<type>]

 Die Systemstatistiken.
 Rückgabewert: *Integer* oder *float*.
 Unterstützte Plattformen: AIX.

Parameter:

- **ent** - die Anzahl der Prozessoreinheiten, 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, die in der Warteschlange des Virtual Memory Manager warten (float)
- **memory,<type>** - Informationen über die Nutzung von virtuellem und realem Speicher:
avm - aktive virtuelle Seiten (integer)
fre - Größe der Freiliste (integer)
- **page,<type>** - Informationen über Seitenfehler und Paging-Aktivität:
fi - Datei-Seiteneinlagerungen pro Sekunde (float)
fo - Datei-Seitenauslagerungen pro Sekunde (float)
pi - aus dem Auslagerungsspeicher eingelagerte Seiten (float)
po - in den Auslagerungsspeicher ausgelagerte Seiten (float)
fr - freigegebene Seiten (Seitenersetzung) (float)
sr - vom Seitenersetzungsalgorithmus gescannte Seiten (float)
- **faults,<type>** - Trap- und Interrupt-Rate:
in - Geräte-Interrupts (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-Anfrage(n) hatte (float)
pc - Anzahl der genutzten physischen Prozessoren (float)
ec - der Prozentsatz der genutzten zugewiesenen Kapazität (float)
lbusy - gibt den Prozentsatz der Auslastung logischer Prozessoren an, der bei der Ausführung auf Benutzer- und Systemebene auftrat (float)
app - gibt die verfügbaren physischen Prozessoren im Shared Pool an (float)
- **disk,<type>** - Festplattenstatistiken:
bps - gibt die Datenmenge an, die pro Sekunde zum Laufwerk übertragen wurde (gelesen oder geschrieben), in Byte (integer)
tps - gibt die Anzahl der Übertragungen pro Sekunde an, die an die physische Festplatte bzw. das Bandgerät gesendet wurden (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>]

 Die Informationen zum Betriebssystem.
 Rückgabewert: *String*.
 Unterstützte Plattformen: Linux, Windows.

Parameter:

- **info** – mögliche Werte: *full* (Standard), *short* oder *name*

Die Informationen werden aus folgenden Quellen bezogen (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 dies 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 (alphabetisch) installierte Pakete auf, deren Name dem angegebenen regulären Ausdruck entspricht (*all* listet alle auf).
- Unterstützte Paketmanager (ausgeführter Befehl):
  - dpkg (dpkg --get-selections)
  - pkgtool (ls /var/log/packages)
  - rpm (rpm -qa)
  - pacman (pacman -Q)
portage
- Wenn `format` als *full* angegeben ist, werden Pakete nach Paketmanagern gruppiert (jeder Manager in einer separaten Zeile, beginnend mit seinem Namen in eckigen Klammern).
- Wenn `format` als *short* angegeben ist, werden 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 Statistik für Swap-In-Vorgänge (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-In-Vorgänge, Standard auf Nicht-Linux-Plattformen), *sectors* (eingelagerte Sektoren) oder *pages* (eingelagerte 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 Parameter *sectors* für den Typ wird nur unter Linux unterstützt.

Beispiel:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Die Statistiken zum Auslagern (aus dem Speicher auf das Gerät).
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, OpenBSD.

Parameter:

- **device** - gibt das zum Auslagern verwendete Gerät an (nur Linux) oder *all* (Standard);
- **type** - mögliche Werte: *count* (Anzahl der Auslagerungsvorgänge, Standard auf Nicht-Linux-Plattformen), *sectors* (ausgelagerte Sektoren) oder *pages* (ausgelagerte 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 Parameter *sectors* für den Typ wird nur unter Linux unterstützt.

Beispiel:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 Die Größe des Swap-Speichers in Byte oder als Prozentsatz der Gesamtkapazität.
 Rückgabewert: *Integer* - für Byte; *Float* - für Prozentangaben.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parameter:

- **device** - gibt das zum Swapping verwendete Gerät an (nur FreeBSD) oder *all* (Standard);
- **type** - mögliche Werte: *free* (freier Swap-Speicher, Standard), *pfree* (freier Swap-Speicher in Prozent), *pused* (verwendeter Swap-Speicher in Prozent), *total* (gesamter Swap-Speicher) oder *used* (verwendeter Swap-Speicher).

Kommentare:

- Beachten Sie, dass *pfree* und *pused* unter Windows nicht unterstützt werden, wenn die Swap-Größe 0 ist.
- Wenn kein Gerät angegeben ist, berücksichtigt der Zabbix Agent nur Swap-Geräte (-Dateien); der physische Speicher wird ignoriert. Zum Beispiel umfasst auf Solaris-Systemen der Befehl `swap -s` einen Teil des physischen Speichers und der Swap-Geräte (im Gegensatz zu `swap -l`).

Beispiel:

```
system.swap.size[,pfree] → Prozentsatz des freien Swap-Speichers
```

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.read[<device>,<type>,<mode>]

 Die Festplatten-Lesestatistiken.
 Rückgabewert: *Integer* - mit `type` in *sectors*, *operations*, *bytes*; *Float* - mit `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: Sektoren, Operationen, Bytes pro Sekunde);
- **mode** - mögliche Werte: *avg1* (Ein-Minuten-Durchschnitt, Standard), *avg5* oder *avg15*. Dieser Parameter wird nur unterstützt, wenn `type` einer der folgenden Werte ist: sps, ops, bps.

Kommentare:

- Bei Verwendung eines Aktualisierungsintervalls von drei Stunden oder mehr[2] 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`) ebenso wie ein optionales Präfix `/dev/` (zum Beispiel `/dev/sda`) verwenden.
- Logische LVM-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 Schreibstatistiken des Datenträgers.
 Rückgabewert: *Integer* - mit `type` in *sectors*, *operations*, *bytes*; *Float* - mit `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 unterstützt, wenn `type` einer der folgenden Werte ist: sps, ops, bps.

Kommentare:

- Bei Verwendung eines Aktualisierungsintervalls von drei Stunden oder mehr[2] 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`) ebenso wie ein optionales Präfix `/dev/` (zum Beispiel `/dev/sda`) verwenden.
- Logische LVM-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; wenn leer, werden alle eingeschlossen (Standardwert);
- **regex excl** - ein regulärer Ausdruck, der das Namensmuster der auszuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; wenn leer, wird nichts ausgeschlossen (Standardwert);

- **types incl** - Typen von Verzeichniseinträgen, die gezählt werden sollen; mögliche Werte: *file* - reguläre Datei, *dir* - Unterverzeichnis, *sym* - symbolischer Link, *sock* - Socket, *bdev* - Blockgerät, *cdev* - Zeichengerät, *fifo* - FIFO, *dev* - synonym zu "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 - kein Abstieg in Unterverzeichnisse.
- **min size** - die Mindestgröße (in Byte), die eine Datei haben muss, um gezählt zu werden. Kleinere Dateien werden nicht gezählt. Speicher-Suffixe können verwendet werden.
- **max size** - die Maximalgröße (in Byte), die eine Datei haben darf, um gezählt zu werden. Größere Dateien werden nicht gezählt. Speicher-Suffixe können verwendet werden.
- **min age** - das Mindestalter (in Sekunden) eines Verzeichniseintrags, damit er gezählt wird. Neuere Einträge werden nicht gezählt. Zeit-Suffixe können verwendet werden.
- **max age** - das Höchstalter (in Sekunden) eines Verzeichniseintrags, damit er gezählt wird. So alte und ältere Einträge 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)

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Umgebungsvariablen, z. B. %APP_HOME%, $HOME und %TEMP%, werden nicht unterstützt.
- Pseudo-Verzeichnisse "." und ".." werden niemals gezählt.
- Symbolische Links werden beim Durchlaufen von Verzeichnissen niemals verfolgt.
- Sowohl `regex incl` als auch `regex excl` werden bei der Berechnung der Eintragsanzahl auf Dateien und Verzeichnisse angewendet, aber bei der Auswahl der zu durchlaufenden Unterverzeichnisse ignoriert (wenn `regex incl` "(?i)^.+\.zip$" ist und `max depth` nicht gesetzt ist, werden alle Unterverzeichnisse durchlaufen, aber nur Dateien vom Typ zip gezählt).
- Die Ausführungszeit ist durch den Timeout-Wert in der Agent-Konfiguration begrenzt (3 Sek.). Da das Durchlaufen großer Verzeichnisstrukturen länger dauern kann, werden keine Daten zurückgegeben und der Datenpunkt wird nicht unterstützt. Eine teilweise Zählung wird nicht zurückgegeben.
- Beim Filtern nach Größe haben nur reguläre Dateien aussagekräftige Größen. Unter Linux und BSD haben auch Verzeichnisse Größen ungleich null (typischerweise einige KB). Geräte haben die Größe null, z. B. spiegelt die Größe von **/dev/sda1** nicht die Größe der entsprechenden Partition wider. Daher wird bei Verwendung von <`min_size`> und <`max_size`> empfohlen, <`types_incl`> als "*file*" anzugeben, um Überraschungen 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-String*.
 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; wenn leer, werden alle eingeschlossen (Standardwert);
- **regex excl** - ein regulärer Ausdruck, der das Namensmuster der auszuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; wenn leer, wird nichts ausgeschlossen (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 zu "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 - kein Abstieg in Unterverzeichnisse.
- **min size** - die Mindestgröße (in Byte), die eine Datei haben muss, um aufgelistet zu werden. Kleinere Dateien werden nicht aufgelistet. Speichergrößensuffixe können verwendet werden.
- **max size** - die maximale Größe (in Byte), die eine Datei haben darf, um aufgelistet zu werden. Größere Dateien werden nicht aufgelistet. Speichergrößensuffixe können verwendet werden.

- **min age** - das Mindestalter (in Sekunden) eines Verzeichniseintrags, damit er aufgelistet wird. Neuere Einträge werden nicht aufgelistet. Zeitsuffixe können verwendet werden.
- **max age** - das Höchstalter (in Sekunden) eines Verzeichniseintrags, damit er aufgelistet wird. So alte und ältere Einträge 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`)

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Umgebungsvariablen, z. B. %APP_HOME%, $HOME und %TEMP%, werden nicht unterstützt.
- Pseudo-Verzeichnisse "." und ".." werden niemals aufgelistet.
- Symbolischen Links wird beim Durchlaufen von Verzeichnissen niemals gefolgt.
- Sowohl `regex incl` als auch `regex excl` werden beim Erzeugen der Eintragsliste auf Dateien und Verzeichnisse angewendet, aber bei der Auswahl der zu durchlaufenden Unterverzeichnisse ignoriert (wenn `regex incl` "(?i)^.+\.zip$" ist und `max depth` nicht gesetzt ist, werden alle Unterverzeichnisse durchlaufen, aber nur Dateien vom Typ zip gezählt).
- Die Ausführungszeit ist 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 unvollständige Liste wird nicht zurückgegeben.
- Beim Filtern nach Größe haben nur reguläre Dateien aussagekräftige Größen. Unter Linux und BSD haben auch Verzeichnisse Größen ungleich null (typischerweise einige KB). Geräte haben die Größe null, z. B. spiegelt die Größe von **/dev/sda1** nicht die Größe der entsprechenden Partition wider. Daher wird bei Verwendung von `min size` und `max size` empfohlen, `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 Byte).
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, Windows. Der Datenpunkt kann auch auf anderen UNIX-ähnlichen Plattformen funktionieren.

Parameter:

- **dir** - der absolute Pfad zum Verzeichnis;
- **regex incl** - ein regulärer Ausdruck, der das Namensmuster der einzuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; wenn leer, werden alle eingeschlossen (Standardwert);
- **regex excl** - ein regulärer Ausdruck, der das Namensmuster der auszuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; wenn leer, wird nichts ausgeschlossen (Standardwert);
- **mode** - mögliche Werte: *apparent* (Standard) - ermittelt die scheinbare Dateigröße statt der Plattenbelegung (entspricht `du -sb dir`), *disk* - ermittelt die Plattenbelegung (entspricht `du -s -B1 dir`). Im Gegensatz zum Befehl du berücksichtigt der Datenpunkt vfs.dir.size bei der Berechnung der Verzeichnisgröße auch versteckte Dateien (entspricht `du -sb .[^.]* *` innerhalb von dir).
- **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 dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Es werden nur Verzeichnisse berechnet, für die der Benutzer *zabbix* mindestens Leseberechtigung hat. 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 für diesen Datenpunkt aufgrund der Einstellung Timeout in den Konfigurationsdateien von Agent und Server/Proxy eine Zeitüberschreitung auftreten. Erhöhen Sie die Timeout-Werte bei Bedarf.
- Die Dateigrößenbegrenzung hängt von der Unterstützung für große Dateien ab.

Beispiele:

```
vfs.dir.size[/tmp,log] #berechnet die Größe aller Dateien in /tmp, die 'log' in ihrem Namen enthalten
vfs.dir.size[/tmp,log,^.+\.old$] #berechnet die Größe aller Dateien in /tmp, die 'log' in ihrem Namen enth
```

vfs.file.cksum[file,<mode>]

 Die Prüfsumme der Datei, 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 Dateigrößenbeschränkung 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** - die Kennung der Codepage.

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Der Rückgabewert ist auf 16 MB begrenzt (einschließlich nachgestellter Leerraumzeichen, die abgeschnitten werden); Datenbankbeschränkungen gelten ebenfalls.
- Eine leere Zeichenfolge wird zurückgegeben, wenn die Datei leer ist oder nur LF-/CR-Zeichen enthält.
- Die Bytereihenfolgemarkierung (BOM) wird von der Ausgabe ausgeschlossen.

Beispiel:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Prüft, ob die Datei existiert.
 Rückgabewert: 0 - nicht gefunden; 1 - Datei des angegebenen Typs existiert.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **types incl** - die Liste der einzuschließenden Dateitypen, mögliche Werte: *file* (reguläre Datei, Standardwert (wenn types_excl nicht gesetzt ist)), *dir* (Verzeichnis), *sym* (symbolischer Link), *sock* (Socket), *bdev* (Blockgerät), *cdev* (Zeichengerät), *fifo* (FIFO), *dev* (synonym zu "bdev,cdev"), *all* (alle genannten Typen, Standardwert, wenn types_excl gesetzt ist).
- **types excl** - die Liste der auszuschließenden Dateitypen, mögliche Werte siehe types_incl (standardmäßig werden keine Typen ausgeschlossen)

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Mehrere Typen müssen durch Kommas getrennt werden, und die gesamte Menge muss in doppelte Anführungszeichen "" eingeschlossen werden.
- Wenn derselbe Typ sowohl in <types_incl> als auch in <types_excl> enthalten ist, werden Dateien dieses Typs ausgeschlossen.
- Die Dateigrößenbeschränkung hängt von der Unterstützung für große 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.
- Das Limit für die Dateigröße hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.get[/etc/passwd] #gibt ein JSON mit Informationen über die Datei /etc/passwd zurück (Typ, Benutze
```

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 dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbeschränkung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Beispiel für den Rückgabewert:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Ruft den Eigentümer 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* wird unter Unix uid/gid, unter Windows SID zurückgegeben.

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbeschränkung hängt von der Unterstützung für große Dateien ab.

Beispiele:

```
vfs.file.owner[/tmp/zabbix_server.log] #return the file owner of /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #return the file owner ID of /tmp/zabbix_server.log
```

vfs.file.permissions[file]

 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>]

 Ruft eine Zeichenkette in der Datei ab[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** - der Bezeichner der Codepage;
- **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 zur Formatierung der Ausgabe. Die Escape-Sequenz **\0** wird durch den übereinstimmenden Teil des Textes ersetzt (vom ersten Zeichen, bei dem die Übereinstimmung beginnt, bis zu dem Zeichen, bei dem die Übereinstimmung endet), während eine Escape-Sequenz **\N** (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 dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbeschränkung hängt von der Unterstützung für große 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 Extraktion des Inhalts 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>]

 Sucht eine Zeichenkette in der Datei[7].
 Rückgabewerte: 0 - keine Übereinstimmung gefunden; 1 - gefunden.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt;
- **encoding** - der Codepage-Bezeichner;
- **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 Dateigrößenbeschränkung hängt von der Unterstützung für große Dateien ab.
- Die Bytereihenfolgemarkierung (BOM) wird ignoriert.

Beispiel:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 Die Dateigröße (in Byte).
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **mode** - mögliche Werte: *bytes* (Standard) oder *lines* (leere Zeilen werden ebenfalls gezählt).

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Datei muss Leseberechtigungen für den Benutzer *zabbix* haben.
- Die Größenbeschränkung der Datei hängt von der Unterstützung für große Dateien ab.

Beispiel:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Die Zeitinformationen der Datei.
 Rückgabewert: *Integer* (Unix-Zeitstempel).
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **mode** - mögliche Werte:
modify (Standard) - der letzte Zeitpunkt der Änderung des Dateiinhalts,
access - der letzte Zeitpunkt des Lesens der Datei,
change - der letzte Zeitpunkt der Änderung der Dateieigenschaften

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbeschränkung 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

 Die Liste der eingehängten Dateisysteme mit ihrem Typ, verfügbarem Festplattenspeicher, Inode-Statistiken und Einhängeoptionen. Kann für Low-Level-Discovery verwendet werden.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Kommentare:

- Dateisysteme mit einer Inode-Anzahl von null, was bei Dateisystemen mit dynamischen Inodes der Fall sein kann (z. B. btrfs), werden ebenfalls gemeldet.
- Siehe auch: Discovery eingehängter Dateisysteme.

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>]

 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* obligatorisch ist). Zulässige URL-Schemata: *http*, *https*[4]. Ein fehlendes Schema wird als *http* behandelt. Wenn eine URL angegeben ist, müssen `path` und `port` leer sein. Die Angabe von Benutzername/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)

Kommentare:

- Dieser Datenpunkt wird auf nicht unterstützt gesetzt, wenn die in `host` angegebene Ressource nicht existiert oder nicht verfügbar ist.
- `host` kann ein Hostname, Domainname, eine IPv4- oder IPv6-Adresse sein. Für IPv6-Adressen muss der Zabbix Agent jedoch mit aktivierter IPv6-Unterstützung kompiliert sein.

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.perf[host,<path>,<port>]

 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* obligatorisch ist). Zulässige URL-Schemata: *http*, *https*[4]. Ein fehlendes Schema wird als *http* behandelt. Wenn eine URL angegeben ist, müssen `path` und `port` leer sein. Die Angabe von Benutzername/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)

Kommentare:

- Dieser Datenpunkt wird auf nicht unterstützt gesetzt, wenn die in `host` angegebene Ressource nicht existiert oder nicht verfügbar ist.
- `host` kann ein Hostname, Domainname, eine IPv4- oder IPv6-Adresse sein. Für IPv6-Adressen muss der Zabbix Agent jedoch mit aktivierter IPv6-Unterstützung kompiliert sein.

Beispiele:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Sucht eine Zeichenkette auf der Webseite.
 Rückgabewert: Die gefundene Zeichenkette 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* obligatorisch ist). Zulässige URL-Schemata: *http*, *https*[4]. Ein fehlendes Schema wird als *http* behandelt. Wenn eine URL angegeben ist, müssen `path` und `port` leer sein. Die Angabe von Benutzername/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 von zurückzugebenden Zeichen;
- **output** - eine optionale Vorlage für die Ausgabeformatierung. Die Escape-Sequenz **\0** wird durch den übereinstimmenden Teil des Textes ersetzt (vom ersten Zeichen, bei dem die Übereinstimmung beginnt, bis zu dem Zeichen, bei dem die Übereinstimmung endet), während eine Escape-Sequenz **\N** (wobei N=1...9) durch die N-te gefundene Gruppe ersetzt wird (oder durch eine leere Zeichenkette, wenn N die Anzahl der erfassten Gruppen überschreitet).

Kommentare:

- Dieser Datenpunkt wird nicht unterstützt, wenn die in `host` angegebene Ressource nicht existiert oder nicht verfügbar ist.
- `host` kann ein Hostname, Domainname, eine IPv4- oder IPv6-Adresse sein. Für IPv6-Adressen muss der Zabbix Agent jedoch mit aktivierter IPv6-Unterstützung kompiliert sein.
- Die Extraktion von Inhalten mit dem Parameter `output` erfolgt auf dem Agent.

Beispiele:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 Die Host-Metadaten des Agent.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Gibt den Wert der Parameter HostMetadata oder HostMetadataItem zurück oder eine leere Zeichenfolge, wenn keine definiert sind.

agent.hostname

 Der Hostname des Agent.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Gibt zurück:

- Als passive Prüfung – den Namen des ersten Host, der im Parameter Hostname der Agent-Konfigurationsdatei aufgeführt ist;
- Als aktive Prüfung – den Namen des aktuellen Hostnamens.

agent.ping

 Die Verfügbarkeitsprüfung des Agent.
 Rückgabewert: Nichts – nicht verfügbar; 1 – verfügbar.
 Siehe unterstützte Plattformen.

Verwenden Sie die Auslöserfunktion **nodata()**, 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 Reihe interner Metriken des Zabbix Server oder Proxy 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-/Netzwerkmaskenliste von Servern/Proxys, die per Fernabfrage abgefragt werden sollen (Standard ist 127.0.0.1);
- **port** - der Port des Server/Proxy, der per Fernabfrage abgefragt werden soll (Standard ist 10051)

Kommentare:

- Eine ausgewählte Menge interner Metriken wird von diesem Datenpunkt zurückgegeben. Einzelheiten finden Sie unter Remote monitoring of Zabbix stats.
- Beachten Sie, dass die Statistikabfrage nur von den Adressen akzeptiert wird, die im Parameter 'StatsAllowedIP' des Server/Proxy auf der Zielinstanz 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 Metriken von der entfernten Instanz abgerufen, andernfalls von der lokalen Instanz.
 Rückgabewert: *JSON-String*.
 Siehe unterstützte Plattformen.

Parameter:

- **ip** - die IP-/DNS-/Netzwerkmaskenliste der Server/Proxys, die per Fernabfrage abgefragt werden sollen (Standard ist 127.0.0.1);
- **port** - der Port des Server/Proxy, der per Fernabfrage abgefragt werden soll (Standard ist 10051)
- **queue** - Konstante (unverändert zu verwenden)
- **from** - um mindestens verzögert (Standard ist 6 Sekunden)
- **to** - um höchstens verzögert (Standard ist unendlich)

Beachten Sie, dass die Statistikabfrage nur von den Adressen akzeptiert wird, die im Parameter 'StatsAllowedIP' server/proxy auf der Zielinstanz aufgeführt sind.

Fußnoten

[1] Ein Linux-spezifischer Hinweis. Der Zabbix Agent muss schreibgeschützten Zugriff auf das Dateisystem */proc* haben. Kernel-Patches von www.grsecurity.org beschränken die Zugriffsrechte nicht privilegierter Benutzer.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: Der Zabbix Agent beendet „veraltete" Geräteverbindungen, wenn auf die Werte der Datenpunkte länger als 3 Stunden nicht zugegriffen wird. Dies kann passieren, wenn ein System Geräte mit dynamisch wechselnden Pfaden hat oder wenn ein Gerät manuell entfernt wird. Beachten Sie auch, 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 zusammenfassende Statistiken zurück, einschließlich aller Blockgeräte wie sda, sdb und ihrer Partitionen (sda1, sda2, sdb3 ...) sowie mehrerer Geräte (MD raid), die auf diesen Blockgeräten/Partitionen basieren, und 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] SSL (HTTPS) wird nur unterstützt, wenn der Agent mit cURL-Unterstützung kompiliert wurde. Andernfalls wird der Datenpunkt nicht unterstützt.

[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 jedoch ein Solaris-System über net-snmp überwachen, können Werte zurückgegeben werden, da net-snmp Legacy-Code aus cmu-snmp enthält, der bis ins Jahr 1997 zurückreicht und beim Fehlschlagen des Lesens von Byte-Werten aus den Schnittstellenstatistiken den Paketzähler (der auf Loopback-Schnittstellen vorhanden ist) mit einem beliebigen Wert von 308 multipliziert. Dabei wird angenommen, dass die durchschnittliche Länge eines Pakets 308 Oktette beträgt, was eine sehr grobe Schätzung ist, da das MTU-Limit auf Solaris-Systemen für Loopback-Schnittstellen 8892 Byte beträgt. Es sollte nicht davon ausgegangen werden, dass diese Werte korrekt oder auch nur annähernd genau sind. Es sind grobe Schätzwerte. Der Zabbix Agent nimmt keine Schätzungen vor, aber net-snmp gibt für diese Felder einen Wert zurück.

[6] Die Befehlszeile auf 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] Die Datenpunkte `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` können zum Abrufen von Dateiinhalten verwendet werden. Wenn Sie den Zugriff auf bestimmte Dateien mit sensiblen Informationen einschränken möchten, führen Sie den Zabbix Agent unter einem Benutzer aus, der keine Berechtigung zum Anzeigen dieser Dateien hat.

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 angeben (z. B. „vfs.file.contents“). Die Liste der unterstützten Kodierungen (Codepage-Bezeichner) finden Sie in der Dokumentation zu libiconv (GNU Project) oder in der Microsoft Windows SDK-Dokumentation zu „Code Page Identifiers“. Beachten Sie, dass Microsoft einige Codepages manchmal als „nur für verwaltete Anwendungen verfügbar“ kennzeichnet — d. h. sie werden nur über die .NET-Laufzeitumgebung bereitgestellt und sind über die nativen Win32-APIs nicht verfügbar. 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 es sich um eine leere Zeichenfolge handelt), wird UTF-8 angenommen und die Daten werden „wie sie sind“ verarbeitet;
- BOM-Analyse – anwendbar auf Datenpunkte „vfs.file.contents“, „vfs.file.regexp“, „vfs.file.regmatch“. Es wird versucht, die korrekte Kodierung anhand der Byte Order Mark (BOM) am Anfang der Datei zu bestimmen. Wenn keine BOM vorhanden ist, wird stattdessen die Standardauflösung angewendet (siehe oben).

Fehlerbehebung bei Agent-Datenpunkten

Bei passiven Prüfungen ist Folgendes zu beachten, damit ein Datenpunkt keinen Wert verliert, weil die Anfrage des Servers an den Agent zuerst eine Zeitüberschreitung erreicht:

- Wenn die Agent-Version älter ist als die Server-Version, muss der Wert *Timeout* in der Datenpunkt-Konfiguration (oder das globale Timeout) möglicherweise höher sein als der `Timeout`-Wert in der Agent-Konfigurationsdatei.
- Wenn die Agent-Version neuer ist als die Server-Version, muss der `Timeout`-Wert in der Server-Konfigurationsdatei möglicherweise höher sein als der `Timeout`-Wert in der Agent-Konfigurationsdatei.

1 Zabbix Agent 2

Zabbix Agent 2 unterstützt alle Datenpunktschlüssel, die für den Zabbix Agent unter Unix und Windows unterstützt werden. Diese Seite enthält Details zu den zusätzlichen Datenpunktschlüsseln, die Sie nur mit Zabbix Agent 2 verwenden können, gruppiert nach dem Plugin, zu dem sie gehören.

Die Datenpunktschlüssel sind ohne Parameter und zusätzliche Informationen aufgeführt. Klicken Sie auf einen Datenpunktschlüssel, um die vollständigen Details anzuzeigen.

Datenpunktschlüssel	Beschreibung	Plugin
ceph.df.details	Datennutzung des Clusters und Verteilung auf die Pools.	Ceph
ceph.osd.stats	Aggregierte und OSD-spezifische Statistiken.	
ceph.osd.discovery	Die Liste der erkannten OSDs.	
ceph.osd.dump	Die Nutzungsschwellenwerte 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	Detaillierte Informationen über einen Container.	Docker
docker.container_stats	Die Statistiken zur Ressourcennutzung des Containers.	
docker.containers	Gibt die Liste der Container zurück.	
docker.containers.discovery	Gibt die Liste der Container zurück. Wird für Low-Level-Discovery verwendet.	
docker.data.usage	Informationen über die aktuelle Datennutzung.	
docker.images	Gibt die Liste der Images zurück.	
docker.images.discovery	Gibt die Liste der Images zurück. Wird für Low-Level-Discovery verwendet.	
docker.info	Die Systeminformationen.	

Datenpunktschlüssel	Beschreibung	Plugin
docker.ping	Prüft, ob der Docker-Daemon aktiv ist oder nicht.	
ember.get	Gibt das Ergebnis des angeforderten Geräts zurück.	Ember+
memcached.ping	Prüft, ob eine Verbindung aktiv ist oder nicht.	Memcached
memcached.stats	Ruft die Ausgabe des Befehls STATS ab.	
mongodb.collection.stats	Gibt verschiedene Speicherstatistiken für eine angegebene Collection zurück.	MongoDB
mongodb.cfg.discovery	Gibt eine Liste der erkannten Konfigurationsserver zurück.	MongoDB
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 von der aktuellen Datenbankinstanz zu anderen Mitgliedern des Sharded-Clusters oder Replica-Sets zurück.	
mongodb.db.stats	Gibt Statistiken zurück, die den Zustand eines bestimmten Datenbanksystems widerspiegeln.	
mongodb.db.discovery	Gibt eine Liste der erkannten Datenbanken zurück.	
mongodb.jumbo_chunks.count	Gibt die Anzahl der Jumbo-Chunks zurück.	
mongodb.oplog.stats	Gibt den Status des Replica-Sets zurück, basierend auf aus dem Oplog abgefragten Daten.	
mongodb.ping	Prüft, ob eine Verbindung aktiv ist oder nicht.	
mongodb.rs.config	Gibt die aktuelle Konfiguration des Replica-Sets zurück.	
mongodb.rs.status	Gibt den Status des Replica-Sets aus Sicht des Mitglieds zurück, auf dem die Methode ausgeführt wird.	
mongodb.server.status	Gibt den Zustand der Datenbank 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 Wildcards) des angegebenen Brokers und wartet auf Veröffentlichungen.	MQTT
mssql.availability.group.get	Gibt Verfügbarkeitsgruppen zurück.	MSSQL
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 einem Replikat (primär oder sekundär) beteiligt sind und sich auf dem Server befinden, zu dem die Verbindung hergestellt wurde.	
mssql.mirroring.get	Gibt Informationen zur Spiegelung zurück.	
mssql.nonlocal.db.get	Gibt Datenbanken zurück, die an einer Always On-Verfügbarkeitsgruppe und einem Replikat (primär oder sekundär) beteiligt sind und sich auf anderen Servern befinden (die Datenbank ist nicht lokal für die SQL-Server-Instanz, zu der die Verbindung hergestellt wurde).	
mssql.perfcounter.get	Gibt die Leistungsindikatoren zurück.	
mssql.ping	Prüft, ob eine Verbindung aktiv ist oder nicht.	
mssql.quorum.get	Gibt Informationen zum Quorum zurück.	
mssql.quorum.member.get	Gibt die Quorummitglieder 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 Größe der Datenbank in Byte.	
mysql.get_status_variables	Werte der globalen Statusvariablen.	
mysql.ping	Prüft, ob eine Verbindung aktiv ist oder nicht.	
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
nvml.device.decoder.utilization	Auslastung des GPU-Gerätedecoders.	
nvml.device.ecc.mode	ECC-Modus des GPU-Geräts.	
nvml.device.encoder.stats.get	Encoder-Statistiken des GPU-Geräts.	
nvml.device.encoder.utilization	Auslastung des GPU-Geräteencoders.	

Datenpunktschlüssel	Beschreibung	Plugin
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	Grafiktaktrate 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	Speichertaktrate 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.	
nvml.device.power.usage	Leistungsaufnahme des GPU-Geräts.	
nvml.device.serial	Seriennummer des GPU-Geräts.	
nvml.device.sm.frequency	Taktrate des Streaming-Multiprocessors des GPU-Geräts.	
nvml.device.temperature	Temperatur des GPU-Geräts.	
nvml.device.utilization	Auslastungsstatistiken des GPU-Geräts.	
nvml.device.video.frequency	Videotaktrate 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)-Datenträgergruppen zurück.	Oracle
oracle.diskgroups.discovery	Gibt die Liste der ASM-Datenträgergruppen 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 der 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 der 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 Kontrolldatei.	
oracle.sga.stats	Gibt die Statistiken der System Global Area (SGA) zurück.	
oracle.sessions.stats	Gibt die Sitzungsstatistiken zurück.	
oracle.sys.metrics	Gibt eine Reihe von Systemmetrikwerten 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
pgsql.archive	Die Informationen über archivierte Dateien.	
pgsql.bgwriter	Die kombinierte Anzahl der Checkpoints für den Datenbankcluster, aufgeschlüsselt nach Checkpoint-Typ.	
pgsql.cache.hit	Die Trefferquote des PostgreSQL-Puffercaches.	
pgsql.connections	Gibt Verbindungen nach Typ zurück.	
pgsql.custom.query	Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.	
pgsql.db.age	Das Alter der ältesten FrozenXID der Datenbank.	
pgsql.db.bloating_tables	Die Anzahl aufgeblähter Tabellen pro Datenbank.	
pgsql.db.discovery	Die Liste der PostgreSQL-Datenbanken.	
pgsql.db.size	Die Größe der Datenbank in Byte.	
pgsql.dbstat	Erfasst die Statistiken pro Datenbank.	
pgsql.dbstat.sum	Die zusammengefassten Daten für alle Datenbanken in einem Cluster.	
pgsql.locks	Die Informationen über gewährte Sperren pro Datenbank.	
pgsql.oldest.xid	Das Alter der ältesten XID.	
pgsql.ping	Prüft, ob eine Verbindung aktiv ist oder nicht.	
pgsql.queries	Abfragemetriken nach Ausführungszeit.	

Datenpunktschlüssel	Beschreibung	Plugin
pgsql.replication.count	Die Anzahl der Standby-Server.	
pgsql.replication.process	Der Flush-Lag, Write-Lag und Replay-Lag für jeden Senderprozess.	
pgsql.replication.process.discovery	Die Erkennung von Replikationsprozessnamen.	
pgsql.replication.recovery_role	Der Wiederherstellungsstatus.	
pgsql.replication.status	Der Status der Replikation.	
pgsql.replication_lag.b	Der Replikationsverzug in Byte.	
pgsql.replication_lag.sec	Der Replikationsverzug in Sekunden.	
pgsql.uptime	Die PostgreSQL-Betriebszeit in Millisekunden.	
pgsql.version	Gibt die PostgreSQL-Version zurück.	
pgsql.wal.stat	Die WAL-Statistiken.	
redis.config	Ruft die Konfigurationsparameter einer Redis-Instanz ab, die dem Muster entsprechen.	Redis
redis.info	Ruft die Ausgabe des Befehls INFO ab.	
redis.ping	Prüft, ob eine Verbindung aktiv ist oder nicht.	
redis.slowlog.count	Die Anzahl der Slow-Log-Einträge seit dem Start von Redis.	
smart.attribute.discovery	Gibt eine Liste von S.M.A.R.T.-Geräteattributen zurück.	S.M.A.R.T.
smart.disk.discovery	Gibt eine Liste von S.M.A.R.T.-Geräten zurück.	
smart.disk.get	Gibt alle verfügbaren Eigenschaften von S.M.A.R.T.-Geräten zurück.	
systemd.unit.get	Gibt alle Eigenschaften einer systemd-Unit zurück.	Systemd
systemd.unit.info	Informationen zur systemd-Unit.	
systemd.unit.discovery	Die Liste der systemd-Units und ihre Details.	
web.certificate.get	Validiert die Zertifikate und gibt Zertifikatsdetails zurück.	Web certificates

Siehe auch:

- Integrierte Plugins
- Ladbare Plugins

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 URL des MQTT-Brokers 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, wird das Standardprotokoll (`tcp`) bzw. der Standardport (1883) verwendet;
- **topic** - das MQTT-Topic (erforderlich). Platzhalter (+,#) werden unterstützt;
- **user, password** - die Anmeldedaten zur Authentifizierung (falls erforderlich).

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein (Datenpunkttyp „Zabbix Agent (active)“);
- TLS-Verschlüsselungszertifikate können verwendet werden, indem sie an einem Standardspeicherort 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 Unterstützung für mehr Eintragstypen und anpassbare Flags.
 Rückgabewerte: *JSON-Objekt*

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** - das Timeout 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 die 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 der DNS-Name bei einer PTR-Abfrage 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 die Werte nur in Kleinbuchstaben angegeben werden dürfen; Großbuchstaben oder gemischte Schreibweise werden nicht unterstützt.
- Internationalisierte Domainnamen werden nicht unterstützt; verwenden Sie stattdessen IDNA-kodierte Namen.
- Die Ausgabe ist ein Objekt, das auf Grundlage der angegebenen Parameter DNS-Eintragsinformationen enthält (siehe weitere Details).

Beispiele:

```
net.dns.get[192.0.2.0,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 ASM-Datenträgergruppen (Automatic Storage Management) 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;
- **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` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename.

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 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` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und 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 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;
- **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 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;
- **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 Statistiken 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 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 und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service.

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` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

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 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.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 und 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 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 und 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:
  - der URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn er ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und trotzdem 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.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 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.

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` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename.

oracle.redolog.info[connString,<user>,<password>,<service>]

 Die Protokolldateiinformationen 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` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

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 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.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` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name;
- **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 Menge 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 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 berücksichtigt und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename;
- **duration** - das Erfassungsintervall (in Sekunden) für Systemmetrikwerte. 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` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Gibt die Tabellenbereichsstatistiken 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-Servicename;
- **tablespace** - Name des abzufragenden Tabellenbereichs. 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 Tabellenbereichs. 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 Tabellenbereichsstatistiken 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` (eine Anmeldeoption ist nicht case-sensitiv und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service.

oracle.user.info[connString,<user>,<password>,<service>,<username>]

 Gibt Informationen über den 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` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service;
- **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 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-Service.

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-Zugangsdaten;
- **database name** - der Name der Datenbank (wenn ausgelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet).

pgsql.archive[uri,<username>,<password>,<database name>]

 Informationen über archivierte Dateien.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (wenn ausgelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 Die kombinierte Anzahl der Checkpoints für den Datenbank-Cluster, 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 (wenn 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-Puffercaches.
 Rückgabewert: *Float*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (wenn ausgelassen, 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 (wenn weggelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet).

pgsql.custom.query[uri,<username>,<password>,queryName,<args...>]

 Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **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 der ältesten FrozenXID der Datenbank.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Name der Datenbank (wenn ausgelassen, 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 (wenn ausgelassen, 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 (wenn ausgelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet).

pgsql.db.size[uri,<username>,<password>,<database name>]

 Die Datenbankgröße in Byte.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (wenn ausgelassen, 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-Anmeldedaten;
- **database name** - der Datenbankname (wenn ausgelassen, 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-Zugangsdaten;
- **database name** - der Datenbankname (wenn ausgelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet).

pgsql.locks[uri,<username>,<password>,<database name>]

 Die Informationen über gewährte Sperren 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 (wenn 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.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (wenn ausgelassen, 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 beliebiger Fehler auftritt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (wenn weggelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Fragt Metriken zu Abfragen nach Ausführungszeit ab.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (wenn ausgelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet);
- **time period** - das Ausführungszeitlimit für die Anzahl langsamer Abfragen (muss eine positive Ganzzahl 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-Betriebszeit in Millisekunden.
 Rückgabewert: *Float*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (wenn 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 (wenn ausgelassen, 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-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (wenn ausgelassen, 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 von S.M.A.R.T.-Geräteattributen zurück.
 Rückgabewert: *JSON-Objekt*.

Kommentare:

- Die folgenden Makros und ihre Werte werden zurückgegeben: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH};
- Die Laufwerkstypen HDD, SSD und NVME werden unterstützt. Laufwerke können einzeln oder in einem RAID-Verbund kombiniert sein. {#NAME} erhält im Falle 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-Objekt*.

Parameter:

- **type** - gibt einen Wert an, nach dem die Festplatten durchsucht werden. Mögliche Werte: *id*, *name* (Standard). *id* wird unter Windows nicht unterstützt und gibt einen Fehler zurück, wenn es ausgewählt wird

Kommentare:

- Die folgenden Makros und ihre Werte werden zurückgegeben: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE};
- Die Laufwerkstypen HDD, SSD und NVME werden unterstützt. Wenn ein Laufwerk nicht zu einem RAID gehört, ist {#RAIDTYPE} leer. {#NAME} enthält im Fall von 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-Objekt*.

Parameter:

- **path** - der Festplattenpfad; das Makro {#PATH} kann als Wert verwendet werden;
- **raid_type** - der RAID-Typ; das Makro {#RAID} kann als Wert verwendet werden

Kommentare:

- Die Laufwerkstypen HDD, SSD und NVME werden unterstützt. Laufwerke können einzeln verwendet oder in einem RAID kombiniert werden;
- Die Daten enthalten die smartctl-Version und Aufrufargumente sowie zusätzliche Felder:
disk_name - enthält den Namen mit den für die RAID-Erkennung erforderlichen Zusätzen, z. B.: {"disk_name": "/dev/sda cciss,2"}
disk_type - enthält den Festplattentyp HDD, SSD oder NVME, z. B.: {"disk_type": "ssd"};
- Wenn keine Parameter angegeben sind, gibt der Datenpunkt Informationen über alle Festplatten zurück.

systemd.unit.get[unit name,<interface>]

 Gibt alle Eigenschaften einer systemd-Unit zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **unit name** - der Name der Unit (möglicherweise möchten Sie das Makro {#UNIT.NAME} im Datenpunkt-Prototyp verwenden, um den Namen zu ermitteln);
- **interface** - der Schnittstellentyp der Unit, 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** - 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 kann eine bestimmte Eigenschaft von einem bestimmten Schnittstellentyp abgerufen werden, wie in der dbus API beschrieben.

Beispiele:

```
systemd.unit.info["{#UNIT.NAME}"] #collect active state (active, reloading, inactive, failed, activating,
systemd.unit.info["{#UNIT.NAME}",LoadState] #collect load state info on discovered systemd units
systemd.unit.info[mysqld.service,Id] #retrieve the service technical name (mysqld.service)
systemd.unit.info[mysqld.service,Description] #retrieve the service description (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #retrieve the last time the service entered the act
systemd.unit.info[dbus.socket,NConnections,Socket] #collect the number of connections from this socket uni
```

systemd.unit.discovery[<type>]

 Liste 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-Adresse oder ein DNS-Name sein.
Kann das URL-Schema (*nur https*), einen Pfad (wird ignoriert) und einen Port enthalten.
Wenn ein Port sowohl im ersten als auch im zweiten Parameter angegeben wird, müssen die Werte übereinstimmen.
Wenn address (der 3. Parameter) angegeben ist, wird hostname nur für SNI und die Hostnamenprüfung verwendet;
- **port** - die Portnummer (Standard ist 443 für HTTPS);
- **address** - kann entweder eine IP-Adresse oder ein DNS-Name 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-Adresse und

der 3. Parameter ein DNS-Name ist, wird der 1. Parameter für die Verbindung verwendet, und der 3. Parameter wird für SNI und die Hostprüfung verwendet.

Kommentare:

- Dieser Datenpunkt wird zu nicht unterstützt, wenn das in der Host-Konfiguration angegebene Ziel nicht existiert, nicht verfügbar ist oder wenn der TLS-Handshake mit einem beliebigen Fehler außer einem ungültigen Zertifikat fehlschlägt;
- Derzeit werden die AIA-(Authority Information Access)-X.509-Erweiterung, 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 Antragstellernamen (falls vorhanden), andernfalls `null`.
  - result: enthält das Validierungsergebnis.
    - *value*: der Validierungsstatus (siehe mögliche Werte unten).
    - *message*: detaillierte Validierungsmeldung (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` zeigt 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 stimmt mit seinem Aussteller überein.
  - *invalid* – das Zertifikat ist aufgrund eines Problems wie Ablauf, falschem Hostnamen oder einer unbekannten signierenden Zertifizierungsstelle 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"
}
```

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

Gemeinsam genutzte Datenpunkte

Die folgende Tabelle listet Zabbix-Agent-Datenpunkte auf, die unter Windows unterstützt werden und mit dem UNIX-Zabbix-Agent gemeinsam genutzt werden:

- Der Datenpunktschlüssel ist ein Link zu den vollständigen Details des UNIX-Zabbix-Agent-Datenpunkts
- Für Windows relevante Kommentare zu Datenpunkten sind enthalten

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
log	Die Überwachung einer Protokolldatei. Dieser Datenpunkt wird für das Windows-Ereignisprotokoll nicht unterstützt. Der Parameter `persistent_dir` wird unter Windows nicht unterstützt.	Protokollüberwachung
log.count	Die Anzahl übereinstimmender 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 rotierten Protokolldatei. Dieser Datenpunkt wird für das Windows-Ereignisprotokoll nicht unterstützt. Der Parameter `persistent_dir` wird unter Windows nicht unterstützt.	
logrt.count	Die Anzahl übereinstimmender Zeilen in einer überwachten, rotierten 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 verfügbar ist. Die Parameter `ip`, `timeout` und `count` werden unter Windows ignoriert, sofern nicht Zabbix-Agent 2 verwendet wird.	Netzwerk
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 Netzwerkschnittstellen. Einige Windows-Versionen (zum Beispiel Server 2008) erfordern möglicherweise die Installation der neuesten Updates, um Nicht-ASCII-Zeichen in Schnittstellennamen zu unterstützen.	
net.if.in	Die Statistik des eingehenden Datenverkehrs auf einer Netzwerkschnittstelle. Unter Windows bezieht der Datenpunkt Werte aus 64-Bit-Zählern, falls verfügbar. 64-Bit-Zähler für Schnittstellenstatistiken wurden in Windows Vista und Windows Server 2008 eingeführt. Wenn keine 64-Bit-Zähler verfügbar sind, verwendet der Agent 32-Bit-Zähler. Mehrbyte-Schnittstellennamen werden unter Windows unterstützt. Sie können Netzwerkschnittstellenbeschreibungen unter Windows mit den Datenpunkten net.if.discovery oder net.if.list abrufen.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
net.if.out	Die Statistik des ausgehenden Datenverkehrs auf einer Netzwerkschnittstelle. Unter Windows bezieht der Datenpunkt Werte aus 64-Bit-Zählern, falls verfügbar. 64-Bit-Zähler für Schnittstellenstatistiken wurden in Windows Vista und Windows Server 2008 eingeführt. Wenn keine 64-Bit-Zähler verfügbar sind, verwendet der Agent 32-Bit-Zähler. Mehrbyte-Schnittstellennamen werden unter Windows unterstützt. Sie können Netzwerkschnittstellenbeschreibungen 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 einer Netzwerkschnittstelle. Unter Windows bezieht der Datenpunkt Werte aus 64-Bit-Zählern, falls verfügbar. 64-Bit-Zähler für Schnittstellenstatistiken wurden in Windows Vista und Windows Server 2008 eingeführt. Wenn keine 64-Bit-Zähler verfügbar sind, verwendet der Agent 32-Bit-Zähler. Sie können Netzwerkschnittstellenbeschreibungen unter Windows mit den Datenpunkten net.if.discovery oder net.if.list abrufen.	
net.tcp.listen	Prüft, ob sich dieser TCP-Port im Status LISTEN befindet.	
net.tcp.port	Prüft, ob es möglich ist, eine TCP-Verbindung zum angegebenen Port herzustellen.	
net.tcp.service	Prüft, ob ein Dienst läuft und TCP-Verbindungen akzeptiert. 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 vom Zabbix-Agent unterstützt, unter Windows jedoch nur von Zabbix agent 2 auf 64-Bit-Windows.	
net.udp.service	Prüft, ob ein Dienst läuft 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 vom 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.	Prozesse
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-Last. Wenn auf dem 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 prozentuale CPU-Auslastungswert. 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 Indikator *Processor Time* verwendet wurde (siehe weitere Details). *system* ist der einzige unter Windows unterstützte `type`-Parameter.	
system.hostname	Der System-Host-Name. Der Wert wird unter Windows entweder mit den 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	Die Informationen zur Softwarearchitektur.	
system.sw.os	Die Informationen zum Betriebssystem.	
system.sw.os.get	Detaillierte Informationen zum Betriebssystem (Version, Typ, Name der Distribution, Neben- und Hauptversion usw.).	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
system.swap.size	Die Größe des Auslagerungsspeichers in Byte oder als Prozentsatz der Gesamtgröße. Der Typ-Parameter `pused` wird unter Linux vom 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 Größe/einen falschen Prozentsatz des Auslagerungsspeichers meldet. In diesem Fall können Sie den Schlüssel `perf_counter[\700(_Total)\702]` verwenden, um den korrekten Prozentsatz des Auslagerungsspeichers zu erhalten.	
system.uname	Identifikation des Systems. Unter Windows wird der Wert für diesen Datenpunkt aus den WMI-Klassen Win32_OperatingSystem und Win32_Processor bezogen. Der Betriebssystemname (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 symbolische Verzeichnislinks übersprungen und Hardlinks nur einmal gezählt.	Virtuelle Dateisysteme
vfs.dir.get	Die Liste der Verzeichniseinträge. Unter Windows werden symbolische Verzeichnislinks übersprungen und Hardlinks nur einmal gezählt.	
vfs.dir.size	Die Verzeichnisgröße. Unter Windows wird jeder symbolische Link übersprungen und Hardlinks werden nur einmal berücksichtigt.	
vfs.file.cksum	Die Dateiprüfsumme, berechnet mit dem UNIX-cksum-Algorithmus.	
vfs.file.contents	Ruft den Inhalt einer Datei ab.	
vfs.file.exists	Prüft, ob die Datei existiert. Unter Windows müssen doppelte Anführungszeichen mit einem Backslash '\' maskiert werden, und der gesamte Datenpunktschlüssel muss in doppelte Anführungszeichen eingeschlossen werden, wenn das Befehlszeilenprogramm zum Aufruf von zabbix_get.exe oder agent2 verwendet wird. Beachten Sie, dass der Datenpunkt unter Windows möglicherweise auf nicht unterstützt wechselt, wenn in einem nicht vorhandenen Verzeichnis nach einem Verzeichnis gesucht wird, z. B. `vfs.file.exists[C:\no\dir,dir]` (wobei 'no' nicht existiert).	
vfs.file.get	Gibt Informationen über eine Datei zurück. Unter Windows unterstützte Dateitypen: reguläre Datei, Verzeichnis, symbolischer Link	
vfs.file.md5sum	Die MD5-Prüfsumme der Datei.	
vfs.file.owner	Ruft den Eigentümer 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 Zeitinformationen der Datei. 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 ihren 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 Byte oder als Prozentsatz der Gesamtgröße.	
vm.memory.size	Die Speichergröße in Byte oder als Prozentsatz der Gesamtgröße.	Virtueller Speicher
web.page.get	Ruft den Inhalt einer Webseite ab.	Webüberwachung
web.page.perf	Die Ladezeit einer vollständigen Webseite.	
web.page.regexp	Findet eine Zeichenfolge auf der Webseite.	
agent.hostmetadata	Die Agent-Host-Metadaten.	Zabbix
agent.hostname	Der Agent-Host-Name.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
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 per Fernzugriff eine Reihe interner Metriken des Zabbix-Server oder Proxy zurück.	
zabbix.stats	Gibt per Fernzugriff die Anzahl überwachter Datenpunkte in der Warteschlange zurück, die auf dem Zabbix-Server oder Proxy 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 Ereignisprotokollkanals (*Log Name* in der GUI der Ereignisanzeige);
- **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) oder *skip* - die Verarbeitung älterer Daten überspringen (betrifft nur neu erstellte Datenpunkte).

Kommentare:

- Der Datenpunkt muss als aktive Prüfung 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 Log-Überwachung.

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 Ereignisprotokollkanals (*Log Name* in der GUI der Ereignisanzeige);
- **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);
- **maxproclines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent analysieren wird (darf 10000 nicht überschreiten). Der Standardwert ist 10*’MaxLinesPerSecond’ in zabbix_agentd.conf.
- **mode** - mögliche Werte: *all* (Standard) oder *skip* - Verarbeitung älterer Daten überspringen (betrifft nur neu erstellte Datenpunkte).

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 Informationen zu Protokollschweregrad und Quelle verloren;
- Siehe auch zusätzliche Informationen zur Log-Überwachung.

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-Performance-Counters.
 Rückgabewert: *Integer*, *Float*, *String* oder *Text* (abhängig von der Anfrage).

Parameter:

- **counter** - der Pfad zum Counter;

- **interval** - die letzten N Sekunden zum Speichern des Durchschnittswerts. `interval` muss zwischen 1 und 900 Sekunden (einschließlich) liegen, der Standardwert ist 1.

Kommentare:

- Dieser Datenpunkt folgt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann;
- `interval` wird für Counter verwendet, die mehr als ein Sample benötigen (wie CPU-Auslastung), sodass die Prüfung jedes Mal einen Durchschnittswert für die letzten „interval"-Sekunden zurückgibt;
- Performance Monitor kann verwendet werden, um die Liste der verfügbaren Counter zu erhalten.
- Siehe auch: Windows-Performance-Counter.

perf_counter_en[counter,<interval>]

 Der Wert eines beliebigen Windows-Leistungsindikators auf Englisch.
 Rückgabewert: *Integer*, *Float*, *String* oder *Text* (abhängig von der Anfrage).

Parameter:

- **counter** - der Pfad zum Leistungsindikator auf Englisch;
- **interval** - die letzten N Sekunden zum Speichern des Durchschnittswerts. `interval` muss zwischen 1 und 900 Sekunden (einschließlich) liegen, der Standardwert ist 1.

Kommentare:

- Dieser Datenpunkt folgt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann;
- `interval` wird für Leistungsindikatoren verwendet, die mehr als eine Stichprobe benötigen (wie CPU-Auslastung), sodass die Prüfung jedes Mal einen Durchschnittswert für die letzten „interval"-Sekunden zurückgibt;
- Dieser Datenpunkt wird nur unter **Windows Server 2008/Vista** und höher unterstützt;
- Die Liste der englischen Zeichenfolgen finden Sie unter folgendem Registrierungsschlüssel: `HKEY_LOCAL_MACHINE\SOFTWARE\Micro 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 (relevant, wenn mehr als ein Prozess mit demselben Namen existiert)

Kommentare:

- Die folgenden `attributes` werden unterstützt:
vmsize (Standard) - Größe des virtuellen Prozessspeichers in KByte
wkset - Größe des Working Sets des Prozesses (Menge des vom Prozess verwendeten physischen Speichers) in KByte
pf - Anzahl der Seitenfehler
ktime - Kernel-Zeit des Prozesses in Millisekunden
utime - Benutzerzeit des Prozesses in Millisekunden
io_read_b - Anzahl der vom Prozess bei I/O-Operationen gelesenen Bytes
io_read_op - Anzahl der vom Prozess ausgeführten Leseoperationen
io_write_b - Anzahl der vom Prozess bei I/O-Operationen geschriebenen Bytes
io_write_op - Anzahl der vom Prozess ausgeführten Schreiboperationen
io_other_b - Anzahl der vom Prozess bei anderen Operationen als Lese- und Schreiboperationen übertragenen Bytes
io_other_op - Anzahl der vom Prozess ausgeführten I/O-Operationen außer Lese- und Schreiboperationen
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 für alle Prozesse mit dem Namen <process>;
- Auf einem 64-Bit-System ist ein 64-Bit-Zabbix-Agent erforderlich, damit dieser Datenpunkt korrekt funktioniert.

Beispiele:

```
proc_info[iexplore.exe,wkset,sum] #Ruft die Menge des physischen Speichers ab, die von allen Internet-Expl
proc_info[iexplore.exe,pf,avg] #Ruft die durchschnittliche Anzahl der Seitenfehler für Internet-Explorer-P
```

registry.data[key,<Wertname>]

 Gibt Daten für den angegebenen Wertnamen im Windows-Registrierungsschlüssel zurück.
 Rückgabewert: *Integer*, *string* oder *text* (abhängig vom Werttyp)

Parameter:

- **key** - der Registrierungsschlüssel einschließlich des Stammschlüssels; Stammabkürzungen (z. B. HKLM) sind zulässig;
- **Wertname** - der Name des Registrierungswerts im Schlüssel (standardmäßig leerer String ””). Der Standardwert wird zurückgegeben, wenn kein Wertname 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 Registrierungsschlüssel einschließlich des Stammschlüssels; Abkürzungen für Stammschlüssel (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** - nur Werte mit Namen ermitteln, die dem regulären Ausdruck entsprechen (Standard – alle Werte ermitteln). Nur zusammen 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$
The JSON will include details of the key, last subkey, value name, value type and value data.
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #gibt die Daten aller Werte
The JSON will include details of the key, last subkey, value name, value type and value data.
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #gibt alle Unterschlüssel dies
The JSON will include details of the key and last subkey.
```

service.discovery

 Die Liste der Windows-Dienste. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

service.info[service,<param>]

 Informationen über einen Dienst.
 Rückgabewert: *Integer* - mit `param` als *state*, *startup*; *String* - mit `param` als *displayname*, *path*, *user*; *Text* - mit `param` als *description*
Speziell für *state*: 0 - läuft, 1 - pausiert, 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

Parameter:

- **service** - ein tatsächlicher Dienstname oder sein Anzeigename, wie im MMC-Snap-In „Dienste“ angezeigt;
- **param** - *state* (Standard), *displayname*, *path*, *user*, *startup* oder *description*.

Kommentare:

- Datenpunkte wie `service.info[service,state]` und `service.info[service]` geben dieselben Informationen zurück;

- Nur wenn param den Wert *state* hat, gibt dieser Datenpunkt einen Wert für nicht vorhandene Dienste zurück (255).

Beispiele:

```
service.info[SNMPTRAP] - Status des Dienstes SNMPTRAP;
service.info[SNMP Trap] - Status desselben Dienstes, jedoch 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 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* sollten Sie ein Feld mit der Bezeichnung „Dienstname“ sehen. Der darauf 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** sein.

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.

2 SNMP Agent

Übersicht

Möglicherweise möchten Sie SNMP-Monitoring auf Geräten wie Druckern, Netzwerk-Switches, Routern oder USVs verwenden, die in der Regel SNMP-fähig sind und auf denen es unpraktisch wäre, vollständige Betriebssysteme und Zabbix-Agenten einzurichten.

Damit Daten abgerufen werden können, die von SNMP-Agenten auf diesen Geräten bereitgestellt werden, muss der Zabbix-Server zunächst mit SNMP-Unterstützung konfiguriert werden, indem das Flag `--with-net-snmp` angegeben wird. Es wird außerdem empfohlen, MIB-Dateien zu installieren, damit Werte von Datenpunkten im korrekten Format angezeigt werden. Ohne die MIB-Dateien können Formatierungsprobleme auftreten, z. B. dass Werte in HEX statt in UTF-8 oder umgekehrt angezeigt werden.

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 wiederholt SNMP-walk- und get-Datenpunkte bis zu 5-mal. Der Wiederholungsmechanismus gilt nicht für DNS-Auflösungsfehler.

Bei veralteten SNMP-Prüfungen (einzelne OID-Nummer oder Zeichenfolge) wiederholt Zabbix-Server/Proxy nach einem fehlgeschlagenen Abfrageversuch mindestens einmal: 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 als snmpEngineID oder "Engine ID" bezeichnet) 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 einem solchen Fall muss der SNMP-Cache auf einem Server/Proxy manuell geleert werden (mit -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 wieder, anstatt jedes Mal eine Probe zu senden, wodurch der Netzwerkverkehr reduziert wird. Wenn eine EngineID nicht wiederverwendet werden kann, wird ein Wiederholungsversuch mit einer Probe 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 den SNMP-String (oder die OID) des Datenpunkts, den Sie überwachen möchten.

Um eine Liste von SNMP-Strings zu erhalten, verwenden Sie den Befehl **snmpwalk** (Teil der Software net-snmp, die Sie im Rahmen der Zabbix-Installation installiert haben sollten) oder ein gleichwertiges Werkzeug:

```
snmpwalk -v 2c -c public <host IP> .
```

Da '2c' hier für die SNMP-Version steht, können Sie es auch durch '1' ersetzen, um SNMP Version 1 auf dem Gerät anzugeben.

Dies sollte Ihnen eine Liste von SNMP-Strings und deren letztem Wert liefern. Falls nicht, ist es möglich, dass die SNMP-'community' vom Standardwert 'public' abweicht; in diesem Fall müssen Sie herausfinden, welcher Wert verwendet wird.

Anschließend können Sie die Liste durchgehen, bis Sie den String finden, den Sie überwachen möchten, z. B. wenn Sie die eingehenden Bytes auf Port 3 Ihres Switches überwachen möchten, würden Sie den String 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 im String 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
```

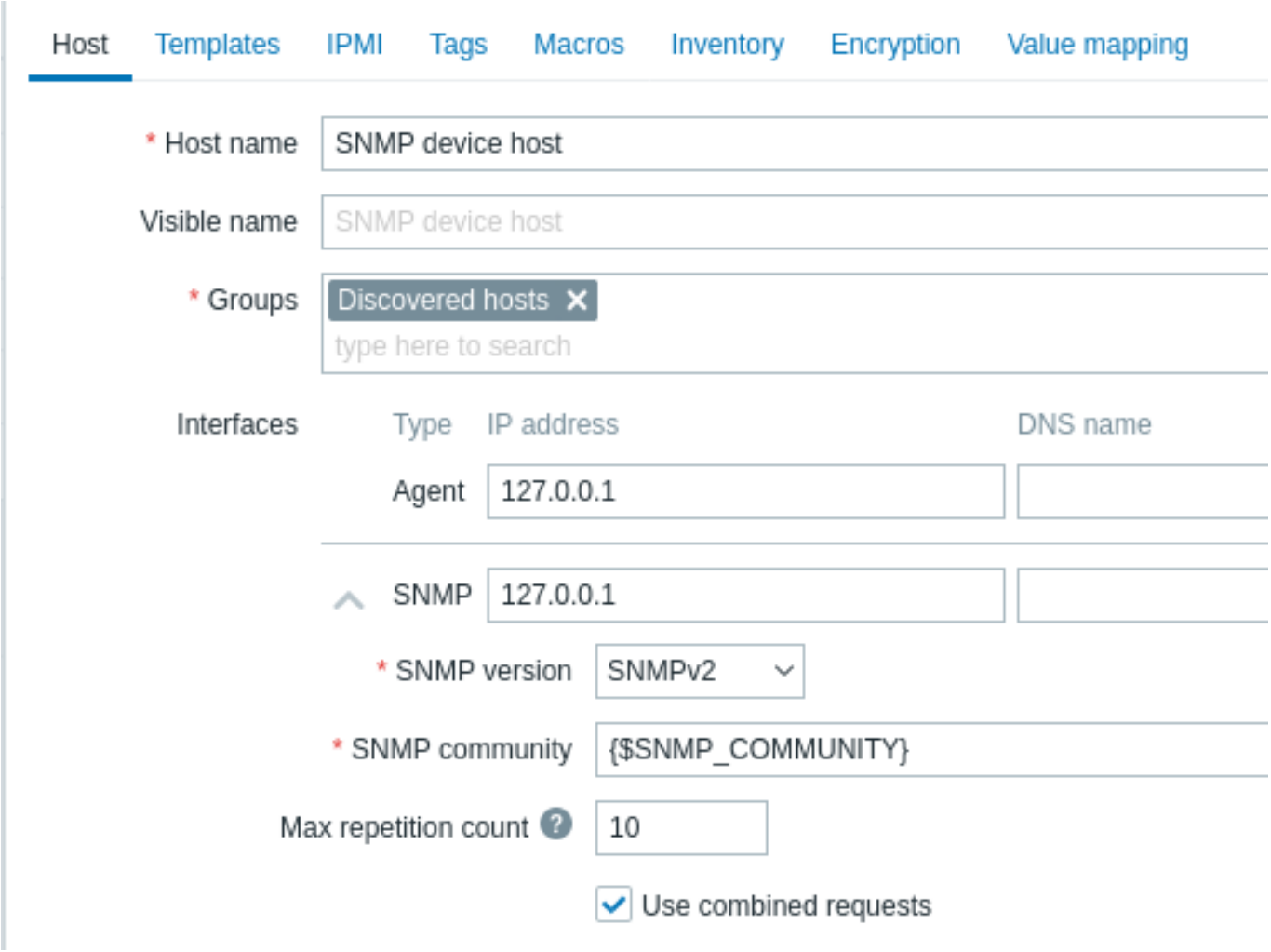
Auch hier ist die letzte Zahl in der OID die Portnummer.

Note:
Einige der am häufigsten verwendeten SNMP-OIDs werden von Zabbix automatisch in eine numerische Darstellung übersetzt.

Im letzten obigen Beispiel ist der Werttyp ”Counter64”, was intern dem Typ ASN_COUNTER64 entspricht. Die vollständige Liste der unterstützten Typen lautet 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 in etwa ”Counter32”, ”Counter64”, ”UInteger32”, ”INTEGER”, ”Float”, ”Double”, ”Timeticks”, ”Gauge32”, ”IpAddress”, ”OCTET STRING”, ”OBJECT IDENTIFIER” in der Ausgabe von **snmpget**, können jedoch je nach Vorhandensein eines Anzeigehinweises auch als ”STRING”, ”Hex-STRING”, ”OID” und anders dargestellt werden.

Schritt 2

Erstellen Sie einen Host, der einem Gerät entspricht.



Fügen Sie eine SNMP-Schnittstelle für den Host hinzu:

- Geben Sie die IP-Adresse/den DNS-Namen und die Portnummer ein.
- Wählen Sie die *SNMP-Version* aus der Dropdown-Liste aus.
- Fügen Sie je nach ausgewählter SNMP-Version Schnittstellen-Zugangsdaten hinzu:
  - SNMPv1, v2 erfordern nur die Community (normalerweise ’public’).
  - SNMPv3 erfordert spezifischere Optionen (siehe unten).
- Geben Sie den Wert für die maximale Wiederholung (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 hoch eingestellter Wert zu einem Timeout bei der Prüfung des SNMP-Agenten führen kann.
- Aktivieren Sie das Kontrollkästchen *Kombinierte Anfragen verwenden*, um die kombinierte Verarbeitung von SNMP-Anfragen zu erlauben (nicht bezogen auf native SNMP-Bulk-Anfragen ”walk” und ”get”).

SNMPv3-Parameter	Beschreibung
Kontextname	Geben Sie den Kontextnamen ein, um den Datenpunkt im SNMP-Subnetz zu identifizieren. Benutzermakros werden in diesem Feld aufgelöst.
Sicherheitsname	Geben Sie den Sicherheitsnamen ein. Benutzermakros werden in diesem Feld aufgelöst.

SNMPv3-Parameter	Beschreibung
Sicherheitsstufe	Wählen Sie die Sicherheitsstufe aus: **noAuthNoPriv** - es werden weder Authentifizierungs- noch Datenschutzprotokolle verwendet **AuthNoPriv** - das Authentifizierungsprotokoll wird verwendet, das Datenschutzprotokoll nicht **AuthPriv** - sowohl Authentifizierungs- als auch Datenschutzprotokolle werden verwendet
Authentifizierungsprotokoll	Wählen Sie das Authentifizierungsprotokoll aus - *MD5*, *SHA1*; mit net-snmp 5.8 und neuer *SHA224*, *SHA256*, *SHA384* oder *SHA512*.
Authentifizierungs-Passphrase	Geben Sie die Authentifizierungs-Passphrase ein. Benutzermakros werden in diesem Feld aufgelöst.
Datenschutzprotokoll	Wählen Sie das Datenschutzprotokoll aus - *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) oder *AES256C* (Cisco). Siehe Hinweise zur Unterstützung von Datenschutzprotokollen
Datenschutz-Passphrase	Geben Sie die Datenschutz-Passphrase ein. Benutzermakros werden in diesem Feld aufgelöst.

Bei falschen SNMPv3-Zugangsdaten (Sicherheitsname, Authentifizierungsprotokoll/-Passphrase, Datenschutzprotokoll):

- Zabbix erhält von net-snmp einen ERROR, außer bei einer falschen *Datenschutz-Passphrase*; in diesem Fall erhält Zabbix von net-snmp einen TIMEOUT-Fehler.
- Die Verfügbarkeit der SNMP-Schnittstelle wechselt auf Rot (nicht verfügbar).

Note:
Änderungen an *Authentifizierungsprotokoll*, *Authentifizierungs-Passphrase*, *Datenschutzprotokoll* oder *Datenschutz-Passphrase*, die ohne Änderung des *Sicherheitsnamens* vorgenommen werden, werden normalerweise automatisch angewendet, wenn die entsprechende SNMPv3-Schnittstelle in Zabbix aktualisiert wird. In Fällen, in denen auch der *Sicherheitsname* geändert wird, werden alle Parameter sofort aktualisiert.

Sie können eine der bereitgestellten SNMP-Vorlagen verwenden, die automatisch eine Reihe von Datenpunkten hinzufügt. Vergewissern Sie sich vor der Verwendung einer Vorlage, dass sie mit dem Host kompatibel ist.

Klicken Sie auf *Hinzufügen*, um den Host zu speichern.

Unterstützung von Privacy-Protokollen

Abhängig von Ihrem Betriebssystem und der net-snmp-Konfiguration sind einige Privacy-Protokolle möglicherweise nicht verfügbar:

- Auf einigen neueren Betriebssystemen (zum Beispiel RHEL9) wurde die Unterstützung von DES für das net-snmp-Paket eingestellt.
- Die Verschlüsselungsprotokolle AES192 und stärker werden auf Betriebssystemen älter als RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04 und 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 standardmäßig nicht 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 dann den Zabbix Server neu, wobei Sie die Option `--with-net-snmp=/home/user/y` angeben.

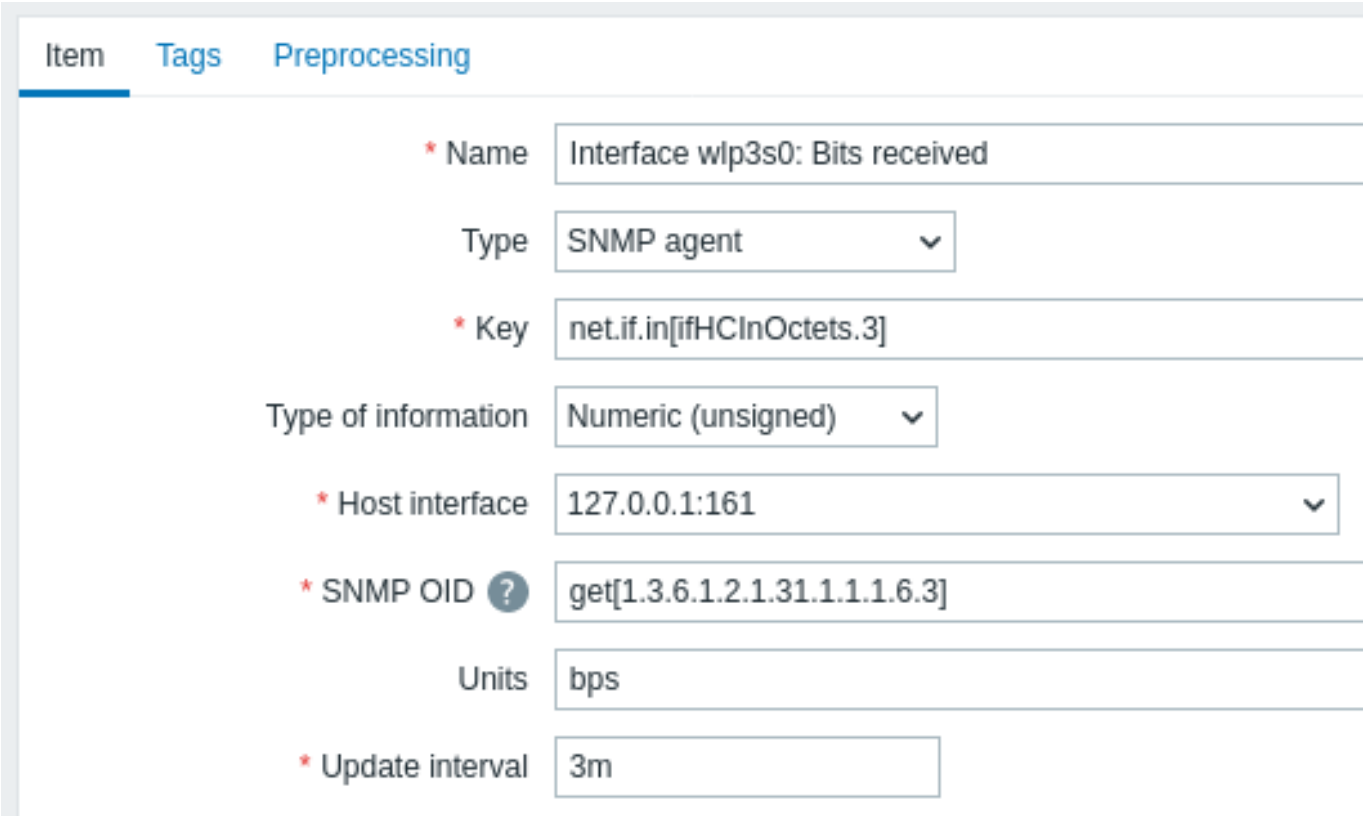
Schritt 3

Erstellen Sie einen Datenpunkt für die Überwachung.

Gehen Sie nun zurück zu Zabbix und klicken Sie bei dem 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 von SNMP-Datenpunkten, die Ihrem

Host zugeordnet sind, oder nur eine leere Liste. Wir gehen davon aus, dass Sie den Datenpunkt selbst mithilfe 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.
Type	Wählen Sie hier **SNMP agent** aus.
Key	Geben Sie einen aussagekräftigen Schlüssel ein.
Host interface	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-Wert(e) einzugeben: **walk[OID1,OID2,...]** – ruft einen Teilbaum von Werten ab. Zum 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 Datenpunkt-Konfiguration festgelegt werden. Ziehen Sie in Betracht, einen niedrigen Timeout-Wert festzulegen, um lange Verzögerungen zu vermeiden, falls das Gerät nicht erreichbar ist, da bis zu 5 Wiederholungsversuche durchgeführt werden, wenn frühere Versuche ein Timeout erreichen oder fehlschlagen (z. B. kann ein Timeout von 3 Sekunden zu einer Wartezeit von 15 Sekunden führen). Sie können diesen als Master-Datenpunkt verwenden, mit abhängigen Datenpunkten, die Daten mithilfe der Vorverarbeitung aus dem Master 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 zurückgibt, wird versucht, ohne Bulk-Anfrage einen einzelnen Datensatz 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, d. h. `walk[ifDescr,ifType,ifPhysAddress]`, dann ist die Ausgabe eine verkettete Liste. GetBulk-Anfragen werden mit SNMPv2- und v3-Schnittstellen verwendet, GetNext bei SNMPv1-Schnittstellen; die maximale Anzahl von Wiederholungen für Bulk-Anfragen wird auf Schnittstellenebene konfiguriert. Der Parameter für maximale Wiederholungen beeinflusst Bulk-Anfragen, indem er die maximale Anzahl von OIDs bestimmt, 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 asynchron einen *einzelnen* Wert ab. Zum Beispiel: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Die Timeout-Einstellungen für diesen Datenpunkt können im Formular Datenpunkt-Konfiguration festgelegt werden. Ziehen Sie in Betracht, einen niedrigen Timeout-Wert festzulegen, um lange Verzögerungen zu vermeiden, falls das Gerät nicht erreichbar ist, da bis zu 5 Wiederholungsversuche durchgeführt werden, wenn frühere Versuche ein Timeout erreichen oder fehlschlagen (z. B. kann ein Timeout von 3 Sekunden 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. Zum Beispiel: `1.3.6.1.2.1.31.1.1.1.6.3`. Bei dieser Option entspricht das Timeout der Datenpunkt-Prüfung dem Wert, der in der Konfigurationsdatei des Server festgelegt ist. Es wird **empfohlen**, Datenpunkte vom Typ `walk[OID]` und `get[OID]` für eine bessere Leistung zu verwenden. Alle Datenpunkte vom Typ `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 asynchroner SNMP-Poller wird durch den Parameter StartSNMPPollers definiert. Beachten Sie, dass bei Netzwerkverkehrsstatistiken, die mit einer der Methoden zurückgegeben werden, im Reiter *Vorverarbeitung* ein Schritt *Änderung pro Sekunde* hinzugefügt werden muss; andernfalls erhalten Sie vom SNMP-Gerät den kumulativen Wert anstelle der letzten Änderung.

Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Speichern Sie nun den Datenpunkt und gehen Sie für Ihre SNMP-Daten zu *Überwachung* > *Letzte Daten*.

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 Datenpunkt **walk[OID1,OID2,...]** ermöglicht die Nutzung nativer SNMP-Funktionalität für Bulk-Anfragen (GetBulkRequest-PDUs), die in den SNMP-Versionen 2/3 verfügbar ist.

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 sammelt, zusammen mit abhängigen Datenpunkten, die die Antwort bei Bedarf mittels Vorverarbeitung auswerten.

Beachten Sie, dass die Verwendung nativer SNMP-Bulk-Anfragen nicht mit der Option zum Kombinieren von SNMP-Anfragen zusammenhängt, was eine Zabbix-eigene Methode zum Zusammenfassen mehrerer SNMP-Anfragen ist (siehe nächsten Abschnitt).

Für SNMP-Bulk-Datenpunkte erfolgen bis zu fünf Wiederholungsversuche, um einen Fehler zu vermeiden, falls eines der Pakete verloren geht. Das Timeout für SNMP-Datenpunkte mit `get` und `walk` (festgelegt im Formular Datenpunktkonfiguration) wird für eine gesamte Sitzung gesetzt. Das Timeout wird unabhängig davon angewendet, ob die Daten vollständig abgerufen werden; wenn Daten nur teilweise empfangen werden (zum Beispiel wenn Daten erfolgreich nur für eine von mehreren OIDs erfasst werden), wird der Datenpunkt mit der Meldung „Only partial data received“ nicht unterstützt. Wird das Timeout erreicht, erfolgt ein Wiederholungsversuch, das Timeout wird zurückgesetzt und die letzte Anfrage wird erneut gesendet, sodass die Sitzung ab der letzten Anfrage fortgesetzt werden kann, wenn 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 5 Wiederholungsversuche durchgeführt werden, wenn frühere Versuche ein Timeout erreichen oder fehlschlagen (z. B. kann ein Timeout von 3 Sekunden zu einer Wartezeit von 15 Sekunden führen).

Interne Funktionsweise der kombinierten Verarbeitung

Zabbix Server und Proxy können SNMP-Geräte mit 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 so eingeplant, dass sie gleichzeitig abgefragt werden. Die ersten beiden Datenpunkt-Typen werden von Pollern stapelweise mit höchstens 128 Datenpunkten verarbeitet, während Low-Level-Discovery-Regeln wie bisher einzeln verarbeitet werden.

Auf niedrigerer Ebene gibt es zwei Arten von Operationen zur Abfrage 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 bei SNMPv1 eine GetNextRequest-PDU verwendet, und bei SNMPv2 und SNMPv3 wird GetBulkRequest mit einem Feld „max-repetitions“ von höchstens 128 verwendet.

Daher ergeben sich die Vorteile der kombinierten Verarbeitung für jeden SNMP-Datenpunkt-Typ wie folgt:

- reguläre SNMP-Datenpunkte profitieren von Verbesserungen beim „Abrufen“;
- SNMP-Datenpunkte mit dynamischen Indizes profitieren sowohl von Verbesserungen beim „Abrufen“ als auch beim „Durchlaufen“: „Abrufen“ wird zur Indexverifizierung verwendet und „Durchlaufen“ zum Aufbau des Cache;
- SNMP-Low-Level-Discovery-Regeln profitieren von Verbesserungen beim „Durchlaufen“.

Es gibt jedoch ein technisches Problem: Nicht alle Geräte sind in der Lage, 128 Werte pro Anfrage zurückzugeben. Einige liefern immer eine korrekte Antwort, andere antworten jedoch entweder mit einem Fehler „tooBig(1)“ oder antworten überhaupt nicht mehr, sobald die potenzielle Antwort eine bestimmte Grenze überschreitet.

Um eine optimale Anzahl von Objekten zu finden, die für ein bestimmtes Gerät abgefragt werden kann, verwendet Zabbix die folgende Strategie. Es beginnt vorsichtig mit der Abfrage von 1 Wert in einer Anfrage. Ist dies erfolgreich, werden 2 Werte in einer Anfrage abgefragt. Ist auch dies erfolgreich, werden 3 Werte in einer Anfrage abgefragt, und es wird auf ähnliche Weise fortgefahren, indem die Anzahl der abgefragten Objekte mit 1,5 multipliziert wird, was zu der folgenden Folge von Anfragegrößen führt: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Sobald ein Gerät jedoch keine korrekte Antwort mehr liefert (zum Beispiel bei 42 Variablen), tut Zabbix zwei Dinge.

Erstens halbiert es für den aktuellen Datenpunkt-Stapel die Anzahl der Objekte in einer einzelnen Anfrage und fragt 21 Variablen ab. Wenn das Gerät erreichbar ist, sollte die Abfrage in den allermeisten Fällen funktionieren, da bekannt ist, dass 28 Variablen funktionierten und 21 deutlich darunter liegt. Falls dies jedoch ebenfalls fehlschlägt, greift Zabbix auf die Abfrage einzelner Werte nacheinander zurück. Falls es auch dann noch fehlschlägt, antwortet das Gerät definitiv nicht, und die Anfragegröße ist nicht das Problem.

Zweitens beginnt Zabbix für nachfolgende Datenpunkt-Stapel mit der zuletzt erfolgreichen Anzahl von Variablen (28 in unserem Beispiel) und erhöht die Anfragegrößen dann jeweils um 1, bis die Grenze erreicht ist. Wenn beispielsweise die größte Antwortgröße 32 Variablen beträgt, 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 Anfragen bei dieser Variablenanzahl fehlschlagen, kann das zwei Ursachen haben. Die genauen Kriterien, die ein Gerät zur Begrenzung der Antwortgröße verwendet, sind nicht bekannt, aber wir versuchen, dies anhand der Anzahl der Variablen anzunähern. Die erste Möglichkeit ist also, dass diese Variablenanzahl im allgemeinen Fall ungefähr an der tatsächlichen Grenze der Antwortgröße des Geräts liegt: Manchmal liegt die Antwort unter der Grenze, manchmal darüber. Die zweite Möglichkeit ist, dass einfach ein UDP-Paket in eine der beiden Richtungen verloren gegangen ist. Aus diesen Gründen reduziert Zabbix bei einer fehlgeschlagenen Abfrage die maximale Anzahl von Variablen, um tiefer in den stabilen Bereich des Geräts zu gelangen, jedoch nur bis zu zwei Mal.

Im obigen Beispiel reduziert Zabbix die Anzahl auf 31, wenn eine Abfrage mit 32 Variablen fehlschlägt. Falls auch dies fehlschlägt, reduziert Zabbix die Anzahl auf 30. Zabbix reduziert die Anzahl jedoch nicht unter 30, da dann angenommen wird, dass weitere Fehler eher auf verlorene UDP-Pakete als auf die Grenze des Geräts zurückzuführen sind.

Wenn ein Gerät kombinierte Anfragen jedoch aus anderen Gründen nicht korrekt verarbeiten kann und die oben beschriebene Heuristik nicht funktioniert, gibt es für jede Schnittstelle die Einstellung „Use combined requests“, mit der kombinierte Anfragen für dieses Gerät deaktiviert werden können.

Wenn kombinierte Anfragen partielle oder fehlerhafte Antworten verursachen, die zu falschen Berechnungen pro Sekunde (Delta) führen (zum Beispiel scheinbare Spitzen in Schnittstellenzählern), deaktivieren Sie *Use combined requests* für die betroffene Schnittstelle, um separate Abfragen pro Datenpunkt zu erzwingen; dies verhindert häufig falsche Spitzen. Alternativ können Sie asynchrone `get[]`- oder `walk[]`-Datenpunkte verwenden, die asynchron ausgeführt werden und nicht dem schnittstellenbezogenen Batching von *Use combined requests* unterliegen — sie können anstelle älterer synchroner OID-Prüfungen verwendet werden, um Probleme im Zusammenhang mit kombinierten Anfragen zu vermeiden. Achten Sie auf Server-/Proxy-Protokolleinträge ähnlich dem im Abschnitt Overview gezeigten Eintrag, um betroffene Geräte zu identifizieren.

Wenn die Schnittstelle außerdem häufig nicht verfügbar wird, kann es erforderlich sein, den Parameter `UnavailableDelay` in den Konfigurationsdateien von Zabbix server oder Zabbix proxy zu erhöhen, um die Häufigkeit der Anfragen zu verringern. Datenpunkte können auf „nicht unterstützt“ gesetzt werden, wenn während der Discovery oder bei OID-Walks nur partielle Daten empfangen werden.

1 Dynamische Indizes

Übersicht

Auch wenn Sie die erforderliche Indexnummer (zum Beispiel die einer Netzwerk- schnittstelle) 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 einer Änderung der Indexnummer berücksichtigt.

Wenn Sie beispielsweise den Indexwert ermitteln 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

Für OID wird eine spezielle Syntax 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 abgefragt, um den Indexwert zu erhalten, der der Zeichenfolge entspricht.
zu suchende Zeichenfolge	Die Zeichenfolge, die bei der Suche für eine exakte Übereinstimmung mit einem Wert 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 dynamischer Index-Datenpunkt angefordert wird, ruft Zabbix die gesamte SNMP-Tabelle unter der Basis-OID für den Index ab und speichert sie im Cache, selbst wenn eine Übereinstimmung früher gefunden würde. Dies geschieht für den Fall, dass sich ein anderer Datenpunkt später auf dieselbe Basis-OID bezieht – Zabbix würde den Index im Cache nachschlagen, 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 Index-SNMP-Tabelle erneut.

2 Spezielle 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 dem/den physischen Verbindungsprotokoll(en) unmittelbar „unterhalb“ der Netzwerkschicht im Protokollstapel.
ifMtu	1.3.6.1.2.1.2.2.1.4	Die Größe des größten Datagramms, das über die 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 Bit 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 Subnetz-Unicast-Pakete, die an ein Protokoll einer höheren Schicht übergeben wurden.
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 Protokoll einer höheren Schicht übergeben wurden.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Die Anzahl der eingehenden Pakete, die zum Verwerfen ausgewählt wurden, obwohl keine Fehler erkannt wurden, um zu verhindern, dass sie an ein Protokoll einer höheren Schicht übergeben werden können. Ein möglicher Grund für das Verwerfen eines solchen Pakets könnte das Freigeben von Pufferspeicher sein.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Die Anzahl der eingehenden Pakete, die Fehler enthielten, welche ihre Übergabe an ein Protokoll einer höheren Schicht verhinderten.
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 über die Schnittstelle gesendeten Oktette, einschließlich Framing-Zeichen.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Die Gesamtzahl der Pakete, deren Übertragung von Protokollen höherer Schichten angefordert wurde und die auf dieser Unterschicht nicht an eine Multicast- oder Broadcast-Adresse adressiert waren, einschließlich derjenigen, die verworfen oder nicht gesendet wurden.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Die Gesamtzahl der Pakete, deren Übertragung von Protokollen höherer Schichten angefordert wurde und die auf dieser Unterschicht an eine Multicast- oder Broadcast-Adresse adressiert waren, einschließlich derjenigen, die verworfen oder nicht gesendet wurden.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Die Anzahl der ausgehenden Pakete, die zum Verwerfen ausgewählt 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 Pufferspeicher 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 Warteschlange für ausgehende Pakete (in Paketen).

3 MIB-Dateien

Einführung

MIB steht für Management Information Base. MIB-Dateien ermöglichen die Verwendung einer textuellen Darstellung einer OID (Object Identifier). Bei der Überwachung von SNMP-Geräten mit Zabbix können rohe OIDs verwendet werden, wenn Sie jedoch lieber die textuelle Darstellung verwenden möchten, 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.

3 SNMP-Traps

Übersicht

Der Empfang von SNMP-Traps ist das Gegenteil der Abfrage 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 bzw. „abgefangen".

Üblicherweise werden Traps bei einer Zustandsänderung gesendet, und der Agent verbindet sich mit dem Server über Port 162 (im Gegensatz zu Port 161 auf der Agent- Seite, der für Abfragen verwendet wird). Durch die Verwendung von Traps können kurze Probleme erkannt werden, die innerhalb des Abfrageintervalls auftreten und von den Abfragedaten möglicherweise übersehen werden.

Der Empfang von SNMP-Traps in Zabbix ist für die Zusammenarbeit mit **snmptrapd** und einem der Mechanismen zur Übergabe 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 Verwendung der Bash-Skript-Lösung, da Perl und SNMPTT in modernen Distributionen oft fehlen und eine komplexere Konfiguration erfordern. Diese Lösung verwendet jedoch ein als `traphandle` konfiguriertes Skript. 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 Empfang eines Traps:

1. `snmptrapd` empfängt einen Trap
2. `snmptrapd` übergibt den Trap an das Empfängerskript (Bash, Perl) oder an SNMPTT
3. Der Empfänger analysiert, formatiert und schreibt den Trap in eine Datei
4. Der Zabbix-SNMP-Trapper liest und analysiert die Trap-Datei
5. Für jeden Trap findet Zabbix alle „SNMP-Trapper"-Datenpunkte mit Host- Schnittstellen, die mit der Adresse des empfangenen Traps übereinstimmen. Beachten Sie, dass bei der Zuordnung nur das in der Host-Schnittstelle ausgewählte „IP" oder „DNS" verwendet wird.
6. Für jeden gefundenen Datenpunkt wird der Trap mit dem regulären Ausdruck in `snmptrap[regexp]` verglichen. Der Trap wird als Wert für **alle** übereinstimmenden Datenpunkte gesetzt. Wenn kein übereinstimmender Datenpunkt gefunden wird und es einen `snmptrap.fallback`-Datenpunkt gibt, 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 > General > Other 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.

Konfiguration von SNMP-Traps

Dieser Datenpunkttyp erfordert die folgende Konfiguration im Frontend.

1. Erstellen Sie eine SNMP-Schnittstelle für Ihren Host

- Erstellen/bearbeiten Sie unter *Datenerfassung* > *Hosts* den Host und fügen Sie im Feld *Schnittstellen* den Schnittstellentyp „SNMP" hinzu, wobei Sie die IP- oder DNS-Adresse angeben.

Die Adresse aus jedem empfangenen Trap wird mit den IP- und DNS-Adressen aller SNMP-Schnittstellen verglichen, um die entsprechenden Hosts zu finden.

2. Konfigurieren Sie den Datenpunkt

- Erstellen/bearbeiten Sie unter *Datenerfassung* > *Hosts* den erforderlichen Datenpunkt.
- Verwenden Sie im Feld *Key* einen der folgenden SNMP-Trap-Keys:

Key		
Beschreibung	Rückgabewert	Kommentare
snmptrap[regexp] Erfasst alle SNMP-Traps, die dem in **regexp** angegebenen regulären Ausdruck entsprechen. Wenn regexp nicht angegeben ist, wird jeder Trap erfasst.	SNMP-Trap	Dieser Datenpunkt kann nur für SNMP-Schnittstellen festgelegt werden. Benutzermakros und globale reguläre Ausdrücke werden im Parameter dieses Datenpunkt-Keys unterstützt.
snmptrap.fallback Erfasst alle SNMP-Traps, die von keinem der Datenpunkte snmptrap[] für diese Schnittstelle erfasst wurden.	SNMP-Trap	Dieser Datenpunkt kann nur für SNMP-Schnittstellen festgelegt werden.

Note:
Der Abgleich mehrzeiliger regulärer Ausdrücke wird derzeit nicht unterstützt.

- Setzen Sie *Art der Informationen* auf „Log“, damit die Zeitstempel geparst werden. Andere Formate wie „Numerisch“ sind ebenfalls zulässig, erfordern jedoch möglicherweise einen benutzerdefinierten Trap-Handler.

Einrichten der SNMP-Trap-Überwachung

Konfiguration 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 verweist, in die von SNMPTT oder einem Bash-/Perl-Trap-Empfänger geschrieben wird. Bearbeiten Sie dazu 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.

Konfiguration des Bash-Trap-Empfängers

Voraussetzungen: nur snmptrapd.

Ein Bash-Trap-Empfänger-Skript kann verwendet werden, um Traps aus snmptrapd über die Trapper-Datei an den Zabbix Server weiterzuleiten. Um ihn zu konfigurieren, fügen Sie die Option `traphandle` zur snmptrapd-Konfigurationsdatei (`snmptrapd.conf`) hinzu, siehe Beispiel.

Note:
snmptrapd muss möglicherweise neu gestartet werden, damit Änderungen an seiner Konfiguration übernommen werden.

Konfiguration des Perl-Trap-Empfängers

Voraussetzungen: Perl, Net-SNMP kompiliert mit --enable-embedded-perl (seit Net-SNMP 5.4 standardmäßig aktiviert)

Ein Perl-Trap-Empfänger (siehe misc/snmptrap/zabbix_trap_receiver.pl) kann verwendet werden, um Traps direkt von snmptrapd an den Zabbix Server weiterzuleiten. Zur Konfiguration:

- Fügen Sie das Perl-Skript zur snmptrapd-Konfigurationsdatei (snmptrapd.conf) hinzu, z. B.:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- Konfigurieren Sie den Empfänger, 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 ähnlich den folgenden:


```
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 zu übergeben. Siehe Anweisungen zum Konfigurieren von SNMPTT.

Wenn SNMPTT für den Empfang der Traps konfiguriert ist, 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-/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; in RHEL 8.3 wieder hinzugefügt. Weitere Informationen finden Sie unter den bekannten Problemen.

Formatieren Sie nun die Traps so, dass Zabbix sie erkennen kann (bearbeiten Sie snmptt.conf):

1. Jede FORMAT-Anweisung sollte mit ”ZBXTRAP [address]” beginnen, wobei [address] mit den IP- und DNS-Adressen von SNMP-Schnittstellen in Zabbix verglichen wird. Zum Beispiel:

```
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 wird sie nicht erkennen können. Unbekannte Traps können durch Definieren eines allgemeinen Ereignisses in snmptt.conf verarbeitet werden:


```
EVENT general .* "General event" Normal
```

SNMP-Trap-Format

Alle angepassten Perl-Trap-Empfänger und SNMPTT-Trap-Konfigurationen müssen den Trap auf folgende Weise formatieren:

```
[timestamp] [der Trap, Teil 1] ZBXTRAP [address] [der Trap, Teil 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 finden

Beachten Sie, dass "ZBXTRAP" und "[address]" während der Verarbeitung aus der Nachricht entfernt werden. Wenn der Trap anders formatiert ist, könnte Zabbix die Traps 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 zum folgenden 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 eine niedrigere Grenze für die Dateigröße festlegen kann.

Log-Rotation

Zabbix stellt kein System für die Log-Rotation bereit – dies muss vom Benutzer übernommen werden. Die Log-Rotation sollte zunächst die alte Datei umbenennen und erst später löschen, damit keine Traps verloren gehen:

1. Zabbix öffnet die Trap-Datei an der zuletzt bekannten Position 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 Datei geöffnet ist, setzt Zabbix die letzte Position zurück und geht zu Schritt 1.
3. Zabbix liest die Daten aus der aktuell geöffneten Datei und setzt die neue Position.
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, wartet Zabbix 1 Sekunde und geht zurück zu Schritt 2.

Dateisystem

Aufgrund der Implementierung der Trap-Datei benötigt Zabbix ein Dateisystem, das Inodes unterstützt, 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 wird, und legen Sie die Trap-Datei fest. Fügen Sie zu `zabbix_server.conf` Folgendes 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/Dock
```

Passen Sie bei Bedarf die Variable ZABBIX_TRAPS_FILE im Skript an. Um den Standardwert zu verwenden, erstellen Sie zuerst das übergeordnete Verzeichnis:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Fügen Sie Folgendes zu `snmtrapd.conf` hinzu (siehe funktionierendes Beispiel)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

Note:
snmptrapd muss möglicherweise neu gestartet werden, damit Änderungen an seiner 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"]`
   Log-Zeitformat: yyyyMMdd.hhmmss

Beachten Sie, dass das Datums- und Zeitformat ISO 8601 verwendet wird.

5. Als Nächstes konfigurieren wir `snmptrapd` für die gewählte SNMP-Protokollversion und senden Test-Traps mit dem Dienstprogramm `snmptrap`.

SNMPv1, SNMPv2

Die Protokolle SNMPv1 und SNMPv2 basieren auf der Authentifizierung per "Community-String". Im folgenden Beispiel verwenden wir "secret" als Community-String. Er muss auf SNMP-Trap-Sendern auf denselben Wert gesetzt sein.

Bitte beachten Sie, dass SNMPv2 zwar in Produktionsumgebungen noch weit verbreitet ist, jedoch weder Verschlüsselung noch eine echte Authentifizierung des Senders bietet. Die Daten werden im Klartext gesendet, daher sollten diese Protokollversionen nur in sicheren Umgebungen wie privaten Netzwerken verwendet und niemals über öffentliche oder fremde Netzwerke eingesetzt werden.

SNMP Version 1 wird heutzutage praktisch nicht mehr verwendet, da sie keine 64-Bit-Zähler unterstützt und als veraltetes Protokoll gilt.

Um das Annehmen von SNMPv1- oder SNMPv2-Traps zu aktivieren, sollten Sie die folgende Zeile zu `snmptrapd.conf` hinzufügen. Ersetzen Sie "secret" durch den auf den SNMP-Trap-Sendern konfigurierten SNMP-Community-String:

```
authCommunity log,execute,net secret
```

Als Nächstes können wir mit `snmptrap` einen Test-Trap senden. In diesem Beispiel verwenden wir die gängige "link up"-OID:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 behebt die Sicherheitsprobleme von SNMPv1/v2 und bietet Authentifizierung sowie Verschlüsselung. Sie können MD5 oder mehrere SHA-Authentifizierungsmethoden sowie DES/mehrere AES-Varianten als Verschlüsselung verwenden.

Um das Annehmen von SNMPv3 zu aktivieren, fügen Sie die folgenden Zeilen zu `snmptrapd.conf` hinzu:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

Attention:
Beachten Sie bitte das Schlüsselwort "execute", das die Ausführung von Skripten für dieses Benutzersicherheitsmodell erlaubt.

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

Warning:
Wenn Sie starke Verschlüsselungsmethoden wie AES192 oder AES256 verwenden möchten, nutzen 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.

Verifizierung

In beiden Beispielen sehen Sie ähnliche Zeilen in Ihrer `/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 Datenpunktwert in Zabbix wird sein:

```
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)

4 IPMI-Prüfungen

Übersicht

Sie können den Zustand und die Verfügbarkeit von Intelligent Platform Management Interface (IPMI)-Geräten in Zabbix überwachen. Um IPMI-Prüfungen durchzuführen, muss der Zabbix-Server zunächst mit IPMI-Unterstützung konfiguriert werden.

IPMI ist eine standardisierte Schnittstelle für die Remote-Verwaltung (”lights-out” oder ”out-of-band”) von Computersystemen. Sie ermöglicht die Überwachung des Hardwarestatus direkt über sogenannte ”out-of-band”-Management-Karten, unabhängig vom Betriebssystem und davon, ob die Maschine 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 durch IPMI-Poller. Ein Host wird dabei immer nur von einem einzigen IPMI-Poller gleichzeitig abgefragt, wodurch die Anzahl offener Verbindungen zu BMC-Controllern reduziert wird. Daher kann die Anzahl der IPMI-Poller bedenkenlos erhöht werden, ohne eine Überlastung der BMC-Controller befürchten zu müssen. Der IPMI-Manager-Prozess wird automatisch gestartet, wenn mindestens ein IPMI-Poller gestartet wird.

Siehe auch bekannte Probleme für IPMI-Prüfungen.

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 IPMI-Abfrager gestartet werden. Daher funktionieren hinzugefügte IPMI-Datenpunkte nicht. Um dies zu ändern, öffnen Sie als root die Zabbix-Server-Konfigurationsdatei (zabbix_server.conf) und suchen Sie nach der folgenden Zeile:

```
# StartIPMIPollers=0
```

Entfernen Sie das Kommentarzeichen und setzen Sie die Anzahl der Abfrager beispielsweise auf 3, sodass die Zeile wie folgt aussieht:

```
StartIPMIPollers=3
```

Speichern Sie die Datei und starten Sie anschließend zabbix_server 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.

5 Einfache Prüfungen

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 sind 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 verlorener Pakete.
icmppingretry	Die Erreichbarkeit des Hosts per ICMP-Ping mit Wiederholungsversuchen.
icmppingsec	Die Antwortzeit des ICMP-Pings.
net.tcp.service	Prüft, ob ein Dienst läuft und TCP-Verbindungen akzeptiert.
net.tcp.service.perf	Prüft die Leistung eines TCP-Dienstes.
net.udp.service	Prüft, ob ein Dienst läuft 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 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.

Beispiel:

```
icmpping[,4] #Wenn mindestens eines der vier Pakete zurückgegeben wird, liefert der Datenpunkt 1.
```

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 in 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 Konfigurationsdatei von Zabbix Server/Proxy an (oder im Parameter *Fping6Location* für die Verwendung von IPv6-Adressen).

fping sollte durch den Benutzer ausführbar sein, unter dem Zabbix Server/Proxy ausgeführt wird, und dieser Benutzer sollte über ausreichende Rechte verfügen.

Siehe auch: Bekannte Probleme zur Verarbeitung einfacher Prüfungen mit fping-Versionen unter 3.10.

Standardwerte

Standardwerte, Grenzwerte und Beschreibung der Werte für ICMP-Prüfparameter:

Parameter	Einheit	Beschreibung	Flag von Fping	Standardwerte gesetzt 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
timeout	Millisekunden	**fping v3.x** - Wartezeit nach dem Senden des letzten Pakets, beeinflusst durch das Flag *-C* **fping v4.x** - individuelles Timeout 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 weiteren Anfrage multipliziert wird	-B	1.5	1.0	1.0	5.0

Die Standardwerte können je nach Plattform und Version leicht abweichen.

Zusätzlich verwendet Zabbix die fping-Optionen *-i interval ms* (nicht zu verwechseln mit dem oben in der Tabelle genannten Datenpunkt-Parameter *interval*, 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 minimalen Wert in Millisekunden zu erkennen, den fping mit *-i* zulässt, indem 3 Werte ausprobiert werden: 0, 1 und 10. Der zuerst erfolgreiche Wert wird dann für nachfolgende ICMP-Prüfungen verwendet. Dieser Prozess wird von jedem ICMP pinger-Prozess einzeln durchgeführt.

Automatisch erkannte fping-Optionen werden jede Stunde verworfen und beim nächsten Versuch, eine ICMP-Prüfung durchzuführen, erneut erkannt. Setzen Sie DebugLevel>=4, um Details dieses Prozesses in der Protokolldatei des Server oder Proxy anzuzeigen.

Zabbix schreibt IP-Adressen, die mit einem der Schlüssel *icmpping** geprüft werden sollen, in eine temporäre Datei, die dann an fping übergeben wird. Wenn Datenpunkte unterschiedliche Schlüsselparameter haben, werden nur diejenigen mit identischen Schlüsselparametern in eine einzelne Datei geschrieben. Alle IP-Adressen, die in dieselbe Datei geschrieben werden, werden von fping parallel geprüft, sodass der Zabbix-ICMP-pinger-Prozess unabhängig von der Anzahl der IP-Adressen in der Datei eine feste Zeitspanne benötigt.

1 VMware-Monitoring-Datenpunktschlüssel

Die Liste der VMware-Monitoring-Datenpunktschlüssel wurde in den Abschnitt VMware-Monitoring verschoben.

6 Überwachung von Protokolldateien

Übersicht

Zabbix kann für die zentrale Überwachung und Analyse von Protokolldateien mit/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 folgende Voraussetzungen erfüllt sein:

- Zabbix Agent läuft auf dem Host
- Datenpunkt für die Protokollüberwachung ist eingerichtet

Attention:
Die Größenbeschränkung einer überwachten Protokolldatei hängt von der Unterstützung für große Dateien ab.

Konfiguration

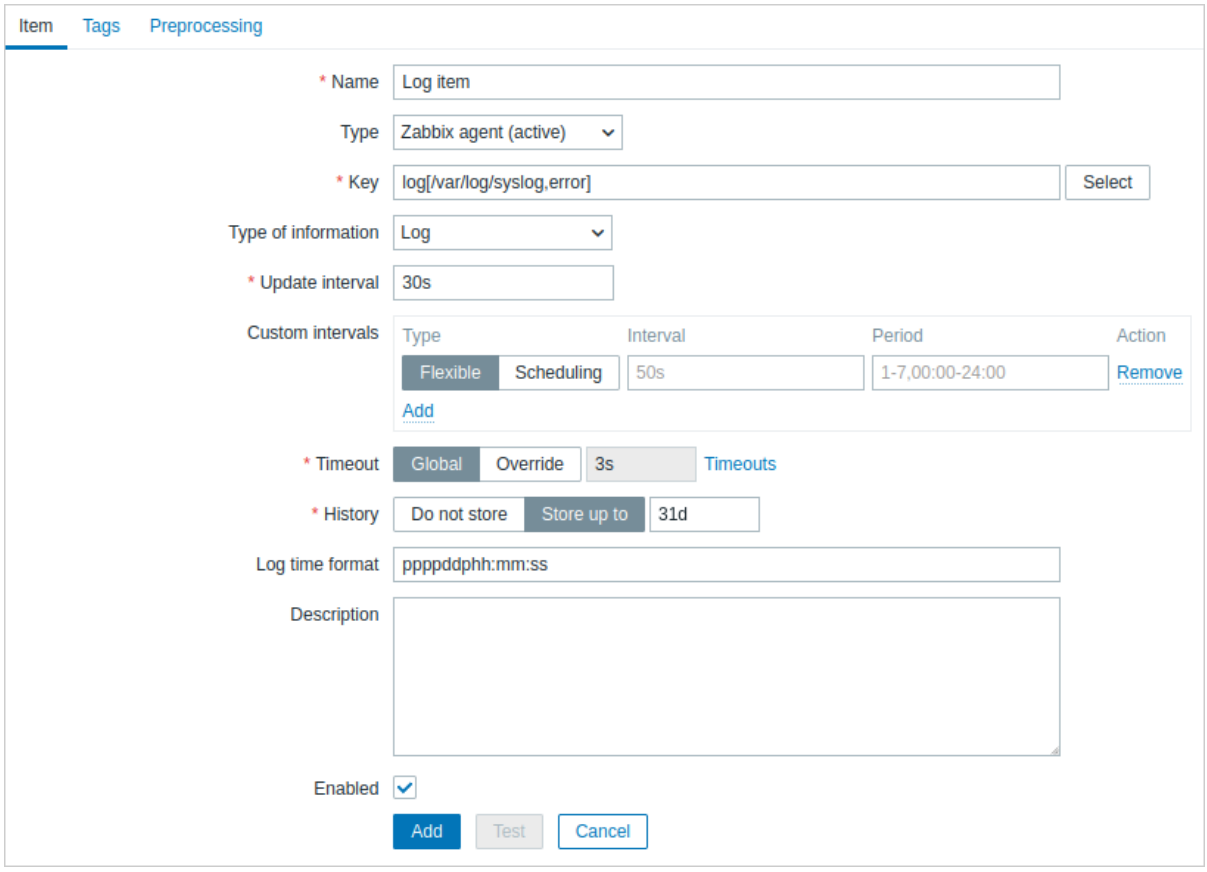
Agent-Parameter überprüfen

Stellen Sie sicher, dass in der Agent-Konfigurationsdatei Folgendes gilt:

- Der Parameter `Hostname` stimmt mit dem Host-Namen im Frontend überein.
- Im Parameter `ServerActive` sind Server für die Verarbeitung aktiver Prüfungen angegeben.

Konfiguration von Datenpunkten

Konfigurieren Sie einen Log-Überwachungs-Datenpunkt.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Speziell für Log-Überwachungs-Datenpunkte geben Sie Folgendes ein:

Type	Wählen Sie hier **Zabbix agent (active)** aus.

Key	Verwenden Sie einen der folgenden Datenpunktschlüssel: **log[]** oder **logrt[]**: Diese beiden Datenpunktschlüssel ermöglichen die Überwachung von Logs und das Filtern von Log-Einträgen anhand des Inhalts-RegExp, falls vorhanden. Zum Beispiel: `log[/var/log/syslog,error]`. Stellen Sie sicher, dass die Datei Leseberechtigungen für den Benutzer 'zabbix' hat, andernfalls wird der Datenpunktstatus auf 'unsupported' gesetzt. **log.count[]** oder **logrt.count[]**: Diese beiden Datenpunktschlüssel ermöglichen es, nur die Anzahl übereinstimmender Zeilen zurückzugeben. Im Abschnitt zu unterstützten Schlüsseln für Zabbix agent-Datenpunkte finden Sie Details zur Verwendung dieser Datenpunktschlüssel und ihrer Parameter.
Type of information	Automatisch vorausgefüllt: Für log[]- oder logrt[]-Datenpunkte - Log; Für log.count[]- oder logrt.count[]-Datenpunkte - `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.
Update interval (in sec)	Der Parameter definiert, wie oft Zabbix agent auf Änderungen in der Log-Datei prüft. Wenn Sie ihn auf 1 Sekunde setzen, erhalten Sie neue Einträge so schnell wie möglich.
Log time format	In diesem Feld können Sie optional das Muster zum Parsen des Zeitstempels einer Log-Zeile 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 Unix-Zeit auf 0 gesetzt, was dem 1. Januar 1970 entspricht. Betrachten Sie zum Beispiel die folgende Zeile aus der Zabbix agent-Log-Datei: " 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 Log-Zeitformat für diese Zeile wäre "ppppppp:yyyyMMdd:hhmmss". Beachten Sie, dass die Zeichen "p" und ":" Platzhalter sind und jedes beliebige Zeichen außer "yMdhms" sein können.

Wichtige Hinweise

- Der Server und der Agent speichern die Größe einer überwachten Protokolldatei und den Zeitpunkt der letzten Änderung (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 Byte 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 zur Nachverfolgung von Dateien verwendet werden können.
  - Unter Microsoft Windows ermittelt der Zabbix Agent den Typ des Dateisystems, auf dem sich die Protokolldateien befinden, und verwendet:
    * Auf NTFS-Dateisystemen 64-Bit-Dateiindizes.
    * Auf ReFS-Dateisystemen (erst 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 unter unsicheren Bedingungen sinnvoll vorzugehen, wenn die Rotation von Protokolldateien zu mehreren Protokolldateien mit derselben letzten Änderungszeit 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 gestoppt wird.
  - Ändern Sie nicht die Zeit der letzten Änderung einer Protokolldatei (zum Beispiel mit `touch`) und ersetzen Sie eine überwachte Protokolldatei nicht, indem Sie eine Datei zurück auf ihren ursprünglichen Namen kopieren (dadurch wird ein neuer Inode erzeugt). In beiden Fällen kann Zabbix die Datei als eine andere Datei behandeln und sie erneut von Anfang an lesen, was zu doppelten Warnmeldungen führen kann.
  - Wenn es mehrere passende Protokolldateien für den `logrt[]`-Datenpunkt gibt und der Zabbix Agent der neuesten

davon folgt und diese neueste Protokolldatei gelöscht wird, wird eine Warnmeldung `"there are no files matching "<regexp mask>" in "<directory>"` protokolliert. Der Zabbix Agent ignoriert Protokolldateien mit einer Änderungszeit, die kleiner ist als die neueste Änderungszeit, die der Agent für den geprüften `logrt[]`-Datenpunkt gesehen hat.

- Der Agent beginnt mit dem Lesen der Protokolldatei an der Stelle, an der er beim letzten Mal aufgehört hat.
- Die Anzahl der bereits analysierten Byte (der Größenzähler) und die Zeit der letzten Änderung (der Zeitzähler) werden in der Zabbix-Datenbank gespeichert und an den Agent gesendet, um sicherzustellen, dass der Agent in Fällen, in denen er gerade gestartet wurde oder Datenpunkte erhalten hat, die zuvor deaktiviert oder nicht unterstützt waren, die Protokolldatei ab dieser Stelle zu lesen beginnt. Wenn der Agent jedoch einen Größenzähler ungleich null vom Server erhalten hat, der `logrt[]`- oder `logrt.count[]`-Datenpunkt aber keine passenden Dateien finden kann, wird der Größenzähler auf 0 zurückgesetzt, damit ab dem Anfang analysiert wird, falls die Dateien später erscheinen.
- Immer wenn die Protokolldatei kleiner wird als der dem Agent bekannte Protokolldatei-Größenzähler, wird der Zähler auf null zurückgesetzt und der Agent beginnt unter Berücksichtigung des Zeitzählers, die Protokolldatei von Anfang an zu lesen.
- Wenn es im Verzeichnis mehrere passende Dateien mit derselben letzten Änderungszeit gibt, versucht der Agent, alle Protokolldateien mit derselben Änderungszeit korrekt zu analysieren und dabei zu vermeiden, Daten zu überspringen oder dieselben Daten zweimal zu analysieren, auch wenn dies nicht in allen Situationen garantiert werden kann. Der Agent geht von keinem bestimmten Schema für die Rotation von Protokolldateien aus und ermittelt auch keines. Wenn mehrere Protokolldateien mit derselben letzten Änderungszeit vorliegen, verarbeitet der Agent sie in lexikografisch absteigender Reihenfolge. Dadurch werden bei einigen Rotationsschemata die Protokolldateien in ihrer ursprünglichen Reihenfolge analysiert und gemeldet. Bei anderen Rotationsschemata wird die ursprüngliche Reihenfolge der Protokolldateien nicht eingehalten, was dazu führen kann, dass übereinstimmende Protokolldatei-Einträge in veränderter Reihenfolge gemeldet werden (dieses Problem tritt nicht auf, wenn Protokolldateien unterschiedliche letzte Änderungszeiten haben).
- Der Zabbix Agent verarbeitet neue Einträge einer Protokolldatei einmal pro *Update interval*-Sekunden.
- Der Zabbix Agent sendet nicht mehr als **maxlines** einer Protokolldatei pro Sekunde. 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 zum Beispiel ein `log[]`- oder `logrt[]`-Datenpunkt ein *Update interval* von 1 Sekunde hat, analysiert der Agent standardmäßig nicht mehr als 200 Protokolldatei-Einträge und sendet in einer Prüfung nicht mehr als 20 passende Einträge an den Zabbix Server. Durch Erhöhen von **MaxLinesPerSecond** in der Agent-Konfigurationsdatei oder durch Setzen des Parameters **maxlines** im Datenpunktschlüssel kann das Limit auf bis zu 10000 analysierte Protokolldatei-Einträge und 1000 an den Zabbix Server gesendete passende Einträge pro Prüfung erhöht werden. Wenn das *Update interval* auf 2 Sekunden gesetzt ist, werden die Grenzwerte für eine Prüfung 2-mal höher gesetzt als bei einem *Update interval* von 1 Sekunde.
- Zusätzlich sind log- und log.count-Werte immer auf 50 % der Größe des Agent-Sendepuffers begrenzt, selbst wenn sich keine Nicht-Log-Werte darin befinden. Damit die **maxlines**-Werte in einer Verbindung gesendet werden können (und nicht in mehreren Verbindungen), muss der Parameter BufferSize des Agent mindestens maxlines x 2 betragen. Der Zabbix Agent kann während der Erfassung von Protokollen Daten hochladen und dadurch den Puffer freigeben, während Zabbix Agent 2 die Protokollerfassung anhält, bis die Daten hochgeladen und der Puffer freigegeben ist; dies erfolgt asynchron.
- Wenn keine Log-Datenpunkte vorhanden sind, wird die gesamte Puffergröße des Agent für Nicht-Log-Werte verwendet. Wenn Log-Werte eingehen, ersetzen sie bei Bedarf ältere Nicht-Log-Werte, bis zu den vorgesehenen 50 %.
- Bei Protokolldatei-Einträgen, die länger als 256 kB sind, werden nur die ersten 256 kB mit dem regulären Ausdruck abgeglichen, der Rest des Eintrags wird ignoriert. Wird der Zabbix Agent jedoch gestoppt, während er einen langen Eintrag verarbeitet, geht der interne Zustand des Agent verloren und der lange Eintrag kann nach dem erneuten Start des Agent erneut und anders analysiert werden.
- Besonderer Hinweis zu \\ als Pfadtrennzeichen: Wenn file_format ”file\.log” ist, darf es kein Verzeichnis ”file” geben, da nicht eindeutig bestimmt werden kann, ob ”.” maskiert ist oder das erste Zeichen des Dateinamens darstellt.
- Reguläre Ausdrücke für `logrt` werden nur im Dateinamen unterstützt; reguläre Ausdrucksvergleiche für Verzeichnisse werden nicht unterstützt.
- Auf UNIX-Plattformen wird ein `logrt[]`-Datenpunkt zu NOTSUPPORTED, wenn ein Verzeichnis, in dem die Protokolldateien erwartet werden, nicht existiert.
- Unter Microsoft Windows wird der Datenpunkt nicht zu NOTSUPPORTED, wenn ein Verzeichnis nicht existiert (zum Beispiel wenn das Verzeichnis im Datenpunktschlüssel falsch geschrieben ist).
- Das Fehlen von Protokolldateien für einen `logrt[]`-Datenpunkt führt nicht dazu, dass er NOTSUPPORTED wird. Fehler beim Lesen von Protokolldateien für einen `logrt[]`-Datenpunkt werden als Warnungen in die Protokolldatei des Zabbix Agent geschrieben, führen aber nicht dazu, dass der Datenpunkt NOTSUPPORTED wird.
- Die Protokolldatei des Zabbix Agent kann hilfreich sein, um herauszufinden, warum ein `log[]`- oder `logrt[]`-Datenpunkt zu NOTSUPPORTED wurde. Zabbix kann seine eigene Agent-Protokolldatei überwachen, außer bei DebugLevel=4 oder DebugLevel=5.
- Die Suche nach einem Fragezeichen mit einem regulären Ausdruck, z. B. \?, kann zu falsch positiven Treffern 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 möchten wir aus einer Zieldatei nur den interessanten Wert extrahieren, anstatt die gesamte Zeile zurückzugeben, wenn eine Übereinstimmung mit einem regulären Ausdruck gefunden wird.

Log-Datenpunkte können gewünschte Werte aus übereinstimmenden Zeilen extrahieren. Dies wird durch den zusätzlichen Parameter **output** in `log`- und `logrt`-Datenpunkten erreicht.

Mit dem Parameter „output“ kann die „Capturing Group“ der Übereinstimmung angegeben werden, für die wir uns interessieren.

Zum Beispiel sollte

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

ermöglichen, die Anzahl der Einträge zurückzugeben, wie sie im folgenden Inhalt gefunden wird:

```
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 Capturing Group bezieht: **([0-9]+)**.

Durch die Möglichkeit, eine Zahl zu extrahieren und zurückzugeben, kann dieser Wert zur Definition 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 können.

> **Warning:**
> Die Angabe von ’maxdelay’ > 0 kann dazu führen, dass **wichtige Logdatei-Einträge ignoriert und Warnmeldungen verpasst werden**. Verwenden Sie ihn nur bei Bedarf und mit Vorsicht auf eigenes Risiko.

Standardmäßig verfolgen Datenpunkte für die Log-Überwachung alle neuen Zeilen, die in den Logdateien erscheinen. Es gibt jedoch Anwendungen, die in bestimmten Situationen beginnen, eine enorme Anzahl von Meldungen in ihre Logdateien zu schreiben. Wenn beispielsweise eine Datenbank oder ein DNS-Server nicht verfügbar ist, überfluten solche Anwendungen die Logdateien mit Tausenden nahezu identischer Fehlermeldungen, bis der normale Betrieb wiederhergestellt ist. Standardmäßig werden all diese Meldungen gewissenhaft analysiert und übereinstimmende Zeilen wie in den Datenpunkten `log` und `logrt` konfiguriert an den Server gesendet.

Der integrierte Schutz vor Überlastung besteht aus einem konfigurierbaren Parameter `maxlines` (schützt den Server vor zu vielen eingehenden übereinstimmenden Logzeilen) und einem Limit von 10*’maxlines’ (schützt CPU und I/O des Hosts vor einer Überlastung durch den Agent bei einer Prüfung). Dennoch gibt es 2 Probleme mit dem 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 von pro Sekunde analysierten Zeilen den neuesten Logeinträgen um Stunden hinterherhinken. Wahrscheinlich möchten Sie lieber früher über die aktuelle Situation in den Logdateien informiert werden, anstatt sich stundenlang durch alte Einträge zu arbeiten.

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 - wie viele Sekunden benötigt würden, um 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 der Logdatei, indem er darüber ”springt”** zu einer neuen geschätzten Position, 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 wird.

Die Tatsache, dass Logdatei-Zeilen übersprungen werden, wird in der Agent-Logdatei 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 nur ungefähr, weil der Agent nach dem ”Sprung” die Position in der Datei auf den Anfang einer Logzeile anpasst, die sich weiter hinten oder auch weiter vorne in der Datei befinden kann.

Je nachdem, wie sich die Wachstumsgeschwindigkeit im Vergleich zur Geschwindigkeit der Analyse der Logdatei verhält, sehen Sie möglicherweise keine ”Sprünge”, seltene oder häufige ”Sprünge”, große oder kleine ”Sprünge” oder sogar bei jeder Prüfung einen kleinen ”Sprung”. Schwankungen der Systemlast und Netzwerklatenz beeinflussen ebenfalls die Berechnung der Verzögerung und damit das Vorspringen, um den Parameter `maxdelay` einzuhalten.

Es wird nicht empfohlen, `maxdelay` < `update interval` festzulegen (dies kann zu häufigen kleinen ”Sprüngen” führen).

Hinweise zur Handhabung der Logdateirotation mit „copytruncate“

`logrt` mit der Option `copytruncate` geht davon aus, dass verschiedene Logdateien unterschiedliche Einträge enthalten (zumindest unterscheiden sich ihre Zeitstempel), daher sind die MD5-Prüfsummen der Anfangsblöcke (bis zu den ersten 512 Byte) unterschiedlich. Zwei Dateien mit denselben MD5-Prüfsummen der Anfangsblöcke bedeuten, dass eine davon das Original und die andere eine Kopie ist.

`logrt` mit der Option `copytruncate` bemüht sich, Kopien von Logdateien korrekt zu verarbeiten, ohne Duplikate zu melden. Es wird jedoch nicht empfohlen, mehrere Kopien von Logdateien mit demselben Zeitstempel zu erzeugen, die Logdateirotation häufiger als das Aktualisierungsintervall des logrt[]-Datenpunkts durchzuführen oder den Agent häufig neu zu starten. Der Agent versucht, all diese Situationen möglichst gut zu handhaben, aber gute Ergebnisse können nicht unter allen Umständen garantiert werden.

Hinweise zu persistenten Dateien für log*[]-Datenpunkte

Zweck persistenter Dateien

Wenn der Zabbix Agent gestartet wird, erhält er vom Zabbix Server oder Proxy eine Liste aktiver Prüfungen. Für log*[]-Metriken erhält er die verarbeitete Log-Größe und die Änderungszeit, um festzustellen, ab welcher Stelle die Überwachung der Log-Datei beginnen soll. Abhängig von der tatsächlichen Größe der Log-Datei und der vom Dateisystem gemeldeten Änderungszeit entscheidet der Agent, ob die Überwachung der Log-Datei ab der verarbeiteten Log-Größe fortgesetzt oder die Log-Datei von Anfang an erneut analysiert werden soll.

Ein laufender Agent verwaltet einen größeren Satz von Attributen, um alle überwachten Log-Dateien zwischen den Prüfungen nachzuverfolgen. Dieser In-Memory-Status geht verloren, wenn der Agent gestoppt wird.

Der neue optionale Parameter **persistent_dir** gibt ein Verzeichnis an, in dem dieser Status eines log[]-, log.count[]-, logrt[]- oder logrt.count[]-Datenpunkts in einer Datei gespeichert wird. Der Status des Log-Datenpunkts wird nach einem Neustart des Zabbix Agent aus der persistenten Datei wiederhergestellt.

Der primäre Anwendungsfall ist die Überwachung einer Log-Datei, die sich auf einem gespiegelten Dateisystem befindet. Bis zu einem bestimmten Zeitpunkt wird die Log-Datei auf beide Spiegel geschrieben. Dann werden die Spiegel getrennt. Auf der aktiven Kopie wächst die Log-Datei weiter und erhält neue Einträge. Der Zabbix Agent analysiert sie und sendet die verarbeitete Log-Größe und die Änderungszeit an den Server. Auf der passiven Kopie bleibt die Log-Datei unverändert und liegt deutlich hinter der aktiven Kopie zurück. Später werden das Betriebssystem und der Zabbix Agent von der passiven Kopie neu gestartet. Die verarbeitete Log-Größe und die Änderungszeit, die der Zabbix Agent vom Server erhält, sind für die Situation auf der passiven Kopie möglicherweise nicht gültig. Damit die Überwachung der Log-Datei an der Stelle fortgesetzt werden kann, an der der Agent zum Zeitpunkt der Trennung der Dateisystemspiegel aufgehört hat, stellt der Agent seinen Status aus der persistenten Datei wieder her.

Betrieb des Agent mit persistenter Datei

Beim Start weiß der Zabbix Agent nichts über persistente Dateien. Erst nach dem Empfang einer Liste aktiver Prüfungen vom Zabbix Server (Proxy) erkennt der Agent, dass einige Log-Datenpunkte durch persistente Dateien in den angegebenen Verzeichnissen unterstützt werden sollen.

Während des Betriebs des Agent werden die persistenten Dateien zum Schreiben geöffnet (mit fopen(filename, ”w”)) und mit den neuesten Daten überschrieben. Die Wahrscheinlichkeit, dass Daten persistenter Dateien verloren gehen, wenn das Überschreiben und eine Aufspaltung der Dateisystemspiegelung gleichzeitig stattfinden, ist sehr gering; es gibt dafür keine spezielle Behandlung. Auf das Schreiben in eine persistente Datei folgt KEINE erzwungene Synchronisierung auf das Speichermedium (fsync() wird nicht aufgerufen).

Das Überschreiben mit den neuesten Daten erfolgt nach erfolgreicher Übermittlung des passenden Logdatei-Datensatzes oder der Metadaten (verarbeitete Loggröße und Änderungszeit) an den Zabbix Server. Dies kann so häufig wie bei jeder Datenpunkt-Prüfung geschehen, wenn sich die Logdatei ständig ändert.

Beim Herunterfahren des Agent werden keine besonderen Maßnahmen durchgeführt.

Nach Erhalt 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.

Die Entfernung erfolgt mit einer Verzögerung von 24 Stunden, da Log-Dateien im Status NOTSUPPORTED nicht in der Liste aktiver Prüfungen enthalten sind, später jedoch wieder SUPPORTED werden können und ihre persistenten Dateien dann nützlich sind.

Wird der Agent gestoppt, bevor die 24 Stunden abgelaufen sind, werden die veralteten Dateien nicht gelöscht, da der Zabbix Agent keine Informationen über ihren Speicherort mehr vom Zabbix Server erhält.

Warning:
Wird **persistent_dir** eines Log-Datenpunkts wieder auf den alten **persistent_dir**-Speicherort zurückgesetzt, während der Agent gestoppt ist, ohne dass die alte persistente Datei vom Benutzer gelöscht wurde, führt dies dazu, dass der Agent-Status aus der alten persistenten Datei wiederhergestellt wird, was zu verpassten Meldungen oder Fehlalarmen führt.

Benennung und Speicherort persistenter Dateien

Der Zabbix Agent unterscheidet aktive Prüfungen anhand ihrer Schlüssel. Zum Beispiel 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, führt dies dazu, dass der Datenpunkt logrt[/home/zabbix/test.log„,10] aus der Liste aktiver Prüfungen des Agent gelöscht und der Datenpunkt logrt[/home/zabbix/test.log„,20] erstellt wird (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 zur Verringerung der Kollisionswahrscheinlichkeit die Länge des Datenpunktschlüssels angehängt wird. Zum Beispiel wird der Status des Datenpunkts logrt[/home/zabbix50/test.log„„„„„/home/zabbix50/agent_private] in der persistenten Datei c963ade4008054813bbc0a650bb8e09266 gespeichert.

Mehrere Log-Datenpunkte können denselben Wert für **persistent_dir** verwenden.

persistent_dir wird unter Berücksichtigung spezifischer Dateisystem-Layouts, Einhängepunkte, Einhängeoptionen und der Konfiguration der Speicherspiegelung angegeben – die persistente Datei sollte sich auf demselben gespiegelten Dateisystem befinden wie die überwachte Protokolldatei.

Wenn das Verzeichnis **persistent_dir** nicht erstellt werden kann oder nicht existiert oder wenn die Zugriffsrechte für den Zabbix Agent das Erstellen/Schreiben/Lesen/Löschen von Dateien nicht erlauben, wird der Log-Datenpunkt zu NOTSUPPORTED.

Wenn Zugriffsrechte auf Dateien des persistenten Speichers während des Betriebs des Agent entfernt werden oder andere Fehler auftreten (z. B. Datenträger voll), werden Fehler in die Agent-Protokolldatei geschrieben, aber der Protokoll-Datenpunkt wird nicht zu NOTSUPPORTED.

Last auf I/O

Die persistente Datei des Datenpunkts wird nach dem erfolgreichen Senden jedes Daten-Batches (der die Daten des Datenpunkts enthält) an den Server aktualisiert. Zum Beispiel ist der Standardwert von `BufferSize` 100. Wenn ein Log-Datenpunkt 70 übereinstimmende Einträge gefunden hat, werden die ersten 50 Einträge in einem Batch gesendet, die persistente Datei wird aktualisiert, dann werden die verbleibenden 20 Einträge (möglicherweise mit einer gewissen Verzögerung, wenn weitere Daten angesammelt werden) im zweiten Batch gesendet, und die persistente Datei wird erneut aktualisiert.

Maßnahmen bei Kommunikationsausfall zwischen Agent und Server

Jede übereinstimmende Zeile aus den `log[]`- und `logrt[]`-Datenpunkten sowie jedes Ergebnis einer Prüfung der `log.count[]`- und `logrt.count[]`-Datenpunkte benötigt einen freien Platz im vorgesehenen 50%-Bereich des Sendepuffers des Agent.

Die Pufferelemente werden regelmäßig an den Server (oder Proxy) gesendet, und die Pufferplätze werden wieder frei.

Solange im vorgesehenen Log-Bereich des Sendepuffers des Agent freie Plätze vorhanden sind und die Kommunikation zwischen Agent und Server (oder Proxy) ausfällt, werden die Ergebnisse der Log-Überwachung im Sendepuffer gesammelt.

Dies hilft, kurze Kommunikationsausfälle abzumildern.

Bei längeren Kommunikationsausfällen werden alle Log-Plätze belegt, und es werden folgende Maßnahmen ergriffen:

- Prüfungen der `log[]`- und `logrt[]`-Datenpunkte werden gestoppt. Wenn die Kommunikation wiederhergestellt ist und freie Plätze im Puffer verfügbar sind, werden die Prüfungen an der vorherigen Position fortgesetzt. Keine übereinstimmenden Zeilen gehen verloren, sie werden lediglich später gemeldet.
- Prüfungen von `log.count[]` und `logrt.count[]` werden gestoppt, wenn `maxdelay = 0` (Standard) ist. Das Verhalten ist ähnlich wie bei den `log[]`- und `logrt[]`-Datenpunkten, wie oben beschrieben. Beachten Sie, dass dies die Ergebnisse von `log.count[]` und `logrt.count[]` beeinflussen kann: Zum Beispiel zählt eine Prüfung 100 übereinstimmende Zeilen in einer Log-Datei, aber da keine freien Plätze im Puffer vorhanden sind, wird die Prüfung gestoppt. Wenn die Kommunikation wiederhergestellt ist, zählt der Agent dieselben 100 übereinstimmenden Zeilen sowie 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[]` mit `maxdelay > 0`: Wenn während der Prüfung kein „Sprung“ stattgefunden hat, ist das Verhalten ähnlich wie oben beschrieben. Wenn ein „Sprung“ über Zeilen der Log-Datei stattgefunden hat, wird die Position nach dem „Sprung“ beibehalten und das gezählte Ergebnis verworfen. So versucht der Agent, auch bei einem Kommunikationsausfall mit einer wachsenden Log-Datei Schritt zu halten.

Behandlung von Fehlern bei der Kompilierung und Laufzeit 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 in den Zustand NOTSUPPORTED mit einer Fehlermeldung. Um die Überwachung des Log-Datenpunkts fortzusetzen, sollte der reguläre Ausdruck korrigiert werden.

Wenn der reguläre Ausdruck erfolgreich kompiliert wird, aber zur Laufzeit fehlschlägt (bei einigen oder bei allen Log-Einträgen), bleibt der Log-Datenpunkt unterstützt und die Überwachung wird fortgesetzt. Der Laufzeitfehler wird in der Zabbix-Agent-Logdatei protokolliert (ohne den Logdatei-Eintrag).

Die Protokollierungsrate ist auf einen Laufzeitfehler pro Prüfung begrenzt, damit der Zabbix-Agent seine eigene Logdatei überwachen kann. Wenn beispielsweise 10 Einträge analysiert werden und bei 3 Einträgen ein Laufzeitfehler des regulären Ausdrucks auftritt, wird ein Eintrag im Agent-Log erzeugt.

Ausnahme: Wenn `MaxLinesPerSecond=1` und das Aktualisierungsintervall=1 ist (nur 1 Eintrag darf pro Prüfung analysiert werden), dann werden Laufzeitfehler regulärer Ausdrücke nicht protokolliert.

zabbix_agentd protokolliert im Fall eines Laufzeitfehlers den Datenpunktschlüssel, zabbix_agent2 protokolliert die Datenpunkt-ID, um zu helfen festzustellen, welcher Log-Datenpunkt Laufzeitfehler hat. Es wird empfohlen, den regulären Ausdruck bei Laufzeitfehlern neu zu entwerfen.

7 Berechnete Datenpunkte

Übersicht

Ein berechneter Datenpunkt ermöglicht es, eine Berechnung auf Grundlage der Werte einiger vorhandener Datenpunkte zu erstellen. Zum Beispiel möchten Sie möglicherweise den stündlichen Durchschnitt eines Datenpunktwerts berechnen oder den Gesamtwert für eine Gruppe von Datenpunkten ermitteln. Genau dafür ist ein berechneter Datenpunkt gedacht.

Berechnungen können Folgendes verwenden:

- einzelne Werte individueller Datenpunkte
- einen komplexen Filter zur Auswahl mehrerer Datenpunkte für die Aggregation (siehe aggregierte Berechnungen für Details)

Berechnete Datenpunkte sind eine Möglichkeit, virtuelle Datenquellen zu erstellen. Alle Berechnungen werden ausschließlich vom Zabbix Server durchgeführt. Die Werte werden periodisch auf Grundlage des verwendeten arithmetischen Ausdrucks berechnet.

Die resultierenden Daten werden wie bei jedem anderen Datenpunkt in der Zabbix-Datenbank gespeichert; sowohl Verlaufs- als auch Trendwerte werden gespeichert, und es können Diagramme erzeugt werden.

> **Note:**
> Wenn das Berechnungsergebnis ein Float-Wert ist, wird es auf eine Ganzzahl gekürzt, wenn der Typ der Informationen des berechneten Datenpunkts *Numerisch (ohne Vorzeichen)* ist.
> Wenn sich außerdem 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.

Berechnete Datenpunkte verwenden dieselbe Syntax wie Auslöser-Ausdrücke. Wenn Sie mit Auslöser-Ausdrücken vertraut sind, können Sie sich berechnete Datenpunkte als Funktionen vorstellen, die einige Werte analysieren, jedoch ohne den Vergleich mit einem Schwellenwert.

Der Vergleich mit Zeichenfolgen ist in berechneten Datenpunkten zulässig. Berechnete Datenpunkte können wie jeder andere Datenpunkttyp durch Makros oder andere Entitäten referenziert werden.

Um berechnete Datenpunkte zu verwenden, wählen Sie den Datenpunkttyp **Calculated**.

Konfigurierbare Felder

Der **Schlüssel** ist eine eindeutige Datenpunkt-Kennung (pro Host). Sie können einen beliebigen Schlüsselnamen unter Verwendung der unterstützten Symbole erstellen.

Die Berechnungsdefinition sollte im Feld **Formel** eingegeben werden. Es besteht keine Verbindung zwischen der Formel und dem Schlüssel. Die Schlüsselparameter werden in der Formel in keiner Weise verwendet.

Die Syntax einer einfachen Formel lautet:

```
function(/host/key,<parameter1>,<parameter2>,...)
```

wobei:

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.
`parameter(s)`	Parameter der Funktion, falls erforderlich. Zeitsuffixe und Speichergrößen-Suffixe werden unterstützt.

Attention:
Benutzermakros in der Formel werden erweitert, wenn sie zur Referenzierung eines Funktionsparameters, eines Datenpunkt-Filterparameters oder einer Konstante verwendet werden. Benutzermakros werden NICHT erweitert, wenn sie sich auf eine Funktion, einen Hostnamen, einen Datenpunkt-Schlüssel, einen Datenpunkt-Schlüsselparameter oder einen Operator beziehen.

Eine komplexere Formel kann eine Kombination aus Funktionen, Operatoren und Klammern verwenden. Sie können alle Funktionen und Operatoren verwenden, die in Auslöser-Ausdrücken unterstützt werden. Die Logik und die Operatorrangfolge sind exakt dieselben.

Im Gegensatz zu Auslöser-Ausdrücken verarbeitet Zabbix berechnete Datenpunkte entsprechend dem Aktualisierungsintervall des Datenpunkts und nicht beim Empfang eines neuen Werts.

Alle Datenpunkte, auf die durch Verlaufsfunktionen in der Formel des berechneten Datenpunkts verwiesen wird, müssen vorhanden sein und Daten erfassen. Wenn Sie außerdem den Datenpunkt-Schlüssel eines referenzierten Datenpunkts ändern, müssen Sie alle Formeln, die diesen Schlüssel verwenden, manuell aktualisieren.

Ein berechneter Datenpunkt kann in mehreren Fällen nicht unterstützt werden:

- referenzierte(r) Datenpunkt(e)
  - wird/werden nicht gefunden
  - ist/sind deaktiviert
  - gehört/gehören zu einem deaktivierten Host
  - wird/werden nicht unterstützt (außer mit der Funktion nodata() und Operatoren mit unbekannten Werten)
- keine Daten zur Berechnung einer Funktion
- Division durch null
- falsche Syntax verwendet

Verwendungsbeispiele

Beispiel 1

Berechnung des Prozentsatzes des freien Speicherplatzes auf '/'.

Use of function **last**:

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix will take the latest values for free and total disk spaces and calculate percentage according to the given formula.

Beispiel 2

Berechnung eines 10-Minuten-Durchschnitts der Anzahl der von Zabbix verarbeiteten Werte.

Verwendung der Funktion **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

Beachten Sie, dass die extensive Verwendung von berechneten Elementen mit langen Zeiträumen die Leistung des Zabbix-Servers beeinträchtigen kann.

Beispiel 3

Berechnung der Gesamtbandbreite auf eth0.

Summe von zwei Funktionen:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Beispiel 4

Berechnung des Prozentsatzes des eingehenden Verkehrs.

Komplexerer Ausdruck:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Siehe auch: Beispiele für Aggregatberechnungen

1 Aggregatberechnungen

Übersicht

Aggregatberechnungen sind ein Typ von berechneten Datenpunkten, mit dem Informationen aus mehreren Datenpunkten vom Zabbix Server erfasst und anschließend abhängig von der verwendeten Aggregatfunktion ein Aggregat berechnet werden können.

Für Aggregatberechnungen ist kein auf dem überwachten Host ausgeführter Agent erforderlich.

Syntax

Um Aggregate abzurufen, verwenden Sie eine der unterstützten Aggregatfunktionen: `avg`, `max`, `min`, `sum` usw. Fügen Sie dann die Funktion **foreach** als einzigen Parameter und ihren Datenpunkt-Filter hinzu, um die erforderlichen Datenpunkte auszuwählen:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Eine Funktion **foreach** (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 Datenpunkt-Historie 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.

> **Note:**
> Wenn das Aggregat zu einem Float-Wert führt, wird dieser auf eine Ganzzahl gekürzt, wenn der aggregierte Informations-Typ des Datenpunkts *Numerisch (Ganzzahl ohne Vorzeichen)* ist.

Benutzermakros und Low-Level-Discovery-Makros werden unterstützt in:

- Datenpunktschlüssel-Parametern
- Funktionsparametern
- Filterbedingungen (Hostgruppenname und Tag-Name)
- Ausdruckskonstanten

Eine Aggregatberechnung kann nicht unterstützt werden, wenn:

- keiner der referenzierten Datenpunkte gefunden wird (dies kann passieren, wenn der Datenpunktschlüssel falsch ist, keiner der Datenpunkte existiert oder alle eingeschlossenen Gruppen falsch sind)
- keine Daten zur Berechnung einer Funktion vorhanden sind

Verwendungsbeispiele

Beispiele für Schlüssel für Aggregatberechnungen.

Beispiel 1

Gesamter Festplattenspeicher der Host-Gruppe „MySQL Servers“.

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Beispiel 2

Summe der letzten Werte aller Datenpunkte, die auf dem Host mit net.if.in[*] übereinstimmen.

```
sum(last_foreach(/host/net.if.in[*]))
```

Beispiel 3

Durchschnittliche Prozessorlast der Host-Gruppe „MySQL Servers“.

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Beispiel 4

5-Minuten-Durchschnitt der Anzahl von Abfragen pro Sekunde für die Host-Gruppe „MySQL Servers“.

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Beispiel 5

Durchschnittliche CPU-Auslastung auf allen Hosts in mehreren Host-Gruppen, die die angegebenen Tags haben.

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Beispiel 6

Berechnung, die auf den Summen der neuesten Datenpunktwerte einer gesamten Host-Gruppe verwendet wird.

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Beispiel 7

Die Gesamtzahl der nicht unterstützten Datenpunkte in der Host-Gruppe „Zabbix servers“.

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Beispiel 8

Summe der letzten numerischen Ergebnisse von DNS-Prüfungen über alle Hosts hinweg. Die angezeigte Datenpunkt-Form ist `net.dns[192.0.2.0,example.com,A]` als Beispiel für einen möglichen Schlüssel.

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Beachten Sie, dass Platzhalter mit der Anzahl der Parameter im Schlüssel übereinstimmen müssen (hier hat `net.dns` drei Parameter: ip, name, type).

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 historischen) 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 in einem Ausdruck wie diesem:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

nicht garantiert werden kann, dass beide Teile der Gleichung immer denselben Wertesatz haben. Während ein Teil des Ausdrucks ausgewertet wird, kann ein neuer Wert für den angeforderten Zeitraum eintreffen, und dann hat der andere Teil des Ausdrucks einen anderen Wertesatz.

8 Interne Prüfungen

Übersicht

Interne Überprüfungen ermöglichen die Überwachung der internen Prozesse von Zabbix. Mit anderen Worten: Sie können überwachen, was im Zabbix Server oder Zabbix Proxy vor sich geht.

Interne Überprüfungen 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 Überprüfungen werden vom Server oder Proxy unabhängig vom Wartungsstatus des Hosts verarbeitet.

Um diesen Datenpunkt zu verwenden, wählen Sie den Datenpunkt-Typ **Zabbix internal**.

> **Note:**
> Interne Überprüfungen 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 sind ohne anpassbare Parameter und zusätzliche Informationen aufgeführt. Klicken Sie auf den Datenpunktschlüssel, um die vollständigen Details anzuzeigen.

Datenpunktschlüssel	Beschreibung
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.
zabbix[discovery_queue]	Die Anzahl der in die Discovery-Warteschlange eingereihten Netzwerkprüfungen.
zabbix[host,,items]	Die Anzahl aktivierter Datenpunkte (unterstützt und nicht unterstützt) auf dem Host.
zabbix[host,,items_unsupported]	Die Anzahl aktivierter nicht unterstützter 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 konfigurierten Schnittstellen des Hosts im Zabbix-Frontend.
zabbix[host,,available]	Die Verfügbarkeit der Hauptschnittstelle eines bestimmten Prüfungstyps auf dem Host.
zabbix[hosts]	Die Anzahl überwachter Hosts.
zabbix[items]	Die Anzahl aktivierter Datenpunkte (unterstützt und nicht unterstützt).
zabbix[items_unsupported]	Die Anzahl nicht unterstützter Datenpunkte.
zabbix[java,,]	Informationen über das Zabbix Java gateway.
zabbix[lld_queue]	Die Anzahl der in die Verarbeitungswarteschlange für Low-Level-Discovery eingereihten Werte.
zabbix[preprocessing]	Statistiken der vom Präprozessierungsmanager empfangenen Werte.
zabbix[preprocessing_queue]	Die Anzahl der in die Präprozessierungswarteschlange eingereihten Werte.
zabbix[process,,,]	Der prozentuale Zeitanteil, den ein bestimmter Zabbix-Prozess oder eine Gruppe von Prozessen (identifiziert durch <type> und <mode>) im Zustand <state> verbracht hat.
zabbix[proxy,,]	Informationen über den Zabbix-Proxy.
zabbix[proxy,discovery]	Die Liste der Zabbix-Proxys.
zabbix[proxy group,,available]	Die Anzahl der Online-Proxys in einer Proxy-Gruppe.
zabbix[proxy group,,pavailable]	Der Prozentsatz der Online-Proxys in einer Proxy-Gruppe.
zabbix[proxy group,,proxies]	Die Liste der Zabbix-Proxys in einer Proxy-Gruppe.
zabbix[proxy group,,state]	Der Status einer Proxy-Gruppe.
zabbix[proxy group,discovery]	Gibt eine Liste von Proxy-Gruppen mit Konfigurationsdaten und Echtzeitdaten zurück.
zabbix[proxy_buffer,buffer,]	Gibt Statistiken zur Nutzung des Proxy-Speicherpuffers zurück.
zabbix[proxy_buffer,state,changes]	Gibt die Anzahl der Statusänderungen zwischen Festplatten-/Speicherpuffermodi seit dem Start zurück.
zabbix[proxy_buffer,state,current]	Gibt den aktuellen Betriebsstatus zurück, in dem die neuen Daten gespeichert werden.

Datenpunktschlüssel	Beschreibung
zabbix[proxy_history]	Die Anzahl der Werte in der Proxy-History-Tabelle, die darauf warten, an den Server gesendet zu werden.
zabbix[queue„]	Die Anzahl überwachter Datenpunkte in der Warteschlange, deren Verzögerung mindestens <from> Sekunden, aber weniger als <to> Sekunden beträgt.
zabbix[rcache„]	Die Verfügbarkeitsstatistiken des Zabbix-Konfigurationscaches.
zabbix[requiredperformance]	Die erforderliche Leistung des Zabbix-Servers oder Zabbix-Proxys, angegeben in erwarteten neuen Werten 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 entfernten 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 entfernten Instanz abgerufen, andernfalls von der lokalen Instanz.
zabbix[tcache„]	Die Effizienzstatistiken des Zabbix-Trendfunktionscaches.
zabbix[triggers]	Die Anzahl aktivierter Auslöser in der Zabbix-Datenbank, bei denen alle Datenpunkte auf aktivierten Hosts aktiviert sind.
zabbix[uptime]	Die Laufzeit des Zabbix-Server- oder Proxy-Prozesses in Sekunden.
zabbix[vcache,buffer,]	Die Verfügbarkeitsstatistiken des Zabbix-Wertecaches.
zabbix[vcache,cache,]	Die Effizienzstatistiken 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 History-Werte.
zabbix[wcache„]	Die Statistiken und die 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[boottime]

 Die Startzeit des Zabbix-Server- oder Zabbix-Proxy-Prozesses in Sekunden.
 Rückgabewert: *Integer*.

zabbix[cluster,discovery,nodes]

 Ermittelt die Knoten des Hochverfügbarkeitsclusters.
 Rückgabewert: *JSON-Objekt*.

Kommentare:

- Dieser Datenpunkt kann in der Low-Level-Discovery verwendet werden.
- Die zurückgegebenen Felder entsprechen größtenteils den Eigenschaften des High availability node object.
- Zusätzliche Felder:
  - `db_timestamp` — aktuelle Server-Zeit (Unix-Zeitstempel), die in die Antwort aufgenommen wird
  - `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* - unbekannt; *1* - verfügbar; *2* - nicht verfügbar.

Parameter:

- **type** - *agent*, *snmp*, *ipmi* oder *jmx*.

Kommentare:

- Der Wert des Datenpunkts wird entsprechend 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>]

 Die 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 mithilfe der Auslöser-Funktion `nodata()` zu prüfen.
- 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 Zeitanteil, den ein bestimmter Zabbix-Prozess oder eine Gruppe von Prozessen (identifiziert durch <type> und <mode>) im Zustand <state> verbracht hat. Er wird nur 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*, *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 geforkten Instanzen; wenn zum Beispiel 4 Trapper laufen, liegt der Wert zwischen 1 und 4);
- **state** - *busy* - der Prozess befindet sich im Busy-Zustand, zum Beispiel bei der Verarbeitung einer Anfrage (Standard);
idle - der Prozess befindet sich im Idle-Zustand und tut nichts.

Kommentare:

- Wenn <mode> eine Zabbix-Prozessnummer ist, die nicht läuft (zum Beispiel wenn 5 Poller laufen und für <mode> der Wert 6 angegeben wird), wird ein solcher Datenpunkt nicht unterstützt.
- Minimum und Maximum beziehen sich auf den prozentualen Nutzungsanteil eines einzelnen Prozesses. Wenn also in einer Gruppe von 3 Pollern die Nutzungsanteile pro Prozess 2, 18 und 66 betragen, gibt min den Wert 2 und max den Wert 66 zurück.
- Prozesse melden in Shared Memory, was sie tun, und der Self-Monitoring-Prozess fasst diese Daten jede Sekunde zusammen. Zustandsänderungen (busy/idle) werden bei einer Änderung registriert - ein Prozess, der also busy wird, wird entsprechend registriert und ändert oder aktualisiert den Zustand nicht, bis er idle wird. Dadurch wird sichergestellt, dass selbst vollständig hängende Prozesse korrekt als zu 100 % busy registriert werden.
- Derzeit bedeutet "busy" "nicht schlafend", künftig könnten jedoch zusätzliche Zustände eingeführt werden - Warten auf Sperren, Ausführen von Datenbankabfragen usw. Beachten Sie, dass asynchrone Poller als busy gelten, wenn sie den durch den Konfigurationsparameter MaxConcurrentChecksPerPoller für Server/Proxy festgelegten Grenzwert erreicht haben.
- Unter Linux und den meisten anderen Systemen beträgt die Auflösung 1/100 Sekunde.

Beispiele:

```
zabbix[process,poller,avg,busy] #die durchschnittliche Zeit, die Poller-Prozesse in der letzten Minute mit
zabbix[process,"icmp pinger",max,busy] #die maximale Zeit, die ein beliebiger ICMP-pinger-Prozess in der l
zabbix[process,"history syncer",2,busy] #die Zeit, die History-Syncer Nummer 2 in der letzten Minute mit e
zabbix[process,trapper,count] #die Anzahl der aktuell laufenden Trapper-Prozesse
```

zabbix[proxy,<name>,<param>]

 Informationen über den Zabbix Proxy.
 Rückgabewert: *Integer*.

Parameter:

- **name** - der Proxy-Name;
- **param** - *lastaccess* - der Zeitstempel der zuletzt vom Proxy empfangenen Heartbeat-Nachricht;
delay - wie lange die erfassten Werte noch nicht gesendet wurden; berechnet als „Proxy-Verzögerung" + („aktuelle Serverzeit" - „letzter Zugriff des Proxy"), wobei die „Proxy-Verzögerung" die Differenz zwischen der aktuellen Proxy-Zeit und dem Zeitstempel des ältesten noch nicht gesendeten Werts auf dem Proxy ist.

Kommentare:

- Dieser Datenpunkt wird immer vom Zabbix Server verarbeitet, unabhängig vom Speicherort des Hosts (auf dem Server oder Proxy).
- Die Funktion `fuzzytime()` kann verwendet werden, um die Verfügbarkeit des Proxy zu prüfen.

Beispiel:

```
zabbix[proxy,"Germany",lastaccess] #der Zeitstempel der zuletzt vom Proxy "Germany" empfangenen Heartbeat-
```

zabbix[proxy,discovery]

 Die Liste der Zabbix-Proxys mit Name, Modus, Verschlüsselung, Komprimierung, Version, zuletzt gesehen, Anzahl der Hosts, Anzahl der Datenpunkte, erforderlichen Werten pro Sekunde (vps), Versionsstatus (aktuell/veraltet/nicht unterstützt), Timeouts nach Datenpunkttyp, Name der Proxy-Gruppe (falls der Proxy zu einer Gruppe gehört), Status (unbekannt/offline/online).
 Rückgabewert: *JSON-Objekt*.

zabbix[proxy group,<name>,available]

 Die Anzahl der online befindlichen Proxys in einer Proxy-Gruppe.
 Rückgabewert: *Integer*.

Parameter:

- **name** - der Name der Proxy-Gruppe.

zabbix[proxy group,<name>,pavailable]

 Der Prozentsatz der online befindlichen Proxys in einer Proxy-Gruppe.
 Rückgabewert: *Float*.

Parameter:

- **name** - der Name der Proxy-Gruppe.

zabbix[proxy group,<name>,proxies]

 Die Liste der Zabbix-Proxys in einer Proxy-Gruppe mit Name, Modus, Verschlüsselung, Komprimierung, Version, zuletzt gesehen, Anzahl der Hosts, Anzahl der Datenpunkte, erforderlichen Werten pro Sekunde (vps), Versionsstatus (aktuell/veraltet/nicht unterstützt), Timeouts, Name der Proxy-Gruppe, Status (unbekannt/offline/online).
 Rückgabewert: *JSON*.

Parameter:

- **name** - der Name der Proxy-Gruppe.

zabbix[proxy group,<name>,state]

 Der Status einer Proxy-Gruppe.
 Rückgabewert: *0* - unbekannt; *1* - offline; *2* - Wiederherstellung; *3* - online; *4* - verschlechtert.

Parameter:

- **name** - der Name der Proxy-Gruppe.

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 online befindlichen Proxys. Zu den Echtzeitdaten gehören der Status der Proxy-Gruppe (siehe Kommentare für Details), die Anzahl der online befindlichen Proxys und der prozentuale Anteil der online befindlichen Proxys.
 Rückgabewert: *JSON*.

Kommentare:

- Dieser Datenpunkt gibt keine Proxys ohne Gruppenzugehörigkeit zurück.
- Wenn für "failover_delay" oder "min_online" ein ungültiger Wert vorliegt, wird der spezielle Wert *-1* gemeldet, um dies anzuzeigen. Ungültige Werte können auftreten, wenn Makros für die Konfiguration verwendet werden und die Makros nicht zu einem gültigen Wert expandiert werden können.
- Der Status der Proxy-Gruppe wird als Ganzzahl gemeldet: *0* - unbekannt; *1* - offline; *2* - Wiederherstellung; *3* - online; *4* - verschlechternd.

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 Betriebszustand zurück, in dem die neuen Daten gespeichert werden.
 Rückgabewerte: *0* - Festplatte; *1* - Speicher.

Kommentare:

- "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*.

Kommentare:

- Dieser Datenpunkt wird auf dem Zabbix Server nicht unterstützt.

zabbix[queue,<from>,<to>]

 Die Anzahl der überwachten Datenpunkte in der Warteschlange, deren Verzögerung mindestens <from> Sekunden beträgt, aber weniger als <to> Sekunden.
 Rückgabewert: *Integer*.

Parameter:

- **from** - verzögert um mindestens (Standard ist 6 Sekunden);
- **to** - verzögert um höchstens (Standard ist unendlich).

Kommentare:

- Zeitsuffixe (s,m,h,d,w) werden in den Parametern unterstützt.

> **Attention:**
> Der Zabbix Proxy berücksichtigt keine Wartungszeiträume; 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 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[requiredperformance]

 Die erforderliche Leistung des Zabbix Server oder Zabbix Proxy, angegeben in erwarteten neuen Werten pro Sekunde.
 Rückgabewert: *Float*.

Kommentare:

- Entspricht ungefähr „Erforderliche Serverleistung, neue Werte pro Sekunde“ in *Berichte* > *Systeminformationen*.

zabbix[stats,<ip>,<port>]

 Gibt die internen Metriken eines Zabbix Server oder Proxy 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-/Netzwerkmaskenliste von Servern/Proxys, die per Fernabfrage abgefragt werden sollen (Standard ist 127.0.0.1);
- **port** - der Port des Server/Proxy, der per Fernabfrage abgefragt werden soll (Standard ist 10051).

Kommentare:

- Die Statistikabfrage wird nur von den Adressen akzeptiert, die im Parameter ’StatsAllowedIP’ des Zielsystems unter server/proxy aufgeführt sind.
- Dieser Datenpunkt gibt eine ausgewählte Menge interner Metriken zurück. Einzelheiten finden Sie unter Remote monitoring of Zabbix stats.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Gibt die internen Warteschlangenmetriken (siehe `zabbix[queue,<from>,<to>]`) eines Zabbix Server oder Proxy 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-/Netzwerkmaskenliste von Servern/Proxys, die per Fernabfrage abgefragt werden sollen (Standard ist 127.0.0.1);
- **port** - der Port des Server/Proxy, der per Fernabfrage abgefragt werden soll (Standard ist 10051);
- **from** - verzögert um mindestens (Standard ist 6 Sekunden);
- **to** - verzögert um höchstens (Standard ist unendlich).

Kommentare:

- Die Statistikabfrage wird nur von den Adressen akzeptiert, die im Parameter ’StatsAllowedIP’ des Zielsystems unter server/proxy aufgeführt sind.
- Eine ausgewählte Menge interner Metriken wird von diesem Datenpunkt zurückgegeben. Einzelheiten finden Sie unter Remote monitoring of Zabbix stats.

Attention:
Der Zabbix Proxy kennt keine Wartungszeiträume; siehe Calculation of queues during maintenance für Details.

zabbix[tcache,cache,<parameter>]

 Die Effektivitätsstatistik 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 Datenpunkte;
requests - die Anzahl der zwischengespeicherten Anfragen;
pitems - Prozentsatz der zwischengespeicherten Datenpunkte an zwischengespeicherten Datenpunkten + Anfragen. Ein niedriger Prozentsatz bedeutet höchstwahrscheinlich, dass die Cache-Größe reduziert werden kann.

Kommentare:

- Dieser Datenpunkt wird auf 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-Werte-Caches.
 Rückgabewerte: *Integer* (für Größe); *Float* (für 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 dem Zabbix Proxy nicht unterstützt.

zabbix[vcache,cache,<parameter>]

 Die Effektivitätsstatistik des Zabbix-Werte-Caches.
 Rückgabewert: *Integer*. Wenn <parameter> auf *mode* gesetzt ist, lautet der Rückgabewert: *0* - normaler Modus; *1* - Modus mit wenig Speicher.

Parameter:

- **parameter** - *requests* - die Gesamtzahl der Anfragen;
hits - die Anzahl der Cache-Treffer (Verlaufswerte aus dem Cache);
misses - die Anzahl der Cache-Fehlzugriffe (Verlaufswerte aus der Datenbank);
mode - der Betriebsmodus des Werte-Caches.

Kommentare:

- Sobald der Modus mit wenig Speicher aktiviert wurde, bleibt der Werte-Cache 24 Stunden in diesem Zustand, auch wenn das Problem, das diesen Modus ausgelöst hat, früher behoben wird.
- Sie können diesen Schlüssel zusammen mit dem Vorverarbeitungsschritt *Änderung pro Sekunde* verwenden, um eine Statistik der Werte pro Sekunde zu erhalten.
- Dieser Datenpunkt wird auf Zabbix Proxy nicht unterstützt.

zabbixversion

 Die Version des Zabbix Server oder Proxy.
 Rückgabewert: *String*. Zum Beispiel: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Die Verfügbarkeitsstatistik des Zabbix-vmware-Cache.
 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-Schreib-Cache.
 Rückgabewerte: *Integer* (für Anzahl/Größe); *Float* (für Prozentsatz).

Parameter:

- **cache** - *values*, *history*, *index* oder *trend*;
- **mode** - (mit *values*) *all* (Standard) - die Gesamtzahl der von Zabbix Server/Proxy verarbeiteten Werte, ausgenommen nicht unterstützte Datenpunkte (Zähler);
float - die Anzahl verarbeiteter Float-Werte (Zähler);
uint - die Anzahl verarbeiteter vorzeichenloser Integer-Werte (Zähler);
str - die Anzahl verarbeiteter Zeichen-/String-Werte (Zähler);
log - die Anzahl verarbeiteter Log-Werte (Zähler);
text - die Anzahl verarbeiteter Textwerte (Zähler);
bin - die Anzahl verarbeiteter Binärwerte (Zähler);
json - die Anzahl verarbeiteter 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 *history*-, *index*-, *trend*-Cache) *pfree* (Standard) - der Prozentsatz des freien Puffers;
total - die Gesamtgröße des Puffers;
free - die Größe des freien Puffers;
used - die Größe des verwendeten Puffers;
pused - der Prozentsatz des verwendeten Puffers.

Kommentare:

- Der Cache-Parameter `trend` wird von 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 zu erwarten und hilft dem System, effizienter zu arbeiten, da der zusätzliche Verarbeitungsaufwand für die ständige Größenanpassung des Speichers vermieden wird.

- 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 Präprozessierungsschritt *Change per second* verwenden, um eine Werte-pro-Sekunde-Statistik zu erhalten.

9 SSH-Prüfungen

Übersicht

SSH-Prüfungen werden als Agent-loses Monitoring 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 Server-Konfiguration erforderlich.

Öffnen Sie die Zabbix-Server-Konfigurationsdatei (*zabbix_server.conf*) als `root` und suchen Sie nach der folgenden Zeile:

```
##### SSHKeyLocation=
```

Entfernen Sie die Auskommentierung und setzen Sie den vollständigen Pfad zu dem Verzeichnis, 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.

In der Regel erstellen Installationspakete des Zabbix-Servers aus verschiedenen OS-Distributionen das Benutzerkonto *zabbix* mit einem Home-Verzeichnis an einem anderen Ort, zum Beispiel */var/lib/zabbix* (wie bei Systemkonten).

Bevor die Schlüssel erzeugt werden, können Sie das Home-Verzeichnis nach */home/zabbix* verlegen, damit es dem oben genannten Zabbix-Server-Konfigurationsparameter `SSHKeyLocation` entspricht.

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, gestoppt 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 am alten Speicherort kein Home-Verzeichnis existiert hat; 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 geschützt ist, können zusätzlich Befehle ausgeführt werden, um die Berechtigungen für das Home-Verzeichnis zu setzen:

```
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 Erzeugen der öffentlichen und privaten Schlüssel mit den folgenden Befehlen durchgeführt werden (zur besseren Lesbarkeit sind die Eingabeaufforderungen in den Kommentaren 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):
<Leer lassen>
##### Enter same passphrase again:
<Leer lassen>
##### 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* erzeugt, was dem Zabbix-Server-Konfigurationsparameter `SSHKeyLocation` entspricht.

Attention:
Andere Schlüsseltypen als "rsa" werden möglicherweise vom Tool ssh-keygen und von SSH-Servern unterstützt, jedoch möglicherweise 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 werden soll.

Mit den folgenden Befehlen kann die Datei mit dem **öffentlichen** Schlüssel 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 Eingabeaufforderungen in den Kommentaren 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.
```

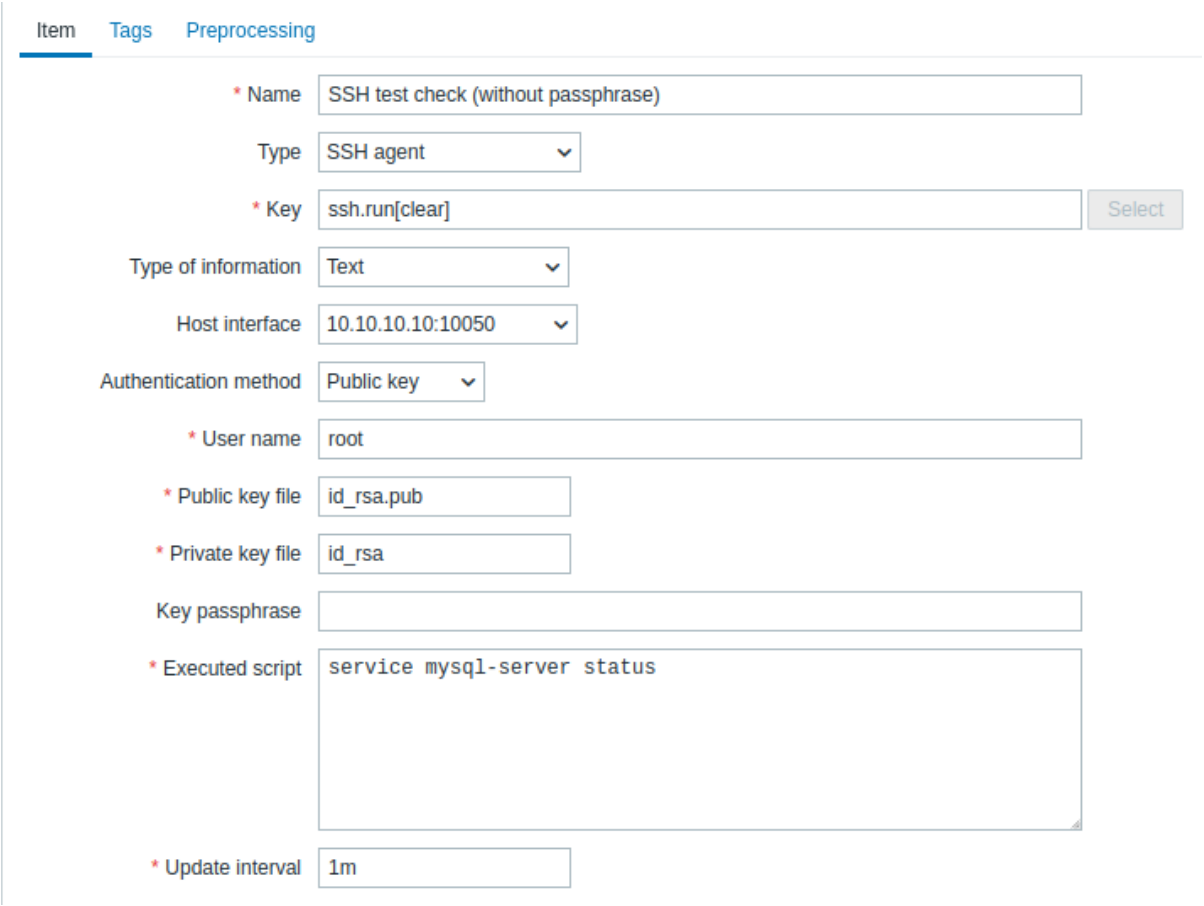
Nun ist es möglich, die SSH-Anmeldung mit dem standardmäßigen privaten Schlüssel (*/home/zabbix/.ssh/id_rsa*) für das Benutzerkonto *zabbix* zu prüfen:

```
sudo -u zabbix ssh root@10.10.10.10
```

Wenn die Anmeldung erfolgreich ist, ist der Konfigurationsteil in der Shell abgeschlossen und die entfernte SSH-Sitzung kann geschlossen werden.

Konfiguration von Datenpunkten

Die tatsächlich auszuführenden Befehle müssen im Feld *Ausgeführtes Skript* in der Datenpunkt-Konfiguration eingetragen werden. Mehrere Befehle können nacheinander ausgeführt werden, indem sie jeweils in eine neue Zeile gesetzt werden. In diesem Fall werden auch die zurückgegebenen Werte mehrzeilig formatiert.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Die Felder, die für SSH-Datenpunkte spezifische Informationen erfordern, sind:

Parameter	Beschreibung	Kommentare
Type	Wählen Sie hier **SSH agent** aus.	

Parameter	Beschreibung	Kommentare

Key	Eindeutiger (pro Host) Datenpunktschlüssel im Format **ssh.run[eindeutige Kurzbeschreibung,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]**	**eindeutige Kurzbeschreibung** ist erforderlich und sollte für jeden SSH-Datenpunkt pro Host eindeutig sein.
		
		Der Standard-Port 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 Komma getrennt übergeben werden (in diesem Fall muss der Parameter in Anführungszeichen gesetzt werden); mehrere Optionsschlüssel können durch Semikolon getrennt übergeben werden.
		
		Die folgenden Optionsschlüssel werden unterstützt: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Die Unterstützung von Optionsschlüssel und -wert hängt von der SSH-Bibliothek ab (zum Beispiel 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 Pluszeichen "+" zum Anhängen von Cipher-Einstellungen und "!" zum Deaktivieren bestimmter Cipher-Einstellungen (wie in GnuTLS und OpenSSL) nicht unterstützt werden.
		
		Beispiele:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-h`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubke`
		
		Mit **subsystem** kann ein SSH-Subsystem übergeben werden, wodurch die SSH-Verbindung auf bestimmte vom Subsystem erlaubte Operationen beschränkt wird (z. B. Dateiübertragungen mit SFTP oder Netzwerkgeräteverwaltung mit NETCONF). Beachten Sie, dass die Verwendung eines Subsystems auch die Verwendung einer spezifischen Skriptsyntax im Parameter *Ausgeführtes Skript* erfordern kann.
		
		Beispiele:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentication method	Eine der Optionen „Password" oder „Public key".	

Parameter	Beschreibung	Kommentare
User name	Benutzername (bis zu 255 Zeichen) zur Authentifizierung auf dem entfernten Host. Erforderlich.	
Public key file	Dateiname des öffentlichen Schlüssels, wenn *Authentication method* auf „Public key“ gesetzt ist. Erforderlich.	Beispiel: *id_rsa.pub* – Standarddateiname des öffentlichen Schlüssels, der mit dem Befehl ssh-keygen erzeugt wird.
Private key file	Dateiname des privaten Schlüssels, wenn *Authentication method* auf „Public key“ gesetzt ist. Erforderlich.	Beispiel: *id_rsa* – Standarddateiname des privaten Schlüssels.
Password oder *Key passphrase*	Passwort (bis zu 255 Zeichen) zur Authentifizierung oder Passphrase, **falls** sie für den privaten Schlüssel verwendet wurde.	Lassen Sie das Feld *Key passphrase* leer, wenn keine Passphrase verwendet wurde. Siehe auch bekannte Probleme bezüglich der Verwendung von Passphrasen.

Executed script	Ausgeführte Shell-Befehle über eine entfernte SSH-Sitzung.	Der Rückgabewert der ausgeführten Shell-Befehle ist auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden); Datenbankgrenzen gelten ebenfalls.
		
		Beachten Sie, dass die Bibliothek libssh2 ausführbare Skripte möglicherweise auf ~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>`
		`]]>]]>`

10 Telnet-Prüfungen

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 gesetzt 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.

Schlüssel	Beschreibung
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.

11 Externe Prüfungen

Übersicht

Externe Prüfung ist eine Prüfung, die vom Zabbix Server durch Ausführen eines Shell-Skripts oder einer Binärdatei ausgeführt wird. Wenn jedoch Hosts von einem Zabbix Proxy überwacht werden, werden die externen Prüfungen vom Proxy ausgeführt.

Externe Prüfungen erfordern keinen Agent, der auf einem überwachten Host läuft.

Die Syntax des Datenpunktschlüssels lautet:

```
script[<parameter1>,<parameter2>,...]
```

Dabei gilt:

ARGUMENT	DEFINITION
script	Name eines Shell-Skripts oder einer Binärdatei.
parameter(s)	Optionale Befehlszeilenparameter.

Wenn Sie keine Parameter an das Skript übergeben möchten, können Sie Folgendes verwenden:

```
script[] or
script
```

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 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 Prüfungen nicht übermäßig, da für jedes Skript vom Zabbix Server/Proxy ein Fork-Prozess gestartet werden muss und die Ausführung vieler Skripte die Leistung von Zabbix erheblich verringern kann.

Anwendungsbeispiel

Ausführen des Skripts **check_oracle.sh** mit dem ersten Parameter '-h'. Der zweite Parameter wird je nach Auswahl in den Host-Eigenschaften durch die IP-Adresse oder den DNS-Namen ersetzt.

```
check_oracle.sh["-h","{HOST.CONN}"]
```

Angenommen, der Host ist für die Verwendung einer IP-Adresse konfiguriert, 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.

12 Trapper-Datenpunkte

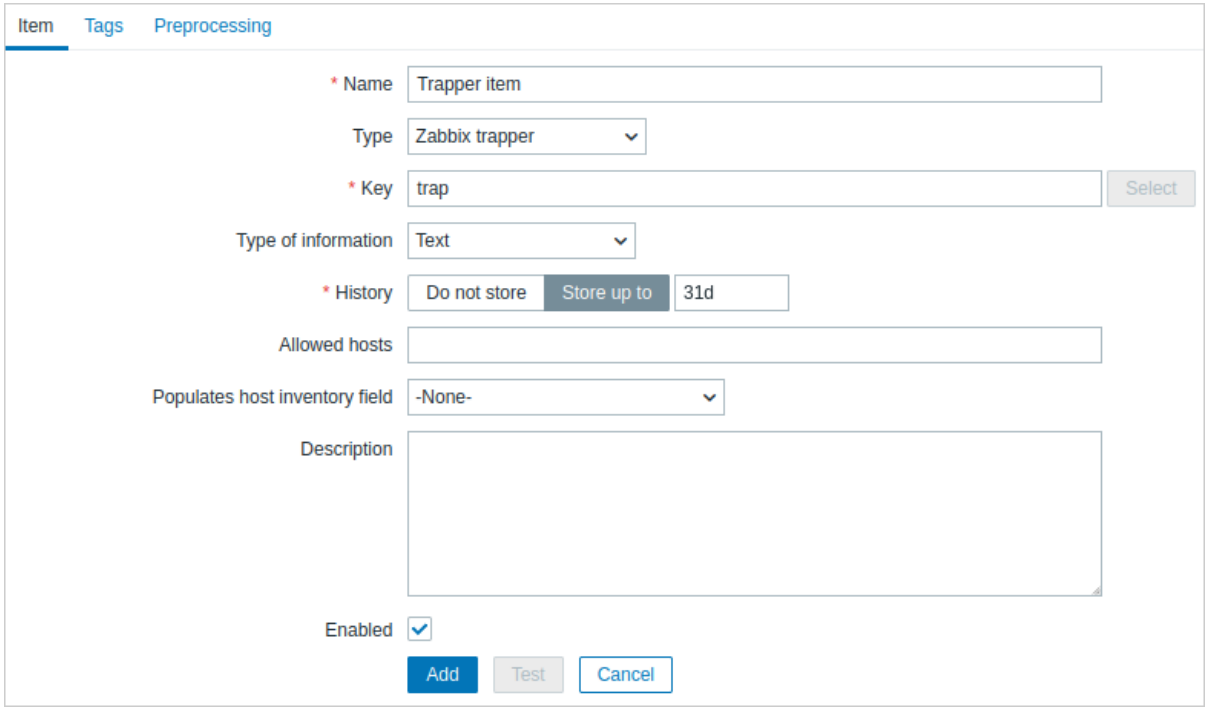
Übersicht

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 *Datensammlung* → *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 erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Die Felder, die für Trapper-Datenpunkte spezifische Informationen erfordern, sind:

Typ	Wählen Sie „Zabbix trapper“ aus.
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.

Erlaubte Hosts	Liste von durch Kommas getrennten IP-Adressen (optional in CIDR-Notation) oder DNS-Namen. Falls angegeben, 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', '::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 werden, aber durch RFC4291 als veraltet eingestuft 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 Sie Werte senden, müssen Sie nach dem Speichern des Datenpunkts möglicherweise bis zu 60 Sekunden warten, bis der Zabbix Server die Änderungen durch eine Aktualisierung des Konfigurationscaches übernommen hat.

Daten senden

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 auch mit der API-Methode `history.push` möglich.

Zabbix sender

Um Daten mit dem Dienstprogramm Zabbix sender an den Zabbix Server oder Proxy zu senden, können Sie den folgenden Befehl ausführen, um den „Testwert" zu senden:

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Um den „Testwert" zu senden, werden die folgenden Befehlsoptionen verwendet:

- `-z` zur Angabe der IP-Adresse des Zabbix Servers
- `-p` zur Angabe der Portnummer des Zabbix Servers (standardmäßig 10051)
- `-s` zur Angabe des Hosts (achten Sie darauf, den technischen statt des sichtbaren Host-Namens zu verwenden)
- `-k` zur Angabe des Schlüssels des im Trapper-Datenpunkt konfigurierten Datenpunkts
- `-o` zur Angabe des zu sendenden Werts

Attention:
Der Zabbix-Trapper-Prozess erweitert keine Makros, die im Schlüssel des Datenpunkts verwendet werden, um die Existenz des entsprechenden Datenpunktschlüssels für den Ziel-Host zu prüfen.

Weitere Informationen zur Kommunikation zwischen Zabbix sender und dem 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:

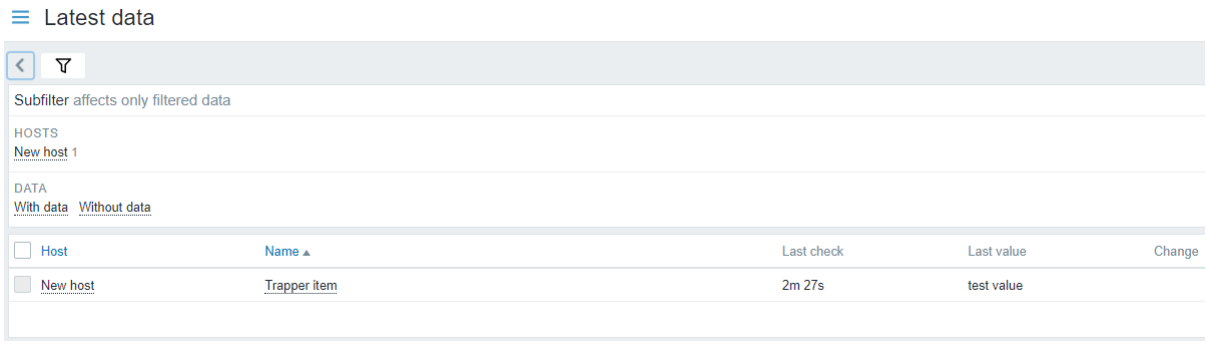
- der Benutzer, der die Daten sendet, hat keine *Leseberechtigung* für den Host des Datenpunkts;
- der Host ist deaktiviert oder befindet sich in der Wartung ohne Datenerfassung;
- der Datenpunkt existiert nicht oder ist noch nicht im Konfigurations-Cache des Servers enthalten;
- der Datenpunkt ist deaktiviert 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.

Daten anzeigen

Sobald Daten gesendet wurden, können Sie zu *Monitoring* → *Letzte Daten* navigieren, um das Ergebnis zu sehen:



Note:
Wenn ein einzelner numerischer Wert gesendet wird, zeigt das Daten-Diagramm links und rechts vom Zeitpunkt des Werts eine horizontale Linie an.

13 JMX-Überwachung

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.

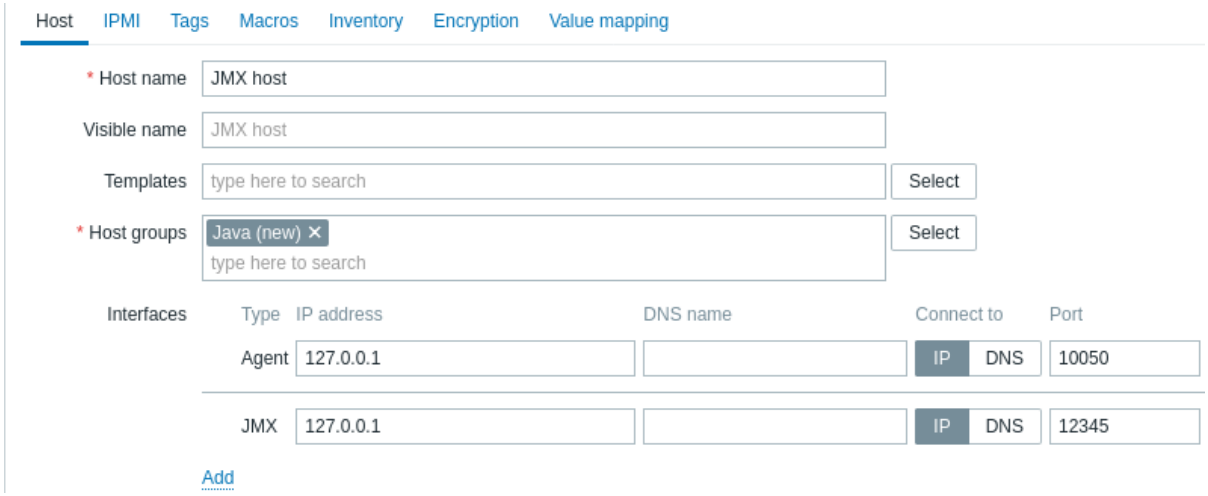
Eine ausführliche Beschreibung finden Sie unter Monitoring and Management Using JMX.

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.

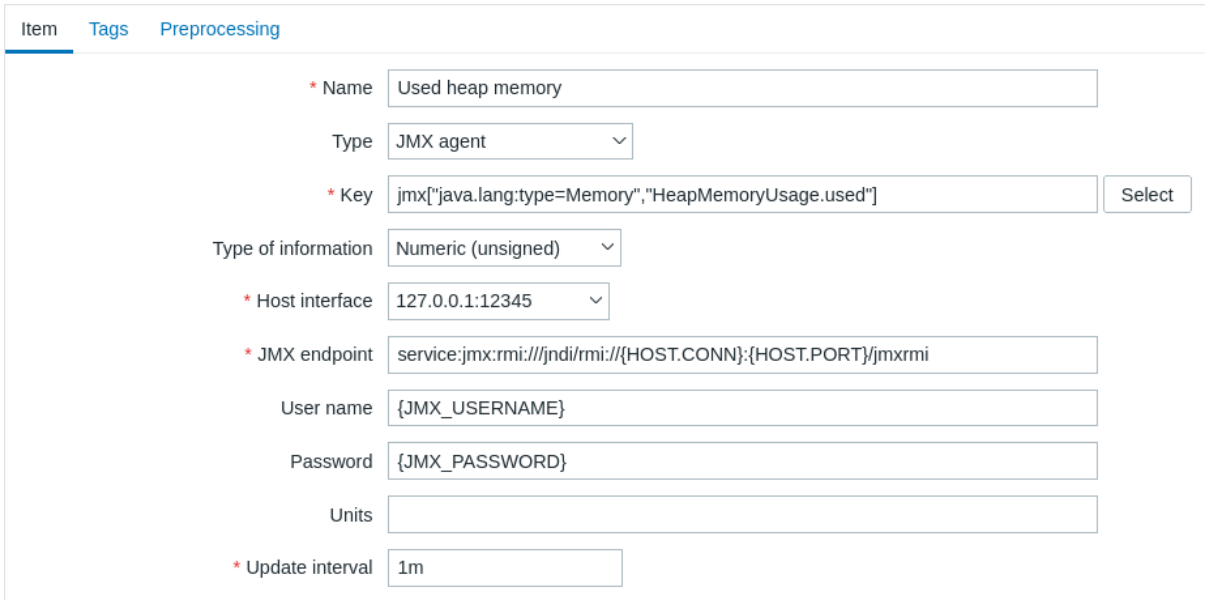


Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

JMX-Agent-Datenpunkt hinzufügen

Für jeden JMX-Zähler, der Sie interessiert, fügen Sie einen **JMX-Agent**-Datenpunkt hinzu, der an diese Schnittstelle angehängt ist.

Der Schlüssel im Screenshot unten lautet `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Die Felder, die für JMX-Datenpunkte spezifische Informationen erfordern, sind:

Typ	Stellen Sie hier **JMX-Agent** ein.
Schlüssel	Der Datenpunktschlüssel `jmx[]` enthält drei Parameter: **Objektname** - der Objektname einer MBean **Attributname** - ein MBean-Attributname mit optionalen zusammengesetzten Datenfeldnamen, die durch Punkte getrennt sind **eindeutige Kurzbeschreibung** - 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 mit einem `jmx.discovery[]`-Low-Level-Discovery-Datenpunkt ermitteln.
JMX-Endpunkt	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 im Standard-JMX-Endpunkt. {HOST.*}-Makros und Benutzermakros werden unterstützt.
Benutzername	Geben Sie den Benutzernamen an (bis zu 255 Zeichen), falls Sie die Authentifizierung für Ihre Java-Anwendung konfiguriert haben. Benutzermakros werden unterstützt.

Passwort	Geben Sie das Passwort an (bis zu 255 Zeichen), falls Sie die Authentifizierung für Ihre Java-Anwendung konfiguriert haben. Benutzermakros werden unterstützt.

Wenn Sie einen booleschen Zähler überwachen möchten, der entweder „true“ oder „false“ ist, geben Sie als Informationstyp „Numerisch (Ganzzahl ohne Vorzeichen)“ an und wählen im Reiter „Vorverarbeitung“ den Vorverarbeitungsschritt „Boolesch in Dezimal“ aus. Der Server speichert boolesche Werte dann jeweils 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]
```

Andere Probleme

Ein Backslash-Zeichen in einem Attributnamen sollte maskiert werden:

```
jmx[com.example:type=Hello,c:\\documents]
```

Informationen zur Behandlung anderer Sonderzeichen im JMX-Datenpunktschlüssel finden Sie im Abschnitt zum Format des Datenpunktschlüssels 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 anderen Transportprotokollen als dem standardmäßigen RMI.

Um diese Möglichkeit zu veranschaulichen, versuchen wir als Beispiel, das Monitoring von JBoss EAP 6.4 zu konfigurieren. Zunächst treffen wir einige Annahmen:

- Sie haben das 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 arbeiten
- 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
```

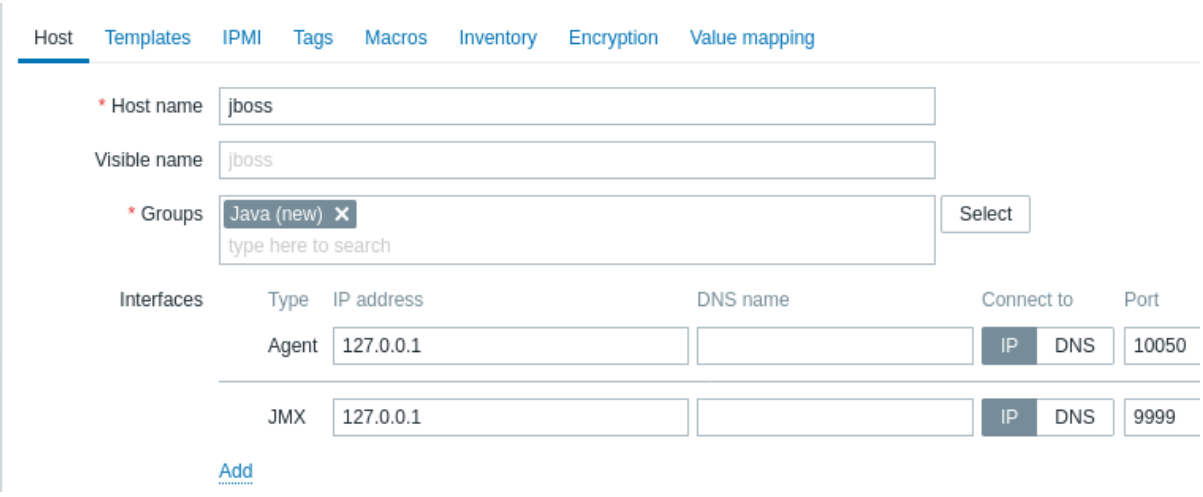
Und in der Konfigurationsdatei `zabbix_java/settings.sh` (oder `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Prüfen Sie, dass JBoss auf seinem standardmäßigen Management-Port lauscht:

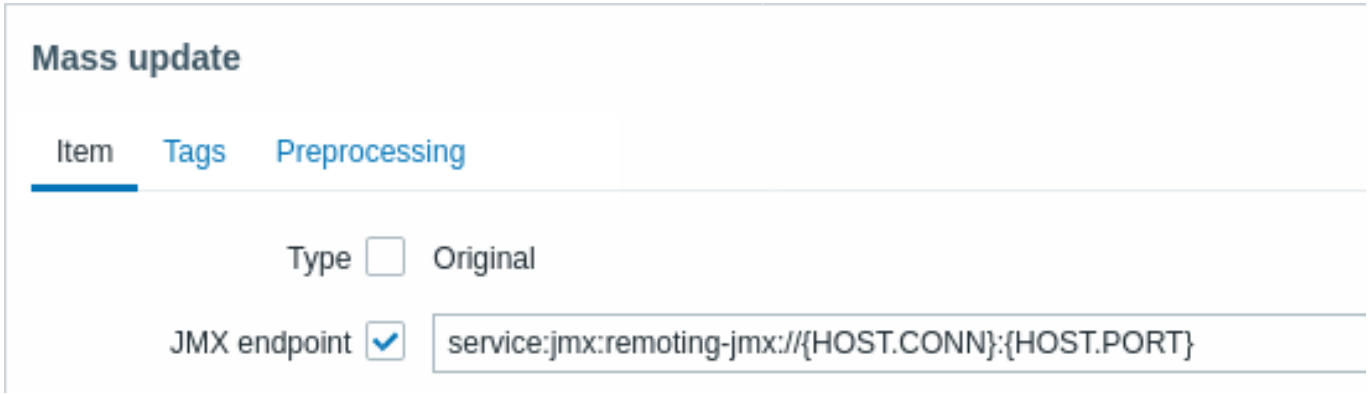
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

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 Protokoll JBoss Remoting anstelle von RMI verwendet, können wir den Parameter JMX endpoint für Datenpunkte in unserer JMX-Vorlage entsprechend per Massenaktualisierung ändern:

```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Aktualisieren wir den Konfigurations-Cache:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Beachten Sie, dass zunächst ein Fehler auftreten kann.

```
3. mc [root@centos7-dev]:/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN  com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885)
as stopped
2017-11-07 13:52:26.167 [main] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has
tarted
```

„Unsupported protocol: remoting-jmx“ bedeutet, dass das Java gateway nicht weiß, wie es mit dem angegebenen Protokoll arbeiten soll. Dies kann behoben werden, indem eine Datei `~/needed_modules.txt` mit folgendem Inhalt erstellt wird:

```
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 der folgende Befehl ausgeführt wird:

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

Damit verfügt das Java gateway über alle erforderlichen Module für die Arbeit mit jmx-remoting. Anschließend muss nur noch das Java gateway neu gestartet werden. Warten Sie dann einen Moment, und wenn Sie alles richtig gemacht haben, werden JMX-Monitoring-Daten in Zabbix eintreffen (siehe auch: Neueste Daten).

14 ODBC-Überwachung

Übersicht

Die ODBC-Überwachung entspricht dem Datenpunkt-Typ **Datenbankmonitor** im Zabbix-Frontend.

ODBC ist eine Middleware-API 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 verbindet sich Zabbix nicht direkt mit den Datenbanken, sondern verwendet die in ODBC eingerichtete ODBC-Schnittstelle und die zugehörigen Treiber. Dies ermöglicht eine effizientere Überwachung verschiedener Datenbanken für mehrere Zwecke (zum Beispiel die Prüfung bestimmter Datenbank-Warteschlangen, Nutzungsstatistiken usw.).

Zabbix unterstützt unixODBC, eine der am häufigsten verwendeten Open-Source-Implementierungen der ODBC-API.

Attention:
Siehe auch: bekannte Probleme für ODBC-Prüfungen.

Installation von unixODBC

Die empfohlene Methode zur Installation von unixODBC ist die Verwendung der standardmäßigen Paket-Repositories des Linux-Betriebssystems. In den gängigsten Linux-Distributionen ist unixODBC standardmäßig im Paket-Repository enthalten. Falls keine Pakete verfügbar sind, können die Quelldateien auf der unixODBC-Homepage bezogen werden: http://www.unixodbc.org/download.html.

Verwenden Sie zur Installation von unixODBC 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 die ODBC-Unterstützung zu aktivieren, sollte Zabbix mit der folgenden Konfigurationsoption kompiliert werden:

`--with-unixodbc[=ARG] # ODBC-Treiber mit dem 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.

Parameter	Beschreibung
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.

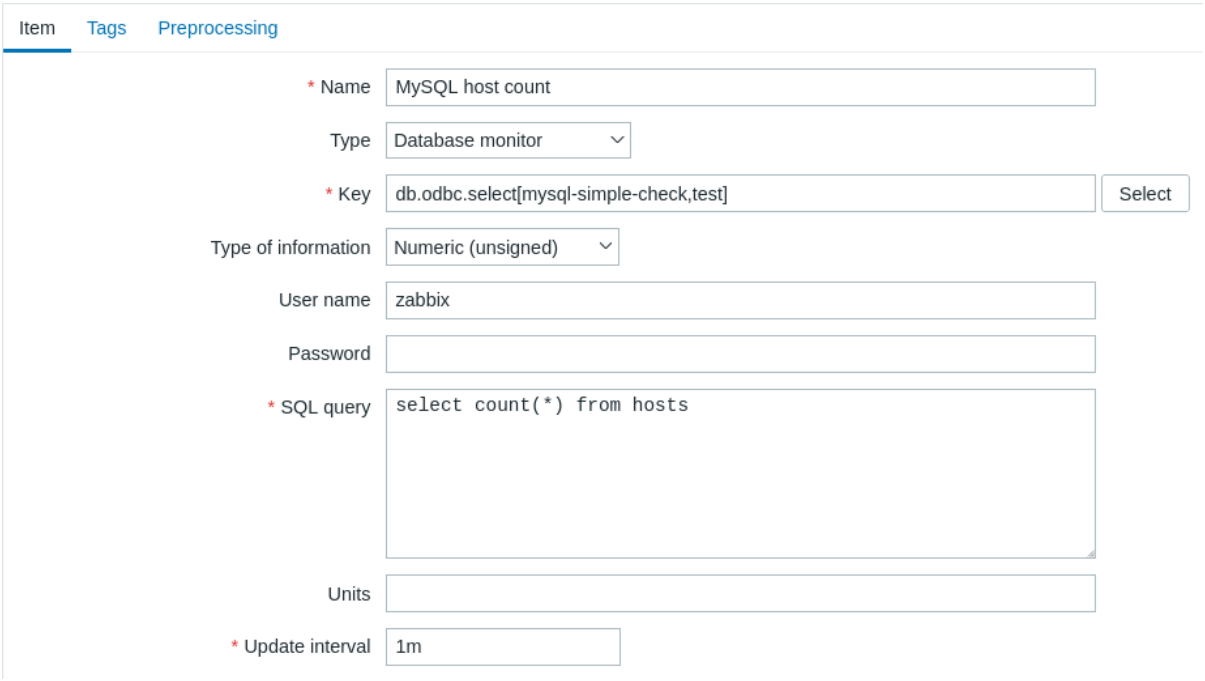
ODBC-Verbindung testen

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                                  |
|                                       |
+---------------------------------------+
```

Datenpunkt-Konfiguration im Zabbix Frontend

Konfigurieren Sie einen **Datenbanküberwachungs**-Datenpunkt.



Alle erforderlichen 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 Datenpunktschlüssel 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 Low-Level-Discovery-Daten zurück.

User name	Geben Sie den Benutzernamen der Datenbank ein (bis zu 255 Zeichen). Dieser Parameter ist optional, wenn der Datenbankbenutzername 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 Passwort des Datenbankbenutzers 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. Das Passwort wird in der Verbindungszeichenfolge nach dem Benutzernamen angehängt, zum Beispiel als `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 Informationstyp aus, der von der Abfrage zurückgegeben wird. Wenn der Informationstyp falsch ausgewählt wird, wird der Datenpunkt nicht unterstützt.

Wichtige Hinweise

- Datenbanküberwachungs-Datenpunkte werden zu „nicht unterstützt“, wenn in der Konfiguration von Server oder Proxy keine *odbc poller*-Prozesse gestartet werden. Um ODBC-Poller zu aktivieren, setzen Sie den Parameter `StartODBCPollers` in der Zabbix-server-Konfigurationsdatei oder, bei Prüfungen, die vom Proxy durchgeführt werden, in der Zabbix-proxy-Konfigurationsdatei.
- Der Wert des Parameters *Timeout* im Formular zur Datenpunkt-Konfiguration wird als ODBC-Anmelde-Timeout und als Timeout für die Abfrageausführung verwendet. Beachten Sie, dass diese Timeout-Einstellungen möglicherweise ignoriert werden, wenn der installierte ODBC-Treiber sie nicht unterstützt.
- Der SQL-Befehl muss wie jede Abfrage, die die Anweisung `select` verwendet, eine Ergebnismenge zurückgeben. Die Abfragesyntax hängt vom RDBMS ab, das sie verarbeitet. Die Syntax einer Anforderung 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>]

 Wandelt das Ergebnis der SQL-Abfrage in ein JSON-Array um.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **unique short description** - eine eindeutige Kurzbeschreibung 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.
- Es können mehrere Zeilen/Spalten im JSON-Format zurückgegeben werden. Dieser Datenpunkt kann als Master-Datenpunkt verwendet werden, der alle Daten in einem Systemaufruf erfasst, 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.

15 Abhängige Datenpunkte

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.

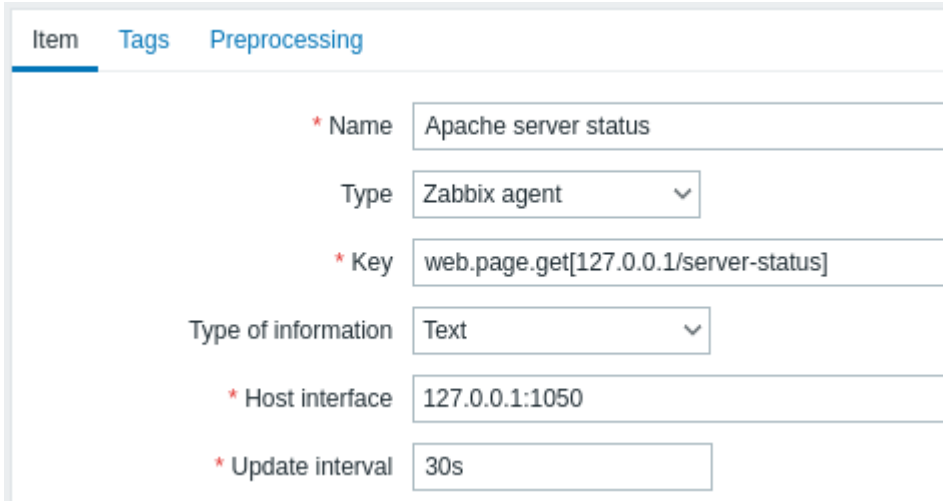
Einschränkungen

- 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

Konfiguration von Datenpunkten

Ein abhängiger Datenpunkt ist für Daten von seinem Master-Datenpunkt abhängig. Deshalb muss der **Master-Datenpunkt** zuerst konfiguriert werden (oder bereits vorhanden sein):

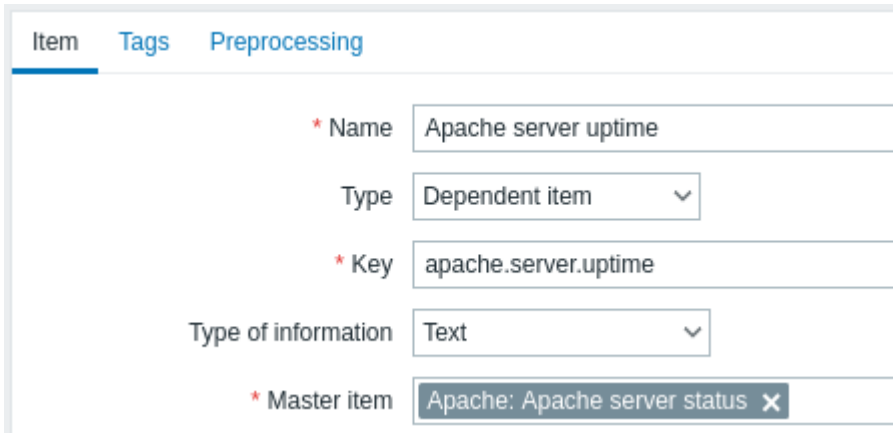
- Gehen Sie zu: *Datenerfassung* → *Hosts*
- Klicken Sie in der Zeile des Hosts auf *Datenpunkte*
- Klicken Sie auf *Datenpunkt erstellen*
- Geben Sie die Parameter des Datenpunkts im Formular ein



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Klicken Sie auf *Hinzufügen*, um den Master-Datenpunkt zu speichern.

Anschließend können Sie einen **abhängigen Datenpunkt** konfigurieren.



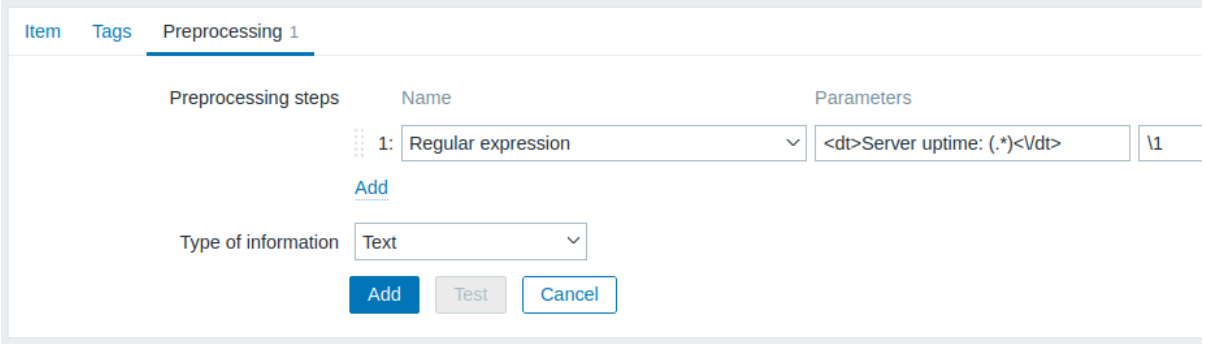
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

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 Erkennung des Datenpunkts verwendet wird.
Master-Datenpunkt	Wählen Sie den Master-Datenpunkt aus. Der Wert des Master-Datenpunkts wird verwendet, um den Wert des abhängigen Datenpunkts zu befüllen.
Informationstyp	Wählen Sie den Informationstyp aus, der dem Format der zu speichernden Daten entspricht.

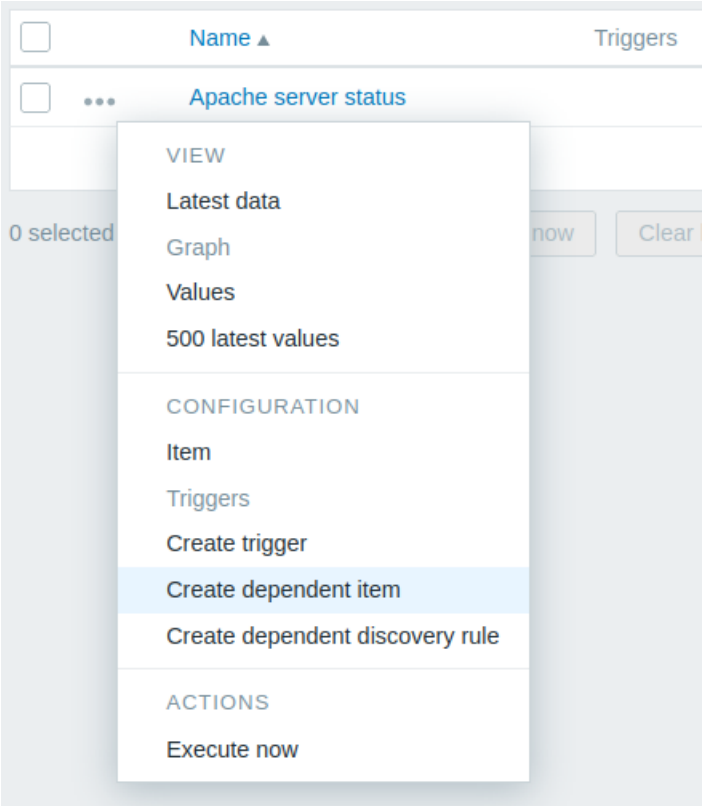
Sie können die Vorverarbeitung von Datenpunktwerten verwenden, um den erforderlichen Teil des Werts des Master-Datenpunkts zu extrahieren.



Ohne Vorverarbeitung ist der Wert des abhängigen Datenpunkts genau derselbe wie der Wert des Master-Datenpunkts.

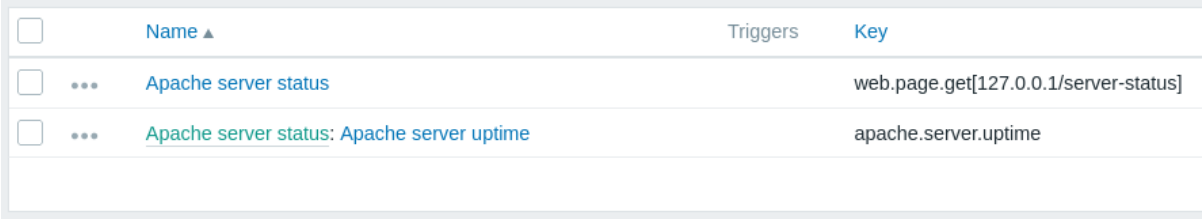
Klicken Sie auf *Hinzufügen*, um den abhängigen Datenpunkt zu speichern.

Eine Verknüpfung zum schnelleren Erstellen eines abhängigen Datenpunkts ist verfügbar, indem Sie in der Datenpunktliste auf die Schaltfläche ••• 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.

16 HTTP-Agent

Übersicht

Dieser Datenpunkttyp ermöglicht die Datenabfrage über das HTTP/HTTPS-Protokoll. Auch Trapping ist mit dem Hilfsprogramm Zabbix sender oder dem Zabbix sender-Protokoll (zum Senden von Daten an den Zabbix Server oder Proxy) oder mit der API-Methode `history.push` (zum Senden von Daten an den Zabbix Server) möglich.

HTTP-Datenpunkt-Prüfungen werden vom Zabbix Server ausgeführt. Wenn Hosts jedoch von einem Zabbix Proxy überwacht werden, werden HTTP-Datenpunkt-Prüfungen vom Proxy ausgeführt.

HTTP-Datenpunkt-Prüfungen erfordern keinen Agent, der auf einem überwachten Host ausgeführt wird.

HTTP Agent unterstützt sowohl HTTP als auch HTTPS. Zabbix folgt Weiterleitungen optional (siehe die Option *Weiterleitungen folgen* unten). Die maximale Anzahl von Weiterleitungen ist fest auf 10 codiert (unter Verwendung der cURL-Option CURLOPT_MAXREDIRS).

> **Attention:**
> 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 asynchroner HTTP-Agent-Poller wird durch den Parameter StartHTTPAgentPollers festgelegt.

Die cURL-Funktion für persistente Verbindungen wurde seit Zabbix 7.0 zu HTTP-Agent-Prüfungen hinzugefügt.

Konfiguration

So konfigurieren Sie einen HTTP-Datenpunkt:

- Gehen Sie zu: *Datenerfassung* → *Hosts*
- Klicken Sie in der Zeile des Hosts auf *Datenpunkte*
- Klicken Sie auf *Datenpunkt erstellen*
- Geben Sie die Parameter des Datenpunkts im Formular ein

Item | Tags | Preprocessing

* Name: HTTP agent item

Type: HTTP agent

* Key: http_value_search [Select]

Type of information: Numeric (unsigned)

* URL: http://localhost:9200/str/values/_search [Parse]

Query fields:

Name		Value	
scroll	⇒	10s	Remove

Add

Request type: POST

Request body type: Raw data | JSON data | XML data

Request body:

```
{
  "query":{
    "bool":{
      "must":{
       "match":{
         "itemid":28275
}
```

Headers:

Name		Value	
name	⇒	value	Remove

Add

Required status codes: 200

Follow redirects: ☑

Retrieve mode: Body | Headers | Body and headers

Convert to JSON: ☐

HTTP proxy: [protocol://][user[:password]@]proxy.example.com[:port]

HTTP authentication: None

SSL verify peer: ☐

SSL verify host: ☐

SSL certificate file:

SSL key file:

SSL key password:

Host interface: 127.0.0.1:10050

Units:

* Update interval: 1m

Custom intervals:

Type	Interval	Period	Action
Flexible | Scheduling	50s	1-7,00:00-24:00	Remove

Add

* Timeout: Global | Override 3 Timeouts

* History: Do not store | Store up to 31d

* Trends: Do not store | Store up to 365d

Value mapping: type here to search [Select]

Enable trapping: ☐

Populates host inventory field: -None-

Description:

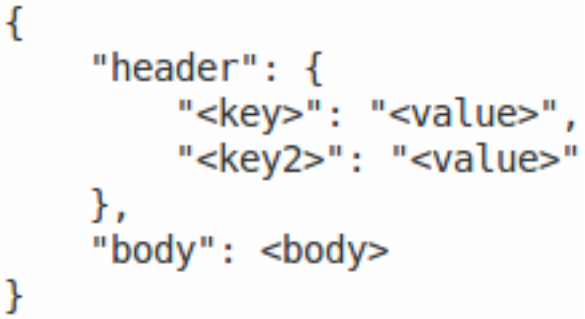
Enabled: ☑

Add | Test | Cancel

Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

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. Zum Beispiel: https://www.example.com http://www.example.com/download Domainnamen können mit Unicode-Zeichen angegeben werden. Sie werden bei der Ausführung der HTTP-Prüfung automatisch per Punycode in ASCII umgewandelt. Mit der Schaltfläche *Parse* können optionale Abfragefelder (wie ?name=Admin&password=mypassword) von der URL getrennt werden, wobei die Attribute und Werte zur automatischen URL-Kodierung in *Abfragefelder* verschoben werden. Begrenzt auf 2048 Zeichen. Unterstützte Makros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, Benutzermakros, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_URL.
Abfragefelder	Variablen für die URL (siehe oben). Werden als Attribut- und Wertpaare 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, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_URL.
Anfragetyp	Wählen Sie den Typ der Anfragemethode: *GET*, *POST*, *PUT* oder *HEAD*
Typ des Anfragetexts	Wählen Sie den Typ des Anfragetexts aus: **Rohdaten** - benutzerdefinierter HTTP-Anfragetext, Makros werden ersetzt, aber es wird keine Kodierung durchgeführt **JSON-Daten** - HTTP-Anfragetext im JSON-Format. Makros können als Zeichenfolge, Zahl, true und false verwendet werden; Makros, die als Zeichenfolgen verwendet werden, müssen in doppelte Anführungszeichen gesetzt werden. Werte aus Makros werden aufgelöst und anschließend automatisch maskiert. Wenn ”Content-Type” nicht in den Headern angegeben ist, wird standardmäßig ”Content-Type: application/json” verwendet **XML-Daten** - HTTP-Anfragetext im XML-Format. Makros können als Textknoten, Attribut oder CDATA-Abschnitt verwendet werden. Werte aus Makros werden aufgelöst und anschließend in einem Textknoten und Attribut automatisch maskiert. Wenn ”Content-Type” nicht in den Headern angegeben ist, wird standardmäßig ”Content-Type: application/xml” verwendet
Anfragetext	Geben Sie den Anfragetext ein. Unterstützte Makros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, Benutzermakros, Low-Level-Discovery-Makros.
Header	Benutzerdefinierte HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Werden als Attribut- und Wertpaare angegeben. Unterstützte Makros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, Benutzermakros, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_HTTPHEADER.
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 leer, wird keine Prüfung durchgeführt. Zum Beispiel: 200,201,210-299 Unterstützte Makros in der Liste: Benutzermakros, Low-Level-Discovery-Makros. Dies verwendet die cURL-Option CURLINFO_RESPONSE_CODE.
Weiterleitungen folgen	Aktivieren Sie das Kontrollkästchen, um HTTP-Weiterleitungen zu folgen. Dies setzt die cURL-Option CURLOPT_FOLLOWLOCATION.
Abrufmodus	Wählen Sie den Teil der Antwort aus, der abgerufen werden soll: **Body** - nur Body **Header** - nur Header **Body und Header** - Body und Header

Parameter	Beschreibung
In JSON umwandeln	Header werden unter dem Schlüssel ”header” als Attribut- und Wertpaare gespeichert, wenn *Abrufmodus* nicht auf *Body* gesetzt ist. Wenn ’Content-Type: application/json’ erkannt wird, wird der Body als Objekt gespeichert, andernfalls als Zeichenfolge, zum Beispiel: 
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 für Protokollpräfixe 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 Port 1080 verwendet. Falls angegeben, überschreibt der Proxy proxy-bezogene Umgebungsvariablen wie http_proxy, HTTPS_PROXY. Falls nicht angegeben, überschreibt der Proxy keine proxy-bezogenen Umgebungsvariablen. Der eingegebene Wert wird unverändert übergeben; es findet keine Plausibilitätsprüfung statt. Beachten Sie, dass beim HTTP-Proxy nur 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, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_PROXY.
HTTP-Authentifizierung	Wählen Sie die Authentifizierungsoption aus: **Keine** - es wird keine Authentifizierung verwendet; **Basic** - Basic-Authentifizierung wird verwendet; **NTLM** - NTLM-Authentifizierung (Windows NT LAN Manager) wird verwendet; **Kerberos** - Kerberos-Authentifizierung wird verwendet (siehe auch: Konfigurieren von Kerberos mit Zabbix); **Digest** - Digest-Authentifizierung wird verwendet. Dies setzt die cURL-Option CURLOPT_HTTPAUTH.
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 Low-Level-Discovery-Makros 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 Low-Level-Discovery-Makros werden unterstützt.
SSL Peer verifizieren	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des Webservers zu verifizieren. Das Serverzertifikat wird automatisch aus dem systemweiten Speicherort der Zertifizierungsstelle (CA) übernommen. Sie können den Speicherort der CA-Dateien mit dem Konfigurationsparameter SSLCALocation von Zabbix Server oder Proxy überschreiben. Dies setzt die cURL-Option CURLOPT_SSL_VERIFYPEER.
SSL Host verifizieren	Aktivieren Sie das Kontrollkästchen, um zu verifizieren, dass das Feld Common Name oder Subject Alternate Name des Webserver-Zertifikats übereinstimmt. Dies setzt die cURL-Option CURLOPT_SSL_VERIFYHOST.

Parameter	Beschreibung
SSL-Zertifikatsdatei	Name der SSL-Zertifikatsdatei, die für die Client-Authentifizierung verwendet wird. Die Zertifikatsdatei muss im PEM[1]-Format vorliegen. Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie das Feld für die SSL-Schlüsseldatei leer. Wenn der Schlüssel verschlüsselt ist, geben Sie das Passwort im Feld für das SSL-Schlüsselpasswort an. Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLCertLocation von Zabbix Server oder Proxy angegeben. Unterstützte Makros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, Benutzermakros, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_SSLCERT.
SSL-Schlüsseldatei	Name der SSL-Datei mit dem privaten Schlüssel, die für die Client-Authentifizierung verwendet wird. Die Datei mit dem privaten Schlüssel muss im PEM[1]-Format vorliegen. Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLKeyLocation von Zabbix Server oder Proxy angegeben. Unterstützte Makros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, Benutzermakros, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_SSLKEY.
SSL-Schlüsselpasswort	Passwort der Datei mit dem privaten SSL-Schlüssel. Unterstützte Makros: Benutzermakros, Low-Level-Discovery-Makros. Dies setzt die cURL-Option CURLOPT_KEYPASSWD.
Timeout	Zabbix verwendet nicht mehr als die festgelegte Zeit für die Verarbeitung der URL (1-600 Sekunden). Tatsächlich definiert dieser Parameter die maximale Zeit für den Verbindungsaufbau zur URL und die maximale Zeit für die Durchführung einer HTTP-Anfrage. Daher verwendet Zabbix für eine Prüfung nicht mehr als 2 x *Timeout* Sekunden. Dies setzt die cURL-Option CURLOPT_TIMEOUT. Weitere Informationen zum Parameter *Timeout* finden Sie unter allgemeine Datenpunktattribute.
Trapping aktivieren	Wenn dieses Kontrollkästchen aktiviert ist, funktioniert der Datenpunkt zusätzlich als Trapper-Datenpunkt und akzeptiert Daten, die mit dem Dienstprogramm Zabbix sender oder dem Zabbix-sender-Protokoll an Zabbix Server oder Proxy gesendet werden, oder akzeptiert Daten, die mit der API-Methode `history.push` an Zabbix Server gesendet werden. Weitere Informationen zum Senden von Daten finden Sie unter: Trapper-Datenpunkte.
Erlaubte Hosts	Nur sichtbar, wenn das Kontrollkästchen *Trapping aktivieren* aktiviert ist. Liste von durch Kommas getrennten IP-Adressen, optional in CIDR-Notation, oder DNS-Namen. Falls angegeben, werden eingehende Verbindungen nur von den hier aufgeführten Hosts akzeptiert. Wenn 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. 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 erlaubt. Host-Makros: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} sind in diesem Feld erlaubt.

Note:
Wenn das Feld *HTTP-Proxy* leer gelassen wird, besteht eine weitere Möglichkeit zur Verwendung eines HTTP-Proxys darin, proxy-bezogene Umgebungsvariablen zu setzen.
Für HTTP - setzen Sie die Umgebungsvariable `http_proxy` für den Zabbix Server-Benutzer. 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 Details erhalten Sie durch Ausführen des Shell-Befehls: *# man curl*.

Attention:
[1] Zabbix unterstützt Zertifikats- und private Schlüsseldateien nur im PEM-Format. Falls Ihre Zertifikats- und privaten Schlüsseldaten im Dateiformat PKCS #12 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 zum Abrufen von Daten von Diensten wie Elasticsearch:

- Erstellen Sie ein GET-Element mit 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-Vorverarbeitungsschritts: `$.version.number`

Beispiel 2

Senden Sie einfache POST-Anfragen zum Abrufen von Daten von Diensten wie Elasticsearch:

- Erstellen Sie ein POST-Element mit URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Konfigurieren Sie den folgenden POST-Body, um die Prozessorlast zu ermitteln (1 Minute-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,
```

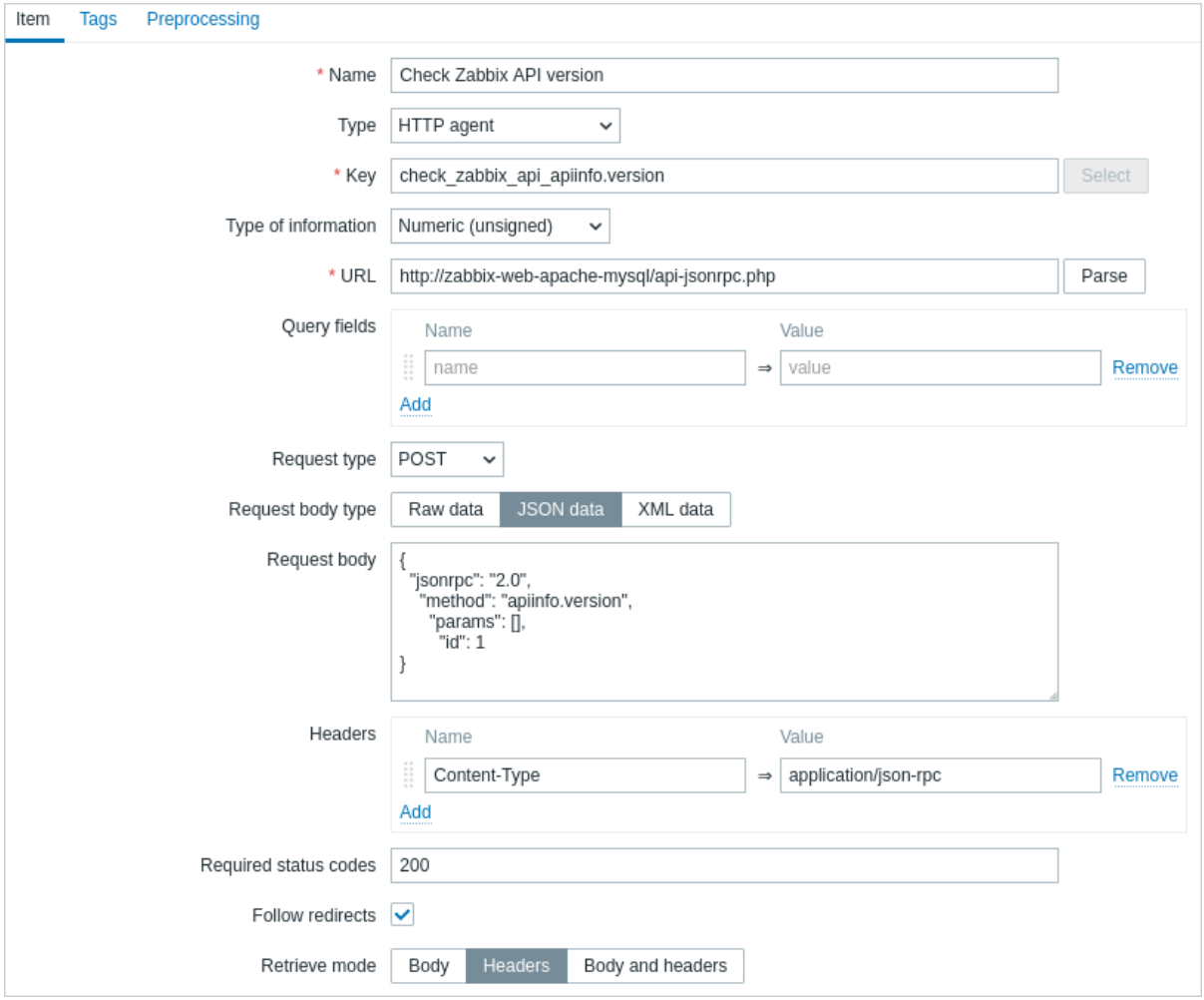
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_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 Elementwert zu erhalten: `$.hits.hits[0]._source.value`

Beispiel 3

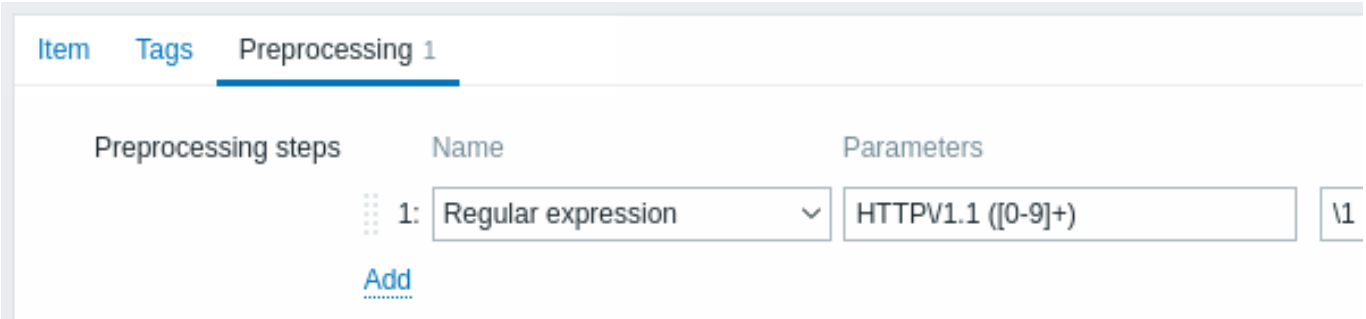
Prüfen, ob die Zabbix API aktiv ist, mit apiinfo.version.

- Konfiguration des Datenpunkts:

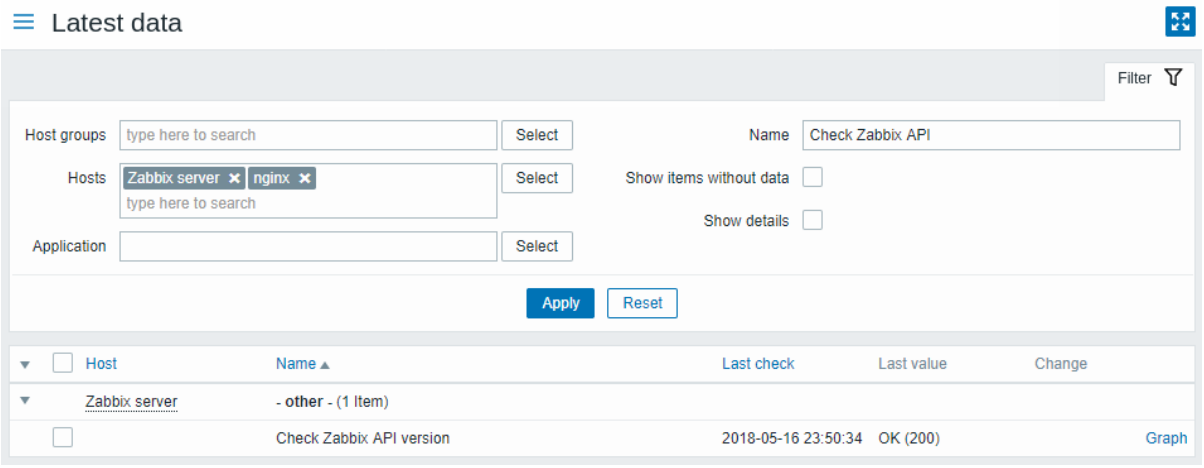


Beachten Sie die Verwendung der POST-Methode mit JSON-Daten, die Festlegung von Anforderungs-Headern und die Aufforderung, nur Header zurückzugeben:

- Vorverarbeitung von Elementwerten mit regulären Ausdrücken, um HTTP-Code zu erhalten:



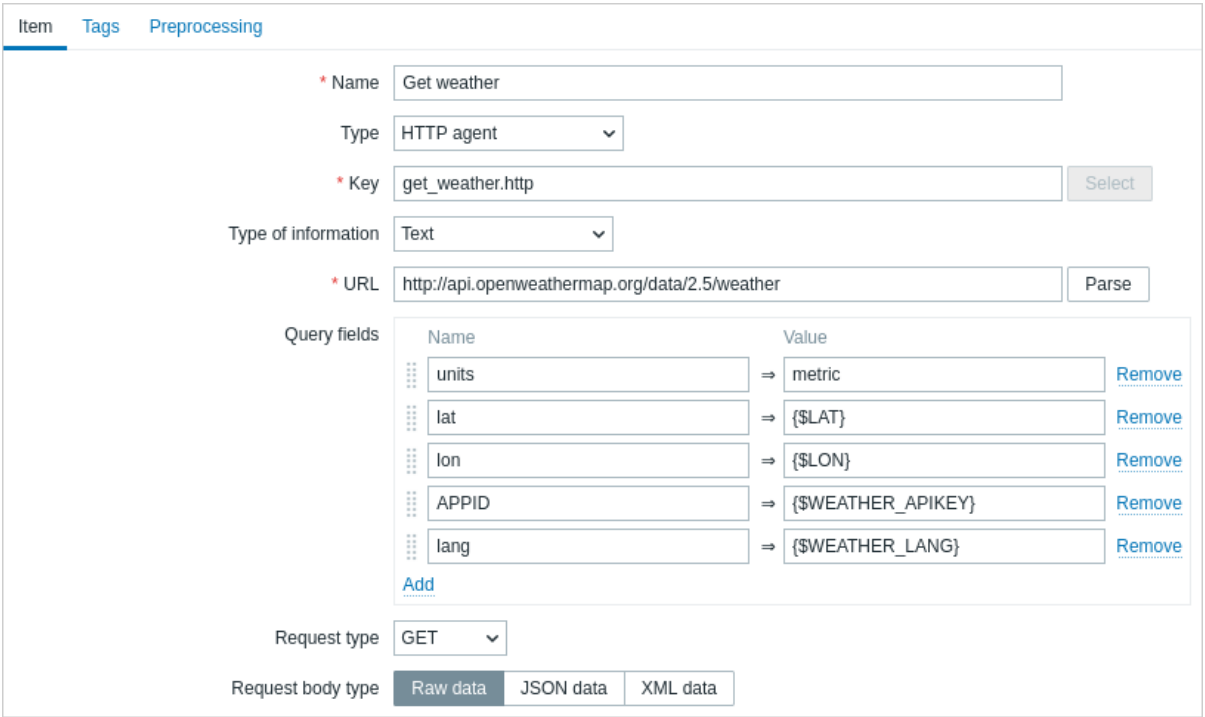
- Prüfen des Ergebnisses in *Latest Data*:



Beispiel 4

Abruf von Wetterinformationen durch Verbindung mit dem öffentlichen Dienst Openweathermap.

- Konfigurieren Sie ein Hauptelement für die Massendatenerfassung in einem einzigen JSON:



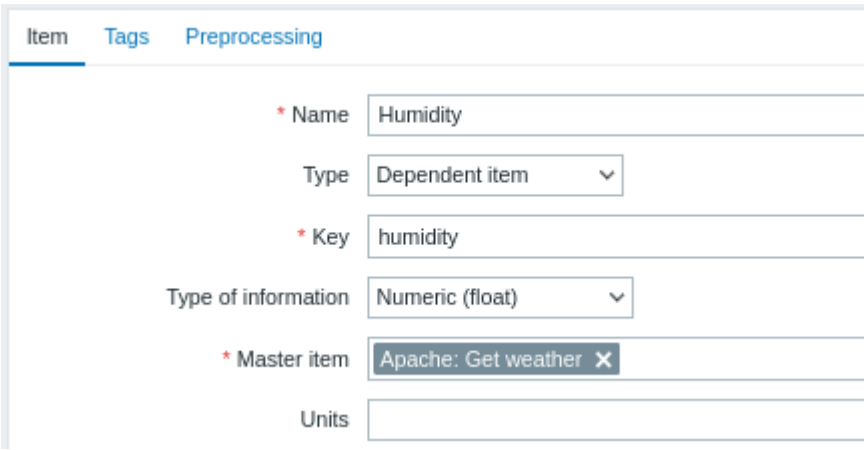
Beachten Sie die Verwendung von Makros in Abfragefeldern. Beachten Sie die Openweathermap API for how to fill them.

Beispiel für JSON, das als Antwort an den HTTP-Agenten 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
    }
}
```

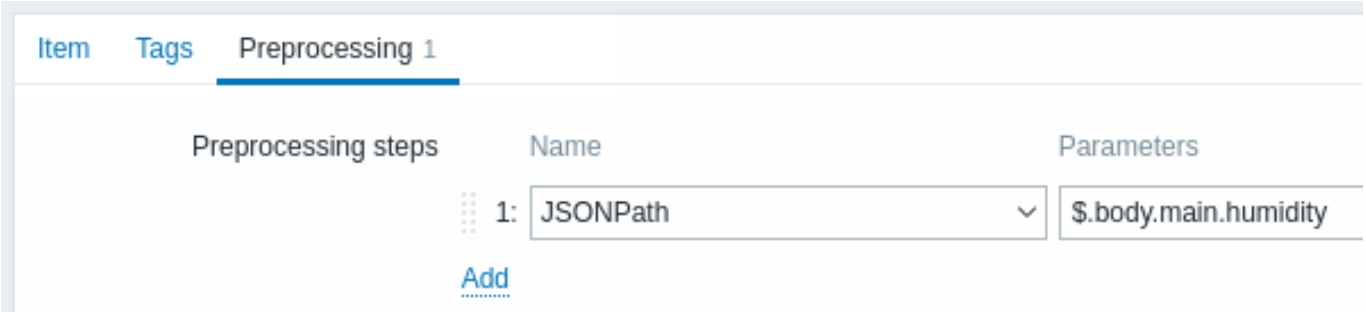

Die nächste Aufgabe besteht darin, abhängige Elemente zu konfigurieren, die Daten aus der JSON-Datei extrahieren.

- Konfigurieren Sie eine probenabhängige Position für Feuchtigkeit:

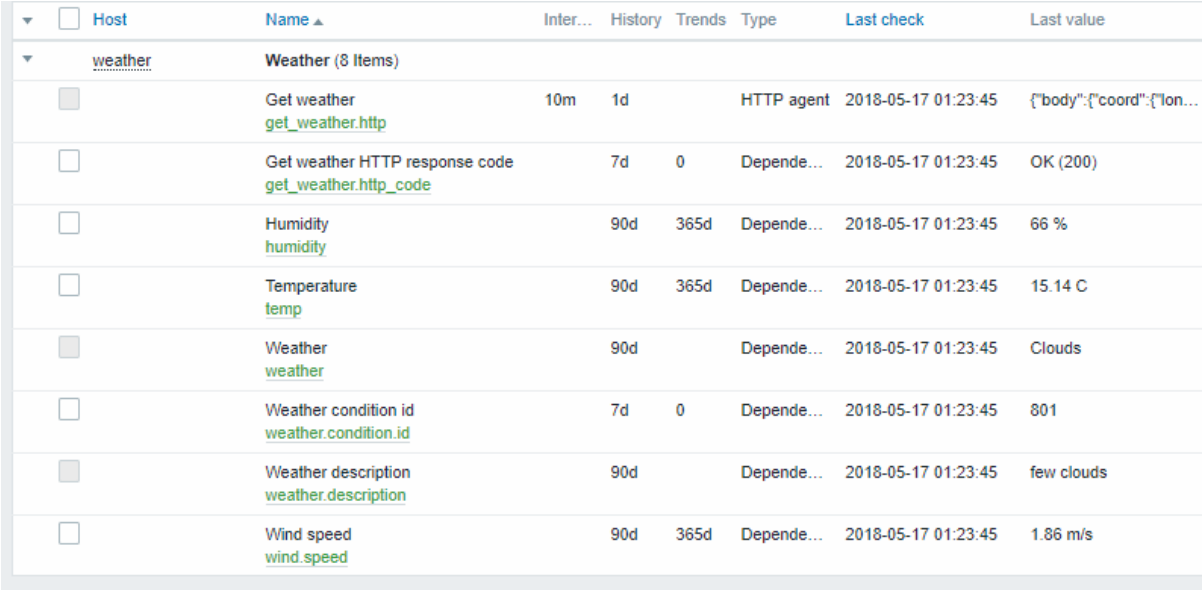


Andere Wetterdaten wie ”Temperatur” werden auf die gleiche Weise hinzugefügt.

- Beispiel für die Vorverarbeitung abhängiger Positionswerte mit JSONPath:



- Prüfen Sie das Ergebnis der Wetterdaten in *Latest data*:

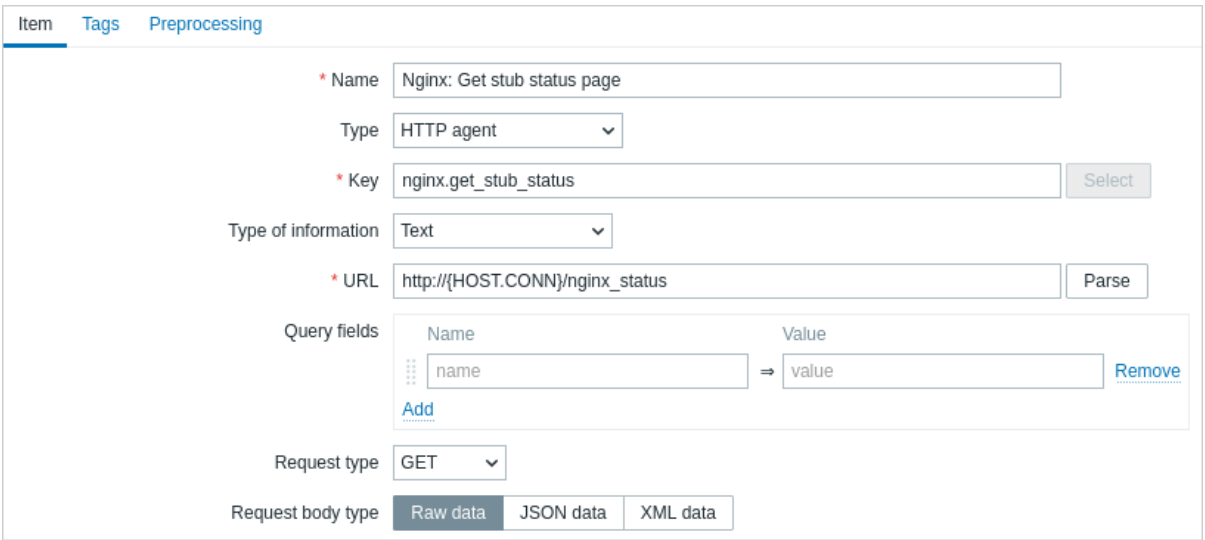


Host	Name	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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 herstellen und ihre Metriken gesammelt abrufen.

- Konfigurieren Sie Nginx gemäß der offiziellen Anleitung.
- Konfigurieren Sie einen Master-Datenpunkt für die Massendatenerfassung:

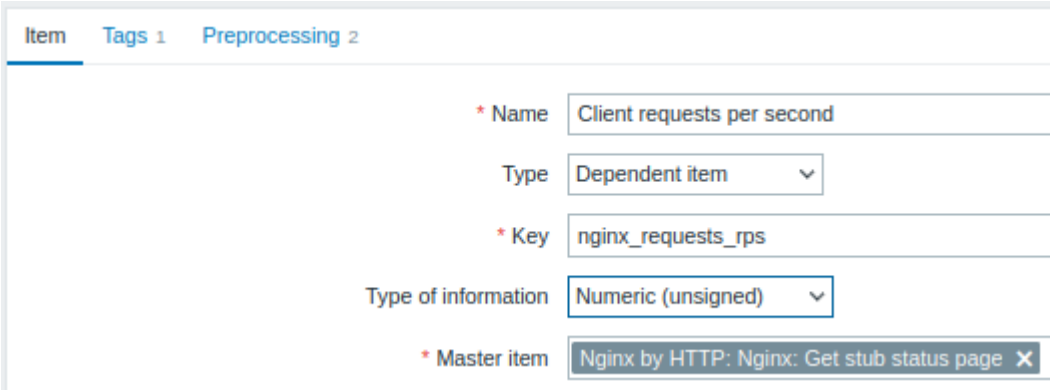


Beispielausgabe des Nginx-Stub-Status:

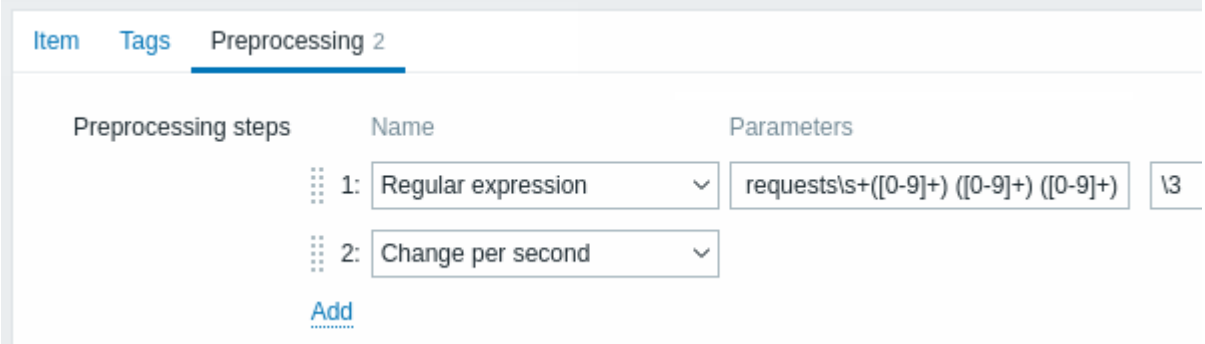
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Die nächste Aufgabe besteht darin, abhängige Datenpunkte zu konfigurieren, die Daten extrahieren.

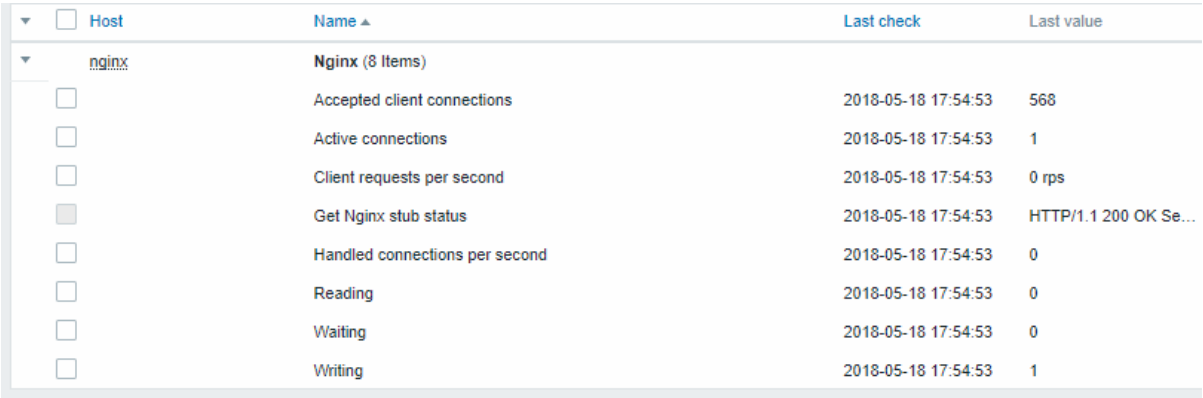
- Konfigurieren Sie einen beispielhaften abhängigen Datenpunkt für Anfragen pro Sekunde:



- Beispiel für die Vorverarbeitung des Werts eines abhängigen Datenpunkts mit dem regulären Ausdruck `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Prüfen Sie das vollständige Ergebnis des Stub-Moduls unter *Neueste Daten*:



17 Prometheus-Prüfungen

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 Vorverarbeitungsoptionen für Prometheus-Daten:

- *Prometheus-Muster* - wird in normalen Datenpunkten verwendet, um Prometheus-Daten abzufragen
- *Prometheus zu JSON* - wird in normalen Datenpunkten und für Low-Level-Discovery verwendet. In diesem Fall werden die abgefragten Prometheus-Daten im JSON-Format zurückgegeben.

Stapelverarbeitung

Die Stapelverarbeitung wird für abhängige Datenpunkte unterstützt. Um Caching und Indizierung zu aktivieren, muss die Vorverarbeitung *Prometheus pattern* der **erste** Vorverarbeitungsschritt sein. Wenn *Prometheus pattern* der erste Vorverarbeitungsschritt ist, werden die geparsten Prometheus-Daten zwischengespeichert und anhand der ersten Bedingung `<label>==<value>` im

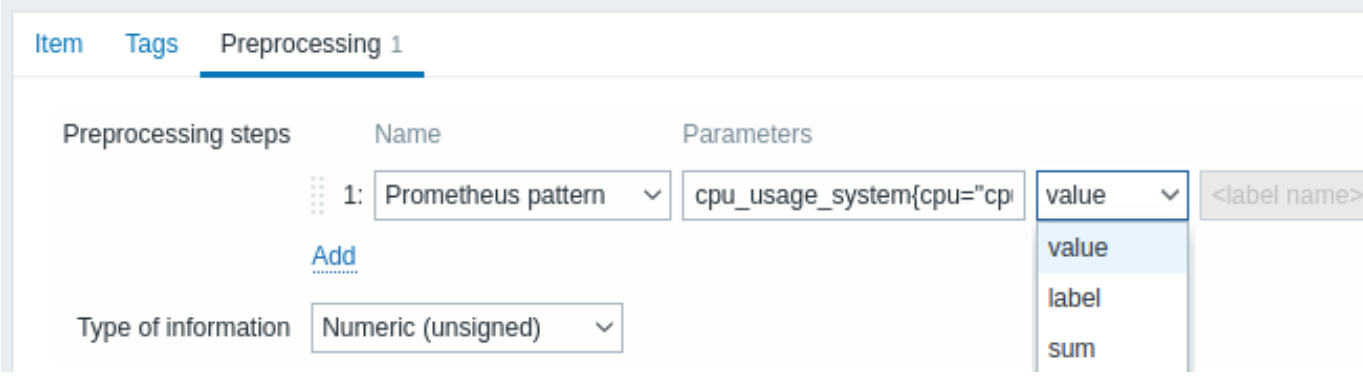
Vorverarbeitungsschritt *Prometheus pattern* indiziert. Dieser Cache wird bei der Verarbeitung anderer abhängiger Datenpunkte in diesem Stapel wiederverwendet. Für eine optimale Leistung sollte das erste Label dasjenige mit den meisten unterschiedlichen Werten sein.

Falls vor dem ersten Schritt weitere Vorverarbeitung erforderlich ist, sollte sie entweder in den Master-Datenpunkt verschoben werden oder in einen neuen abhängigen Datenpunkt, der als Master-Datenpunkt für die abhängigen Datenpunkte verwendet wird.

Konfiguration

Vorausgesetzt, dass Sie den HTTP-Master-Datenpunkt konfiguriert haben, müssen Sie einen abhängigen Datenpunkt erstellen, der einen Prometheus-Präprozessierungsschritt verwendet:

- Geben Sie die allgemeinen Parameter für den abhängigen Datenpunkt im Konfigurationsformular ein
- Wechseln Sie auf die Registerkarte „Preprocessing“
- Wählen Sie eine Prometheus-Präprozessierungsoption aus (*Prometheus pattern* oder *Prometheus to JSON*)



Die folgenden Parameter sind spezifisch für die Präprozessierungsoption *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 Label-Namen {<label name>=~”<regex>”,...} - Auswahl nach Label-Namen, die einem regulären Ausdruck entsprechen {__name__=~”.*”}==<value> - Auswahl nach Metrikwert Oder eine Kombination aus den obigen Angaben: <metric name>{<label1 name>=”<label1 value>”,<label2 name>=~”<regex>”,...}==<value> Der Label-Wert kann eine beliebige Folge von UTF-8-Zeichen sein, aber Backslash, doppeltes Anführungszeichen und Zeilenvorschub müssen jeweils als \\, \" und \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 mit mehreren Zeilen übereinstimmt 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 zur Verwendung der Parameter.
Output	Definieren Sie den Label-Namen (optional). In diesem Fall wird der dem Label-Namen entsprechende Wert zurückgegeben. Dieses Feld ist nur für die Option *Prometheus pattern* verfügbar, wenn im Feld *Result processing* „Label“ ausgewählt ist.	

Beispiele für die Verwendung von Parametern

1. Der häufigste Anwendungsfall ist die Rückgabe des **Werts**. Um den Wert von `/var/db` aus

   `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`

zurückzugeben, verwenden Sie die folgenden Parameter:

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - wählen Sie „value“

2. Möglicherweise interessiert Sie auch der **Durchschnittswert** aller Parameter `node_disk_usage_bytes`:

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - wählen Sie „avg“

3. Obwohl Prometheus nur numerische Daten unterstützt, ist es üblich, einen Workaround zu verwenden, der es ermöglicht, auch die relevante textuelle Beschreibung zurückzugeben. Dies kann mit einem Filter und durch Angabe des Labels erreicht werden. Um also den Wert des Labels „color“ aus

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

zurückzugeben, verwenden Sie die folgenden Parameter:

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - wählen Sie „label“
- *Label* - geben Sie „color“ an

Der Filter (basierend auf dem numerischen Wert „1“) findet die entsprechende Zeile, während das Label die Beschreibung des Gesundheitsstatus zurückgibt (derzeit „green“, potenziell aber auch „red“ oder „yellow“).

Prometheus zu JSON

Daten aus Prometheus können für die Low-Level-Discovery verwendet werden. In diesem Fall werden Daten im JSON-Format benötigt, und die Vorverarbeitungsoption *Prometheus zu JSON* liefert genau diese.

Weitere Details finden Sie unter Discovery using Prometheus data.

Vergleich der Abfragesprachen

Die folgende Tabelle listet Unterschiede und Gemeinsamkeiten zwischen PromQL und der Zabbix-Prometheus-Preprocessing-Abfragesprache auf.

PromQL instant vector selector		Zabbix-Prometheus-Preprocessing
Unterschiede		
Abfrageziel	Prometheus-Server	Klartext im Prometheus-Expositionsformat
Gibt zurück	Instant-Vektor	Metrik- oder Label-Wert (Prometheus-Muster) Array von Metriken für einen einzelnen Wert in JSON (Prometheus zu JSON)
Label-Matching-Operatoren	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Für das Matching von Labeln oder Metriknamen verwendeter regulärer Ausdruck	RE2	PCRE
Vergleichsoperatoren	Siehe Liste	Für die Wertfilterung wird nur **==** (gleich) unterstützt
Gemeinsamkeiten		
Auswahl nach Metrikname, der einer Zeichenfolge entspricht	<metric name> oder {__name__=”<metric name>”}	<metric name> oder {__name__=”<metric name>”}
Auswahl nach Metrikname, 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>

18 Skript-Datenpunkte

Übersicht

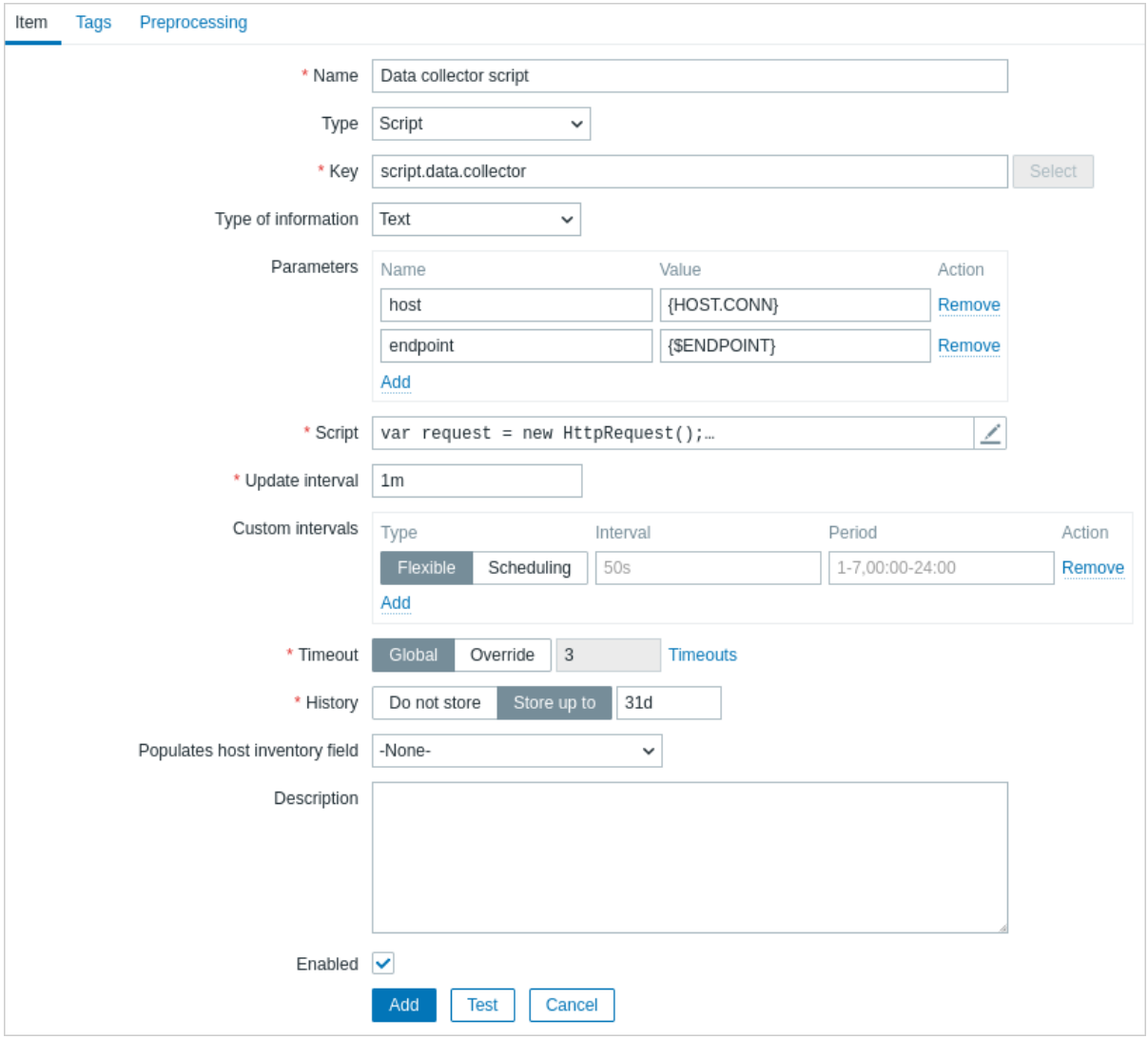
Script-Datenpunkte können verwendet werden, um Daten zu erfassen, indem benutzerdefinierter JavaScript-Code ausgeführt wird, mit der Möglichkeit, Daten über HTTP/HTTPS abzurufen. Zusätzlich zum Script können eine optionale Liste von Parametern (Name-Wert-Paare) sowie ein Timeout angegeben werden.

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“.

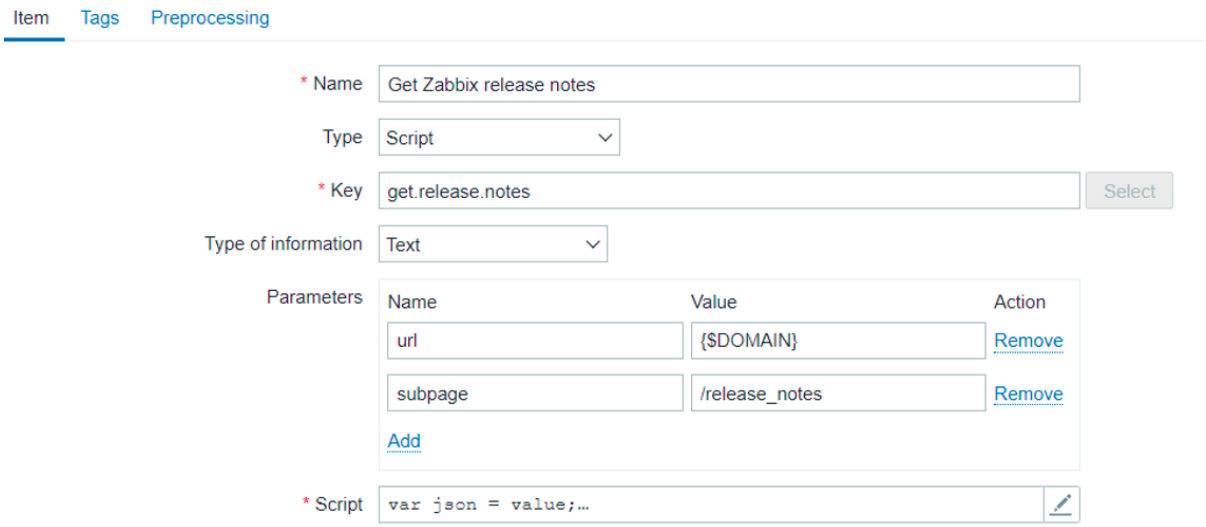
- Geben Sie im Feld *Script* Folgendes ein:

```
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;
```

19 Browser-Datenpunkte

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 einen benutzerdefinierten JavaScript-Code ausführen und Daten über HTTP/HTTPS abrufen. Dieser Datenpunkt kann browserbezogene Aktionen wie Klicken, Texteingabe, Navigation durch Webseiten und andere Benutzerinteraktionen mit Websites oder Webanwendungen simulieren.

Zusätzlich zum Skript kann eine optionale Liste von Parametern (Name-Wert-Paare) sowie ein Timeout angegeben werden.

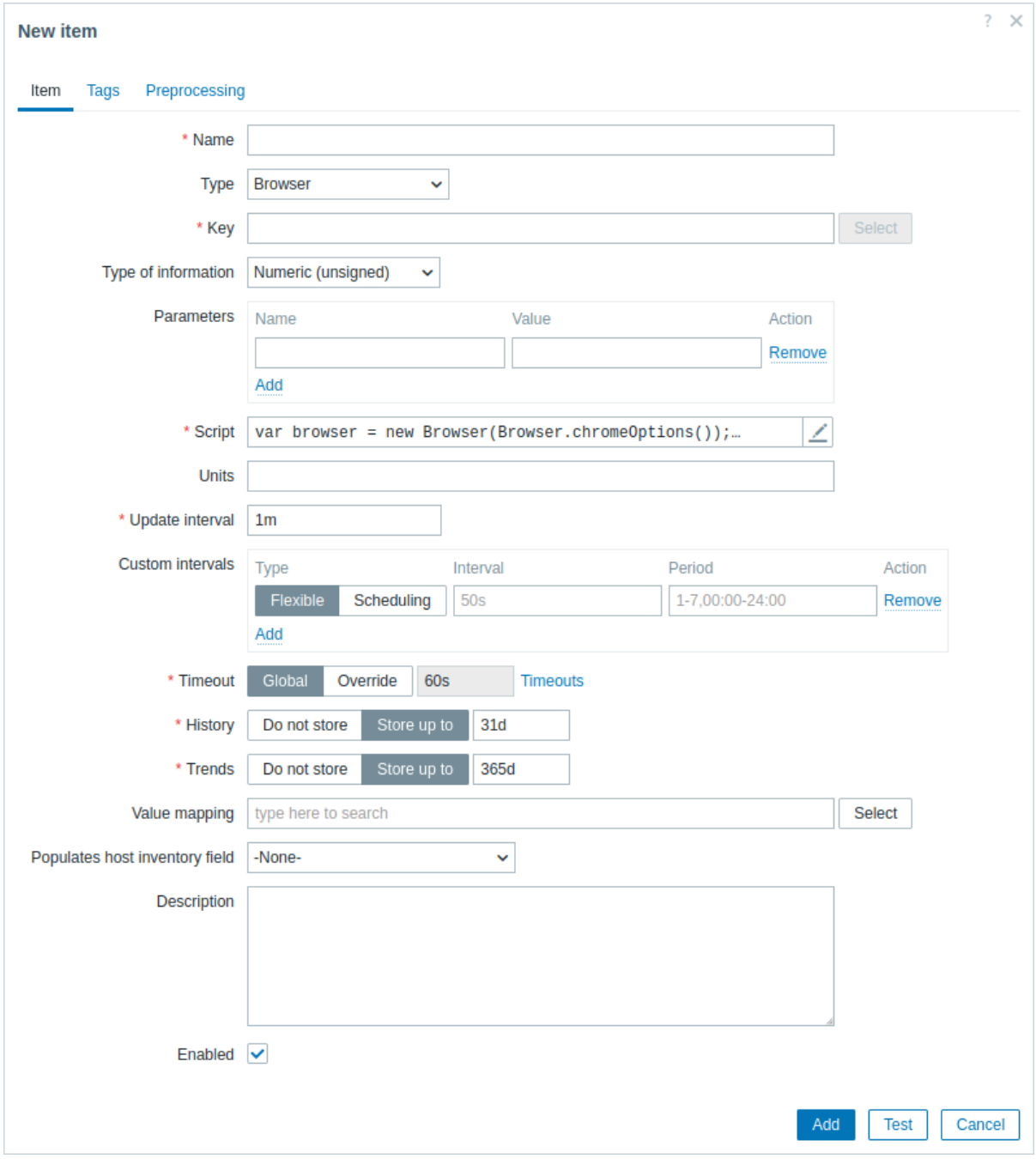
Attention:
Der Datenpunkt implementiert den W3C WebDriver-Standard teilweise und verwendet dabei entweder Selenium Server oder einen einfachen WebDriver (zum Beispiel ChromeDriver) als Endpunkt für Webtests. Damit der Datenpunkt funktioniert, legen Sie den Endpunkt im Konfigurationsparameter `WebDriverURL` der Zabbix-Server-/Proxy-Konfigurationsdatei fest (bei Verwendung von ChromeDriver siehe Security Considerations). Für eine bessere Leistung sollten Sie einen dedizierten Server für die Webtestumgebung in Betracht ziehen.

Prüfungen von Browser-Datenpunkten werden von Browser-Pollern des Zabbix Server oder Proxy ausgeführt und verarbeitet. Falls erforderlich, können Sie die Anzahl der vorab geforkten Instanzen von Browser-Pollern im Konfigurationsparameter `StartBrowserPollers` der Zabbix-Server-/Proxy-Konfigurationsdatei anpassen.

Für die Überwachung komplexer Websites und Webanwendungen steht die Vorlage Website by Browser als vorkonfigurierte 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 für die Einrichtung von Zabbix zur Überwachung von Websites unter Verwendung der Vorlage *Website by Browser* finden Sie unter [Websites mit Browser-Elementen überwachen].(/manual/guides/monitor_browser).

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);
}
```

4 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.

Verlauf aufbewahren

Sie können festlegen, für wie viele Tage der Verlauf aufbewahrt 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 aufzubewahren, um die Datenbank nicht mit zu vielen Verlaufswerten zu überlasten.

Anstatt einen langen Verlauf aufzubewahren, können Sie Trenddaten länger speichern. Zum Beispiel könnten Sie den Verlauf 14 Tage und Trends 5 Jahre lang aufbewahren.

Einen guten Eindruck davon, wie viel Speicherplatz Verlaufsdaten im Vergleich zu Trenddaten benötigen, erhalten Sie auf der Seite zur Datenbankdimensionierung.

Auch bei einer kürzeren Aufbewahrung des Verlaufs können Sie ältere Daten weiterhin in Diagrammen prüfen, 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 history export 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.

5 Benutzerparameter

Übersicht

Manchmal möchten Sie möglicherweise eine Agent-Prüfung ausführen, die nicht bereits vordefiniert mit Zabbix geliefert wird. Hier kommen Benutzerparameter ins Spiel.

Sie können einen Befehl schreiben, der die benötigten Daten abruft, und ihn im Benutzerparameter in der Agent-Konfigurationsdatei (Konfigurationsparameter 'UserParameter') einfügen.

Ein Benutzerparameter hat die folgende Syntax:

```
UserParameter=<key>,<command>
```

Wie Sie sehen, enthält ein Benutzerparameter auch einen Schlüssel. Der Schlüssel wird bei der Konfiguration eines Datenpunkts benötigt. Geben Sie einen Schlüssel Ihrer Wahl ein, auf den leicht verwiesen werden kann (er muss innerhalb eines Hosts eindeutig sein).

Starten Sie den Agent neu oder verwenden Sie die Option Laufzeitsteuerung des Agenten, damit der neue Parameter übernommen wird, z. B.:

```
zabbix_agentd -R userparameter_reload
```

Geben Sie dann beim Konfigurieren eines Datenpunkts den Schlüssel ein, um auf den Befehl aus dem Benutzerparameter zu verweisen, den Sie ausführen möchten.

Benutzerparameter sind Befehle, die vom Zabbix Agent ausgeführt werden. Beachten Sie, dass vor den Schritten der Datenpunkt-Wertvorverarbeitung bis zu 16 MB Daten zurückgegeben werden können.

Unter UNIX-Betriebssystemen wird **/bin/sh** als Befehlszeileninterpreter verwendet. Benutzerparameter unterliegen dem Timeout der Agent-Prüfung; wenn das Timeout erreicht wird, wird der durch den Benutzerparameter gestartete Prozess beendet.

Siehe auch:

- Schritt-für-Schritt-Anleitung zur Verwendung von Benutzerparametern
- Befehlsausführung

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'.

Flexible Benutzerparameter

Flexible Benutzerparameter akzeptieren Parameter mit dem Schlüssel. Auf diese Weise kann ein flexibler Benutzerparameter die Grundlage für die Erstellung mehrerer Datenpunkte sein.

Flexible Benutzerparameter haben die folgende Syntax:

```
UserParameter=key[*],command
```

Parameter	Beschreibung
Key	Eindeutiger Datenpunktschlüssel. Das [*] definiert, dass dieser Schlüssel Parameter innerhalb der Klammern akzeptiert. Die Parameter werden bei der Konfiguration des Datenpunkts angegeben.
Command	Befehl, der zur Ermittlung des Werts des Schlüssels ausgeführt wird. *Nur für flexible Benutzerparameter*: Sie können Positionsreferenzen $1…$9 im Befehl verwenden, um auf den jeweiligen Parameter im Datenpunktschlüssel zu verweisen. Zabbix analysiert die in [ ] des Datenpunktschlüssels eingeschlossenen Parameter und ersetzt $1,…,$9 im Befehl entsprechend. $0 wird durch den ursprünglichen Befehl ersetzt (vor der Erweiterung von $0,…,$9), der ausgeführt werden soll. Positionsreferenzen werden unabhängig davon interpretiert, ob sie in doppelte (") oder einfache (') Anführungszeichen eingeschlossen sind. Um Positionsreferenzen unverändert zu verwenden, geben Sie ein doppeltes Dollarzeichen an - zum Beispiel awk '{print $$2}'. In diesem Fall wird $$2 bei der Ausführung des Befehls tatsächlich zu $2.

Attention:
Positionsreferenzen mit dem Zeichen $ werden nur bei flexiblen Benutzerparametern vom Zabbix Agent gesucht und ersetzt. Bei einfachen Benutzerparametern wird diese Referenzverarbeitung übersprungen, daher ist kein Escaping von $ erforderlich.

Attention:
Bestimmte Zeichen sind in Benutzerparametern standardmäßig nicht erlaubt. Die vollständige Liste der Zeichen finden Sie unter UnsafeUserParameters.

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]
```

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“.

1 Zabbix-Agenten erweitern

Dieses Tutorial bietet eine Schritt-für-Schritt-Anleitung dazu, wie die Funktionalität des Zabbix-Agenten mithilfe eines Benutzerparameters erweitert werden kann.

Schritt 1

Schreiben Sie ein Skript oder eine Befehlszeile, um den erforderlichen Parameter abzurufen.

Zum Beispiel können wir den folgenden Befehl schreiben, um die Gesamtzahl der von einem MySQL-Server ausgeführten Abfragen zu erhalten:

```
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 zu zabbix_agentd.conf hinzu:

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions ist eine eindeutige Kennung. Sie kann ein beliebiger gültiger Schlüsselbezeichner sein, zum Beispiel *queries*.

Testen Sie diesen Parameter mit dem Zabbix Agent unter Verwendung des Schalters „-t“ (wenn er jedoch unter root ausgeführt wird, beachten Sie, dass der Agent beim Start als Daemon möglicherweise andere Berechtigungen hat):

```
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 anstelle des Laufzeitsteuerungsbefehls auch den Agent neu starten.

Testen Sie den Parameter mit dem Dienstprogramm zabbix_get.

Schritt 4

Fügen Sie dem überwachten Host einen neuen Datenpunkt mit dem Schlüssel 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 festgelegt sein muss. Andernfalls akzeptiert Zabbix diese nicht.

6 Windows-Leistungsindikatoren

Übersicht

Sie können Windows-Leistungsindikatoren mit dem Schlüssel *perf_counter[]* 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 nur auf Englisch verfügbaren Äquivalents `perf_counter_en` finden Sie unter Windows-spezifische Datenpunkt-Schlüssel.

Um eine vollständige Liste der zur Überwachung verfügbaren Leistungsindikatoren zu erhalten, können Sie Folgendes ausführen:

```
typeperf -qx
```

Sie können auch Low-Level-Discovery verwenden, um mehrere Objektinstanzen von Windows-Leistungsindikatoren zu erkennen und die Erstellung von perf_counter-Datenpunkten für Objekte mit mehreren Instanzen zu automatisieren.

Numerische Darstellung

Windows verwaltet numerische Darstellungen (Indizes) für Objekt- und Performance-Counter-Namen. Zabbix unterstützt diese numerischen Darstellungen als Parameter für die Datenpunktschlüssel `perf_counter`, `perf_counter_en` sowie in den Konfigurationsparametern `PerfCounter`, `PerfCounterEn`.

Es wird jedoch nicht empfohlen, sie zu verwenden, es sei denn, Sie können garantieren, dass Ihre numerischen Indizes auf bestimmten Hosts den korrekten Zeichenfolgen zugeordnet sind. Wenn Sie portable Datenpunkte erstellen müssen, die auf verschiedenen Hosts mit unterschiedlichen lokalisierten Windows-Versionen funktionieren, können Sie den Schlüssel `perf_counter_en` oder den Konfigurationsparameter `PerfCounterEn` verwenden, die die Verwendung englischer Namen unabhängig vom Gebietsschema des Systems ermöglichen.

Um die numerischen Entsprechungen herauszufinden, führen Sie **regedit** aus und suchen Sie dann den *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 Zahlen für jeden Zeichenfolgenteil des Performance-Counters, wie in '\System\% Processor Time':

```
System → 2
% Processor Time → 6
```

Anschließend können Sie diese Zahlen verwenden, um den Pfad numerisch darzustellen:

```
\2\6
```

Parameter für Performance Counter

Sie können einige PerfCounter-Parameter für die Überwachung von Windows-Performance-Countern bereitstellen.

Zum Beispiel können Sie diese zur Zabbix-Agent-Konfigurationsdatei hinzufügen:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
or
PerfCounter=UserPerfCounter2,"\4\24",30
```

Wenn solche Parameter eingerichtet sind, können Sie anschließend einfach *UserPerfCounter1* oder *UserPerfCounter2* als Schlüssel zum Erstellen der jeweiligen Datenpunkte verwenden.

Denken Sie daran, den Zabbix-Agent nach Änderungen an der Konfigurationsdatei neu zu starten.

7 Massenaktualisierung

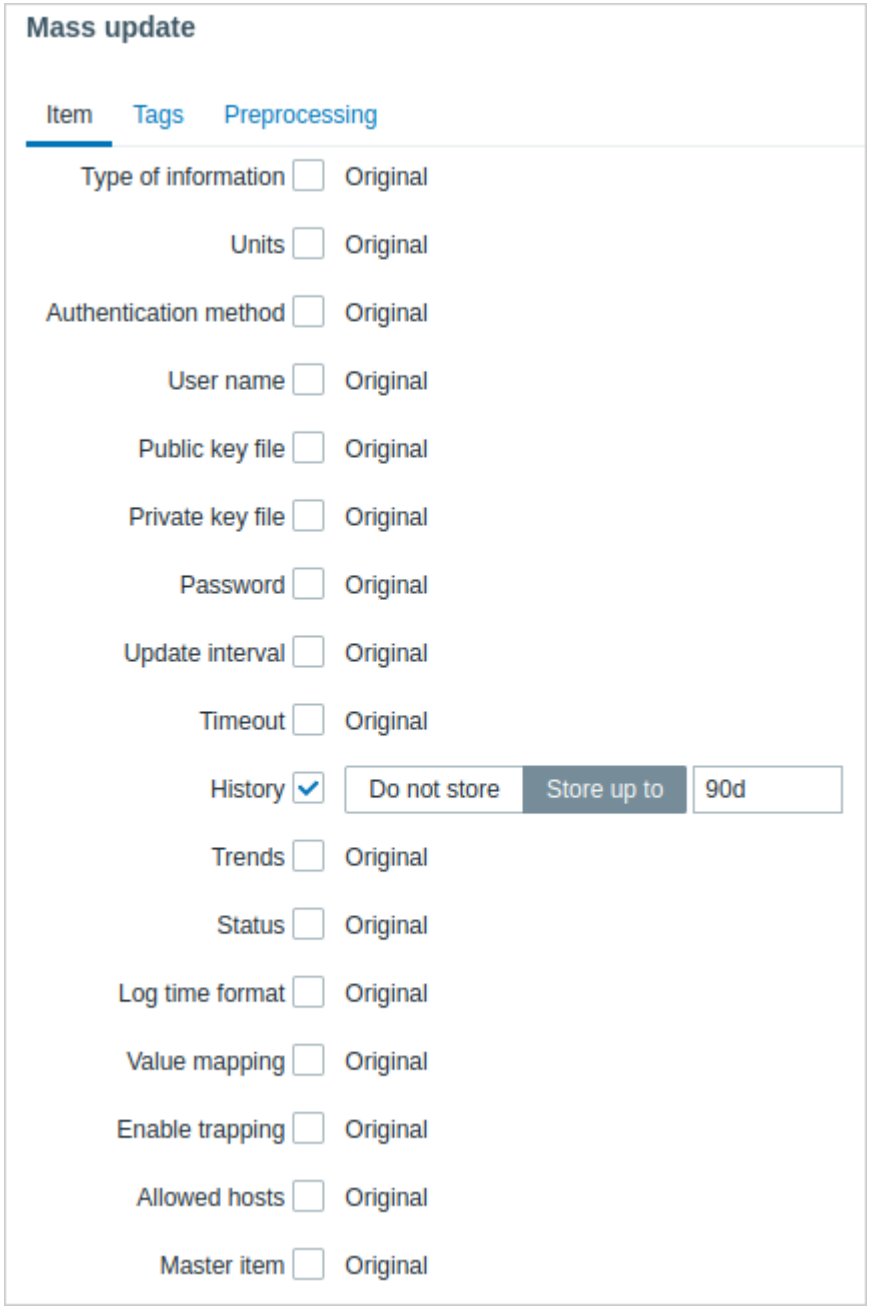
Übersicht

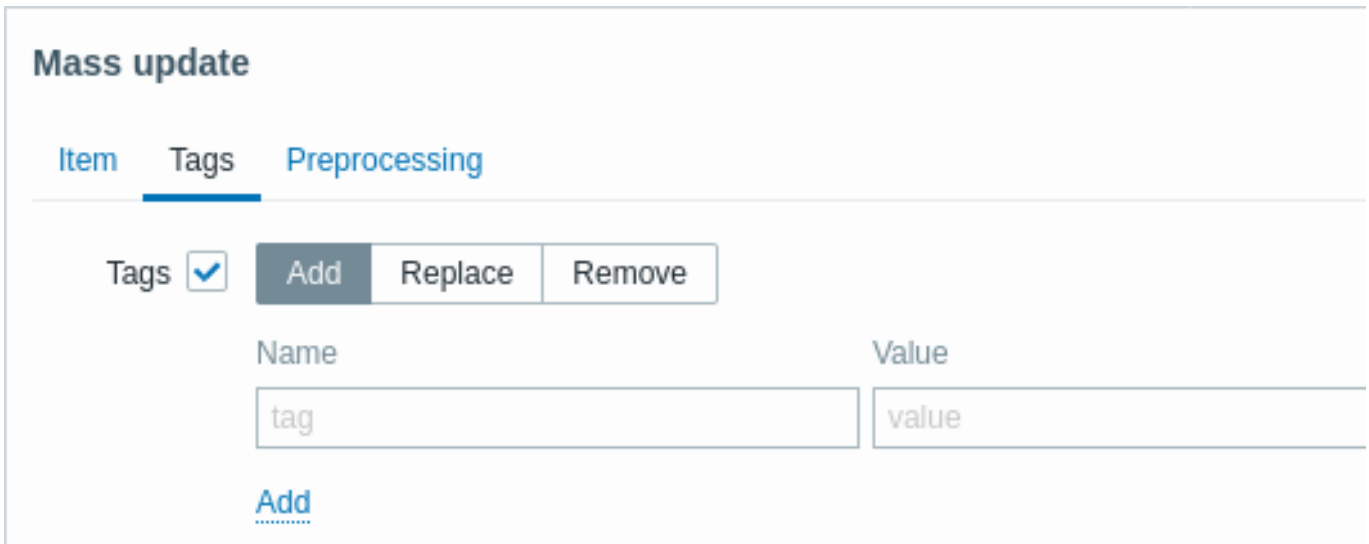
Manchmal möchten Sie ein Attribut für mehrere Datenpunkte gleichzeitig ändern. Anstatt jeden einzelnen Datenpunkt zur Bearbeitung zu öffnen, können Sie dafür die Funktion zur Massenaktualisierung verwenden.

Massenaktualisierung verwenden

Um mehrere Datenpunkte per Massenaktualisierung zu aktualisieren, gehen Sie wie folgt vor:

- Markieren Sie in der Liste die Kontrollkästchen der zu aktualisierenden Datenpunkte
- Klicken Sie unterhalb der Liste auf *Massenaktualisierung*
- Wechseln Sie zur Registerkarte mit den erforderlichen Attributen (*Datenpunkt*, *Tags* oder *Vorverarbeitung*)
- Markieren Sie die Kontrollkästchen der zu aktualisierenden Attribute
- Geben Sie neue Werte für die Attribute ein



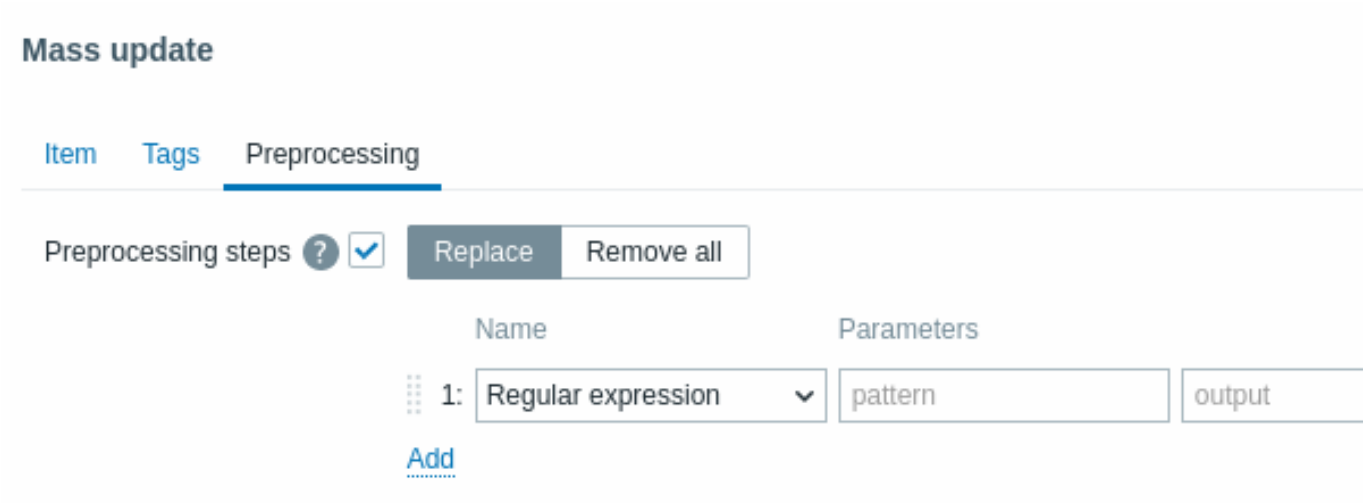


Die Option *Tags* ermöglicht Folgendes:

- *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* - angegebene 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.



Markieren Sie das Kontrollkästchen neben *Vorverarbeitungsschritte*, um die Massenaktualisierung für Vorverarbeitungsschritte zu aktivieren.

Die Massenaktualisierung *Vorverarbeitung* ermöglicht Folgendes:

- *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*.

8 Wertezuordnung

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.

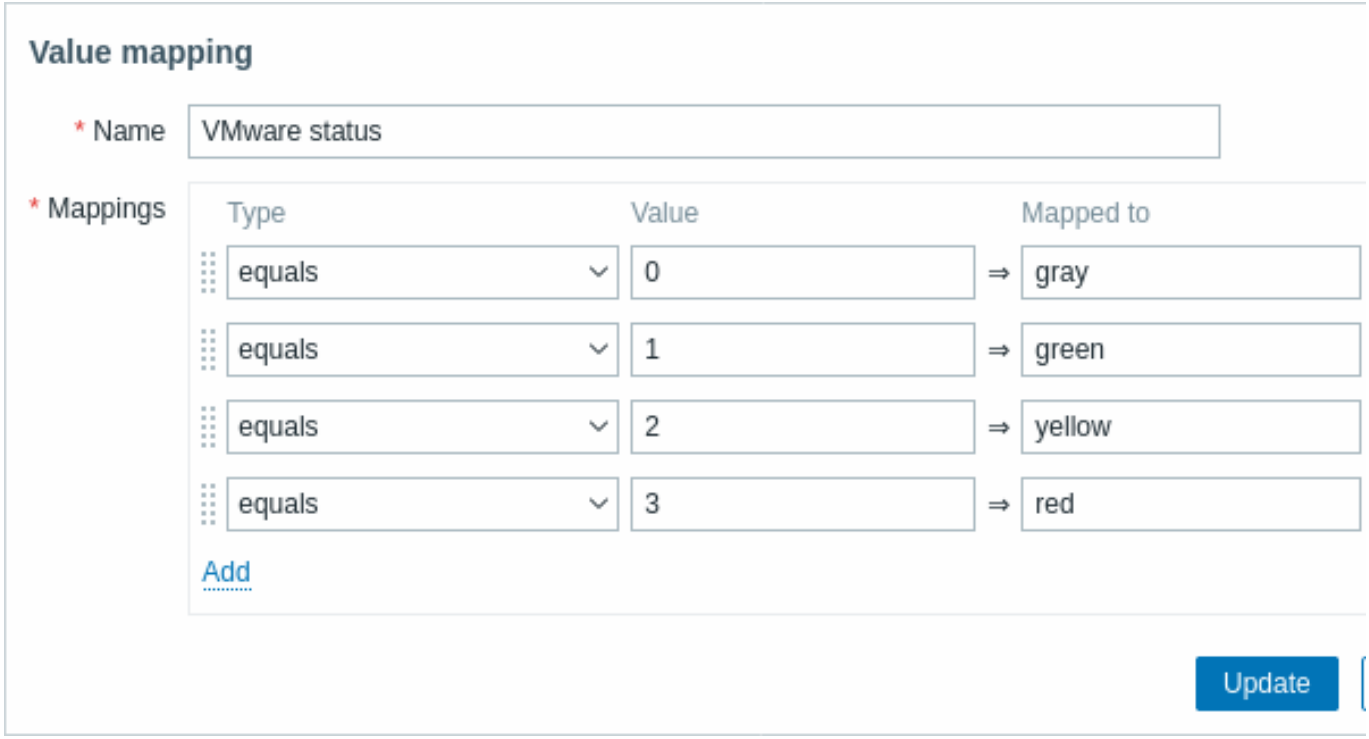
Wertezuordnungen werden zusammen mit der jeweiligen Vorlage oder dem jeweiligen Host exportiert/importiert. Sie können auch mithilfe der Formulare zur Massenaktualisierung für Hosts und Vorlagen massenhaft aktualisiert werden.

Konfiguration

Gehen Sie wie folgt vor, um eine Wertezuordnung zu konfigurieren:

1. Öffnen Sie das Konfigurationsformular des Hosts oder der Vorlage.

2. Klicken Sie auf der Registerkarte *Wertezuordnung* auf *Hinzufügen*, um eine neue Wertezuordnung hinzuzufügen, oder klicken Sie auf den Namen einer vorhandenen Zuordnung, um sie zu bearbeiten.



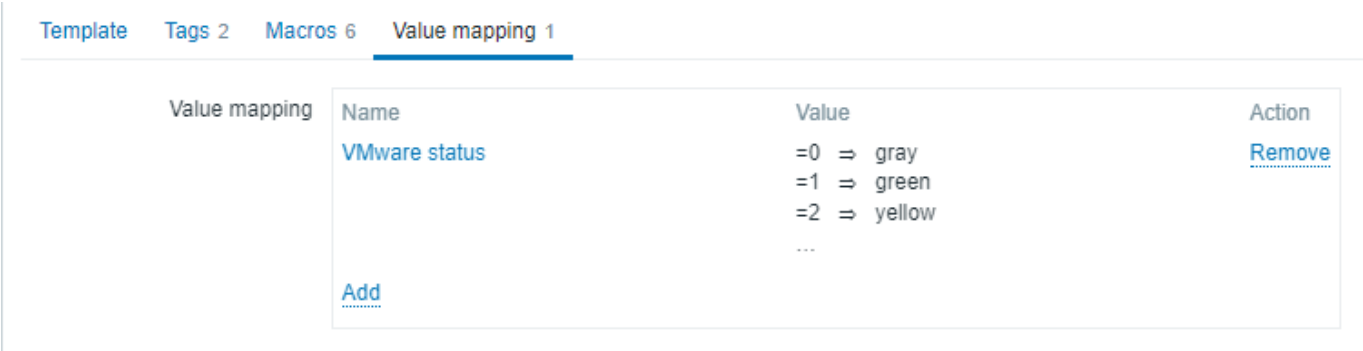
Parameter einer Wertezuordnung:

Parameter	Beschreibung
Name	Eindeutiger Name für die Menge der Wertezuordnungen.
Zuordnungen	Einzelne Regeln für die Zuordnung numerischer/String-Werte zu String-Darstellungen. Die Zuordnung wird in der Reihenfolge der Regeln angewendet, die per Ziehen neu angeordnet werden können.

Parameter	Beschreibung
Typ	Zuordnungstyp: **gleich** - gleiche Werte werden zugeordnet; **ist größer oder gleich** - gleiche oder größere Werte werden zugeordnet; **ist kleiner oder gleich** - gleiche oder kleinere Werte werden zugeordnet; **im Bereich** - Werte im Bereich 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); **Standard** - alle übrigen Werte werden zugeordnet, außer denen mit spezifischen Zuordnungen. Für Bereichszuordnungen werden nur numerische Werttypen (*ist größer oder gleich*, *ist kleiner oder gleich*, *im Bereich*) unterstützt.
Wert	Eingehender Wert (kann je nach Zuordnungstyp einen Bereich oder einen regulären Ausdruck enthalten).
Zugeordnet zu	String-Darstellung (bis zu 64 Zeichen) für den eingehenden Wert.

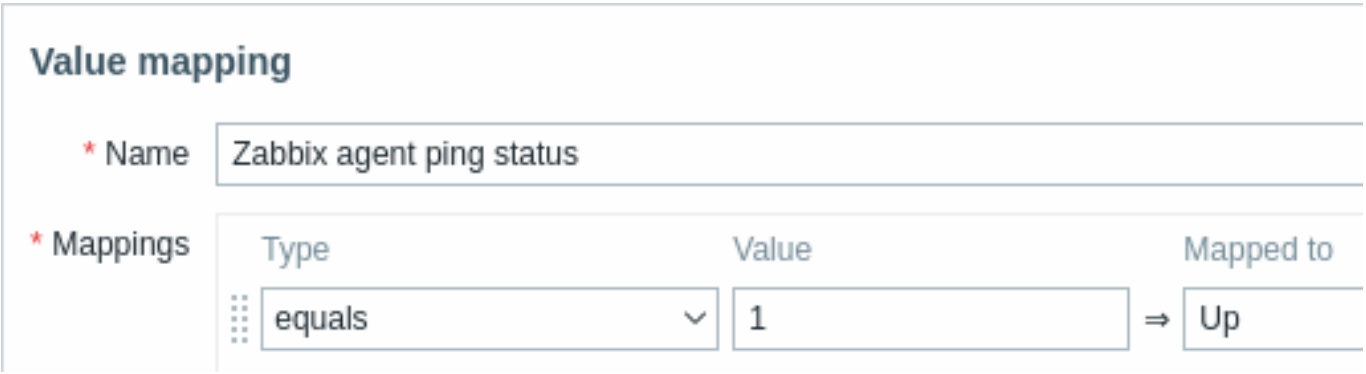
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Bei der Anzeige der Wertezuordnung in der Liste sind nur die ersten drei Zuordnungen sichtbar; drei Punkte zeigen an, dass weitere Zuordnungen vorhanden sind.



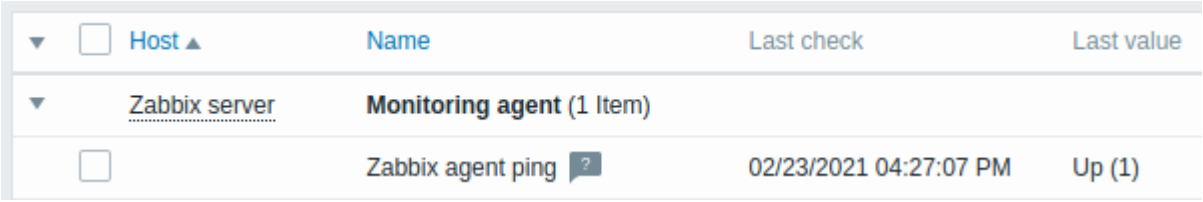
Beispiel für Wertzuordnung

Einer der vordefinierten Agent-Datenpunkte, *Zabbix agent ping*, verwendet eine Wertzuordnung auf Vorlagenebene „Zabbix agent ping status“, um seine Werte anzuzeigen.



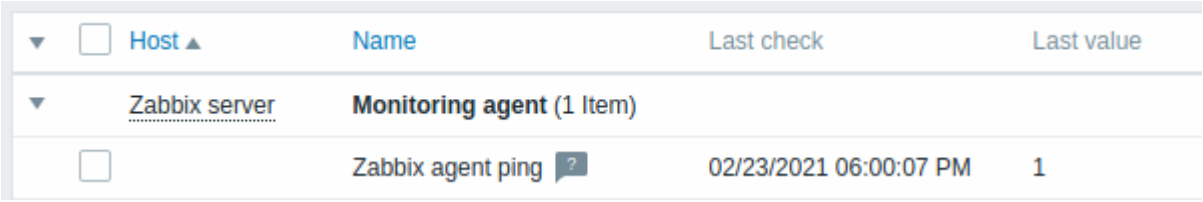
Im Konfigurationsformular des Datenpunkts finden Sie im Feld *Value mapping* einen Verweis auf diese Wertzuordnung:

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 angewendet, sondern nur auf den Rohwert (in Klammern angezeigt).

Ohne eine vordefinierte Wertzuordnung würden Sie nur „1“ sehen, was möglicherweise schwer zu verstehen ist.



9 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

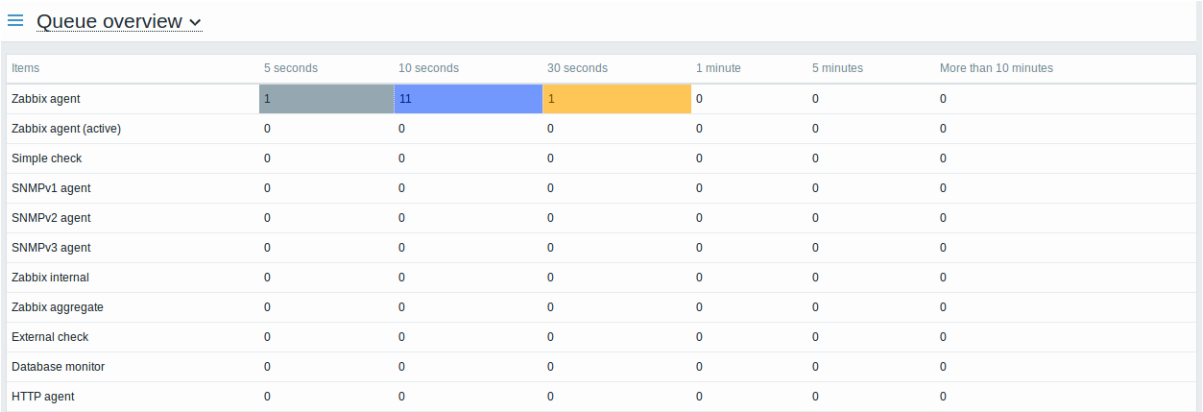
Die von der Warteschlange angezeigten Statistiken sind ein guter Indikator für die Leistung des Zabbix-Servers.

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 Queue

Um die Queue zu lesen, gehen Sie zu *Administration* > *Queue* > *Queue overview*.



Die Abbildung hier ist im Allgemeinen „ok“, daher können wir annehmen, dass der Server ordnungsgemäß arbeitet.

Die Queue zeigt einige Datenpunkte, die bis zu 30 Sekunden warten. Es wäre hilfreich zu wissen, um welche Datenpunkte es sich handelt.

Um genau das zu tun, wählen Sie im Dropdown-Menü des Titels *Queue details* aus. Nun sehen Sie eine Liste dieser verzögerten Datenpunkte.

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 bereitgestellten Details lässt sich möglicherweise herausfinden, warum diese Datenpunkte verzögert sein könnten.

Bei einem oder zwei verzögerten Datenpunkten besteht vielleicht kein Grund zur Beunruhigung. Sie werden möglicherweise in einer Sekunde aktualisiert. Wenn Sie jedoch sehen, dass eine größere Anzahl von Datenpunkten zu lange verzögert wird, könnte ein ernsteres Problem vorliegen.

Siehe auch: Ausrichten von Zeitzonen bei der Verwendung von Planungsintervallen.

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 den festgelegten Zeitraum verzögert sind. Weitere Informationen finden Sie unter Interne Datenpunkte.

10 Wertespeicher

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 Wertespeichers zu aktivieren, wird der optionale Parameter **ValueCacheSize** in der configuration-Datei des Zabbix Server unterstützt.

Zwei interne Datenpunkte werden zur Überwachung des Wertespeichers unterstützt: **zabbix[vcache,buffer,<mode>]** und **zabbix[vcache,cache,<parameter>]**. Weitere Details finden Sie unter internal items.

11 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

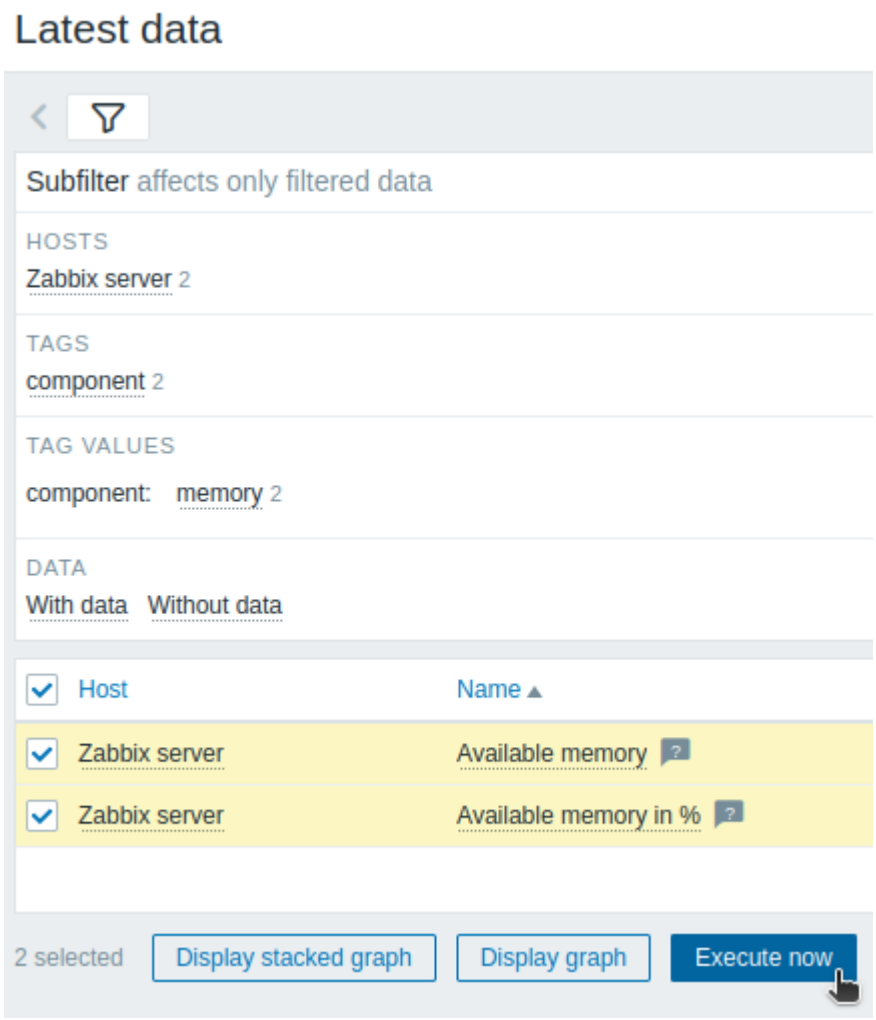
Sie können passive Prüfungen ausführen von:

- *Monitoring* > *Aktuelle Daten*
- *Datenerfassung* > *Hosts* > *Datenpunkte* oder *Discovery*
- Das Datenpunkt-Menü
- Beim Bearbeiten von Datenpunkten oder Low-Level-Discovery-Regeln (in deren Konfigurationsformularen)

Attention:
Die Prüfung muss im Konfigurations-Cache vorhanden sein, damit sie ausgeführt werden kann; siehe auch CacheUpdate-Frequency. Der Cache wird vor der Ausführung der Prüfung **nicht aktualisiert**, daher werden aktuelle Änderungen an der Konfiguration von Datenpunkten oder Low-Level-Discovery-Regeln nicht berücksichtigt. Um Datenpunkte oder LLD-Regeln zu testen, die gerade erstellt werden oder neu erstellt wurden, können Sie stattdessen die Option Test verwenden.

Zum Beispiel, um eine passive Prüfung in *Monitoring* > *Aktuelle Daten* auszuführen:

1. Wählen Sie Datenpunkte aus; Sie können die Prüfung für mehrere Datenpunkte gleichzeitig ausführen.
2. Klicken Sie auf *Jetzt ausführen*.



Note:
Unter *Aktuelle Daten* können Benutzer Datenpunkte auf Hosts *jetzt ausführen*, für die sie über Berechtigungen mit *Lese-/Schreibzugriff* verfügen oder über *Lesezugriff* und für deren Rolle die Aktion *„Jetzt ausführen“ auf schreibgeschützten Hosts aufrufen* aktiviert ist.

12 Einschränken von Agent-Prüfungen

Übersicht

Sie können steuern, welche Datenpunktschlüssel der Zabbix Agent oder Agent 2 bei der Ausführung von Datenpunkt-Prüfungen, Remote-Befehlen oder Skripten verwenden darf oder verweigert.

Verwenden Sie dazu diese Parameter der Agent-Konfiguration, um Zulassungs-/Verweigerungsregeln zu definieren:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`

Das `<pattern>` muss genau einen Datenpunktschlüssel enthalten und unterstützt Platzhalter (`*`). Der Platzhalter entspricht an seiner Position einer beliebigen Anzahl beliebiger Zeichen und kann verwendet werden, um Datenpunktschlüssel oder Parameter abzugleichen (z. B. `vfs.file.*[*]`).

Attention:
Zur Verbesserung der Sicherheit wird empfohlen, exakte Datenpunktschlüssel anstelle von Platzhaltern zu verwenden. Weitere Informationen finden Sie unter Absichern von Zulassungs-/Verweigerungsregeln.

Im Gegensatz zu anderen Parametern der Agent-Konfiguration können Sie eine unbegrenzte Anzahl von `AllowKey`- oder `DenyKey`-Parametern angeben.

Wichtige Hinweise

- Alle `system.run`-Datenpunkte sind standardmäßig deaktiviert (auch wenn `DenyKey` leer ist), als wäre `DenyKey=system.run[*]`

als letzte Regel gesetzt. Daher können Sie bestimmte system.run-Datenpunkte erlauben, ohne andere system.run-Datenpunkte ausdrücklich zu verbieten.

- Ein in AllowKey angegebener Datenpunkt muss auch in DenyKey angegeben werden (außer bei system.run-Datenpunkten); andernfalls **startet** der Zabbix Agent **nicht**.
- Verwenden Sie nach Möglichkeit AllowKey, um nur die erforderlichen Datenpunkte zu erlauben und alles andere zu verbieten. Einige Schlüssel können missbraucht werden, um über Path Traversal unbeabsichtigte Dateien zu lesen (z. B. vfs.file.contents["../../../../etc/passwd"]), und neue Versionen des Zabbix Agent können Schlüssel einführen, die nicht durch Ihre DenyKey-Regeln abgedeckt sind.
- Die Konfiguration von AllowKey und DenyKey wirkt sich nicht auf die Agent-Parameter HostnameItem, HostMetadataItem oder HostInterfaceItem aus.
- Verbotene Datenpunkte werden ohne Hinweise oder Fehlermeldungen nicht unterstützt; zum Beispiel:
  - Der Befehlszeilenparameter --print (-p) des Zabbix Agent zeigt keine verbotenen Datenpunktschlüssel an.
  - Der Befehlszeilenparameter --test (-t) des Zabbix Agent gibt für verbotene Datenpunktschlüssel "Unsupported item key." zurück.
  - Wenn die Protokollierung aktiviert ist (LogRemoteCommands=1), protokolliert die Logdatei des Zabbix Agent keine verbotenen Remote-Befehle.

Reihenfolge von Allow-/Deny-Regeln

Sie können eine unbegrenzte Anzahl von AllowKey- oder DenyKey-Regeln angeben, wobei ihre Reihenfolge wichtig ist.

- Regeln werden nacheinander von oben nach unten ausgewertet.
- Wenn ein Datenpunktschlüssel mit einer Regel übereinstimmt, wird er entweder zugelassen oder verweigert, und die Regelauswertung wird beendet.

Zum Beispiel werden bei der Auswertung von vfs.file.contents[/etc/passwd] die Regeln wie folgt verarbeitet:

```
AllowKey=vfs.file.contents[/tmp/app.log]    # Das Muster des Datenpunktschlüssels stimmt nicht überein, Ag
AllowKey=vfs.file.contents[/etc/passwd]     # Das Muster des Datenpunktschlüssels stimmt überein; Agent er
DenyKey=vfs.file.*[*]                       # Agent ignoriert die Regel, da die Auswertung bereits beendet
```

Die folgende Reihenfolge der Regeln verweigert die Datenpunktprüfung:

```
DenyKey=vfs.file.*[*]                       # Das Muster des Datenpunktschlüssels stimmt überein; Agent ve
AllowKey=vfs.file.contents[/etc/passwd]     # Agent ignoriert die Regel, da die Auswertung bereits beendet
AllowKey=vfs.file.contents[/tmp/app.log]    # Agent ignoriert die Regel, da die Auswertung bereits beendet
```

Beispiele

Die folgenden Beispiele zeigen gängige Konfigurationsmuster für AllowKey und DenyKey.

Zulassen bestimmter Prüfungen und Befehle

Erlauben Sie nur zwei vfs.file-Datenpunkt-Prüfungen und zwei system.run-Befehle:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

Note:
Die Einstellung DenyKey=system.run[*] ist nicht erforderlich, da alle anderen system.run-Befehle standardmäßig verweigert werden.

Ausführen von Skripten erlauben

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 (diese Methode verwendet immer den Schlüssel system.run[myscript.sh])
- Remote-Befehle aus Aktionsoperationen (diese Methode verwendet immer den Schlüssel system.run[myscript.sh,nowait])
- 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,*]
```

Um den Parameter wait/nowait zu steuern, müssen Sie eine andere Regel festlegen. Sie können zum Beispiel nur system.run[myscript.sh,wait]-Datenpunkte erlauben und dadurch andere Methoden ausschließen:

```
AllowKey=system.run[myscript.sh,wait]
```

Absichern von Allow-/Deny-Regeln

Dieses Beispiel zeigt, wie sich zu großzügige AllowKey- oder DenyKey-Regeln absichern lassen.

Betrachten Sie die folgenden Regeln:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

Note:
Unter Windows müssen Leerzeichen im Pfad mit einem Caret-Zeichen (^) maskiert werden.

Diese Regeln enthalten ein Platzhalterzeichen (*), das missbraucht werden kann:

- Das Skript test.bat kann mit beliebigen Argumenten ausgeführt werden, auch mit unbeabsichtigten.
- Das Muster vfs.file.* entspricht nur Datenpunkt-Schlüsseln ohne Parameter; allerdings erfordern alle vfs.file-Datenpunkte Parameter.
- Das Muster system.cpu.load[*] entspricht nur Datenpunkt-Schlüsseln mit Parametern; allerdings benötigen system.cpu.load-Datenpunkte keine Parameter.

Um diese Regeln abzusichern, erlauben Sie die Ausführung von test.bat explizit nur mit bestimmten Argumenten und verweigern Sie die korrekten Datenpunkt-Schlüsselmuster; zum Beispiel:

```
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[*]
```

Sie können die Regeln testen, indem Sie die folgenden Befehle ausführen; diese geben ZBX_NOTSUPPORTED zurück.

```
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 folgende Tabelle zeigt, wie Datenpunktschlüssel-Muster abgeglichen werden:

- Ein Schlüssel entspricht dem Muster nur, wenn er **alle** Bedingungen in der Spalte *Entspricht* erfüllt.
- Parameter müssen vollständig in eckige Klammern eingeschlossen sein (z. B. sind vfs.file.contents[* und vfs.file.contents*utf8] ungültige Muster).

Muster	Entspricht	Beispiele
*	Beliebiger Schlüssel mit oder ohne Parameter	
vfs.file.*	Schlüssel beginnt mit vfs.file. Keine Parameter	Entspricht: vfs.file.size vfs.file.contents Entspricht nicht: vfs.file.contents[] vfs.file.size[/var/log/app.log]

Muster	Entspricht	Beispiele

`vfs.*.contents`	Schlüssel beginnt mit `vfs.`
	Schlüssel endet mit `.contents`
	Keine Parameter
		Entspricht:
		`vfs..contents`
		`vfs.mount.point.file.contents`
		
		Entspricht nicht:
		`vfs.contents`
		`vfs.file.contents[]`

`vfs.file.*[*]`	Schlüssel beginnt mit `vfs.file.`
	Beliebige oder leere Parameter
		Entspricht:
		`vfs.file.get.custom[]`
		`vfs.file.size[/var/log/app.log, utf8]`
		
		Entspricht nicht:
		`vfs.file.get.custom`

`vfs.file.contents`	Schlüssel ist `vfs.file.contents`
	Keine Parameter
		Entspricht:
		`vfs.file.contents`
		
		Entspricht nicht:
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[]`	Schlüssel ist `vfs.file.contents[]`
	Leere Parameter
		Entspricht:
		`vfs.file.contents[]`
		
		Entspricht nicht:
		`vfs.file.contents`

`vfs.file.contents[*]`	Schlüssel ist `vfs.file.contents`
	Beliebige oder leere Parameter
		Entspricht:
		`vfs.file.contents[/path/to/file]`
		
		Entspricht nicht:
		`vfs.file.contents`

`vfs.file.contents[/etc/passwd,*]`	Schlüssel ist `vfs.file.contents`
	Erster Parameter ist `/etc/passwd`
	Beliebiger oder leerer zweiter Parameter
		Entspricht:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Entspricht nicht:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[*passwd]`	Schlüssel ist `vfs.file.contents`
	Erster Parameter enthält `passwd`
	Kein zweiter Parameter
		Entspricht:
		`vfs.file.contents[/etc/passwd]`
		
		Entspricht nicht:
		`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
		Entspricht:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Entspricht nicht:
		`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`
		Entspricht:
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Entspricht nicht:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

3 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 erfasst.
2. Definieren Sie den Auslöser für diesen Datenpunkt. Ohne einen Auslöser erfasst Zabbix lediglich Daten, reagiert jedoch nie darauf.
3. Beobachten Sie erkannte Probleme:
    - Unter *Monitoring* > *Probleme*
    - Im *Probleme*-Widget
4. Erhalten Sie Benachrichtigungen und führen Sie Remote-Befehle aus (falls definiert).

Beachten Sie, dass offizielle Zabbix-Vorlagen vordefinierte Auslöser enthalten. Siehe die Anleitung zum Anwenden einer Vorlage, um Linux zu 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.

1 Auslöser-Beispiele

Diese Seite ist 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
- Instabile Schnittstelle
- Hoher eingehender Datenverkehr
- Wenig Speicherplatz
- Wenig Speicherplatz (dynamischer Schwellenwert)
- Hohe CPU-Last
- Vergleich der CPU-Last
- Langfristiger Vergleich der CPU-Last
- 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[192.0.2.0,{$WEBSITE_NAME},{$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 normalerweise zurückgegeben wird:

```
example.com             MX       0 mail.example.com
```

Funktion: last
 Datenpunkt: net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], mit 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-Agenten als passiver Check konfiguriert werden muss; bei Zabbix-Agent 2 kann er als aktiver Check 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`

2 Einen Auslöser konfigurieren

Diese Seite beschreibt, wie ein Auslöser im Zabbix Frontend konfiguriert wird.

Beachten Sie, dass offizielle Zabbix-Vorlagen vordefinierte Auslöser enthalten. Siehe die Anleitung zum Anwenden einer Vorlage auf Linux überwachen. Es ist auch möglich, konfigurierte Auslöser per Massenaktualisierung zu aktualisieren.

Weitere Ressourcen:

- Problemerkennung mit Auslösern (allgemeine Einführung)

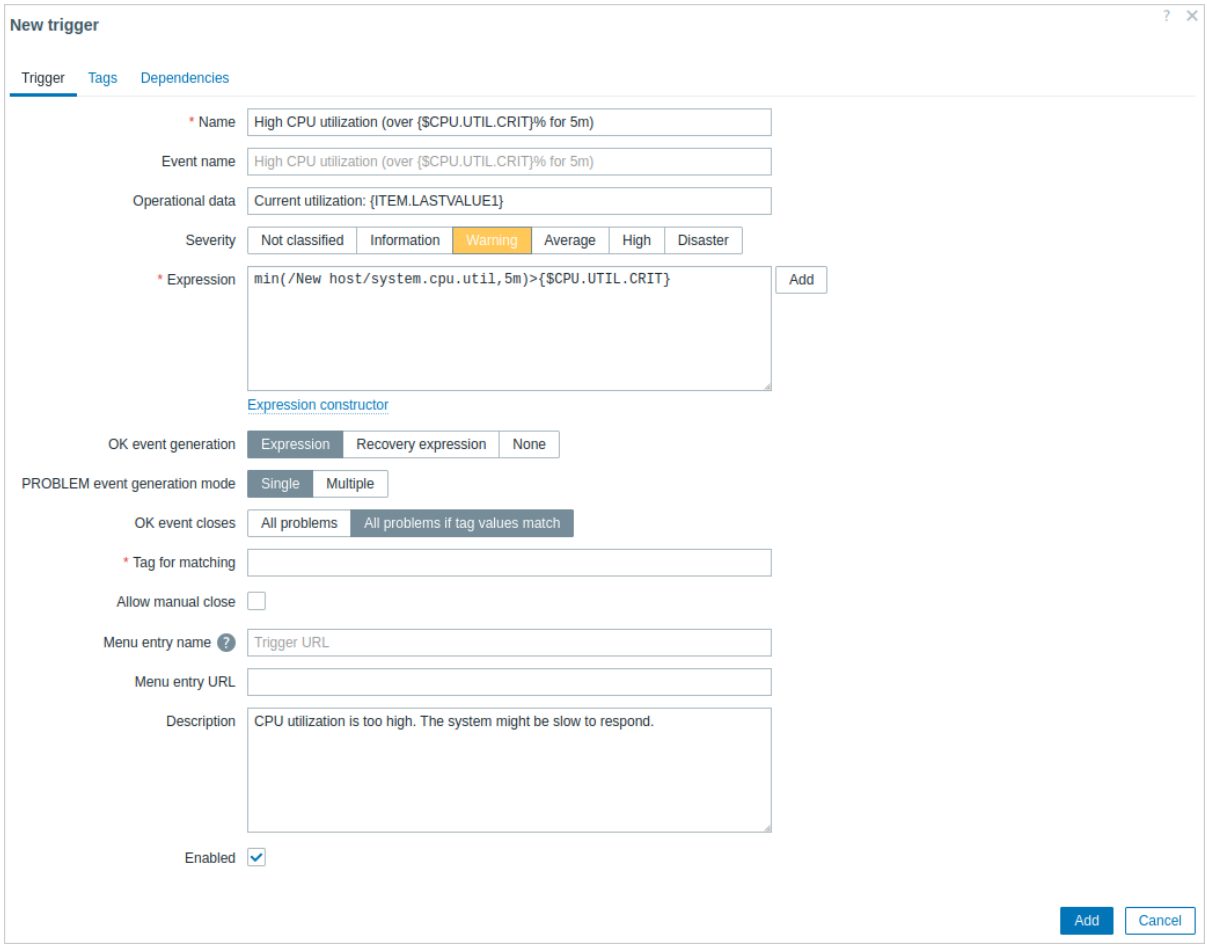
- Auslöser-Beispiele (nach Anwendungsfall)
- Auslöser-Ausdruck (Syntaxdetails)
- Best Practices

Konfiguration

Um einen Auslöser zu konfigurieren, gehen Sie wie folgt vor:

- Gehen Sie zu: *Datenerfassung* > *Hosts*
- Klicken Sie in der Zeile des Hosts auf *Auslöser*
- Klicken Sie rechts auf *Auslöser erstellen* (oder auf den Namen des Auslösers, um einen vorhandenen Auslöser zu bearbeiten)
- Geben Sie die Parameter des Auslösers in das Formular ein

Die Registerkarte **Auslöser** enthält alle wesentlichen Attribute des Auslösers.



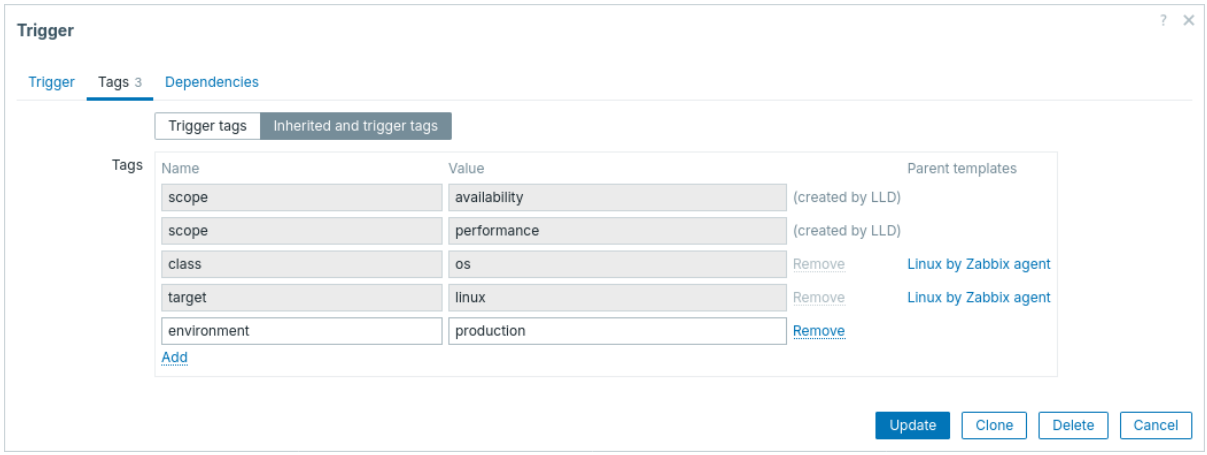
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

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 Benutzermakros {$MACRO}. Die Makros **$1, $2...$9** können verwendet werden, um auf die erste, zweite ... neunte Konstante des Ausdrucks zu verweisen. *Hinweis*: Die Makros $1-$9 werden korrekt aufgelöst, wenn sie sich auf Konstanten in relativ einfachen, geradlinigen Ausdrücken beziehen. Zum Beispiel wird der Name „Prozessorlast über $1 auf {HOST.NAME}“ automatisch in „Prozessorlast über 5 auf New host“ geändert, wenn der Ausdruck last(/New host/system.cpu.load[percpu,avg1])>5 ist.

Parameter	Beschreibung
Ereignisname	Wenn definiert, wird dieser Name zur Erstellung des Problemereignisnamens verwendet. Standardmäßig ist der Ereignisname identisch mit dem Namen des Auslösers. Der Ereignisname kann verwendet werden, um aussagekräftige Benachrichtigungen mit Problemdaten zu erstellen (siehe Beispiel). Es wird derselbe Satz von Makros wie im Auslösernamen unterstützt, zusätzlich {TIME}, {TIMESTAMP} und Ausdrucksmakros {?EXPRESSION}.
Betriebsdaten	Mit Betriebsdaten können beliebige Zeichenfolgen zusammen mit Makros definiert werden. Die Makros werden in *Überwachung* > *Probleme* dynamisch in Echtzeitdaten aufgelöst. Während Makros im Auslösernamen (siehe oben) zum Zeitpunkt des Auftretens eines Problems in ihre Werte aufgelöst werden und die Grundlage eines statischen Problemnamens bilden, behalten die Makros in den Betriebsdaten die Fähigkeit, dynamisch die neuesten Informationen anzuzeigen. Wenn auf Auslöser-Ebene keine Betriebsdaten konfiguriert sind, werden die letzten Werte aller Datenpunkte aus dem Ausdruck angezeigt. Es wird derselbe Satz von Makros wie im Auslösernamen unterstützt.
Schweregrad	Legen Sie den erforderlichen Schweregrad des Auslösers durch Klicken auf die Schaltflächen fest.
Ausdruck	Logischer Ausdruck, der zur Definition der Bedingungen eines Problems verwendet wird. Zeitsuffixe und Speichergrößensuffixe werden unterstützt. Ein Problem wird erstellt, nachdem alle im Ausdruck enthaltenen Bedingungen erfüllt sind, d. h. der Ausdruck den Wert TRUE ergibt. Das Problem wird gelöst, sobald der Ausdruck den Wert FALSE ergibt, sofern in *Wiederherstellungsausdruck* keine zusätzlichen Wiederherstellungsbedingungen angegeben sind.
Erzeugung von OK-Ereignissen	Optionen für die Erzeugung von OK-Ereignissen: **Ausdruck** - OK-Ereignisse werden auf Grundlage desselben Ausdrucks wie Problemereignisse erzeugt; **Wiederherstellungsausdruck** - OK-Ereignisse werden erzeugt, wenn der Problemausdruck den Wert FALSE ergibt und der Wiederherstellungsausdruck den Wert TRUE ergibt; **Keine** - in diesem Fall kehrt der Auslöser niemals selbstständig in einen OK-Zustand zurück.
Wiederherstellungsausdruck	Logischer Ausdruck (optional), der zusätzliche Bedingungen definiert, die erfüllt sein müssen, bevor das Problem gelöst wird, nachdem der ursprüngliche Problemausdruck bereits als FALSE ausgewertet wurde. Der Wiederherstellungsausdruck ist nützlich für die Hysterese von Auslösern. Es ist **nicht** möglich, ein Problem allein durch den Wiederherstellungsausdruck zu lösen, wenn der Problemausdruck weiterhin TRUE ist. Dieses Feld ist nur verfügbar, wenn für *Erzeugung von OK-Ereignissen* „Wiederherstellungsausdruck“ ausgewählt ist.
Modus für die Erzeugung von PROBLEM-Ereignissen	Modus zur Erzeugung von Problemereignissen: **Einzeln** - ein einzelnes Ereignis wird erzeugt, wenn ein Auslöser erstmals in den Zustand „Problem“ wechselt; **Mehrfach** - bei *jeder* Auswertung des Auslösers als „Problem“ wird ein Ereignis erzeugt.
OK-Ereignis schließt	Wählen Sie aus, ob das OK-Ereignis Folgendes schließt: **Alle Probleme** - alle Probleme dieses Auslösers; **Alle Probleme, wenn Tag-Werte übereinstimmen** - nur die Probleme dieses Auslösers mit übereinstimmenden Ereignis-Tag-Werten.
Tag für Abgleich	Geben Sie den Namen des Ereignis-Tags ein, der für die Ereigniskorrelation verwendet werden soll. Dieses Feld wird angezeigt, wenn für die Eigenschaft *OK-Ereignis schließt* „Alle Probleme, wenn Tag-Werte übereinstimmen“ ausgewählt ist, und ist in diesem Fall erforderlich.
Manuelles Schließen erlauben	Aktivieren Sie dieses Kontrollkästchen, um das manuelle Schließen von Problemereignissen zu erlauben, die von diesem Auslöser erzeugt wurden. Manuelles Schließen ist beim Bestätigen von Problemereignissen möglich.
Name des Menüeintrags	Wenn nicht leer, wird der hier eingegebene Name (bis zu 64 Zeichen) an mehreren Stellen im Frontend als Bezeichnung für die im Parameter *URL des Menüeintrags* angegebene Auslöser-URL verwendet. Wenn leer, wird der Standardname *Auslöser-URL* verwendet. Es wird derselbe Satz von Makros wie im Auslösernamen unterstützt, zusätzlich {EVENT.ID}, {HOST.ID} und {TRIGGER.ID}.
URL des Menüeintrags	Wenn nicht leer, ist die hier eingegebene URL (bis zu 2048 Zeichen) als Link im Ereignismenü an mehreren Stellen im Frontend verfügbar, zum Beispiel beim Klicken auf den Problemnamen in *Überwachung* > *Probleme* oder im Dashboard-Widget *Probleme*. Es wird derselbe Satz von Makros wie im Auslösernamen unterstützt, zusätzlich {EVENT.ID}, {HOST.ID} und {TRIGGER.ID}. Hinweis: Benutzermakros mit geheimen Werten werden in der URL nicht aufgelöst.

Parameter	Beschreibung
Beschreibung	Textfeld, das verwendet wird, um weitere Informationen zu diesem Auslöser bereitzustellen. Kann Anweisungen zur Behebung eines bestimmten Problems, Kontaktdaten des zuständigen Personals usw. enthalten. Es wird derselbe Satz von Makros wie im Auslösernamen unterstützt.
Aktiviert	Wenn dieses Kontrollkästchen deaktiviert wird, 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, Auslöser-Tags zu definieren. Alle Probleme dieses Auslösers werden mit den hier eingegebenen Werten getaggt.



Zusätzlich ermöglicht die Option *Vererbte und Auslöser-Tags* die Anzeige von Tags, die auf Vorlagenebene definiert sind, wenn der Auslöser aus dieser Vorlage stammt. Wenn es mehrere Vorlagen mit demselben Tag gibt, werden diese Tags einmal angezeigt und die Vorlagennamen durch Kommas getrennt. Ein Auslöser „erbt“ keine Tags auf Host-Ebene und zeigt diese auch nicht an.

Parameter	Beschreibung
Name/Wert	Legen Sie benutzerdefinierte Tags fest, um Auslöserereignisse zu kennzeichnen. Tags sind ein Paar aus Tag-Name 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. Benutzermakros, Benutzermakros mit Kontext, Makros der Low-Level-Discovery und Makro-Funktionen mit `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` werden in Ereignis-Tags unterstützt. Makros der Low-Level-Discovery können innerhalb des Makrokontexts 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 aus diesen Auslösern während der Wartung zu unterdrücken. Wenn die Gesamtlänge des erweiterten Werts 255 überschreitet, wird sie auf 255 Zeichen gekürzt. Siehe alle Makros, die für Ereignis-Tags unterstützt werden. Ereignis-Tags können für die Ereigniskorrelation, in Aktionsbedingungen verwendet werden und sind außerdem in *Überwachung* > *Probleme* oder im Widget *Probleme* sichtbar.

Die Registerkarte **Abhängigkeiten** enthält alle Abhängigkeiten des Auslösers.

Klicken Sie auf *Hinzufügen*, um eine neue Abhängigkeit hinzuzufügen.

Note:
Sie können einen Auslöser auch konfigurieren, indem Sie einen vorhandenen öffnen, auf die Schaltfläche *Klonen* klicken und ihn dann unter einem anderen Namen speichern.

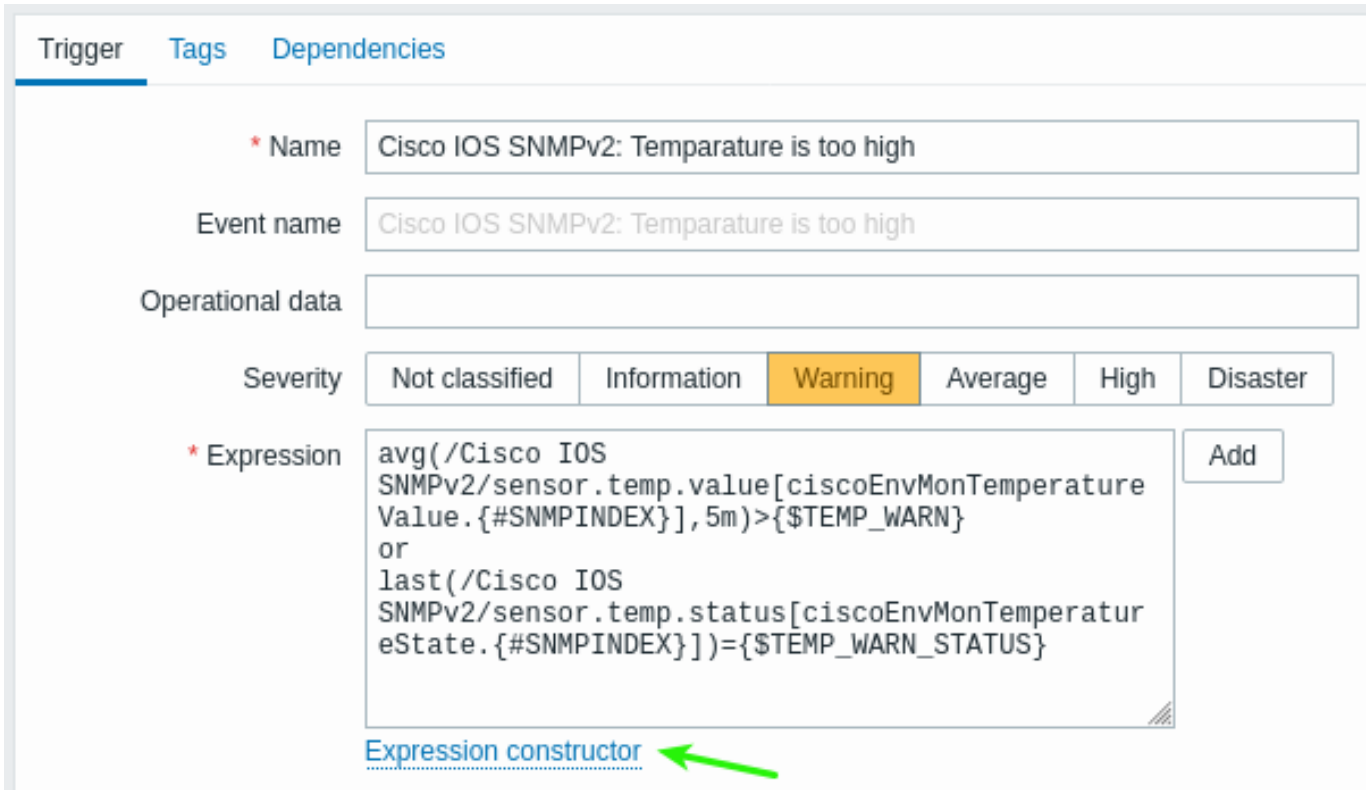
Testen von Ausdrücken für Auslöser

Es ist möglich zu testen, wie das Ergebnis des Ausdrucks in Abhängigkeit vom empfangenen Wert ausfallen würde.

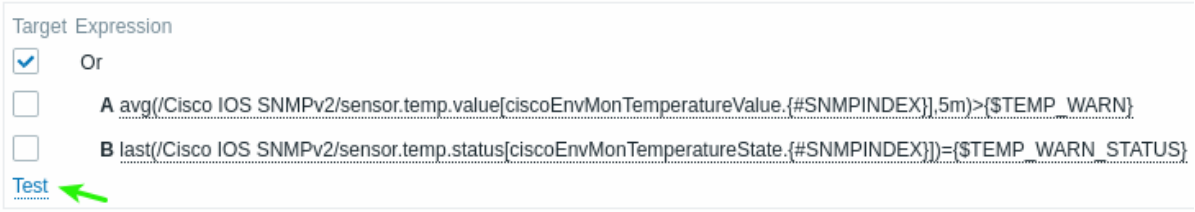
Der folgende Ausdruck aus einer offiziellen Vorlage wird als Beispiel verwendet:

```
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}
```

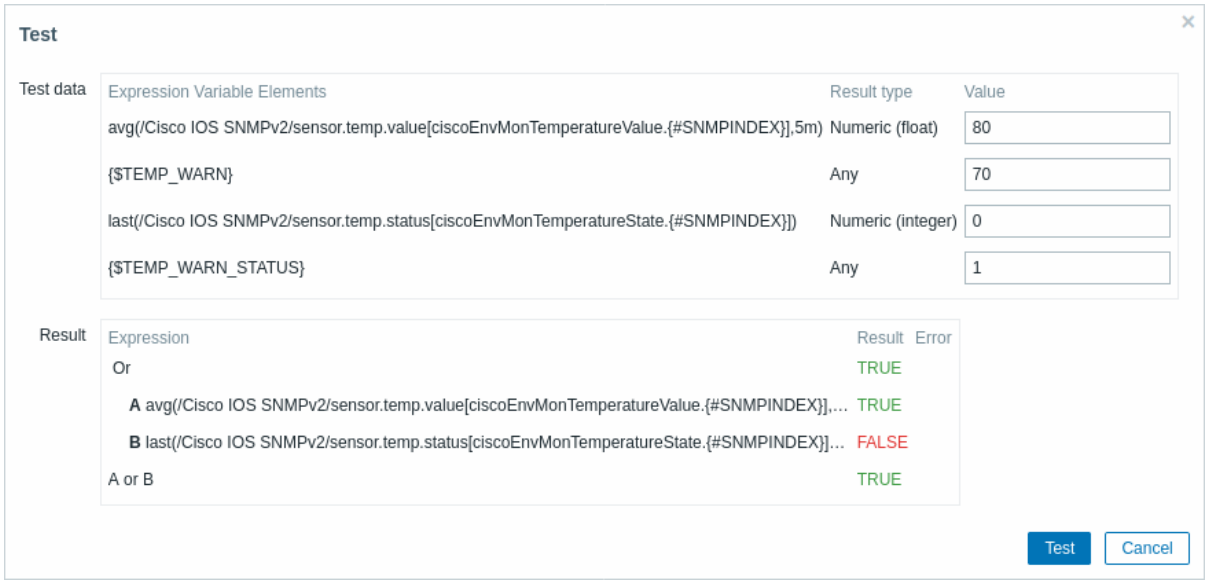
Um den Ausdruck zu testen, klicken Sie unter dem Ausdrucksfeld auf *Expression constructor*.



Im Expression constructor werden alle einzelnen Ausdrücke aufgelistet. Um das Testfenster zu öffnen, klicken Sie unter der Ausdrucksliste auf *Test*.



Im Testfenster können Sie Beispielwerte eingeben (’80’, ’70’, ’0’, ’1’ in diesem Beispiel) und dann durch Klicken auf die Schaltfläche *Test* das Ergebnis des Ausdrucks anzeigen.



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}, in diesem Beispiel ”70”. Wie erwartet erscheint ein Ergebnis von ”TRUE”.

”FALSE” bedeutet, dass der angegebene Ausdruck nicht korrekt ist. In diesem speziellen Fall B muss {$TEMP_WARN_STATUS} ”1” dem angegebenen Wert entsprechen, in diesem Beispiel ”0”. Wie erwartet erscheint ein Ergebnis von ”FALSE”.

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.

3 Auslöserausdruck

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

Mit Funktionen können erfasste Werte analysiert werden, zum Beispiel zur Berechnung des Durchschnitts oder zum Finden einer bestimmten Zeichenfolge.

Klicken Sie auf die jeweilige 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, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate
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, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, trim, xmlxpath

Sofern nicht anders angegeben, werden diese Funktionen unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Foreach-Funktionen werden nur für Aggregatberechnungen unterstützt.

Typischerweise geben Funktionen numerische Werte zum Vergleich zurück. Wenn Zeichenfolgen zurückgegeben werden, ist ein Vergleich mit den Operatoren **=** und **<>** möglich (siehe Beispiel Nicht übereinstimmende Software auf verschiedenen Hosts erkennen).

Funktionsparameter

Mit Funktionsparametern kann Folgendes angegeben werden:

- Datenpunktschlüssel (als `/Host/Schlüssel`) für Funktionen, die auf den Verlauf eines Host-Datenpunkts verweisen
- Zeitraum (und andere 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.

Zeitperiode

Funktionsspezifische Parameter werden nach dem Datenpunktschlüssel platziert und durch ein Komma vom Datenpunktschlüssel getrennt.

Die meisten numerischen Funktionen akzeptieren **Zeitperiode** als Parameter. Damit kann das Intervall angegeben werden, für das wir uns interessieren. Es kann als Zeitperiode (30s, 10m, 1h) oder als Wertebereich (#5 - für die fünf neuesten Werte) angegeben werden.

Sie können Sekunden oder Zeitsuffixe verwenden, um die Zeitperiode anzugeben. Wenn dem Parameter eine Raute vorangestellt ist, hat er eine andere Bedeutung:

Expression	Description
sum(/host/key,**10m)**	Summe der Werte in den letzten 10 Minuten.
sum(/host/key,**#10)**	Summe der letzten zehn Werte.

Parameter mit einer Raute 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 also:

- `last(/host/key,#2)` würde '70' zurückgeben
- `last(/host/key,#5)` würde '50' zurückgeben

Die Zeitperiode wird bis zu „jetzt" gemessen - wobei „jetzt" der Zeitpunkt der letzten Neuberechnung des Auslösers ist (siehe Berechnungshäufigkeit); „jetzt" ist nicht die „jetzt"-Zeit des Servers.

Die Zeitperiode gibt entweder Folgendes an:

- Es werden alle Werte zwischen „jetzt-Zeitperiode" und „jetzt" berücksichtigt (oder bei Zeitverschiebung zwischen „jetzt-Zeitverschiebung-Zeitperiode" und „jetzt-Zeitverschiebung")
- Es werden nicht mehr als die num Anzahl von Werten aus der Vergangenheit berücksichtigt, bis zu „jetzt"
  - Wenn für die angegebene Zeitperiode oder num Anzahl 0 Werte verfügbar sind, wird der Auslöser oder berechnete Datenpunkt, der diese Funktion verwendet, nicht unterstützt

Beachten Sie:

- Wenn im Auslöser nur eine einzelne Funktion (die auf den Datenverlauf verweist) verwendet wird, ist „jetzt" immer der zuletzt empfangene Wert. Wenn beispielsweise der letzte Wert vor einer Stunde empfangen wurde, wird die Zeitperiode so betrachtet, dass sie bis zum letzten Wert vor einer Stunde reicht.
- Ein neuer Auslöser wird berechnet, sobald der erste Wert empfangen wird (Verlaufsfunktionen); für Funktionen zu Datum und Uhrzeit und nodata() erfolgt die Berechnung innerhalb von 30 Sekunden. Somit wird der Auslöser auch dann berechnet, obwohl die festgelegte Zeitperiode (zum Beispiel eine Stunde) seit der Erstellung des Auslösers möglicherweise noch nicht verstrichen ist. Der Auslöser wird auch nach dem ersten Wert berechnet, selbst wenn der Zeitbereich beispielsweise auf die zehn neuesten Werte gesetzt wurde.

Zeitverschiebung

Eine optionale Zeitverschiebung wird mit Zeit- oder Wertanzahl als Funktionsparameter unterstützt. Dieser Parameter ermöglicht es, auf Daten aus einem vergangenen Zeitraum zu verweisen.

Die Zeitverschiebung beginnt mit `now` – zur Angabe der aktuellen Zeit – und wird gefolgt von `+N<time unit>` oder `-N<time unit>`, um N Zeiteinheiten zu addieren oder zu subtrahieren.

Zum Beispiel 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, zum Beispiel Mitternacht bis Mitternacht für einen Tag, Montag bis Sonntag für eine Woche, erster Tag bis letzter Tag des Monats für einen Monat.

Die Zeitverschiebung für absolute Zeiträume beginnt mit `now` – zur Angabe der aktuellen Zeit – und wird gefolgt von einer beliebigen Anzahl von Zeitoperationen: `/<time unit>` definiert den Anfang und das Ende der Zeiteinheit, zum Beispiel Mitternacht bis Mitternacht für einen Tag; `+N<time unit>` oder `-N<time unit>` dienen dazu, N Zeiteinheiten zu addieren oder zu subtrahieren.

Bitte beachten Sie, dass der Wert der Zeitverschiebung größer oder gleich 0 sein kann, während der Mindestwert des Zeitraums 1 ist.

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 Auslöser-Ausdruck erscheint, wenn:

- ein nicht unterstützter Datenpunkt verwendet wird
- die Funktionsauswertung für einen unterstützten Datenpunkt zu einem Fehler führt

In diesem Fall wird der Auslöser-Ausdruck im Allgemeinen zu `Unknown` ausgewertet (da er nicht ausgewertet werden kann)

Es ist möglich, bei unbekannten Auslösern benachrichtigt zu werden.

Ausnahmen

Trotz eines unbekannten Operanden können Auslöser-Ausdrücke 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 nimmt als unbekannter Operand an der weiteren Auswertung des Ausdrucks teil.

Unbekannte Operanden können nur in logischen Ausdrücken „verschwinden“, wie oben beschrieben. In arithmetischen Ausdrücken führen unbekannte Operanden immer zu `Unknown` (außer bei Division durch 0).

Attention:
Der unbekannte Ausdruckszustand ändert den Auslöser-Zustand (Problem/OK) nicht. Wenn der Auslöser-Zustand also „Problem“ war (siehe Fall 1), bleibt er im Problemzustand, selbst wenn der bekannte Teil behoben wird ('1' wird zu '0'), weil der Ausdruck jetzt zu `Unknown` ausgewertet wird und das den Auslöser-Zustand nicht ändert.

Wenn ein Auslöser-Ausdruck 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.

1 Aggregatfunktionen

Sofern nicht anders angegeben, werden alle hier aufgeführten Funktionen unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Aggregatfunktionen können arbeiten mit:

- der Historie von Datenpunkten, zum Beispiel `min(/host/key,1h)`
- foreach-Funktionen als einzigem Parameter, zum Beispiel `min(last_foreach(/*/key))` (nur in berechneten Datenpunkten; kann nicht in Auslösern verwendet werden)

Die Funktionen sind 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 vorhandener Datenpunkte in der Konfiguration, die den Filterkriterien entsprechen.
kurtosis	Die „Schwanzlastigkeit“ der Wahrscheinlichkeitsverteilung der erfassten Werte innerhalb des definierten Auswertungszeitraums.
mad	Die mediane absolute Abweichung der erfassten 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.
skewness	Die Asymmetrie der Wahrscheinlichkeitsverteilung der erfassten Werte innerhalb des definierten Auswertungszeitraums.

Function	Description
stddevpop	Die Populationsstandardabweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums.
stddevsamp	Die Stichprobenstandardabweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums.
sum	Die Summe der erfassten Werte innerhalb des definierten Auswertungszeitraums.
sumofsquares	Die Summe der Quadrate der erfassten Werte innerhalb des definierten Auswertungszeitraums.
varpop	Die Populationsvarianz der erfassten Werte innerhalb des definierten Auswertungszeitraums.
varsamp	Die Stichprobenvarianz der erfassten Werte innerhalb des definierten Auswertungszeitraums.

Allgemeine Parameter

- `/host/key` ist ein allgemeiner obligatorischer erster Parameter für die Funktionen, die auf die Historie des Host-Datenpunkts verweisen
- `(sec|#num)<:time shift>` ist ein allgemeiner zweiter Parameter für die Funktionen, die auf die Historie des Host-Datenpunkts verweisen, wobei:
  - **sec** - maximaler Auswertungszeitraum in Sekunden (Zeit-Suffixe können verwendet werden), oder
  - **#num** - maximaler Auswertungsbereich in den zuletzt erfassten Werten (wenn ihm ein Hash-Zeichen vorangestellt ist)
  - **time shift** (optional) ermöglicht es, den Auswertungszeitpunkt zeitlich zurückzuverschieben. Siehe weitere Details zur Angabe von time shift.

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 eines früheren Zeitpunkts verglichen werden soll.

Beispiele:

```
avg(/host/key,1h) #der Durchschnittswert der letzten Stunde bis jetzt
avg(/host/key,1h:now-1d) #der Durchschnittswert für eine Stunde von vor 25 Stunden bis vor 24 Stunden
avg(/host/key,#5) #der Durchschnittswert der fünf neuesten Werte
avg(/host/key,#5:now-1d) #der Durchschnittswert der fünf neuesten Werte ohne die in den letzten 24 Stunden
```

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))`.

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. Siehe foreach-Funktionen für Details. 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 als
ge - größer oder gleich
lt - kleiner als
le - kleiner oder gleich
like - trifft zu, 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 ignorierender Abgleich mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden); unterstützt, wenn *operator* im dritten Parameter angegeben ist.

Kommentare:

- Die Verwendung von **count()** mit einer verlaufsbezogenen foreach-Funktion (max_foreach, avg_foreach usw.) kann Auswirkungen auf die Performance haben, während die Verwendung von **exists_foreach()**, das nur mit Konfigurationsdaten arbeitet, keinen solchen Effekt hat.
- Optionale Parameter *operator* oder *pattern* können nach einem Komma nicht leer gelassen, sondern nur vollständig weggelassen werden.
- Wenn *bitand* als dritter Parameter verwendet wird, kann der vierte Parameter `pattern` als zwei durch '/' getrennte Zahlen angegeben werden: **number_to_compare_with/mask**. count() berechnet aus dem Wert und der *mask* ein „bitweises UND" 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 '/').
- Wenn *regexp* oder *iregexp* als dritter Parameter verwendet wird, kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler (mit '@' beginnender) regulärer Ausdruck sein. Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung aus den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Zweck des regexp-Abgleichs werden Float-Werte immer mit 4 Dezimalstellen nach dem '.' dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede zwischen der dezimalen Darstellung (in der Datenbank gespeichert) und der binären Darstellung (vom Zabbix Server verwendet) die 4. Dezimalstelle beeinflussen können.

Beispiele:

```
count(max_foreach(/*/net.if.in[*],1h)) #die Anzahl der net.if.in-Datenpunkte, die in der letzten Stunde bi
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #die Anzahl der Dateisysteme mit mehr als 95 % belegt
```

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' von 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"]) #die Anzahl der Hosts mit dem Datenpunkt *agent.ping* in
```

kurtosis(/host/key,(sec|#num)<:time shift>)

Die „Schwänzigkeit“ der Wahrscheinlichkeitsverteilung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Kurtosis.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiel:

```
kurtosis(/host/key,1h) #Kurtosis für die letzte Stunde bis jetzt
```

mad(/host/key,(sec|#num)<:time shift>)

Die mediane absolute Abweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Mediane absolute Abweichung.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiel:

```
mad(/host/key,1h) #mediane absolute Abweichung für die letzte Stunde bis jetzt
```

max(/host/key,(sec|#num)<:time shift>)

Der höchste 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.

Beispiel:

```
max(/host/key,1h) - min(/host/key,1h) #berechnet die Differenz zwischen den maximalen und minimalen Werten
```

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.

Beispiel:

```
max(/host/key,1h) - min(/host/key,1h) #berechnet die Differenz zwischen den maximalen und minimalen Werten
```

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.

Beispiel:

```
skewness(/host/key,1h) #die Schiefe für die letzte Stunde bis jetzt
```

stddevpop(/host/key,(sec|#num)<:time shift>)

Die Populationsstandardabweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Standardabweichung.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiel:

```
stddevpop(/host/key,1h) #die Populationsstandardabweichung für die letzte Stunde bis jetzt
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

Die Stichproben-Standardabweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Standardabweichung.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Damit diese Funktion arbeitet, sind mindestens zwei Datenwerte erforderlich.

Beispiel:

```
stddevsamp(/host/key,1h) #die Stichproben-Standardabweichung für die letzte Stunde bis jetzt
```

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.

Beispiel:

```
sum(/host/key,1h) #die Summe der Werte der letzten Stunde bis jetzt
```

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.

Beispiel:

```
sumofsquares(/host/key,1h) #die Summe der Quadrate der letzten Stunde bis jetzt
```

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.

Beispiel:

```
varpop(/host/key,1h) #die Populationsvarianz der letzten Stunde bis jetzt
```

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.

Damit diese Funktion funktioniert, sind mindestens zwei Datenwerte erforderlich.

Beispiel:

```
varsamp(/host/key,1h) #die Stichprobenvarianz der letzten Stunde bis jetzt
```

Siehe alle unterstützten Funktionen.

Foreach-Funktionen

Übersicht

Foreach-Funktionen werden in aggregierten Berechnungen verwendet, um für jeden Datenpunkt, der durch den verwendeten **Datenpunkt-Filter** ausgewählt wird, einen aggregierten Wert zurückzugeben. Es wird ein Array von Werten zurückgegeben.

> **Attention:**
> Foreach-Funktionen werden nur innerhalb von berechneten Datenpunkten als Teil von Formeln für aggregierte Berechnungen unterstützt. Sie können in diesem Kontext nicht in einfachen Datenpunktschlüsseln oder Auslöser-Ausdrücken außerhalb dieses Kontexts aufgerufen werden.

Zum Beispiel gibt die Funktion *avg_foreach* ein Array von Werten zurück, wobei jeder Wert der *durchschnittliche* Verlaufswert des ausgewählten Datenpunkts während des angegebenen Zeitintervalls ist.

Der Datenpunkt-Filter ist Teil der von Foreach-Funktionen verwendeten Syntax. Die Verwendung von Platzhaltern wird im Datenpunkt-Filter unterstützt, sodass die erforderlichen Datenpunkte sehr flexibel ausgewählt werden können.

Unterstützte Funktionen

Function	Beschreibung
avg_foreach	Gibt den Durchschnittswert für jeden Datenpunkt zurück.

Function	Beschreibung
bucket_rate_foreach	Gibt Paare (obere Bucket-Grenze, Ratenwert) zurück, die für die Verwendung in der Funktion histogram_quantile() geeignet sind, wobei die „obere Bucket-Grenze“ der Wert des Schlüsselsparameters des Datenpunkts ist, der durch den Parameter <parameter number> definiert wird.
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 Maximalwert für jeden Datenpunkt zurück.
min_foreach	Gibt den Minimalwert 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.

Zeitperiode

Der **zweite** Parameter ermöglicht die Angabe der Zeitperiode für die Aggregation. Die Zeitperiode kann nur als Zeitangabe ausgedrückt werden; die Anzahl der Werte (mit **#** vorangestellt) wird nicht unterstützt.

Unterstützte Einheitensymbole können in diesem Parameter der Einfachheit halber verwendet werden, zum Beispiel „5m" (fünf Minuten) statt „300s" (300 Sekunden) oder „1d" (ein Tag) statt „86400" (86400 Sekunden).

Für die Funktion *last_foreach* ist die Zeitperiode ein optionaler Parameter (unterstützt seit Zabbix 7.0), der weggelassen werden kann:

`last_foreach(/*/key?[group="host group"])`

Die Zeitperiode wird von der Funktion *exists_foreach* nicht unterstützt.

Zusätzliche Parameter

bucket_rate_foreach

Ein dritter optionaler Parameter wird von der Funktion *bucket_rate_foreach* unterstützt:

`bucket_rate_foreach(item filter,time period,<parameter number>)`

wobei <parameter number> die Position des „bucket"-Werts im Datenpunktschlüssel ist. Wenn beispielsweise der „bucket"-Wert in myItem[aaa,0.2] „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 Datenpunktwerte (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 unter Beachtung der Groß-/Kleinschreibung mit dem

in `pattern` angegebenen regulären Ausdruck
iregexp - Übereinstimmung ohne Beachtung der Groß-/Kleinschreibung mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** ist das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden); wird unterstützt, wenn *operator* im dritten Parameter angegeben ist.

Kommentare:

- Optionale Parameter *operator* oder *pattern* können nach einem Komma nicht leer gelassen, sondern nur vollständig weggelassen werden.
- Wenn *bitand* als dritter Parameter verwendet wird, kann der vierte Parameter `pattern` als zwei durch „/“ getrennte Zahlen angegeben werden: **number_to_compare_with/mask**. count_foreach() berechnet aus dem Wert und der *mask* ein „bitweises UND“ 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 „/“).
- Wenn *regexp* oder *iregexp* als dritter Parameter verwendet wird, kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler (mit „@“ beginnender) regulärer Ausdruck sein. Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung aus den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Zweck des regexp-Abgleichs werden Float-Werte immer mit 4 Dezimalstellen nach dem „.“ dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede zwischen der dezimalen (in der Datenbank gespeicherten) und der 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.

2 Bitweise Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Die Funktionen sind ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
bitand	Der Wert von „bitweisem UND“ eines Datenpunktwerts und einer Maske.
bitlshift	Die bitweise Linksverschiebung eines Datenpunktwerts.

Function	Description
bitnot	Der Wert von „bitweisem NICHT“ eines Datenpunktwerts.
bitor	Der Wert von „bitweisem ODER“ eines Datenpunktwerts und einer Maske.
bitrshift	Die bitweise Rechtsverschiebung eines Datenpunktwerts.
bitxor	Der Wert von „bitweisem exklusivem ODER“ 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.

3 Datums- und Zeitfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöserausdrücken
- Berechneten Datenpunkten
- Ausdrucksmakros

Attention:
Datums- und Zeitfunktionen können im Ausdruck nicht allein verwendet werden; im Ausdruck muss mindestens eine Funktion aus einer anderen Gruppe enthalten sein, die auf den Host-Datenpunkt verweist (mit Ausnahme der Funktion nodata()). Detaillierte Informationen dazu, wie Datums- und Zeitfunktionen in Ausdrücken funktionieren, finden Sie unter Berechnungszeit.

Die Funktionen sind ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
date	Das aktuelle Datum im Format JJJJMMTT.
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 Epoch (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.

4 Verlaufsfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Die Funktionen sind ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
change	Die Differenz zwischen dem vorherigen und dem neuesten Wert.
changecount	Die Anzahl der Änderungen zwischen benachbarten 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 eine Wertübereinstimmung innerhalb des definierten Auswertungszeitraums.
first	Der erste (älteste) Wert innerhalb des definierten Auswertungszeitraums.
firstclock	Der Zeitstempel des ersten (ältesten) Werts innerhalb des definierten Auswertungszeitraums.
fuzzytime	Prüft, wie stark sich die Zeit des passiven Agent von der Zeit des Zabbix Server/Proxy unterscheidet.
last	Der neueste Wert.
lastclock	Der Zeitstempel des N-tjüngsten 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-tjüngsten Werts eines Log-Datenpunkts.
monodec	Prüft, ob es eine monotone Abnahme der Werte gegeben hat.
monoinc	Prüft, ob es eine monotone Zunahme der Werte gegeben hat.
nodata	Prüft, ob keine Daten empfangen wurden.
percentile	Das P-te Perzentil eines Zeitraums, wobei P (Prozentsatz) durch den dritten Parameter angegeben wird.
rate	Die durchschnittliche Rate pro Sekunde der Zunahme eines monoton steigenden Zählers innerhalb des definierten Zeitraums.

Allgemeine Parameter

- `/host/key` ist ein gemeinsamer obligatorischer erster Parameter für die Funktionen, die auf die Datenpunkthistorie des Hosts verweisen
- `(sec|#num)<:time shift>` ist ein gemeinsamer zweiter Parameter für die Funktionen, die auf die Datenpunkthistorie 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 Hash-Zeichen vorangestellt ist)
  - **time shift** (optional) ermöglicht es, den Auswertungszeitpunkt zeitlich zurückzuverschieben. Siehe weitere Details zur Angabe von time shift.

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 Wertetypen: *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 als
ge - größer oder gleich
lt - kleiner als
le - kleiner oder gleich
like (Standard für string, text, log) - trifft zu, wenn das Muster enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises AND
regexp - Übereinstimmung unter Beachtung der Groß-/Kleinschreibung mit dem in `pattern` angegebenen regulären Ausdruck
iregexp - Übereinstimmung ohne Beachtung der Groß-/Kleinschreibung mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden)

Kommentare:

- Float-Datenpunkte werden mit einer Genauigkeit von 2.22e-16 abgeglichen.
- *like* wird als operator für Integer-Werte nicht unterstützt.

- *like* und *bitand* werden als operatoren für Float-Werte nicht unterstützt.
- Für String-, Text- und Log-Werte werden nur die operatoren *eq*, *ne*, *like*, *regexp* und *iregexp* unterstützt.
- Bei Verwendung von *bitand* als operator kann der vierte Parameter `pattern` als zwei durch '/' getrennte Zahlen angegeben werden: **number_to_compare_with/mask**. count() berechnet aus dem Wert und der *mask* ein "bitweises AND" und vergleicht das Ergebnis mit *number_to_compare_with*. Wenn das Ergebnis des "bitweisen AND" 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 Verwendung von *regexp* oder *iregexp* als operator kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler (mit '@' beginnender) regulärer Ausdruck sein. Bei globalen regulären Ausdrücken wird die Beachtung der Groß-/Kleinschreibung aus den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Zweck des regexp-Abgleichs werden Float-Werte immer mit 4 Nachkommastellen nach '.' dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede zwischen der dezimalen Darstellung (in der Datenbank gespeichert) und der binären Darstellung (vom Zabbix Server verwendet) die 4. Nachkommastelle beeinflussen können.

Beispiele:

```
count(/host/icmpping,30m,,"0")>5 #Ping ist in 30 Minuten mehr als 5 Mal fehlgeschlagen
count(/host/key,10m,"like","error") #die Anzahl der Werte in den letzten 10 Minuten bis jetzt, die 'error'
count(/host/key,10m,,12) #die Anzahl der Werte in den letzten 10 Minuten bis jetzt, die gleich '12' sind
count(/host/key,10m,"gt",12) #die Anzahl der Werte in den letzten 10 Minuten bis jetzt, die größer als '12
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 vor 24 Stunden und vor 24 Stunden minus
count(/host/key,10m,"bitand","6/7") #die Anzahl der Werte in den letzten 10 Minuten bis jetzt, die in den
count(/host/key,10m:now-1d) #die Anzahl der Werte zwischen vor 24 Stunden und vor 24 Stunden minus 10 Minu
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Die Anzahl eindeutiger Werte innerhalb des definierten Auswertungszeitraums.
 Unterstützte Wertetypen: *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 als
ge - größer oder gleich
lt - kleiner als
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 - Groß-/Kleinschreibung beachtende Übereinstimmung mit dem in `pattern` angegebenen regulären Ausdruck
iregexp - Groß-/Kleinschreibung ignorierende Übereinstimmung mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden)

Kommentare:

- Float-Datenpunkte werden mit einer Genauigkeit von 2.22e-16 abgeglichen.
- *like* wird als operator für Integer-Werte nicht unterstützt.
- *like* und *bitand* werden als operatoren für Float-Werte nicht 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 durch '/' getrennte Zahlen angegeben werden: **number_to_compare_with/mask**. countunique() berechnet aus dem Wert und der *mask* ein "bitweises UND" 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 (mit '@' beginnender) regulärer Ausdruck sein. Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung aus 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 zwischen der dezimalen Darstellung (in der Datenbank gespeichert) und der binären Darstellung (vom Zabbix Server verwendet) die 4. Dezimalstelle beeinflussen können.

Beispiele:

```
countunique(/host/key,10m) #die Anzahl eindeutiger Werte in den letzten 10 Minuten bis jetzt
countunique(/host/key,10m,"like","error") #die Anzahl eindeutiger Werte in den letzten 10 Minuten bis jetz
countunique(/host/key,10m,,12) #die Anzahl eindeutiger Werte in den letzten 10 Minuten bis jetzt, die glei
countunique(/host/key,10m,"gt",12) #die Anzahl eindeutiger Werte in den 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 vor 24 Stunden und vor 24
```

```
countunique(/host/key,10m,"bitand","6/7") #die Anzahl eindeutiger Werte in den letzten 10 Minuten bis jetz
countunique(/host/key,10m:now-1d) #die Anzahl eindeutiger Werte zwischen vor 24 Stunden und vor 24 Stunden
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Findet eine Wertübereinstimmung innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.
 Rückgabewert: 1 - gefunden; 0 - andernfalls.

Parameter:

- Siehe gemeinsame Parameter
- **sec** oder **#num** (optional) - standardmäßig der letzte Wert, wenn nicht angegeben
- **operator** (muss in doppelte Anführungszeichen gesetzt werden) Unterstützte `operators`:
eq - gleich (Standard für integer, float)
ne - ungleich
gt - größer als
ge - größer oder gleich
lt - kleiner als
le - kleiner oder gleich
like (Standard für string, text, log) - stimmt überein, wenn die in `pattern` angegebene Zeichenkette enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises AND
regexp - Übereinstimmung unter Beachtung der Groß-/Kleinschreibung mit dem in `pattern` angegebenen regulären Ausdruck
iregexp - Übereinstimmung ohne Beachtung der Groß-/Kleinschreibung mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden); regulärer Ausdruck nach Perl Compatible Regular Expression (PCRE), wenn `operator` *regexp* oder *iregexp* ist

Kommentare:

- Wenn mehr als ein Wert verarbeitet wird, wird '1' zurückgegeben, wenn es mindestens einen übereinstimmenden Wert gibt.
- *like* wird als Operator für Integer-Werte nicht unterstützt.
- *like* und *bitand* werden als Operatoren für Float-Werte nicht unterstützt.
- Für String-, Text- und Log-Werte werden nur die Operatoren *eq*, *ne*, *like*, *regexp* und *iregexp* unterstützt.
- Bei Verwendung von *regexp* oder *iregexp* als Operator kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler (beginnend mit '@') regulärer Ausdruck sein. Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung aus den Einstellungen des globalen regulären Ausdrucks übernommen.

Beispiele:

```
find(/host/key,10m,"like","error") #find a value that contains 'error' within the last 10 minutes until no
find(/host/agent.version,,"like","beta")=1 #Zabbix agent has beta version, must be upgraded
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #internal web server error has been found
```

first(/host/key,sec<:time shift>)

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 sich die Zeit des passiven Agent von der Zeit des Zabbix Server/Proxy unterscheidet.
 Unterstützte Werttypen: *Float*, *Integer*.
 Rückgabewerte: 1 - die Differenz zwischen dem Wert des passiven Datenpunkts (als Zeitstempel) und dem Zeitstempel des Zabbix Server/Proxy (dem Zeitpunkt der Werterfassung) ist kleiner oder gleich *sec* Sekunden; andernfalls 0.

Parameter:

- Siehe allgemeine Parameter.

Kommentare:

- Wird üblicherweise mit dem Datenpunkt 'system.localtime' verwendet, um zu prüfen, dass die lokale Zeit mit der lokalen Zeit des Zabbix Server synchron ist. *Beachten Sie*, dass 'system.localtime' für den Zabbix Agent als passive Prüfung konfiguriert sein muss; bei 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, dass die Datei lange Zeit nicht aktualisiert wurde.
- Diese Funktion wird nicht zur Verwendung in komplexen Auslöserausdrücken empfohlen (mit mehreren beteiligten Datenpunkten), da dies zu unerwarteten Ergebnissen führen kann (die Zeitdifferenz wird anhand der neuesten Metrik gemessen), z. B. in `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Beispiele:

```
fuzzytime(/host/system.localtime,5m)=0 #lokale Client-Zeit unterscheidet sich um mehr als 5 Minuten von de
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #lokale Client-Zeit unters
```

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 Log-Eintrags mit einem regulären Ausdruck übereinstimmt.
 Unterstützte Werttypen: *Log*.
 Rückgabewerte: 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) (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden)

Beispiele:

```
logeventid(/host/eventlog[Security],,"4625")=1 #a log entry with ID matching "4625" (failed authentication
logeventid(/host/eventlog[System],,"6008|41")=1 #a log entry with ID matching "6008" or "41" found
```

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 Log-Quelle des letzten Log-Eintrags mit einem regulären Ausdruck übereinstimmt.
 Unterstützte Wertetypen: *Log*.
 Rückgabewerte: 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) (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden)

Wird normalerweise für Windows-Ereignisprotokolle verwendet.

Beispiele:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #ein Log-Eintrag mit einer Quelle, die mit "MSSQLS
logsource(/host/eventlog[System],,"Service Control Manager")=1 #ein Log-Eintrag mit einer Quelle, die mit
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"703
```

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 Wertetypen: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Rückgabewert: 1 - wenn während des definierten Zeitraums keine Daten empfangen wurden; andernfalls 0.

Parameter:

- Siehe allgemeine Parameter
- **sec** - der Zeitraum sollte nicht kürzer als 30 Sekunden sein, da der History-Syncer-Prozess diese Funktion nur alle 30 Sekunden berechnet; `nodata(/host/key,0)` ist nicht zulässig
- **mode** - wenn auf *strict* gesetzt (in doppelten Anführungszeichen), ist diese Funktion unempfindlich gegenüber der Verfügbarkeit des Proxy (siehe Kommentare für Details)

Kommentare:

- die von einem Proxy überwachten 'nodata'-Auslöser reagieren standardmäßig empfindlich auf die Verfügbarkeit des Proxy - wenn der Proxy nicht mehr verfügbar ist, werden die 'nodata'-Auslöser nicht sofort nach Wiederherstellung der Verbindung ausgelöst, sondern überspringen die Daten für den verzögerten Zeitraum. Beachten Sie, dass bei passiven Proxys die Unterdrückung aktiviert wird, wenn die Verbindung mehr als 15 Sekunden und nicht weniger als 2 Sekunden später wiederhergestellt wird. Bei aktiven Proxys wird die Unterdrückung aktiviert, wenn die Verbindung mehr als 15 Sekunden später wiederhergestellt wird. Um die Empfindlichkeit gegenüber der Verfügbarkeit des Proxy 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 verstrichen ist.
- Diese Funktion zeigt einen Fehler an, wenn innerhalb des im 1. Parameter angegebenen Zeitraums:
- keine Daten vorhanden sind und der Zabbix Server neu gestartet wurde
- keine Daten vorhanden sind und eine Wartung abgeschlossen wurde
- keine Daten vorhanden sind und der Datenpunkt hinzugefügt oder wieder aktiviert wurde.
- Fehler werden in der Spalte *Info* in der Auslöser-Konfiguration angezeigt.
- Diese Funktion funktioniert möglicherweise nicht korrekt, wenn es Zeitunterschiede zwischen Zabbix Server, Proxy und Agent gibt. Siehe auch: Anforderung an die Zeitsynchronisierung.

Beispiel:

```
nodata(/host/agent.ping,5m)=1 #Auslöser, wenn 5 Minuten lang keine Daten vom Zabbix Agent empfangen wurden
```

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
```

Siehe alle unterstützten Funktionen.

5 Trendfunktionen

Trendfunktionen verwenden im Gegensatz zu Verlaufsfunktionen Trend-Daten für Berechnungen.

Trends speichern stündlich aggregierte Werte. Trendfunktionen verwenden diese stündlichen Durchschnittswerte und sind daher für Langzeitanalysen nützlich.

Die Ergebnisse von Trendfunktionen werden zwischengespeichert, sodass bei mehreren Aufrufen derselben Funktion mit denselben Parametern Informationen nur einmal aus der Datenbank abgerufen werden. Der Cache für Trendfunktionen wird durch den Server-Parameter TrendFunctionCacheSize gesteuert.

Auslöser, die **nur** auf Trendfunktionen verweisen, werden einmal pro kleinstem Zeitraum im Ausdruck ausgewertet. Zum Beispiel 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 Verlaufsfunktionen (oder Funktionen für Datum und Uhrzeit und/oder nodata()) enthält, wird er gemäß den üblichen Prinzipien berechnet.

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöserausdrücken
- Berechneten Datenpunkten
- Ausdrucksmakros

Die Funktionen sind hier ohne zusätzliche Informationen aufgeführt. Klicken Sie auf eine Funktion, um die vollständigen Details anzuzeigen.

Function	Description
baselinedev	Gibt die Anzahl der Abweichungen (nach dem Algorithmus stddevpop) zwischen dem letzten Datenzeitraum und denselben Datenzeiträumen in vorhergehenden Saisons zurück.
baselinewma	Berechnet die Baseline durch Mittelung von Daten aus demselben Zeitrahmen in mehreren gleich langen Zeiträumen („Saisons“) unter Verwendung des gewichteten gleitenden Durchschnitts.
trendavg	Der Durchschnitt der Trendwerte innerhalb des definierten Zeitraums.
trendcount	Die Anzahl der erfolgreich abgerufenen Verlaufswerte, die zur Berechnung des Trendwerts innerhalb des definierten Zeitraums verwendet werden.

Function	Description
trendmax	Das Maximum der Trendwerte innerhalb des definierten Zeitraums.
trendmin	Das Minimum der Trendwerte innerhalb des definierten Zeitraums.
trendstl	Gibt die Rate der Anomalien während des Erkennungszeitraums zurück – ein Dezimalwert zwischen 0 und 1, der `((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.

6 Mathematische Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Mathematische Funktionen werden mit den Werttypen Gleitkommazahl und Ganzzahl unterstützt, sofern nicht anders angegeben.

Die Funktionen sind ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
abs	Der Absolutwert eines Wertes.
acos	Der Arkuskosinus eines Wertes als Winkel, ausgedrückt in Radiant.
asin	Der Arkussinus eines Wertes als Winkel, ausgedrückt in Radiant.
atan	Der Arkustangens eines Wertes als Winkel, ausgedrückt in Radiant.
atan2	Der Arkustangens der angegebenen Ordinaten- (Wert) und Abszissenkoordinaten als Winkel, ausgedrückt in Radiant.
avg	Der Durchschnittswert der referenzierten Datenpunktwerte.
cbrt	Die Kubikwurzel eines Wertes.
ceil	Den Wert auf die nächstgrößere oder gleiche Ganzzahl aufrunden.
cos	Der Kosinus eines Wertes, wobei der Wert ein in Radiant ausgedrückter Winkel ist.
cosh	Der hyperbolische Kosinus eines Wertes.
cot	Der Kotangens eines Wertes, wobei der Wert ein in Radiant ausgedrückter Winkel ist.
degrees	Wandelt einen Wert von Radiant in Grad um.
e	Die Eulersche Zahl (2.718281828459045).
exp	Die Eulersche Zahl hoch einem Wert.
expm1	Die Eulersche Zahl hoch einem Wert minus 1.
floor	Den Wert auf die nächstkleinere oder gleiche Ganzzahl abrunden.
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 Wertes.
radians	Wandelt einen Wert von Grad in Radiant um.
rand	Gibt einen zufälligen Ganzzahlwert zurück.
round	Den Wert auf Dezimalstellen runden.
signum	Gibt '-1' zurück, wenn ein Wert negativ ist, '0', wenn ein Wert null ist, und '1', wenn ein Wert positiv ist.
sin	Der Sinus eines Wertes, wobei der Wert ein in Radiant ausgedrückter Winkel ist.
sinh	Der hyperbolische Sinus eines Wertes, wobei der Wert ein in Radiant ausgedrückter Winkel ist.
sqrt	Die Quadratwurzel eines Wertes.
sum	Die Summe der referenzierten Datenpunktwerte.
tan	Der Tangens eines Wertes.
truncate	Den Wert auf Dezimalstellen abschneiden.

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.

7 Operatorfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöserausdrücken
- Berechneten Datenpunkten
- Ausdrucksmakros

Die Funktionen sind hier ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
between	Prüfen, ob der Wert zum angegebenen Bereich gehört.
in	Prüfen, 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.

8 Prädiktive Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Die Funktionen sind ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
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 sich auf die Historie des Host-Datenpunkts beziehen
- `(sec|#num)<:time shift>` ist ein gemeinsamer zweiter Parameter für die Funktionen, die sich auf die Historie des Host-Datenpunkts beziehen, wobei:
  - **sec** - maximaler Auswertungszeitraum in Sekunden (Zeit-Suffixe können verwendet werden), oder
  - **#num** - maximaler Auswertungsbereich in den zuletzt erfassten Werten (wenn eine Raute vorangestellt ist)
  - **time shift** (optional) ermöglicht es, den Auswertungszeitpunkt in die Vergangenheit zu verschieben. Siehe weitere Details zur Angabe von time shift.

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, Minimum, Delta oder der Durchschnitt des Datenpunkts.
 Unterstützte Wertetypen: *Float*, *Integer*.

Parameter:

- Siehe allgemeine Parameter;
- **time** - der Prognosehorizont in Sekunden (Zeitsuffixe können verwendet werden); negative Werte werden unterstützt;
- **fit** (optional; muss in doppelte Anführungszeichen gesetzt werden) - die Funktion, die zum Anpassen 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* gleichbedeutend mit *linear* ist;
- **mode** (optional; muss in doppelte Anführungszeichen gesetzt 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`.

Kommentare:

- Wenn der zurückzugebende Wert größer als 1.7976931348623158E+308 oder kleiner als -1.7976931348623158E+308 ist, wird der Rückgabewert entsprechend auf 1.7976931348623158E+308 bzw. -1.7976931348623158E+308 begrenzt;
- Wird nur dann nicht unterstützt, wenn die Funktion im Ausdruck falsch verwendet wird (falscher Datenpunkttyp, ungültige Parameter); andernfalls wird im Fehlerfall -1 zurückgegeben;
- Siehe auch zusätzliche Informationen zu prädiktiven Auslöserfunktionen.

Beispiele:

```
forecast(/host/key,#10,1h) #prognostiziert den Datenpunktwert in einer Stunde auf Basis der letzten 10 Wer
forecast(/host/key,1h,30m) #prognostiziert den Datenpunktwert in 30 Minuten auf Basis der Daten der letzte
forecast(/host/key,1h:now-1d,12h) #prognostiziert den Datenpunktwert in 12 Stunden auf Basis einer Stunde
forecast(/host/key,1h,10m,"exponential") #prognostiziert den Datenpunktwert in 10 Minuten auf Basis der Da
forecast(/host/key,1h,2h,"polynomial3","max") #prognostiziert den maximalen Wert, den der Datenpunkt in de
forecast(/host/key,#2,-20m) #schätzt den Datenpunktwert vor 20 Minuten auf Basis der letzten zwei Werte (d
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Die in Sekunden benötigte Zeit, bis ein Datenpunkt den angegebenen Schwellenwert erreicht.
 Unterstützte Wertetypen: *Float*, *Integer*.

Parameter:

- Siehe allgemeine Parameter;
- **threshold** - der zu erreichende Wert (Einheitensuffixe können verwendet werden);
- **fit** (optional; muss in doppelte Anführungszeichen gesetzt 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 im Fehlerfall -1 zurückgegeben;
- Siehe auch zusätzliche Informationen zu prädiktiven Auslöserfunktionen.

Beispiele:

```
timeleft(/host/key,#10,0) #die Zeit, bis der Datenpunktwert auf null sinkt, basierend auf den letzten 10 W
timeleft(/host/key,1h,100) #die Zeit, bis der Datenpunktwert 100 erreicht, basierend auf den Daten der let
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 in historischen Daten gefundenen Abhängigkeiten extrapolieren soll. 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 der Gegenwart liegt und rechts von der Gegenwart liegt, oder 1.7976931348623158E+308, falls es keine solche Nullstelle gibt.

Note:
Wenn sich Datenpunktwerte dem Schwellenwert annähern und ihn dann überschreiten, geht **timeleft** davon aus, dass der 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 also die Daten geringfügig unterscheiden, wird auch die angepasste Funktion geringfügig anders sein. Wenn wir einfach den Wert der angepassten Funktion zu einem angegebenen Zeitpunkt in der Zukunft berechnen, erfahren Sie nichts darüber, wie sich der analysierte Datenpunkt voraussichtlich zwischen jetzt und diesem Zeitpunkt in der Zukunft 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)`

9 String-Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Berechneten Datenpunkten
- Ausdrucks-Makros

Die Funktionen sind hier ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
ascii	Der ASCII-Code des ganz linken 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 das Verketten der referenzierten Datenpunktwerte oder konstanter Werte entsteht.
insert	Fügt angegebene Zeichen oder Leerzeichen in die Zeichenfolge ein, beginnend an der angegebenen Position in der Zeichenfolge.
jsonpath	Gibt das JSONPath-Ergebnis zurück.
left	Gibt die ganz linken Zeichen des Werts zurück.
length	Die Länge des Werts in Zeichen.
ltrim	Entfernt angegebene Zeichen vom Anfang der Zeichenfolge.
mid	Gibt eine Teilzeichenfolge mit N Zeichen zurück, beginnend an der durch 'start' angegebenen Zeichenposition.
repeat	Wiederholt eine Zeichenfolge.
replace	Findet das Muster im Wert und ersetzt es durch den Ersatzwert.
right	Gibt die ganz rechten Zeichen des Werts zurück.
rtrim	Entfernt angegebene Zeichen vom Ende der Zeichenfolge.
trim	Entfernt angegebene 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])
```

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.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **start** - Startposition der Teilzeichenfolge;
- **length** - in der Teilzeichenfolge zurückzugebende Positionen.

Zum Beispiel ist es möglich, aus einem Wert wie 'Zabbix' 'abbi' zurückzugeben, wenn die Startposition 2 ist und die Anzahl der zurückzugebenden Positionen 4 beträgt.

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
```

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.

4 Best Practices für Auslöser

Auslöser sind ein leistungsstarkes Werkzeug, können aber auch unerwünschtes Alarmrauschen verursachen. Um mehr echte Signale und weniger Rauschen zu sehen, befolgen Sie diese Tipps:

1. Machen Sie Auslöser unempfindlicher. Statt beim neuesten Wert zu alarmieren (zu hoch/zu niedrig), analysieren Sie den Durchschnitt über einen längeren Zeitraum, indem Sie Funktionen wie **avg**, **min** und **max** verwenden.
2. Erwägen Sie die Verwendung der Funktion **percentile** (setzen Sie sie auf 95 % oder 5 %) in Auslösern, wenn Sie Warnmeldungen bei zufälligen Spitzen und Einbrüchen vermeiden möchten.
3. Verwenden Sie Hysterese, um Flattern von Auslösern zu vermeiden — häufige Änderungen des Auslöserstatus (OK ↔ Problem). Ein Kontinuum für den Problemstatus kann durch Hinzufügen eines Wiederherstellungsausdrucks definiert werden (eine separate Bedingung für die Problembehebung).
4. Verwenden Sie Abhängigkeiten von Auslösern, um Warnmeldungen zu vermeiden, die nicht mit der Grundursache zusammenhängen.
5. Verwenden Sie den Schweregrad von Auslösern, um nur bei schwerwiegenderen Problemen zu alarmieren.
6. Definieren Sie Wartungsfenster.

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, der sich hinter einem Router befindet, wird unerreichbar, wenn der Router ausfällt. Wenn für beide Auslöser konfiguriert sind, erhalten Sie möglicherweise Benachrichtigungen über zwei ausgefallene Hosts – obwohl tatsächlich nur der Router die Ursache war.

Hier kann eine Abhängigkeit zwischen Hosts nützlich sein. Wenn eine Abhängigkeit eingerichtet ist, können Benachrichtigungen für abhängige Hosts zurückgehalten und nur die Benachrichtigung zum eigentlichen Grundproblem gesendet werden.

Obwohl Zabbix Abhängigkeiten zwischen Hosts nicht direkt unterstützt, können sie mit einer anderen, flexibleren Methode definiert werden – Auslöser- Abhängigkeiten. Ein Auslöser kann von einem oder mehreren anderen Auslösern abhängen.

In unserem einfachen Beispiel öffnen wir also das Konfigurationsformular des Server-Auslösers und legen fest, dass er vom entsprechenden Auslöser des Routers abhängt. Mit einer solchen Abhängigkeit ändert der Server-Auslöser seinen Status nicht, solange sich der Auslöser, von dem er abhängt, im Status „PROBLEM“ befindet – und daher werden keine abhängigen Aktionen ausgeführt und keine Benachrichtigungen gesendet.

Wenn sowohl der Server als auch der Router ausgefallen sind und eine Abhängigkeit besteht, führt Zabbix keine Aktionen für den abhängigen Auslöser aus.

Während sich der übergeordnete Auslöser im Status PROBLEM befindet, können seine abhängigen Auslöser Werte melden, 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“ in den Status „OK“ zurückkehrt;
- seinen Status von „PROBLEM“ auf „UNKNOWN“ ändert;
- manuell, durch Korrelation oder mithilfe der Funktionen date and time und/oder nodata() geschlossen wird;
- durch einen Wert eines Datenpunkts aufgelöst wird, der nicht am abhängigen Auslöser beteiligt ist;
- deaktiviert wird, einen deaktivierten Datenpunkt oder einen Host mit deaktiviertem Datenpunkt hat

In allen oben genannten Fällen wird der abhängige Auslöser (Server) erst dann erneut ausgewertet, wenn ein neuer Messwert dafür empfangen wird. Das bedeutet, dass der abhängige Auslöser möglicherweise nicht sofort aktualisiert wird.

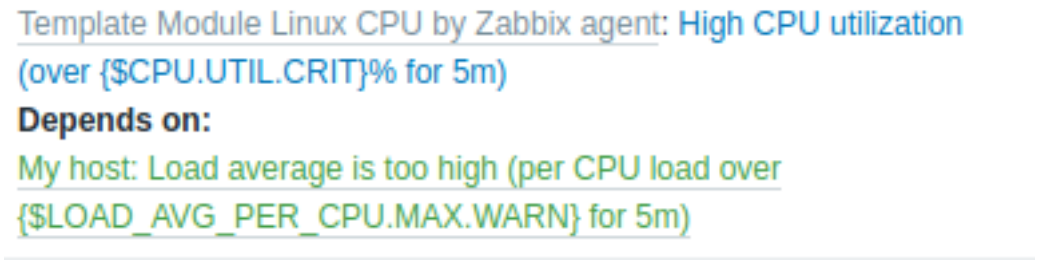
Außerdem gilt:

- Eine Auslöser-Abhängigkeit kann von jedem Host-Auslöser zu jedem anderen Host-Auslöser hinzugefügt werden, solange dadurch keine zirkuläre Abhängigkeit entsteht.
- Eine Auslöser-Abhängigkeit kann von einer Vorlage zu einer anderen 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 Auslöser-Abhängigkeit kann von einem Vorlagen-Auslöser zu einem Host- Auslöser hinzugefügt werden. In diesem Fall erzeugt das Verknüpfen einer solchen Vorlage mit einem Host einen Host-Auslöser, der von derselben Auslöser-Vorlage abhängt, von der der Auslöser abhängig war. Dies ermöglicht es beispielsweise, eine Vorlage zu haben, bei der einige Auslöser von den Router-Auslösern (Host) abhängen. Alle Hosts, die mit dieser Vorlage verknüpft sind, hängen von diesem bestimmten Router ab.
- Eine Auslöser-Abhängigkeit kann nicht von einem Host-Auslöser zu einem Vorlagen-Auslöser hinzugefügt werden.
- Eine Auslöser-Abhängigkeit kann von einem Auslöser-Prototyp zu einem anderen Auslöser-Prototyp (innerhalb derselben Low-Level-Discovery-Regel) oder zu einem echten Auslöser hinzugefügt werden. Ein Auslöser-Prototyp darf nicht von einem Auslöser- Prototyp aus einer anderen LLD-Regel oder von einem aus einem Auslöser-Prototyp erzeugten Auslöser abhängen. Ein Host-Auslöser-Prototyp kann nicht von einem Auslöser aus einer Vorlage abhängen.

Konfiguration

Um eine Abhängigkeit zu definieren, öffnen Sie die Registerkarte „Abhängigkeiten“ im Konfigurationsformular des Auslösers. Klicken Sie im Block „Abhängigkeiten“ auf *Hinzufügen* und wählen Sie einen oder mehrere Auslöser aus, von denen der Auslöser abhängen soll.

Klicken Sie auf *Aktualisieren*. Nun wird der Auslöser mit dem Hinweis auf seine Abhängigkeit in der Liste angezeigt.



Beispiel für mehrere Abhängigkeiten

Zum Beispiel befindet sich der Host hinter Router2, und Router2 befindet sich hinter Router1.

```
Zabbix - Router1 - Router2 - Host
```

Wenn Router1 ausfällt, sind natürlich auch der Host und Router2 nicht erreichbar, doch drei Benachrichtigungen darüber zu erhalten, dass der Host, Router1 und Router2 alle ausgefallen sind, ist übermäßig.

Daher definieren wir in diesem Fall zwei Abhängigkeiten:

```
der Auslöser „Host ist ausgefallen“ hängt vom Auslöser „Router2 ist ausgefallen“ ab
der Auslöser „Router2 ist ausgefallen“ hängt vom Auslöser „Router1 ist ausgefallen“ ab
```

Bevor Zabbix den Status des Auslösers „Host ist ausgefallen“ ändert, prüft es die entsprechenden Auslöser-Abhängigkeiten. 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 Router1 oder 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.

4 Ereignisse

Übersicht

In Zabbix werden mehrere Arten von Ereignissen erzeugt:

- Auslöser-Ereignisse – immer dann, wenn ein Auslöser seinen Status ändert (*OK→PROBLEM→OK*)
- Service-Ereignisse – immer dann, wenn ein Service seinen Status ändert (*OK→PROBLEM→OK*)
- Discovery-Ereignisse – wenn Hosts oder Services erkannt werden
- Autoregistrierungs-Ereignisse – wenn aktive Agenten automatisch durch den Server registriert werden
- interne Ereignisse – wenn ein Datenpunkt bzw. eine Low-Level-Discovery-Regel nicht unterstützt wird oder ein Auslöser in einen unbekannten Zustand übergeht

Ereignisse werden mit einem Zeitstempel versehen und können die Grundlage für Aktionen wie das Senden von Benachrichtigungs-E-Mails usw. sein.

Um Details zu Ereignissen im Frontend anzuzeigen, gehen Sie zu *Monitoring* → *Probleme*. Dort können Sie auf Datum und Uhrzeit des Ereignisses klicken, um Details zu einem Ereignis anzuzeigen.

Weitere Informationen finden Sie unter:

- Auslöser-Ereignisse
- andere Ereignisquellen

1 Erzeugung von Auslöser-Ereignissen

Übersicht

Die Änderung des Auslöser-Status ist die häufigste und wichtigste Quelle von Ereignissen. Jedes Mal, wenn der Auslöser seinen Status ändert, wird ein Ereignis erzeugt. Das Ereignis enthält Details zur Änderung des Auslöser-Status – wann sie stattgefunden hat und was der neue Status ist.

Durch Auslöser werden zwei Arten von Ereignissen erstellt – `Problem` und `OK`.

Problemereignisse

Ein Problemereignis wird erstellt:

- wenn ein Auslöser-Ausdruck zu TRUE ausgewertet wird, falls sich der Auslöser im Status OK befindet;
- jedes Mal, wenn ein Auslöser-Ausdruck zu TRUE ausgewertet wird, falls für den Auslöser die Generierung mehrerer Problemereignisse aktiviert ist.

OK-Ereignisse

Ein OK-Ereignis schließt die zugehörigen Problemereignisse und kann von 3 Komponenten erstellt werden:

- Auslöser – basierend auf den Einstellungen „OK-Ereignisgenerierung“ und „OK-Ereignis schließt“;
- Ereigniskorrelation
- Task-Manager – wenn ein Ereignis manuell geschlossen wird

Auslöser

Auslöser verfügen über eine Einstellung „OK-Ereignisgenerierung“, die steuert, wie OK-Ereignisse erzeugt werden:

- *Ausdruck* - Ein OK-Ereignis wird für einen Auslöser im Problemzustand erzeugt, wenn sein Ausdruck zu FALSE ausgewertet wird. Dies ist die einfachste Einstellung und standardmäßig aktiviert.
- *Wiederherstellungsausdruck* - Ein OK-Ereignis wird für einen Auslöser im Problemzustand erzeugt, wenn sein Ausdruck zu FALSE ausgewertet wird und der Wiederherstellungsausdruck zu TRUE ausgewertet wird. Dies kann verwendet werden, wenn sich die Kriterien für die Wiederherstellung des Auslösers von den Problemkriterien unterscheiden.
- *Keine* - Ein OK-Ereignis wird niemals erzeugt. Dies kann in Verbindung mit der Erzeugung mehrerer Problemereignisse verwendet werden, um einfach eine Benachrichtigung zu senden, wenn etwas passiert.

Zusätzlich verfügen Auslöser über eine Einstellung „OK-Ereignis schließt“, die steuert, welche Problemereignisse geschlossen werden:

- *Alle Probleme* - Ein OK-Ereignis schließt alle offenen Probleme, die vom Auslöser erstellt wurden
- *Alle Probleme, wenn Tag-Werte übereinstimmen* - Ein OK-Ereignis schließt offene Probleme, die vom Auslöser erstellt wurden und mindestens einen übereinstimmenden Tag-Wert haben. Das Tag wird durch die Auslöser-Einstellung „Tag für Abgleich“ definiert. Wenn keine Problemereignisse zum Schließen vorhanden sind, wird kein OK-Ereignis erzeugt. Dies wird oft als Ereigniskorrelation auf Auslöser-Ebene bezeichnet.

Ereigniskorrelation

Die Ereigniskorrelation (auch globale Ereigniskorrelation genannt) ist eine Möglichkeit, benutzerdefinierte Regeln zum Schließen von Ereignissen (was zur Generierung von OK-Ereignissen führt) einzurichten.

Die Regeln definieren, wie neue Problemereignisse mit bestehenden Problemereignissen abgeglichen werden, und ermöglichen es, das neue Ereignis oder die übereinstimmenden Ereignisse durch Generierung entsprechender OK-Ereignisse zu schließen.

Die Ereigniskorrelation muss jedoch sehr sorgfältig konfiguriert werden, da sie die Leistung der Ereignisverarbeitung negativ beeinflussen oder bei Fehlkonfiguration mehr Ereignisse als beabsichtigt schließen kann (im schlimmsten Fall könnten sogar alle Problemereignisse geschlossen werden). Einige Konfigurationstipps:

1. den Korrelationsumfang immer reduzieren, indem Sie ein eindeutiges Tag für das Steuerereignis festlegen (das Ereignis, das mit alten Ereignissen abgeglichen wird), und die Korrelationsbedingung „Tag des neuen Ereignisses“ verwenden
2. vergessen Sie nicht, beim Verwenden der Operation „Altes Ereignis schließen“ eine auf dem alten Ereignis basierende Bedingung hinzuzufügen, da sonst alle bestehenden Probleme geschlossen werden könnten
3. vermeiden Sie die Verwendung allgemeiner Tag-Namen, die von verschiedenen Korrelationskonfigurationen verwendet werden

Task-Manager

Wenn die Einstellung „Allow manual close“ für einen Auslöser aktiviert ist, können durch den Auslöser erzeugte Problemereignisse manuell geschlossen werden. Dies erfolgt im Frontend beim Aktualisieren eines Problems. Das Ereignis wird nicht direkt geschlossen – stattdessen wird eine Aufgabe vom Typ „close event“ erstellt, die kurz darauf vom Task-Manager verarbeitet wird. Der Task-Manager erzeugt ein entsprechendes OK-Ereignis, und das Problemereignis wird geschlossen.

2 Andere Ereignisquellen

Service-Ereignisse

Service-Ereignisse werden nur erzeugt, wenn Service-Aktionen für diese Ereignisse aktiviert sind. In diesem Fall erstellt jede Änderung des Service-Status ein neues Ereignis:

- Problemereignis – wenn der Service-Status von `OK` auf `Problem` geändert wird
- OK-Ereignis – wenn der Service-Status von `Problem` auf `OK` geändert wird

Das Ereignis enthält Details zur Änderung des Service-Status – wann sie stattgefunden hat und wie der neue Status lautet.

Discovery-Ereignisse

Zabbix scannt regelmäßig die in den Regeln für die Netzwerkdiscovery definierten IP-Bereiche. Die Häufigkeit der Prüfung kann für jede Regel einzeln konfiguriert werden. Sobald ein Host oder ein Dienst entdeckt wird, wird ein Discovery-Ereignis (oder mehrere Ereignisse) erzeugt.

Zabbix erzeugt die folgenden Ereignisse:

Ereignis	Wann erzeugt
Dienst aktiv	Jedes Mal, wenn Zabbix einen aktiven Dienst erkennt.
Dienst inaktiv	Jedes Mal, wenn Zabbix einen Dienst nicht erkennen kann.
Host aktiv	Wenn mindestens einer der Dienste für die IP UP ist.
Host inaktiv	Wenn alle Dienste nicht antworten.
Dienst entdeckt	Wenn der Dienst nach einer Ausfallzeit wieder verfügbar ist oder zum ersten Mal entdeckt wird.
Dienst verloren	Wenn der Dienst verloren geht, nachdem er aktiv war.
Host entdeckt	Wenn der Host nach einer Ausfallzeit wieder verfügbar ist oder zum ersten Mal entdeckt wird.
Host verloren	Wenn der Host verloren geht, nachdem er aktiv war.

Ereignisse der aktiven Agent-Autoregistrierung

Die aktive Agent-Autoregistrierung erstellt Ereignisse in Zabbix.

Falls konfiguriert, wird ein Ereignis der aktiven Agent-Autoregistrierung erstellt, wenn ein zuvor unbekannter aktiver Agent Prüfungen anfordert oder wenn sich die Host-Metadaten geändert haben. Der Server fügt einen neuen automatisch registrierten Host hinzu und verwendet dabei die empfangene IP-Adresse und den Port des Agent.

Weitere Informationen finden Sie auf der Seite Autoregistrierung aktiver Agenten.

Interne Ereignisse

Interne Ereignisse treten auf, wenn:

- ein Datenpunkt seinen Status von „normal“ auf „unsupported“ ändert
- ein Datenpunkt seinen Status von „unsupported“ auf „normal“ ändert
- eine Low-Level-Discovery-Regel ihren Status von „normal“ auf „unsupported“ ändert
- eine Low-Level-Discovery-Regel ihren Status von „unsupported“ auf „normal“ ändert
- ein Auslöser seinen Status von „normal“ auf „unknown“ ändert
- ein Auslöser seinen Status von „unknown“ auf „normal“ ändert

Ziel der Einführung interner Ereignisse ist es, Benutzern Benachrichtigungen zu ermöglichen, wenn ein internes Ereignis eintritt, zum Beispiel wenn ein Datenpunkt auf „unsupported“ wechselt und keine Daten mehr erfasst.

Interne Ereignisse werden nur erstellt, wenn interne Aktionen für diese Ereignisse aktiviert sind. Um die Erzeugung interner Ereignisse zu stoppen (zum Beispiel für Datenpunkte, die auf „unsupported“ wechseln), deaktivieren Sie alle Aktionen für interne Ereignisse unter Warnungen → Aktionen → Interne Aktionen.

Note:
Wenn interne Aktionen deaktiviert sind und sich ein Objekt im Status „unsupported“ befindet, wird dennoch ein Wiederherstellungsereignis für dieses Objekt erstellt.

Wenn interne Aktionen aktiviert sind und sich ein Objekt im Status „unsupported“ befindet, wird ein Wiederherstellungsereignis für dieses Objekt erstellt, auch wenn für das Objekt kein „Problemereignis“ erstellt wurde.

Siehe auch: Benachrichtigung bei nicht unterstützten Datenpunkten erhalten

3 Manuelles Schließen von Problemen

Übersicht

Während Problemereignisse im Allgemeinen automatisch behoben werden, wenn der Auslöserstatus von `Problem` auf `OK` wechselt, kann es Fälle geben, in denen sich nur schwer feststellen lässt, ob ein Problem mithilfe eines Auslöserausdrucks behoben wurde. In solchen Fällen muss das Problem manuell behoben werden.

Zum Beispiel kann *syslog* melden, dass einige Kernel-Parameter für eine optimale Leistung angepasst werden müssen. In diesem Fall wird das Problem an Linux-Administratoren gemeldet, diese beheben es und schließen das Problem dann manuell.

Probleme können nur bei Auslösern manuell geschlossen werden, für die 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 an Ereignis-Tags, die für dieses Ereignis aufgelöst wurden (einschließlich Tags, die von Vorlagen, Hosts und Auslösern geerbt wurden). 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 nie wieder in den Status *Problem* wechseln wird. Der Auslöserausdruck wird erneut ausgewertet und kann wieder zu einem Problem führen:

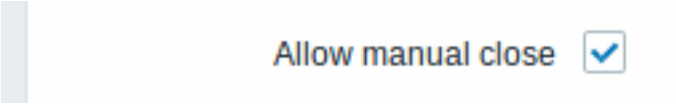
- Wenn neue Daten für einen im Auslöserausdruck enthaltenen Datenpunkt eintreffen (beachten Sie, dass Werte, die durch einen Drosselungs-Preprocessing-Schritt verworfen werden, nicht als empfangen gelten und nicht dazu führen, dass der Auslöserausdruck erneut ausgewertet wird);
- Wenn die Funktionen date and time und/oder nodata() im Ausdruck verwendet werden.

Konfiguration

Zum manuellen Schließen eines Problems sind zwei Schritte erforderlich.

Auslöser-Konfiguration

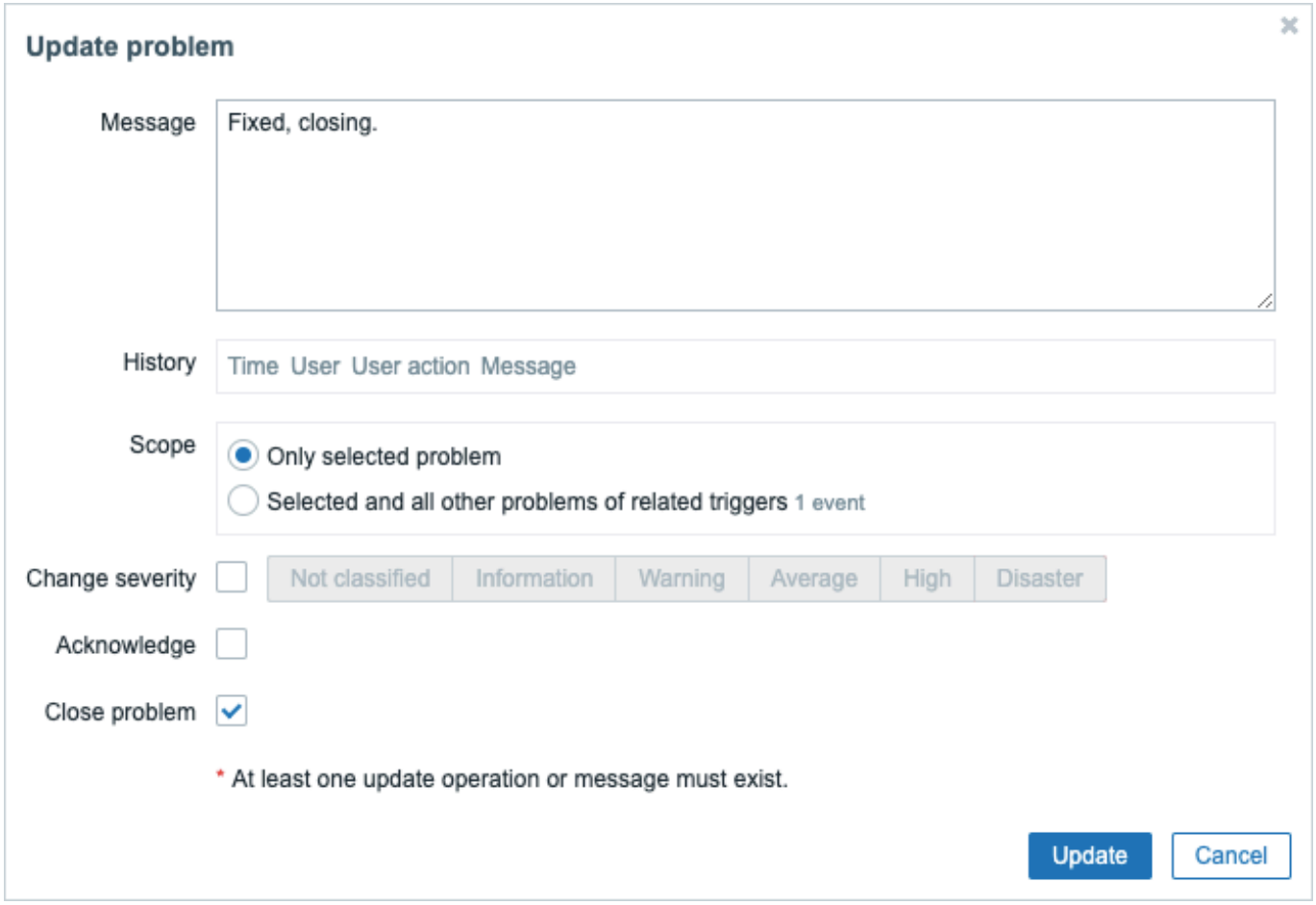
Aktivieren Sie in der Auslöser-Konfiguration die Option *Manuelles Schließen erlauben*.



Fenster „Problem aktualisieren“

Wenn für einen Auslöser mit dem Kennzeichen *Manuelles Schließen* ein Problem auftritt, können Sie das Popup-Fenster Problem aktualisieren 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 erforderlichen Eingabefelder 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 unter *Monitoring* > *Probleme WIRD GESCHLOSSEN* als Status des Problems angezeigt.

Verifizierung

Es kann verifiziert werden, dass ein Problem manuell geschlossen wurde:

- in den Ereignisdetails, verfügbar über *Monitoring* > *Problems*;
- durch die Verwendung des Makros `{EVENT.UPDATE.HISTORY}` in Benachrichtigungsnachrichten, das diese Information bereitstellt.

5 Ereigniskorrelation

Übersicht

Die Ereigniskorrelation ermöglicht es, Problemereignisse auf eine sehr präzise und flexible Weise mit ihrer Auflösung zu korrelieren.

Die Ereigniskorrelation kann definiert werden:

- auf Auslöser-Ebene - ein Auslöser kann verwendet werden, um separate Probleme mit ihrer Lösung in Beziehung zu setzen
- global - Probleme können mithilfe globaler Korrelationsregeln mit ihrer Lösung aus einem anderen Auslöser-/Polling-Verfahren korreliert werden

1 Auslöser-basierte Ereigniskorrelation

Übersicht

Die Auslöser-basierte Ereigniskorrelation ermöglicht es, separate Probleme zu korrelieren, die von einem Auslöser gemeldet werden.

Während ein OK-Ereignis im Allgemeinen alle Problemereignisse schließen kann, die von einem Auslöser erstellt wurden, gibt es Fälle, in denen ein detaillierterer Ansatz erforderlich ist. Zum Beispiel möchten Sie bei der Überwachung von Protokolldateien möglicherweise bestimmte Probleme in einer Protokolldatei erkennen und diese 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 für die Protokollüberwachung, Trap-Verarbeitung usw. verwendet.

In Zabbix ist es möglich, Problemereignisse auf Grundlage von Tags in Beziehung zu setzen. Tags werden verwendet, um Werte zu extrahieren und eine Identifikation für Problemereignisse zu erstellen. Dadurch können Probleme auch einzeln auf Grundlage eines übereinstimmenden Tags geschlossen werden.

Mit anderen Worten: Derselbe Auslöser kann separate Ereignisse erzeugen, die durch das Ereignis-Tag identifiziert werden. Daher können Problemereignisse einzeln identifiziert und separat auf Grundlage der Identifikation durch das Ereignis-Tag geschlossen werden.

Funktionsweise

Bei der Log-Überwachung können Sie auf Zeilen wie diese stoßen:

```
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 abzugleichen 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
```

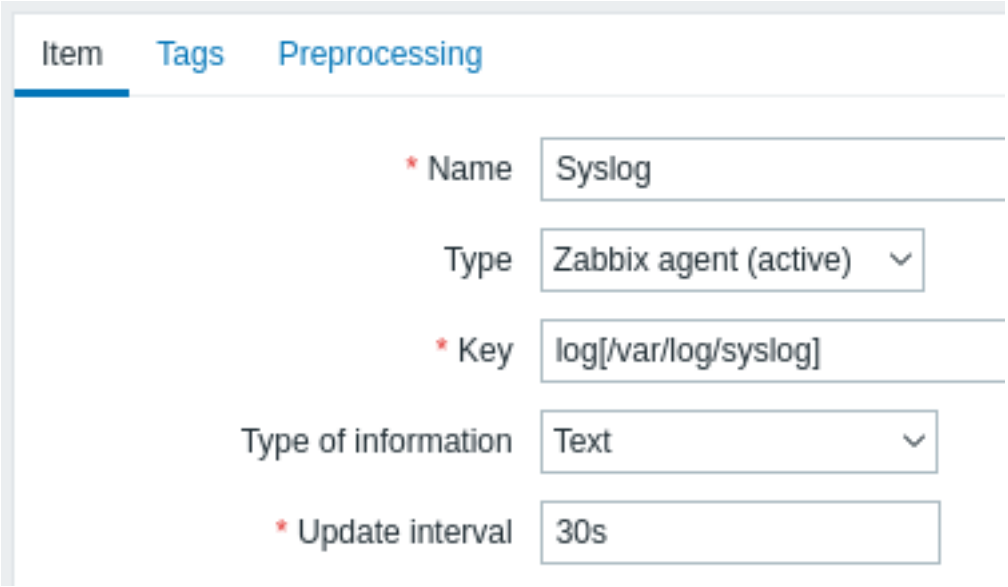
Dazu müssen Sie diese zusammengehörigen Ereignisse zum Beispiel mit „Dienst 1“ und „Dienst 2“ taggen. Das kann durch Anwenden eines regulären Ausdrucks auf die Log-Zeile erfolgen, um den Tag-Wert zu extrahieren. Wenn dann Ereignisse erstellt werden, werden sie entsprechend mit „Dienst 1“ und „Dienst 2“ getaggt, und das Problem kann mit der Behebung abgeglichen werden.

Konfiguration

Datenpunkt

Zunächst sollten Sie einen Datenpunkt einrichten, der eine Protokolldatei überwacht, zum Beispiel:

```
log[/var/log/syslog]
```



Nachdem der Datenpunkt eingerichtet wurde, warten Sie eine Minute, bis die Konfigurationsänderungen übernommen wurden, und gehen Sie dann zu Letzte Daten, um sicherzustellen, dass der Datenpunkt begonnen hat, Daten zu sammeln.

Auslöser

Wenn der Datenpunkt funktioniert, müssen Sie den Auslöser konfigurieren. Es ist wichtig zu entscheiden, welche Einträge in der Protokolldatei beachtet werden sollen. Zum Beispiel sucht der folgende Auslöserausdruck nach einer Zeichenfolge wie ‘Stopping’, um auf potenzielle Probleme hinzuweisen:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Um sicherzustellen, dass jede Zeile mit der Zeichenfolge ”Stopping” als Problem betrachtet wird, setzen Sie auch den *Modus zur Problemerzeugung von Ereignissen* in der Auslöserkonfiguration auf ’Multiple’.

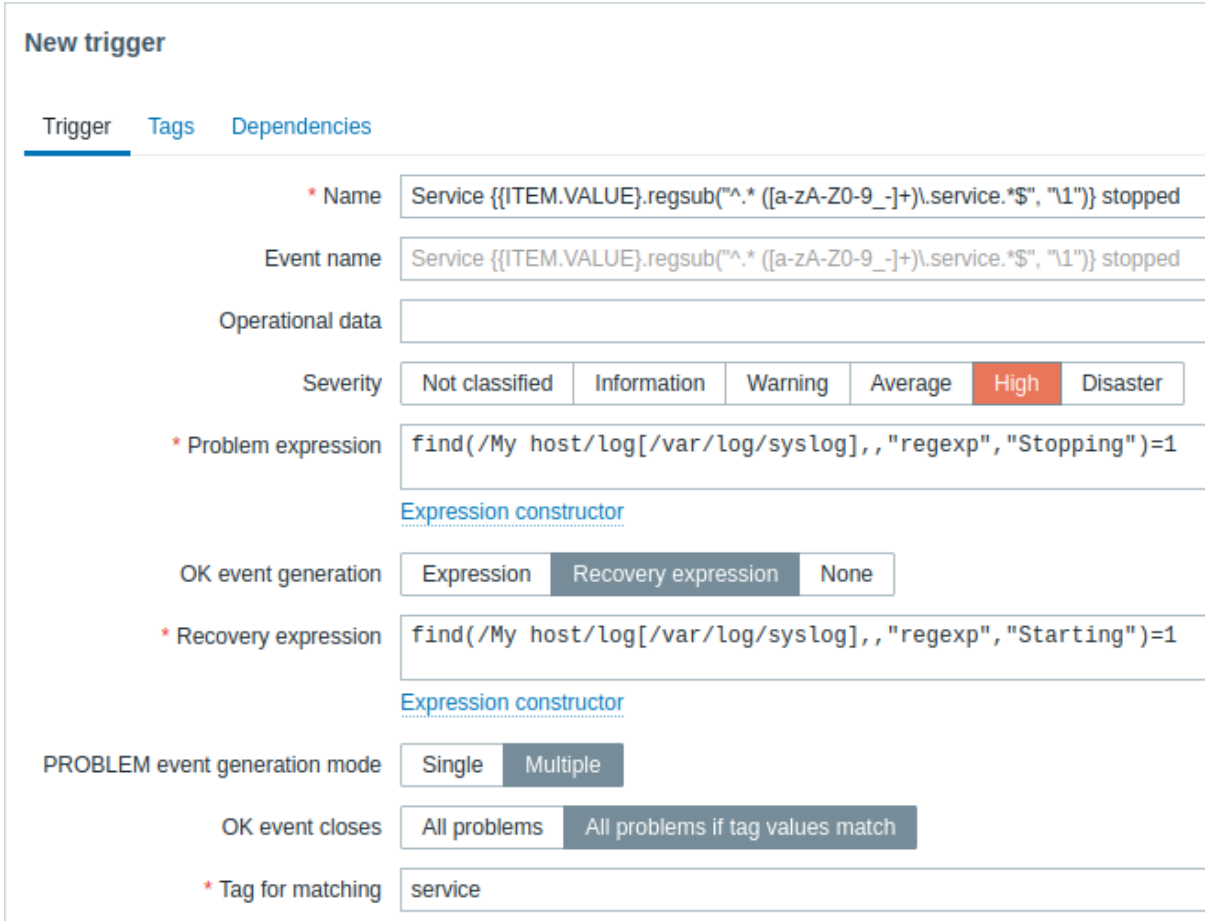
Definieren Sie dann einen Wiederherstellungsausdruck. Der folgende Wiederherstellungsausdruck löst alle Probleme, wenn eine Protokollzeile gefunden wird, die die Zeichenfolge ”Starting” enthält:

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

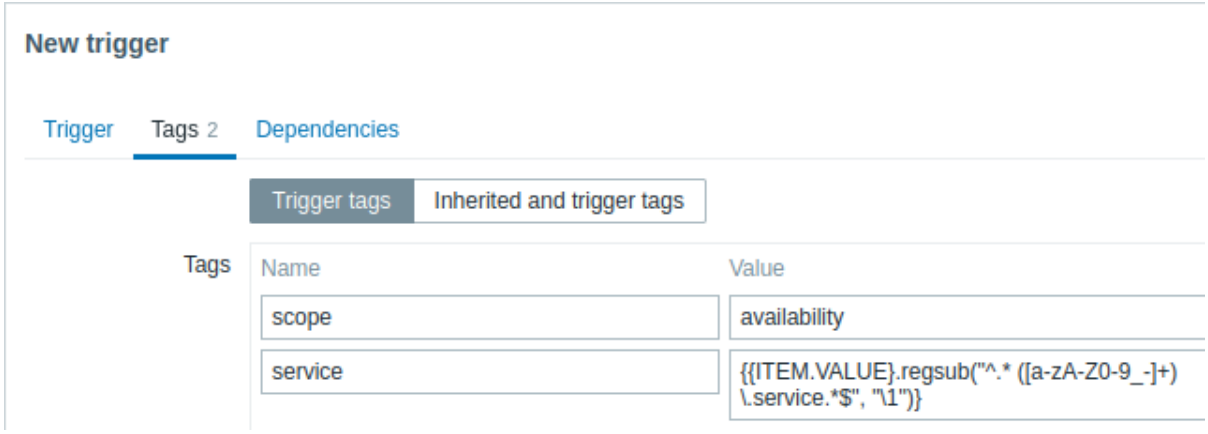
Da wir das nicht möchten, ist es wichtig, irgendwie sicherzustellen, dass die entsprechenden ursprünglichen Probleme geschlossen werden und nicht einfach alle Probleme. Hier kann die Verschlagwortung helfen.

Probleme und Lösungen können durch Angabe eines Tags in der Auslöserkonfiguration abgeglichen werden. Die folgenden Einstellungen müssen vorgenommen werden:

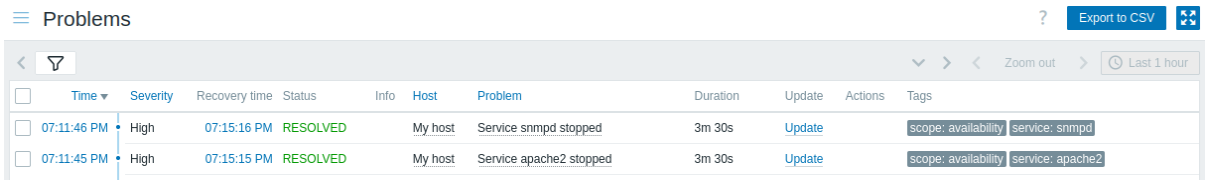
- *Modus zur Problemerzeugung von Ereignissen*: Multiple
- *OK-Ereignis schließt*: Alle Probleme, wenn Tag-Werte übereinstimmen
- Geben Sie den Namen des Tags für den Ereignisabgleich ein



- konfigurieren Sie die Tags, um Tag-Werte aus Protokollzeilen zu extrahieren



Bei erfolgreicher Konfiguration können Sie in *Monitoring → Probleme* Problemereignisse sehen, die nach Anwendung getaggt und ihrer Lösung zugeordnet sind.



Warning:
Da Fehlkonfigurationen möglich sind, wenn ähnliche Ereignis-Tags für **nicht zusammenhängende** Probleme erzeugt werden können, prüfen Sie bitte die unten beschriebenen Fälle!

- Indizierte Makros beziehen sich immer auf das Feld *Ausdruck* der Auslöserkonfiguration, nicht auf den *Wiederherstellungsausdruck*. Zum Beispiel wird **{ITEM.VALUE1}** in einem Wiederherstellungsereignis zum Zeitpunkt der Wiederherstellung zum neuesten Wert des ersten Datenpunkts **im Problemausdruck** aufgelöst. Wenn der Wiederherstellungsausdruck auf einem anderen Datenpunkt basiert und sich der Wert des Datenpunkts aus dem Problemausdruck bis zur Wiederherstellung ändert, erhalten die Ereignisse unterschiedliche Tags und werden nicht korreliert.

- Wenn zwei Anwendungen Fehler- und Wiederherstellungsmeldungen in dieselbe Protokolldatei schreiben, kann ein Benutzer entscheiden, zwei *service*-Tags im selben Auslöser mit unterschiedlichen Tag-Werten zu verwenden, indem separate reguläre Ausdrücke in den Tag-Werten verwendet werden, um die Namen von beispielsweise Service A und Service B aus dem Makro {ITEM.VALUE} zu extrahieren (z. B. wenn sich die Nachrichtenformate unterscheiden). Dies funktioniert jedoch möglicherweise nicht wie geplant, wenn es keine Übereinstimmung mit den regulären Ausdrücken gibt. Nicht übereinstimmende reguläre Ausdrücke ergeben leere Tag-Werte, und ein einzelner leerer Tag-Wert sowohl in Problem- als auch in OK-Ereignissen reicht aus, um sie zu korrelieren. So kann eine Wiederherstellungsmeldung von Service A versehentlich eine Fehlermeldung von Service B schließen.
- Tatsächliche Tags und Tag-Werte werden erst sichtbar, wenn ein Auslöser ausgelöst wird. Wenn der verwendete reguläre Ausdruck ungültig ist, wird er stillschweigend durch die Zeichenfolge *UNKNOWN* ersetzt. Wenn das ursprüngliche Problemereignis mit einem *UNKNOWN*-Tag-Wert übersehen wird, können nachfolgende OK-Ereignisse mit demselben *UNKNOWN*-Tag-Wert erscheinen, die Problemereignisse schließen können, die sie nicht hätten schließen dürfen.
- 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 unspezifisch sind, kann dies ebenfalls zu ähnlichen Ereignis-Tags für nicht zusammenhängende Probleme führen.

2 Globale Ereigniskorrelation

Übersicht

Die globale Ereigniskorrelation ermöglicht es, über alle von Zabbix überwachten Metriken hinweg Korrelationen zu erstellen.

Es ist möglich, Ereignisse zu korrelieren, die von völlig unterschiedlichen Auslösern erstellt wurden, und auf sie alle dieselben Operationen anzuwenden. Durch das Erstellen intelligenter Korrelationsregeln können Sie sich tatsächlich Tausende sich wiederholender Benachrichtigungen ersparen und sich auf die eigentlichen Ursachen eines Problems konzentrieren!

Die globale Ereigniskorrelation ist ein leistungsfähiger Mechanismus, der es Ihnen ermöglicht, sich von einer problem- und lösungslogik zu lösen, die auf einem einzelnen Auslöser basiert. Bisher wurde ein einzelnes Problemereignis von einem Auslöser erstellt, und wir waren für die Problemlösung von genau diesem Auslöser abhängig. Wir konnten ein von einem Auslöser erstelltes Problem nicht mit einem anderen Auslöser lösen. Mit einer auf Ereignis-Tags basierenden Ereigniskorrelation ist dies jedoch möglich.

Beispielsweise kann ein Log-Auslöser Anwendungsprobleme melden, während ein Polling-Auslöser meldet, dass die Anwendung betriebsbereit ist. Unter Nutzung von Ereignis-Tags können Sie den Log-Auslöser mit *status:down* und den Polling-Auslöser mit *status:up* kennzeichnen. Anschließend können Sie diese Auslöser in einer globalen Korrelationsregel miteinander verknüpfen und dieser Korrelation eine passende Operation zuweisen, etwa das Schließen alter Ereignisse.

In einem anderen Anwendungsfall kann die globale Korrelation ähnliche Auslöser identifizieren und dieselbe Operation auf sie anwenden. Was wäre, wenn wir nur einen einzigen Problembericht pro Netzwerkportproblem erhalten könnten? Es ist nicht nötig, sie 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 bestehenden Problemereignissen abgeglichen werden und was im Falle einer Übereinstimmung zu tun ist (das neue Ereignis schließen, übereinstimmende alte Ereignisse durch Erzeugen entsprechender OK-Ereignisse schließen). Wenn ein Problem durch globale Korrelation geschlossen wird, wird dies in der Spalte *Info* unter *Monitoring > Problems* angezeigt.

Die Konfiguration globaler Korrelationsregeln ist nur für Benutzer mit der Berechtigungsstufe Super Admin verfügbar.

Attention:
Die Ereigniskorrelation muss sehr sorgfältig konfiguriert werden, da sie die Leistung der Ereignisverarbeitung negativ beeinflussen oder bei Fehlkonfiguration mehr Ereignisse schließen kann als beabsichtigt (im schlimmsten Fall könnten sogar alle Problemereignisse geschlossen werden).

Um die globale Korrelation **sicher** zu konfigurieren, beachten Sie die folgenden wichtigen Hinweise:

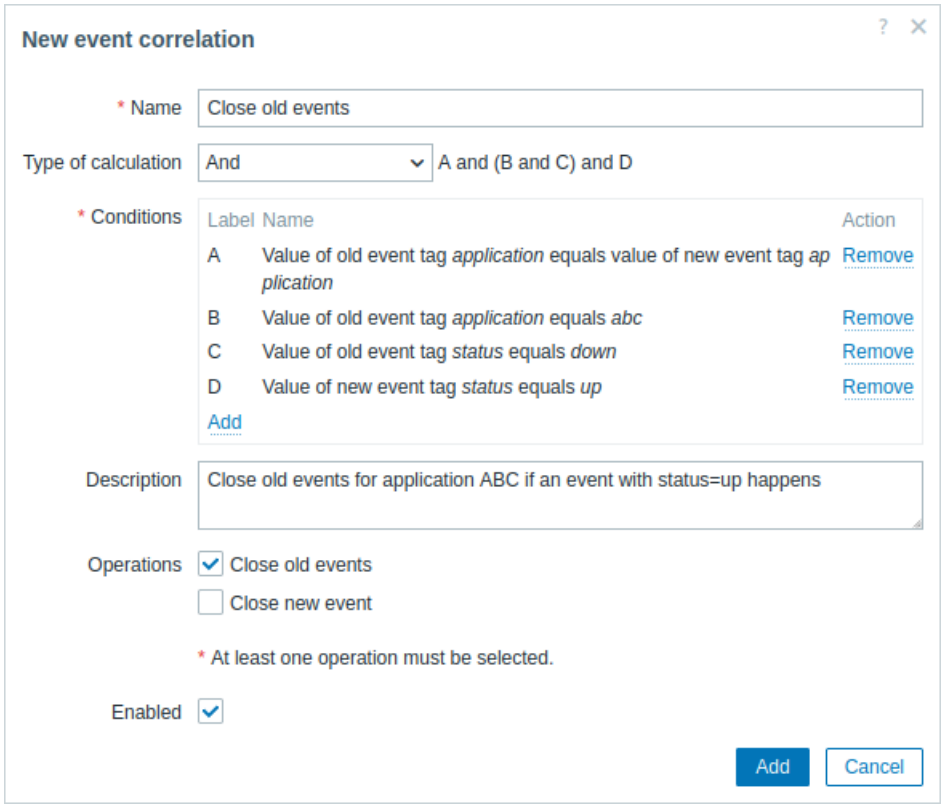
- Reduzieren Sie den Korrelationsumfang. Setzen Sie immer ein eindeutiges Tag für das neue Ereignis, das mit alten Ereignissen abgeglichen wird, und verwenden Sie die Korrelationsbedingung *New event tag name*.
- Fügen Sie bei Verwendung von **Close old events** eine explizite Bedingung für alte Ereignisse hinzu. Fügen Sie immer mindestens eine Bedingung für *Old event* hinzu (zum Beispiel *Old event tag name*, *Old event tag value* oder *Event tag pair*), wenn Sie **Close old events** auswählen — andernfalls kann die Regel übereinstimmen und nicht zusammenhängende bestehende Problemereignisse schließen (im schlimmsten Fall alle Probleme). Bevorzugen Sie **Event tag pair** für den Abgleich von Laufzeitwerten (host:port, session id usw.) und grenzen Sie die Übereinstimmung nach Möglichkeit zusätzlich nach Host oder Hostgruppe ein.
- Vermeiden Sie die Verwendung allgemeiner Tag-Namen, die möglicherweise von verschiedenen Korrelationskonfigurationen verwendet werden.
- Halten Sie die Anzahl der Korrelationsregeln auf die wirklich benötigten Regeln beschränkt.

Siehe auch: bekannte Probleme.

Konfiguration

So konfigurieren Sie Ereigniskorrelationsregeln global:

- Gehen Sie zu *Datenerfassung* > *Ereigniskorrelation*
- Klicken Sie rechts auf *Ereigniskorrelation erstellen* (oder auf den Korrelationsnamen, um eine vorhandene Regel zu bearbeiten)
- Geben Sie die Parameter der Korrelationsregel im Formular ein

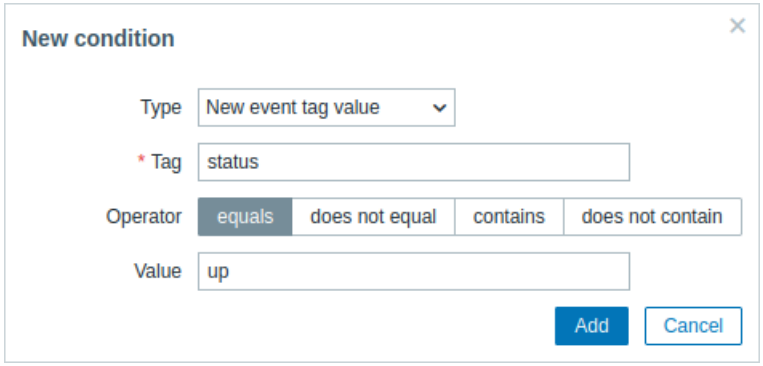


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Eindeutiger Name der Korrelationsregel.

Parameter	Beschreibung
Berechnungstyp	Die folgenden Optionen zur Berechnung von Bedingungen sind verfügbar: **Und** - alle Bedingungen müssen erfüllt sein **Oder** - es genügt, wenn eine Bedingung erfüllt ist **Und/Oder** - UND mit verschiedenen Bedingungstypen und ODER mit demselben Bedingungstyp **Benutzerdefinierter Ausdruck** - eine benutzerdefinierte Berechnungsformel zur Auswertung von Aktionsbedingungen. Sie muss alle Bedingungen enthalten (dargestellt als Großbuchstaben A, B, C, ...) und kann Leerzeichen, Tabulatoren, Klammern ( ), **and** (Groß-/Kleinschreibung beachten), **or** (Groß-/Kleinschreibung beachten), **not** (Groß-/Kleinschreibung beachten) enthalten.
Bedingungen	Liste der Bedingungen. Details zur Konfiguration einer Bedingung finden Sie unten.
Beschreibung	Beschreibung der Korrelationsregel.
Operationen	Aktivieren Sie das Kontrollkästchen der Operation, die ausgeführt werden soll, wenn ein Ereignis korreliert wird. Die folgenden Operationen sind verfügbar: **Alte Ereignisse schließen** - alte Ereignisse schließen, wenn ein neues Ereignis eintritt. Fügen Sie bei Verwendung der Operation *Alte Ereignisse schließen* immer eine Bedingung hinzu, die auf dem alten Ereignis basiert, da sonst alle vorhandenen Problemen geschlossen werden könnten. **Neues Ereignis schließen** - das neue Ereignis schließen, wenn es eintritt. **Warnung!** Lassen Sie die Bedingungen für alte/neue Ereignisse nicht leer, wenn Sie *Alte Ereignisse schließen/Neues Ereignis schließen* verwenden.Wenn Sie die Operation *Alte Ereignisse schließen* auswählen, ohne eine Bedingung hinzuzufügen, die dem alten Ereignis entspricht, kann Zabbix alle vorhandenen alten Ereignisse abgleichen und schließen.Fügen Sie bei Verwendung von *Alte Ereignisse schließen* immer eine explizite Bedingung für alte Ereignisse hinzu (zum Beispiel *Name des Tags des alten Ereignisses* oder *Ereignis-Tag-Paar*). Beispielsweise wird eine Regel, die nur eine *Bedingung für neues Ereignis* und die *Operation Alte Ereignisse schließen* verwendet, mit allen alten Ereignissen übereinstimmen, die die (fehlenden) Kriterien für alte Ereignisse erfüllen - wodurch alte Probleme effektiv geschlossen werden.
Aktiviert	Wenn Sie dieses Kontrollkästchen aktivieren, wird die Korrelationsregel aktiviert.

Um die Details einer neuen Bedingung zu konfigurieren, klicken Sie im Block Bedingungen auf Add. Es öffnet sich ein Popup-Fenster, in dem Sie die Bedingungsdetails bearbeiten können.



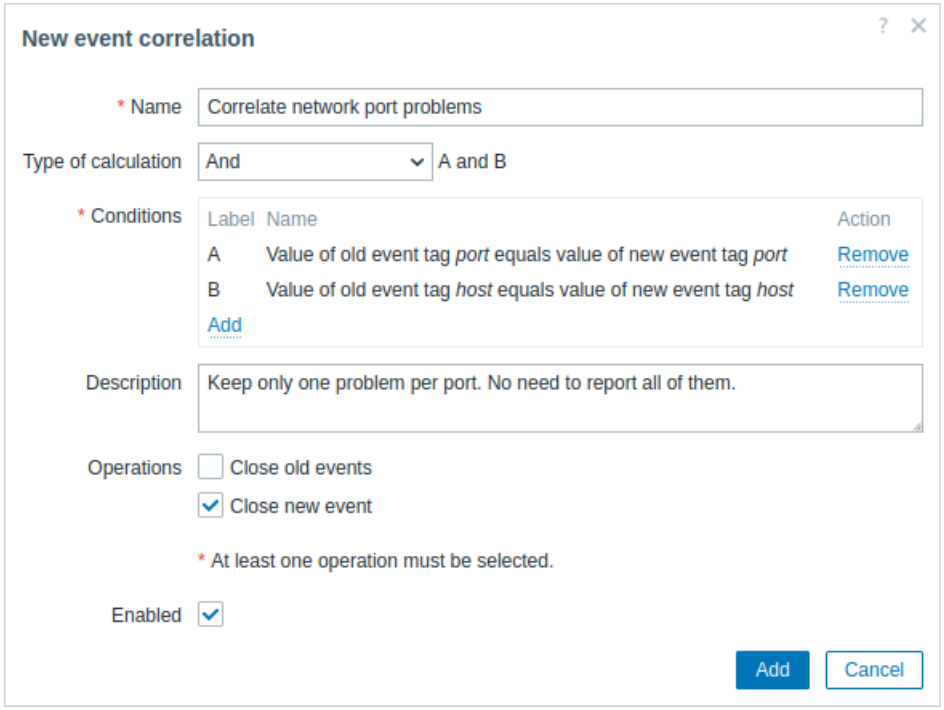
Parameter	Beschreibung
Neue Bedingung	Wählen Sie eine Bedingung für die Korrelation von Ereignissen aus. *Beachten Sie*, dass, wenn keine Bedingung für alte Ereignisse angegeben ist, alle alten Ereignisse abgeglichen und geschlossen werden können. Ebenso können, wenn keine Bedingung für neue Ereignisse angegeben ist, alle neuen Ereignisse abgeglichen und geschlossen werden. Die folgenden Bedingungen sind verfügbar: **Name des Tags des alten Ereignisses** - geben Sie den Namen des Tags des alten Ereignisses für den Abgleich an. **Name des Tags des neuen Ereignisses** - geben Sie den Namen des Tags des neuen Ereignisses für den Abgleich an. **Host-Gruppe des neuen Ereignisses** - geben Sie die Host-Gruppe des neuen Ereignisses für den Abgleich an. **Ereignis-Tag-Paar** - geben Sie den Tag-Namen des neuen Ereignisses und den Tag-Namen des alten Ereignisses für den Abgleich 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, um Laufzeitwerte abzugleichen, die zum Zeitpunkt der Konfiguration möglicherweise nicht bekannt sind (siehe auch Beispiel). **Wert des Tags des alten Ereignisses** - geben Sie den Namen und den Wert des Tags des alten Ereignisses für den Abgleich an, unter Verwendung der folgenden Operatoren: *gleich* - hat den Wert des Tags des alten Ereignisses *ungleich* - hat nicht den Wert des Tags des alten Ereignisses *enthält* - enthält die Zeichenfolge im Wert des Tags des alten Ereignisses *enthält nicht* - enthält die Zeichenfolge nicht im Wert des Tags des alten Ereignisses **Wert des Tags des neuen Ereignisses** - geben Sie den Namen und den Wert des Tags des neuen Ereignisses für den Abgleich an, unter Verwendung der folgenden Operatoren: *gleich* - hat den Wert des Tags des neuen Ereignisses *ungleich* - hat nicht den Wert des Tags des neuen Ereignisses *enthält* - enthält die Zeichenfolge im Wert des Tags des neuen Ereignisses *enthält nicht* - enthält die Zeichenfolge nicht im Wert des Tags des neuen Ereignisses

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 ausgelöst wird. Wenn der verwendete reguläre Ausdruck ungültig ist, wird er stillschweigend durch eine *UNKNOWN*-Zeichenfolge ersetzt. Wenn das ursprüngliche Problemereignis mit einem *UNKNOWN*-Tag-Wert übersehen wird, können nachfolgende OK-Ereignisse mit demselben *UNKNOWN*-Tag-Wert erscheinen, die Problemereignisse schließen können, 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

Wiederholte Problemereignisse vom selben Netzwerkport stoppen.



Diese globale Korrelationsregel korreliert Probleme, wenn die Tag-Werte *Host* und *port* im Auslöser vorhanden sind und sie im ursprünglichen Ereignis und im neuen Ereignis identisch sind.

Die Operation schließt neue Problemereignisse am selben Netzwerkport, sodass nur das ursprüngliche Problem offen bleibt.

6 Tagging

Übersicht

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
- Webszenarien
- Auslöser
- Services
- Vorlagen-Datenpunkte und Auslöser
- Host-, Datenpunkt- und Auslöser-Prototypen

In den Listen *Vorlagen*, *Hosts*, *Datenpunkte*, *Auslöser*, *Webszenarien* und deren Prototypen zeigt die Spalte *Tags* sowohl reguläre als auch geerbte Tags an. Wenn Sie mit der Maus über ein *geerbtes* Tag fahren oder darauf klicken, erscheint ein JavaScript-Tooltip mit dem Text „Inherited tag“. Wenn ein geerbtes Tag auch als reguläres Tag vorhanden ist, wird je nach Liste, in der es erscheint, 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 im Frontend werden reguläre und geerbte Tags zusammen angezeigt (Duplikate werden entfernt), ohne Symbole und ohne zusätzlichen Tooltip-Text.

Note:
Beachten Sie die offiziellen Zabbix-Richtlinien mit allgemeinen Empfehlungen zur Definition von Tags sowie spezifischen Hinweisen für Vorlagen, Datenpunkte, Auslöser und Low-Level-Discovery-Regeln.

Tags haben mehrere Zwecke, insbesondere die Kennzeichnung von Ereignissen. Wenn Entitäten mit Tags versehen sind, erbt jedes neue Ereignis, das mit einer getaggten Entität zusammenhängt, deren Tags. Zum Beispiel:

- bei getaggten Vorlagen erbt jedes Host-Problem (erstellt durch Auslöser aus der Vorlage) die Tags der Vorlage.
- bei getaggten Hosts erbt jedes Host-Problem die Tags des Hosts.
- bei getaggten Datenpunkten/Webszenarien erbt jedes Datenpunkt-/Webszenario-Problem die Tags des Datenpunkts/Webszenarios.
- bei getaggten Auslösern erbt jedes durch den Auslöser erzeugte Problem die Tags des Auslösers.

Ein Problemereignis erbt alle Tags aus der gesamten Kette von Entitäten – Vorlagen, Hosts, Datenpunkte/Webszenarien, Auslöser. Identische `tag:value`-Kombinationen (nach Auflösung von Makros) werden zu einer einzigen zusammengeführt, wodurch Duplikate vermieden werden.

Wiederherstellungsereignisse, die durch ein manuelles Schließen erzeugt werden, enthalten ebenfalls die aufgelösten Ereignis-Tags, die von Vorlagen, Hosts, Datenpunkten/Webszenarien 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:

- die Ereigniskorrelation kann auf Basis von Ereignis-Tags konfiguriert werden.
- Aktionsbedingungen können auf Basis von Ereignis-Tags konfiguriert werden.
- Datenpunkt-Probleme 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 ein Tag ohne Wert und dasselbe Tag mit einem Wert haben (zum Beispiel `database` und `database:postgresql`). Solche Tags gelten nicht als Duplikate.

Anwendungsfälle

Einige gängige Anwendungsfälle für das Tagging sind wie folgt:

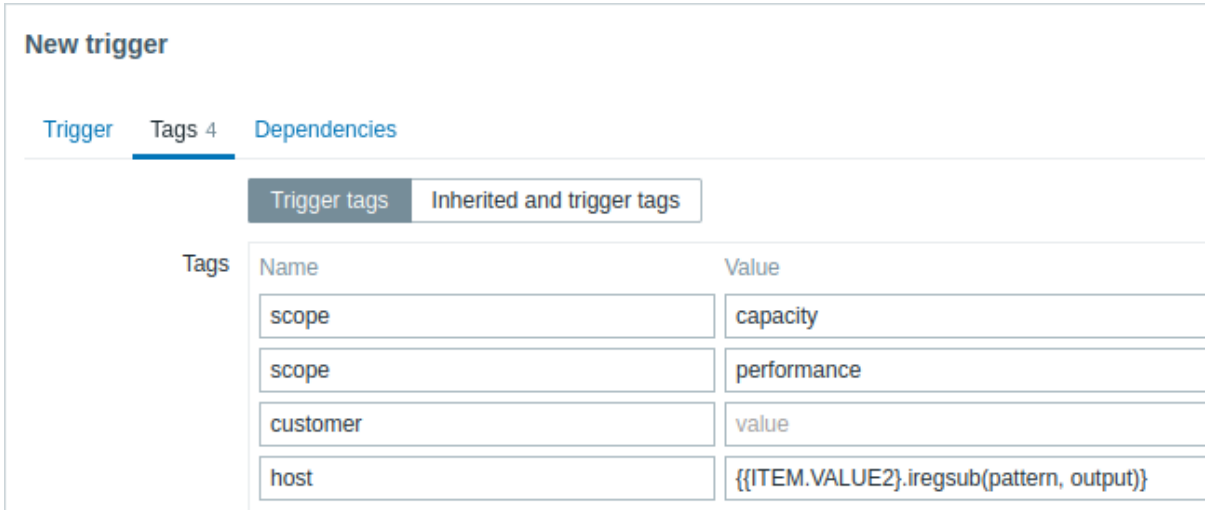
1. Auslöser-Ereignisse kennzeichnen:
   - Definieren Sie ein Auslöser-Tag (zum Beispiel `scope:performance`).
   - Durch diesen Auslöser erzeugte Probleme erhalten das Auslöser-Tag.
2. Von Vorlagen geerbte Probleme kennzeichnen:
   - Definieren Sie ein Vorlagen-Tag (zum Beispiel `target:mysql`).
   - Durch Auslöser aus dieser Vorlage erzeugte Probleme erhalten das Vorlagen-Tag.
3. Host-Probleme kennzeichnen:
   - Definieren Sie ein Host-Tag (zum Beispiel `service:jira`).
   - Durch Auslöser von diesem Host erzeugte Probleme erhalten das Host-Tag.
4. Zugehörige Datenpunkte filtern:
   - Definieren Sie ein Datenpunkt-Tag (zum Beispiel `component:cpu`).
   - Unter *Monitoring* > *Letzte Daten* können Datenpunkte nach dem Tag `component:cpu` gefiltert werden.
5. Aus dem Datenpunktwert extrahierte Informationen als Tag-Wert verwenden:
   - Definieren Sie ein Tag mit einem Makro als Tag-Wert (zum Beispiel `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Unter *Monitoring* > *Probleme* wird bei Problemen der Tag-Wert in die aus dem Datenpunktwert extrahierten Daten aufgelöst.
6. Probleme in einer Protokolldatei identifizieren und separat schließen:
   - Definieren Sie ein Auslöser-Tag für den Auslöser des Log-Monitoring-Datenpunkts, das Werte mithilfe eines Makros aus dem Datenpunktwert extrahiert (zum Beispiel `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Richten Sie in der Auslöser-Konfiguration die Ereigniskorrelation ein:
     - setzen Sie *Modus für die Generierung von PROBLEM-Ereignissen* auf „Mehrere“;
     - setzen Sie *OK-Ereignis schließt* auf „Alle Probleme, wenn Tag-Werte übereinstimmen“;
     - legen Sie das Tag für den Abgleich fest.
   - Durch den Auslöser des Log-Datenpunkts erzeugte Probleme erhalten das Auslöser-Tag und werden einzeln geschlossen.
7. Benachrichtigungen filtern:
   - Definieren Sie Auslöser-Tags (zum Beispiel `scope:security` für Auslöser1 und `scope:availability` für Auslöser2).
   - Verwenden Sie die Tag-Filterung in Aktionsbedingungen, um Benachrichtigungen nur für Ereignisse zu erhalten, die den Tag-Daten entsprechen.
8. Probleme in Benachrichtigungen identifizieren:
   - Definieren Sie Auslöser-Tags.
   - Verwenden Sie das Makro {EVENT.TAGS} in der Problembenachrichtigung.
   - Die Problembenachrichtigung enthält die Auslöser-Tags, wodurch sich leichter erkennen lässt, zu welcher Anwendung/welchem Dienst die Benachrichtigung gehört.
9. Konfigurationsaufgaben durch die Verwendung von Vorlagen-Tags vereinfachen:
   - Definieren Sie ein Vorlagen-Auslöser-Tag.
   - Auslöser, die aus diesem Vorlagen-Auslöser erstellt werden, erhalten dessen Tag.
10. Auslöser mit Tags aus der Low-Level-Discovery (LLD) erstellen:
    - Definieren Sie ein Tag für einen Auslöser-Prototyp mit einem LLD-Makro im Tag-Namen oder -Wert (zum Beispiel `scope:{#FSNAME}`).
    - Auslöser, die aus dem Auslöser-Prototyp erstellt werden, erhalten dessen Tag.
11. Services mithilfe von Service-Tags abgleichen:
    - Definieren Sie Service-Tags.

- Konfigurieren Sie Service-Aktionen für Services mit übereinstimmenden Tags.
- Verwenden Sie Service-Tags außerdem, um einen Service mit einem SLA für SLA-Berechnungen zu verknüpfen.

12. Services über Service-Problem-Tags mit Problemen verknüpfen:
    - Definieren Sie ein Problem-Tag in der Service-Konfiguration (zum Beispiel `target:mysql`).
    - Probleme mit einem übereinstimmenden Tag werden automatisch mit dem Service korreliert, und der Service-Status ändert sich entsprechend den konfigurierten Regeln zur Berechnung des Service-Status.
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 Benutzergruppen-Konfiguration.
    - Benutzer in der Benutzergruppe können nur Probleme mit den definierten Tags anzeigen.

Konfiguration

Tags können in einem eigenen Reiter definiert werden, zum Beispiel in der Auslöser-Konfiguration:



Unterstützung von Makros

Integrierte und Benutzermakros in Tags werden zum Zeitpunkt des Ereignisses aufgelöst. Bis das Ereignis eingetreten ist, werden diese Makros im Zabbix Frontend nicht aufgelöst angezeigt.

Low-level discovery-Makros werden während des Discovery-Prozesses aufgelöst.

Die folgenden Makros können in Namen und Werten von Auslöser-Tags verwendet werden:

- integrierte Makros {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.*} (zum Referenzieren von Host-Inventarwerten von einem oder mehreren Hosts in einem Auslöser-Ausdruck)
- Benutzermakros und Benutzermakros mit Kontext (der Kontext kann Low-level discovery-Makros enthalten)
- Low-level discovery-Makros (nur in Tags von Auslöser-Prototypen)

Die folgenden Makros können in Namen und Werten von Tags von Vorlagen, Hosts und Datenpunkten/Webszenarien verwendet werden:

- integrierte Makros {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID}
- integrierte Makros {INVENTORY.*}
- Benutzermakros
- Low-level discovery-Makros (nur in Tags von Host- und Datenpunkt-Prototypen)

Die folgenden Makros können in Auslöser-basierten Benachrichtigungen verwendet werden:

- integrierte Makros {EVENT.TAGS} und {EVENT.RECOVERY.TAGS} (diese Makros werden zu einer durch Kommas getrennten Liste von Ereignis-Tags bzw. Tags von Wiederherstellungsereignissen aufgelöst)
- integrierte Makros {EVENT.TAGSJSON} und {EVENT.RECOVERY.TAGSJSON} (diese Makros werden zu einem JSON-Array aufgelöst, das Ereignis-Tag-Objekte oder Objekte von Wiederherstellungsereignis-Tags enthält)

Teilzeichenfolgenextraktion in Auslöser-Tags

Die Teilzeichenfolgenextraktion wird zum Befüllen des Tag-Namens oder Tag-Werts mithilfe einer Makro-Funktion unterstützt. Die Funktion wendet einen regulären Ausdruck auf den Wert an, der durch das unterstützte Makro erhalten 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 255 Zeichen überschreitet, wird er auf 255 Zeichen gekürzt.

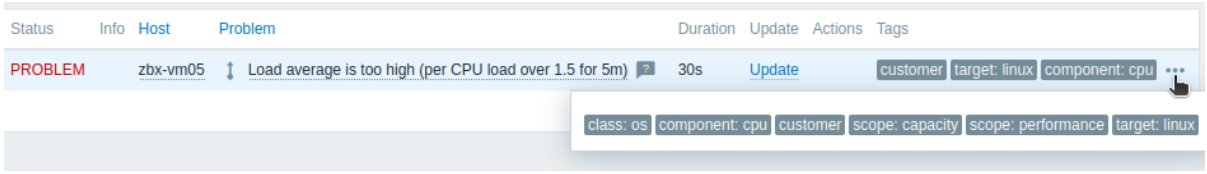
Siehe auch: Verwendung von Makrofunktionen in Low-Level-Discovery-Makros für Event-Tagging.

Anzeigen von Ereignis-Tags

Tags können, sofern definiert, 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 Dashboard-Widget *Probleme* bestimmt. Beachten Sie, dass maximal drei Tags angezeigt werden können; wenn mehr Tags vorhanden sind, werden beim Überfahren der drei Punkte alle Tags in einem Pop-up-Fenster angezeigt.



7 Visualisierung

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte im Abschnitt „Visualisierung“ zuzugreifen.

1 Diagramme

Übersicht

Da viele Daten in Zabbix einfließen, wird es für die Benutzer deutlich einfacher, wenn sie eine visuelle Darstellung dessen sehen können, was vor sich geht, anstatt nur Zahlen.

Hier kommen Diagramme ins Spiel. Diagramme ermöglichen es, den Datenfluss auf einen Blick zu erfassen, Probleme zu korrelieren, zu erkennen, wann etwas begonnen hat, oder darzustellen, 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

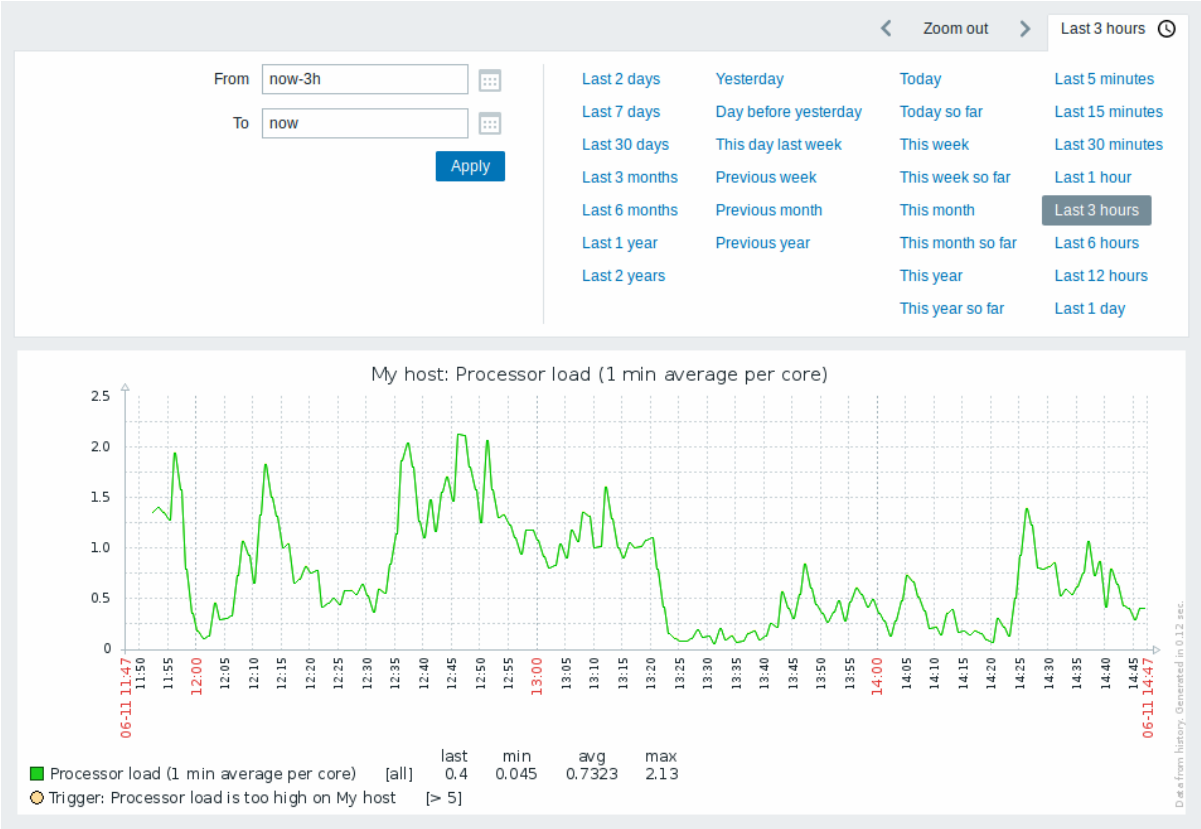
1 Einfache Diagramme

Übersicht

Einfache Diagramme stehen zur Visualisierung der von Datenpunkten erfassten Daten zur Verfügung.

Für die Anzeige einfacher Diagramme ist seitens des Benutzers kein Konfigurationsaufwand erforderlich. Sie werden von Zabbix automatisch bereitgestellt.

Gehen Sie einfach zu *Monitoring* > *Letzte Daten* und klicken Sie beim jeweiligen Datenpunkt auf den Link „Diagramm“, dann wird ein Diagramm angezeigt.



Note:
Einfache Diagramme stehen für alle numerischen Datenpunkte zur Verfügung. Für textuelle Datenpunkte ist in *Monitoring* > *Letzte Daten* ein Link zum Verlauf verfügbar.

Auswahl des Zeitraums

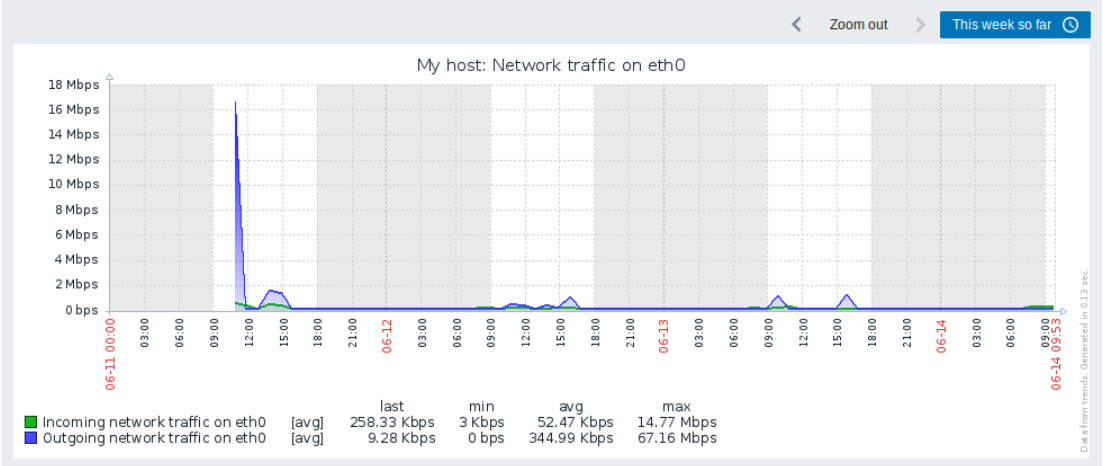
Die Auswahl des Zeitraums oberhalb des Diagramms ermöglicht es, häufig benötigte Zeiträume mit einem Mausklick auszuwählen. Weitere Informationen finden Sie unter Zeit- und Host-Auswahl.

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 gelbem Hintergrund gefüllt.

Die Arbeitszeit (Arbeitstage) wird in Diagrammen mit weißem Hintergrund angezeigt, während die Nicht-Arbeitszeit 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 darstellt.

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; gibt es mehr Auslöser, werden die Auslöser mit niedrigerem Schweregrad priorisiert. Auslöser werden in einfachen Diagrammen immer angezeigt, während ihre Anzeige in benutzerdefinierten Diagrammen eine Benutzereinstellung ist.

Aus Verlauf/Trends erzeugen

Diagramme können entweder auf Basis des Verlaufs oder der Trends eines Datenpunkts gezeichnet werden.

Für Benutzer, bei denen im Frontend der Debug-Modus aktiviert ist, wird unten rechts in einem Diagramm eine graue, vertikale Beschriftung angezeigt, die angibt, woher die Daten stammen.

Mehrere Faktoren beeinflussen, ob Verlauf oder Trends verwendet werden:

- Aufbewahrungsdauer des Datenpunkt-Verlaufs. Beispielsweise kann der Verlauf eines Datenpunkts 14 Tage lang aufbewahrt werden. In diesem Fall stammen alle Daten, die älter als vierzehn Tage sind, aus den Trends.
- Datendichte im Diagramm. Wenn die Anzahl der pro horizontalem Diagramm-Pixel darzustellenden Sekunden 3600/16 überschreitet, werden Trenddaten angezeigt (auch wenn für denselben Zeitraum noch Datenpunkt-Verlauf verfügbar ist).
- Wenn Trends deaktiviert sind, wird zum Erstellen des Diagramms der Datenpunkt-Verlauf 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.

Wechsel zu Rohwerten

Ein Dropdown-Menü oben rechts ermöglicht den Wechsel vom einfachen Diagramm zu den Listen *Werte/500 letzte Werte*. Dies kann nützlich sein, um die numerischen Werte anzuzeigen, aus denen sich das Diagramm zusammensetzt.

Die hier dargestellten Werte sind Rohwerte, d. h. es werden keine Einheiten oder Nachbearbeitungen der Werte verwendet. Die Wertezuordnung wird jedoch angewendet.

Bekannte Probleme

Siehe bekannte Probleme für Graphen.

2 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.

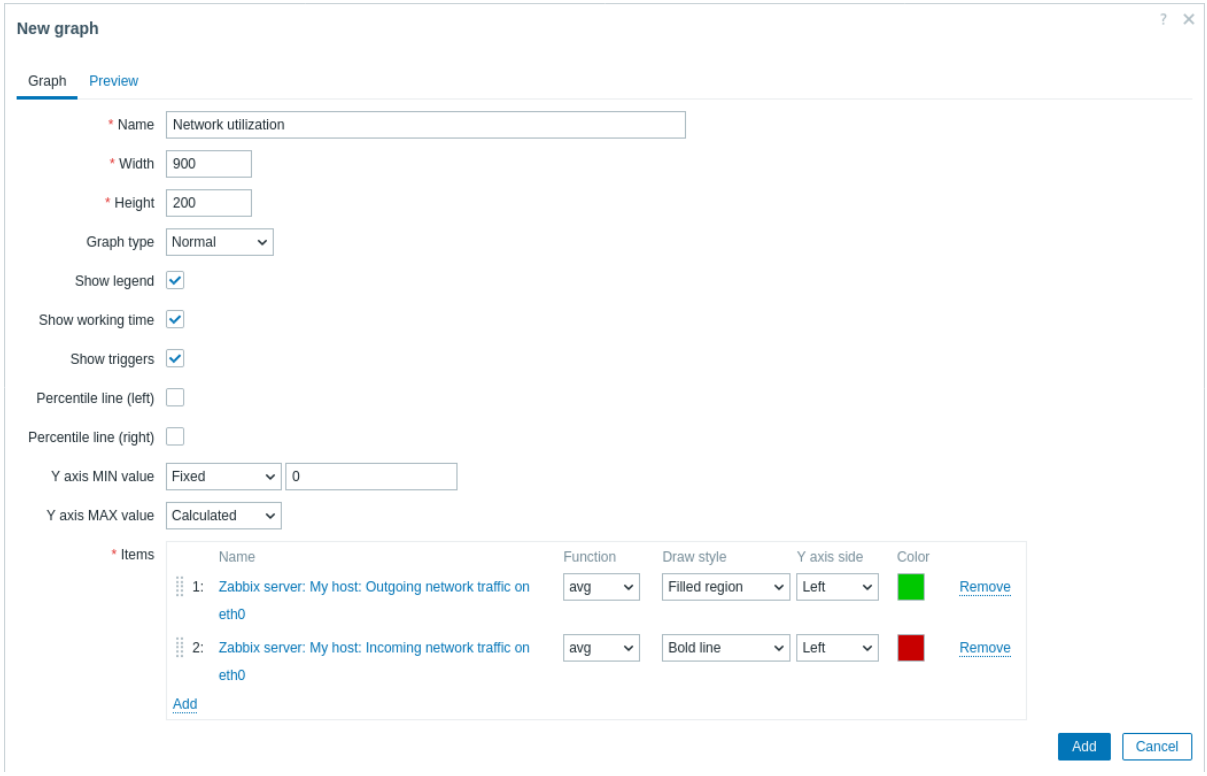
Benutzerdefinierte Diagramme werden manuell konfiguriert.

Sie können für einen Host oder mehrere Hosts oder für eine einzelne Vorlage erstellt werden.

Konfigurieren benutzerdefinierter Diagramme

Um ein benutzerdefiniertes Diagramm zu erstellen, gehen Sie wie folgt vor:

- Gehen Sie zu *Datenerfassung > Hosts (oder Vorlagen)*
- Klicken Sie in der Zeile neben dem gewünschten Host oder der gewünschten Vorlage auf *Diagramme*
- Klicken Sie im Bildschirm „Diagramme“ auf *Diagramm erstellen*
- Bearbeiten Sie die Diagrammattribute



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Diagrammattribute:

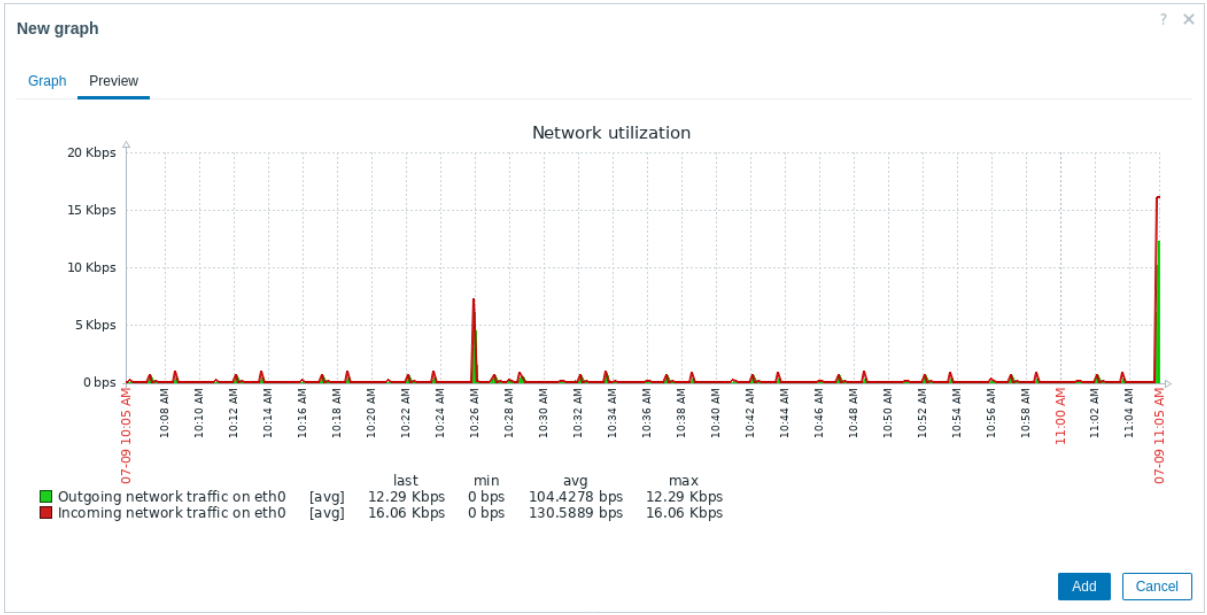
Parameter	Beschreibung
Name	Eindeutiger Diagrammname. Ausdrücke Makros werden in diesem Feld unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max`, wobei Zeit als Parameter verwendet wird (zum Beispiel `{?avg(/host/key,1h)}`). {HOST.HOST<1-9>}-Makros werden zur Verwendung innerhalb 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 redundant ist, da auf den ersten Host implizit verwiesen werden kann, zum Beispiel `{?avg(//key,1h)}`.

Parameter	Beschreibung
Width	Diagrammbreite in Pixeln (nur für Vorschau sowie Kreis-/explodierte Diagramme).
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** - „explodiertes“ Kreisdiagramm, Segmente werden als aus dem Kreis „herausgeschnitten“ dargestellt
Show legend	Wenn dieses Kontrollkästchen aktiviert ist, wird die Diagrammlegende angezeigt.
Show working time	Wenn ausgewählt, werden arbeitsfreie Stunden mit grauem Hintergrund angezeigt. Dieser Parameter ist für Kreis- und explodierte Kreisdiagramme nicht verfügbar.
Show triggers	Wenn ausgewählt, werden einfache Auslöser als Linien mit schwarzen Strichen über der Farbe des Auslöser-Schweregrads angezeigt. Dieser Parameter ist für Kreis- und explodierte Kreisdiagramme nicht verfügbar.
Percentile line (left)	Perzentil für die linke Y-Achse anzeigen. Wenn zum Beispiel das 95%-Perzentil festgelegt ist, befindet sich die Perzentillinie auf der Höhe, unter der 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 zum Beispiel das 95%-Perzentil festgelegt ist, befindet sich die Perzentillinie auf der Höhe, unter der 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 minimale Wert der Y-Achse wird automatisch berechnet. **Fixed** - fester Minimalwert für die Y-Achse. **Item** - der letzte Wert des ausgewählten Datenpunkts wird als Minimalwert verwendet. Dieser Parameter ist für Kreis- und explodierte Kreisdiagramme nicht verfügbar.
Y axis MAX value	Maximalwert der Y-Achse: **Calculated** - der maximale Wert der Y-Achse wird automatisch berechnet. **Fixed** - fester Maximalwert für die Y-Achse. **Item** - der letzte Wert des ausgewählten Datenpunkts wird als Maximalwert verwendet Dieser Parameter ist für Kreis- und explodierte Kreisdiagramme nicht verfügbar.
3D view	3D-Stil aktivieren. Nur für Kreis- und explodierte Kreisdiagramme.
Items	Datenpunkte, deren Daten in diesem Diagramm angezeigt werden sollen. Klicken Sie auf *Hinzufügen*, um Datenpunkte auszuwählen. Sie können auch verschiedene Anzeigeoptionen auswählen (Funktion, Zeichenstil, 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) anderen zu zeichnen. Sie können Datenpunkte per Drag-and-drop mit dem Symbol am Anfang einer Zeile verschieben, um die Sortierreihenfolge festzulegen oder zu bestimmen, welcher Datenpunkt vor dem anderen angezeigt wird.
Name	Der Name des ausgewählten Datenpunkts wird als Link angezeigt. Ein Klick auf den Link öffnet die Liste anderer verfügbarer Datenpunkte.
Type	Typ (nur für Kreis- und explodierte Kreisdiagramme verfügbar): **Simple** - der Wert des Datenpunkts wird proportional im Kreisdiagramm dargestellt **Graph sum** - der Wert des Datenpunkts stellt das gesamte Kreisdiagramm dar Beachten Sie, dass die Färbung des Datenpunkts „graph sum“ nur in dem Maß sichtbar ist, in dem sie nicht von „proportionalen“ Datenpunkten eingenommen wird.

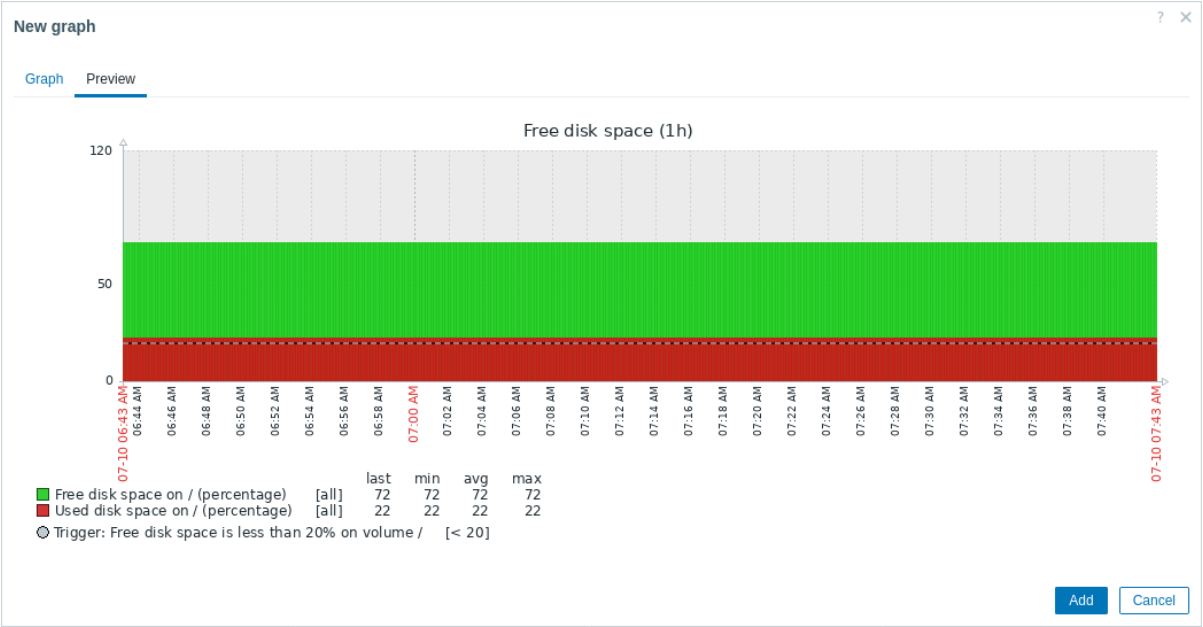
Parameter	Beschreibung
Function	Wählen Sie aus, welche Werte angezeigt werden, wenn für einen Datenpunkt mehr als ein Wert pro vertikalem Diagrammpixel vorhanden ist: **all** - alle möglichen Werte (Minimum, Maximum, Durchschnitt) im Diagramm anzeigen. Beachten Sie, dass diese Einstellung bei kürzeren Zeiträumen keine Auswirkung hat; erst bei längeren Zeiträumen, wenn die Datendichte in einem vertikalen Diagrammpixel zunimmt, beginnt „all“ Minimum-, Maximum- und Durchschnittswerte anzuzeigen. Diese Funktion ist nur für den Diagrammtyp *Normal* verfügbar. Siehe auch: Diagramme aus Verlauf/Trends erzeugen. **avg** - Durchschnittswerte anzeigen **last** - die neuesten Werte anzeigen. Diese Funktion ist nur verfügbar, wenn als Diagrammtyp *Pie/Exploded pie* ausgewählt ist. **max** - Maximalwerte anzeigen **min** - Minimalwerte anzeigen
Draw style	Wählen Sie den Zeichenstil (nur für normale Diagramme verfügbar; bei gestapelten Diagrammen wird immer ein gefüllter Bereich verwendet), der auf die Daten des Datenpunkts angewendet werden soll - *Linie*, *Fette Linie*, *Gefüllter Bereich*, *Punkt*, *Gestrichelte Linie*, *Verlaufslinie*.
Y axis side	Wählen Sie die Seite der Y-Achse aus, auf der die Daten des Datenpunkts angezeigt werden sollen - *Links*, *Rechts*.
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.

3 Ad-hoc-Diagramme

Übersicht

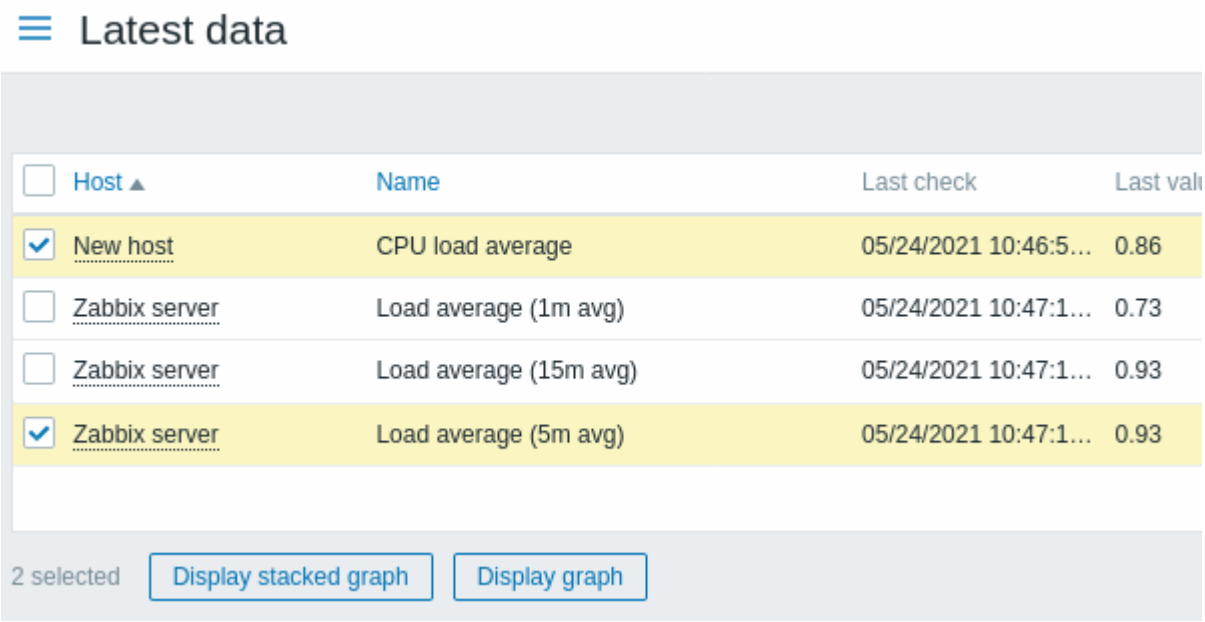
Während ein einfaches Diagramm hervorragend geeignet ist, um auf Daten eines Datenpunkts zuzugreifen, und benutzerdefinierte Diagramme Anpassungsoptionen bieten, ermöglicht keine der beiden Optionen, mit wenig Aufwand und ohne Wartung schnell ein Vergleichsdiagramm für mehrere Datenpunkte zu erstellen.

Um dieses Problem zu lösen, können Ad-hoc-Diagramme für mehrere Datenpunkte sehr schnell erstellt werden.

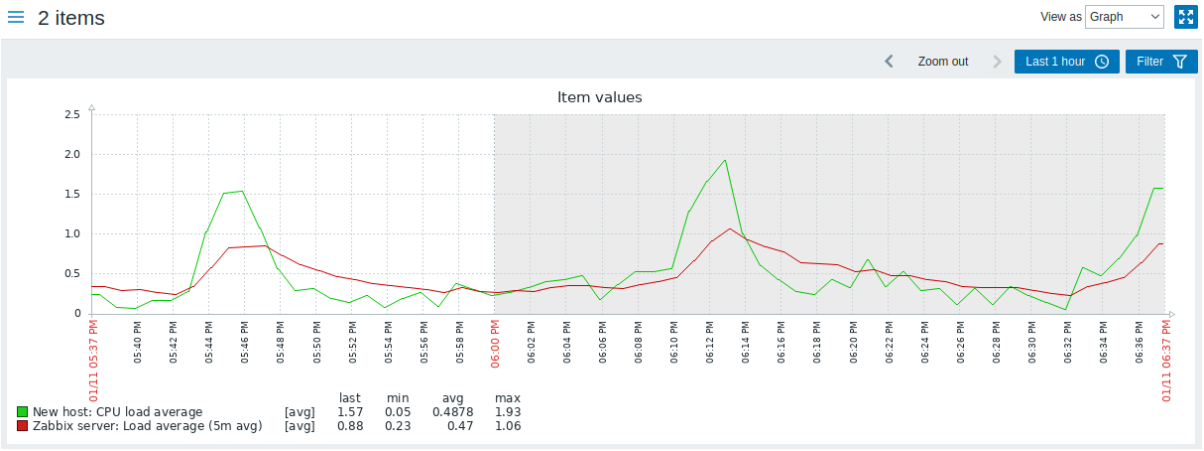
Konfiguration

Gehen Sie wie folgt vor, um ein Ad-hoc-Diagramm zu erstellen:

- Gehen Sie zu *Monitoring* → *Letzte Daten*
- Verwenden Sie den Filter, um die Datenpunkte anzuzeigen, die Sie möchten
- Markieren Sie die Kontrollkästchen der Datenpunkte, die Sie grafisch darstellen möchten
- Klicken Sie auf die Schaltflächen *Gestapeltes Diagramm anzeigen* oder *Diagramm anzeigen*

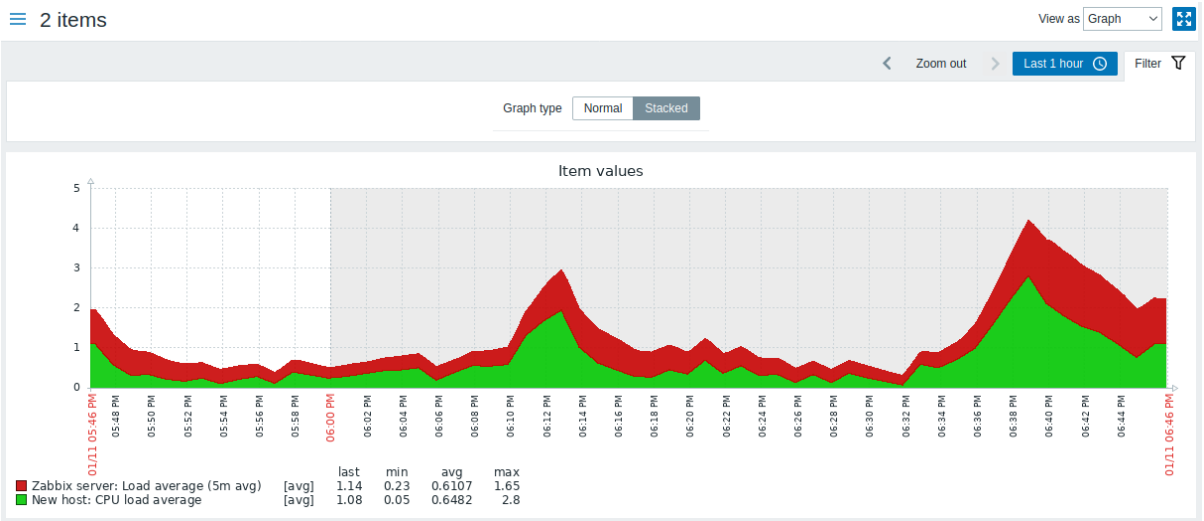


Ihr Diagramm wird sofort erstellt:



Beachten Sie, dass zur Vermeidung der Anzeige zu vieler Linien im Diagramm nur der Durchschnittswert für jeden Datenpunkt angezeigt wird (Linien für Minimal-/Maximalwerte werden nicht angezeigt). Auslöser und Auslöserinformationen werden im Diagramm nicht angezeigt.

Im Fenster des erstellten Diagramms stehen Ihnen der Zeitraumauswahl sowie die Möglichkeit zur Verfügung, vom „normalen“ Liniendiagramm zu einem gestapelten Diagramm zu wechseln (und zurück).



4 Aggregation in Graphen

Übersicht

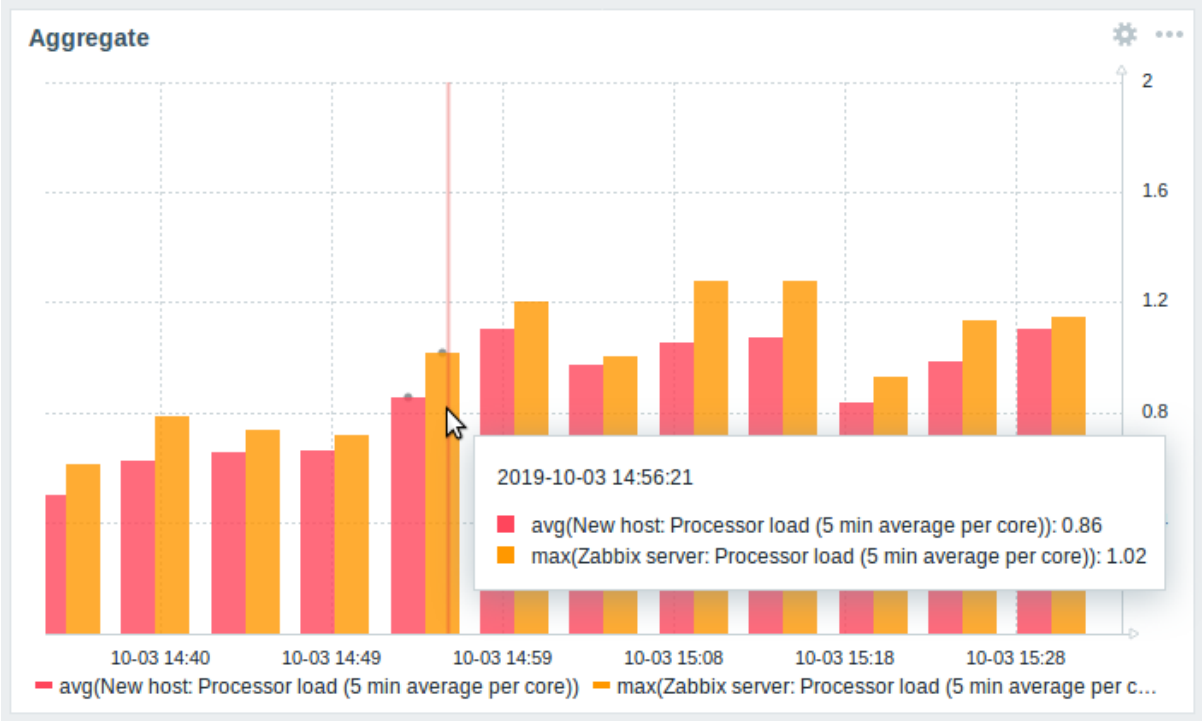
Aggregationsfunktionen, die in den Widgets „Graph“ und „Kreisdiagramm“ des Dashboards verfügbar sind, ermöglichen die Anzeige eines aggregierten Werts für das gewählte Intervall (5 Minuten, eine Stunde, ein Tag) anstelle aller Werte.

Dieser Abschnitt enthält weitere Details zu den Aggregationsoptionen im Graph-Widget.

Die Aggregationsoptionen sind wie folgt:

- min
- max
- avg
- count
- sum
- first (erster angezeigter Wert)
- last (letzter angezeigter Wert)

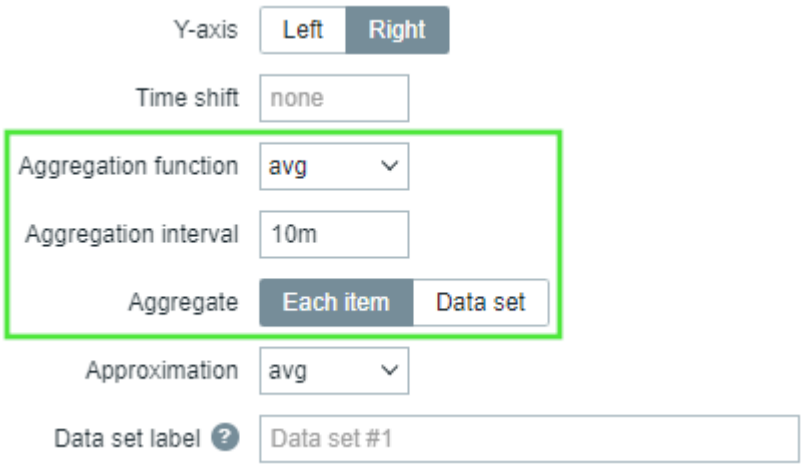
Der interessanteste Anwendungsfall der Datenaggregation ist die Möglichkeit, ansprechende Vergleiche von Daten für einen bestimmten Zeitraum nebeneinander zu erstellen:



Wenn Sie mit der Maus über einen Zeitpunkt im Graphen fahren, werden zusätzlich zu den Datenpunkten und ihren aggregierten Werten auch Datum und Uhrzeit angezeigt. Datenpunkte werden in Klammern angezeigt, mit der verwendeten Aggregationsfunktion als Präfix. Wenn im Graph-Widget ein *Datenreihenbeschriftung* konfiguriert ist, wird die Beschriftung in Klammern angezeigt, mit der verwendeten Aggregationsfunktion als Präfix. Beachten Sie, dass dies das Datum und die Uhrzeit des Punkts im Graphen sind, nicht die der tatsächlichen Werte.

Konfiguration

Die Optionen für die Aggregation sind in den Einstellungen des Datensatzes verfügbar, wenn ein Graph-Widget konfiguriert wird.



Sie können die Aggregationsfunktion und das Zeitintervall auswählen. Da der Datensatz mehrere Datenpunkte umfassen kann, gibt es außerdem eine weitere Option, mit der aggregierte Daten für jeden Datenpunkt separat oder für alle Datenpunkte des Datensatzes als ein aggregierter Wert angezeigt werden können.

Anwendungsfälle

Durchschnittliche Anzahl von Anfragen an den Nginx-Server

Zeigen Sie die durchschnittliche Anzahl von Anfragen pro Sekunde und Tag an den Nginx-Server an:

- Fügen Sie den Datenpunkt für die Anzahl von Anfragen pro Sekunde zum Datensatz hinzu
- Wählen Sie die Aggregatfunktion `avg` aus und geben Sie das Intervall `1d` an
- Es wird ein Balkendiagramm angezeigt, in dem jeder Balken die durchschnittliche Anzahl von Anfragen pro Sekunde pro Tag darstellt

Minimaler wöchentlicher Festplattenspeicher unter Clustern

Zeigt den niedrigsten Festplattenspeicher unter Clustern über eine Woche an.

- zum Datensatz hinzufügen: Hosts `cluster*`, Schlüssel „Freier Festplattenspeicher auf /data“
- die Aggregatfunktion `min` auswählen und das Intervall `1w` angeben
- es wird ein Balkendiagramm angezeigt, wobei jeder Balken den minimalen Festplattenspeicher pro Woche für jedes /data-Volume des Clusters darstellt

2 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.

1 Konfigurieren einer Netzwerkkarte

Übersicht

Die Konfiguration einer Karte in Zabbix erfordert, dass Sie zunächst eine Karte erstellen, indem Sie ihre allgemeinen Parameter definieren. Anschließend beginnen Sie damit, die eigentliche Karte mit Elementen und deren Verknüpfungen zu füllen.

Sie können die Karte mit Elementen füllen, die ein Host, eine Host-Gruppe, ein Auslöser, ein Bild oder eine andere Karte sind.

Symbole werden verwendet, um Kartenelemente darzustellen. Sie können festlegen, welche Informationen zusammen mit den Symbolen angezeigt werden, und einstellen, dass aktuelle Probleme auf besondere Weise dargestellt werden. Sie können die Symbole miteinander 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 Host-Eigenschaften oder ein Kartensymbol mit einer anderen Karte verknüpfen.

Die Anzahl der Probleme in Karten wird nur für ursächliche 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.

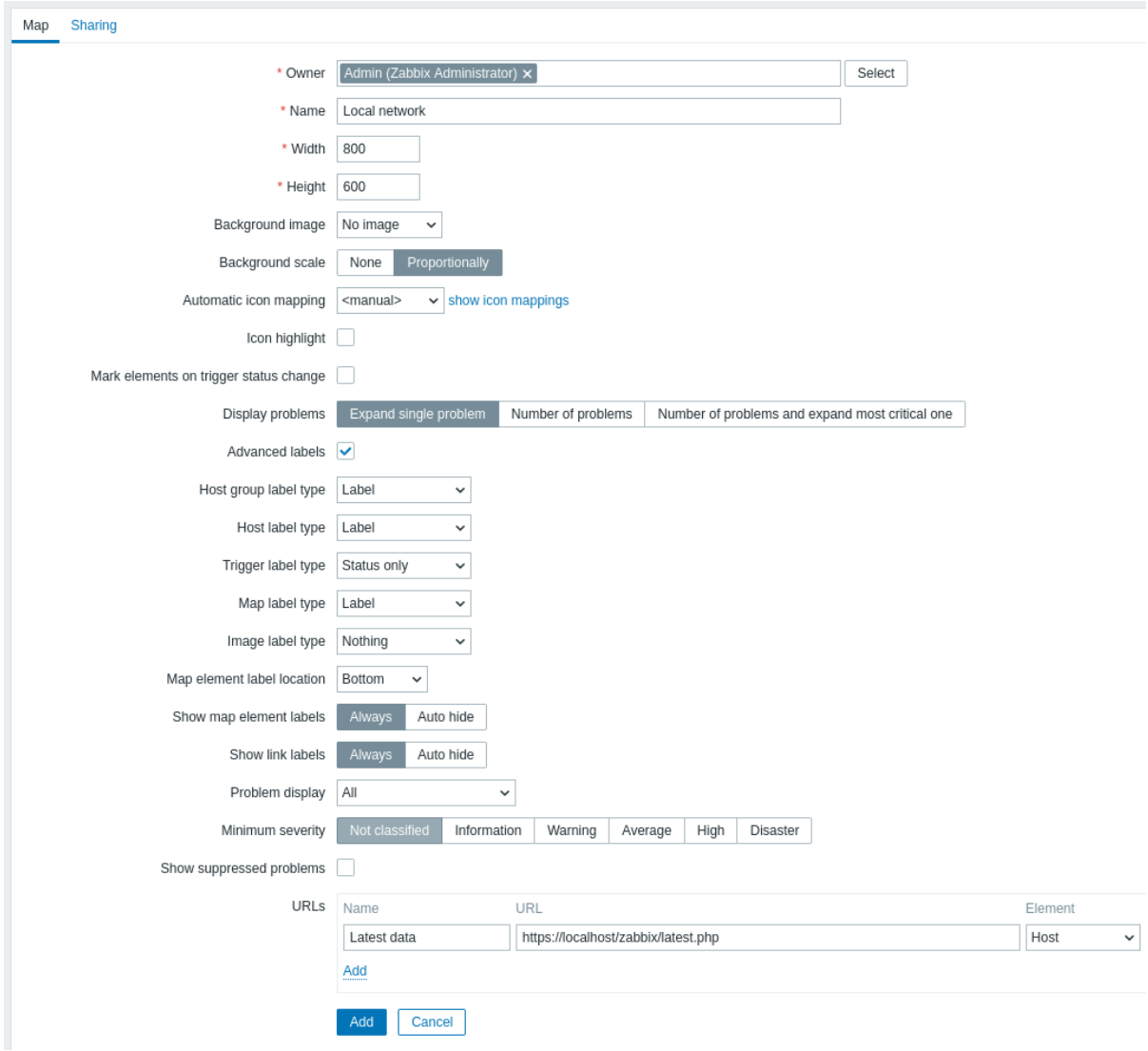
Erstellen einer Karte

Gehen Sie wie folgt vor, um eine Karte zu erstellen:

1. Gehen Sie zu *Monitoring* > *Maps*.
2. Wechseln Sie in die Ansicht mit allen Karten.
3. Klicken Sie auf *Create map*.

Sie können auch die Schaltfläche *Clone* im Konfigurationsformular einer vorhandenen Karte verwenden, um eine neue Karte zu erstellen. Diese Karte übernimmt alle Eigenschaften der vorhandenen Karte, einschließlich allgemeiner Layout-Attribute sowie der Elemente der vorhandenen Karte.

Die Registerkarte **Map** enthält allgemeine Kartenattribute:



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

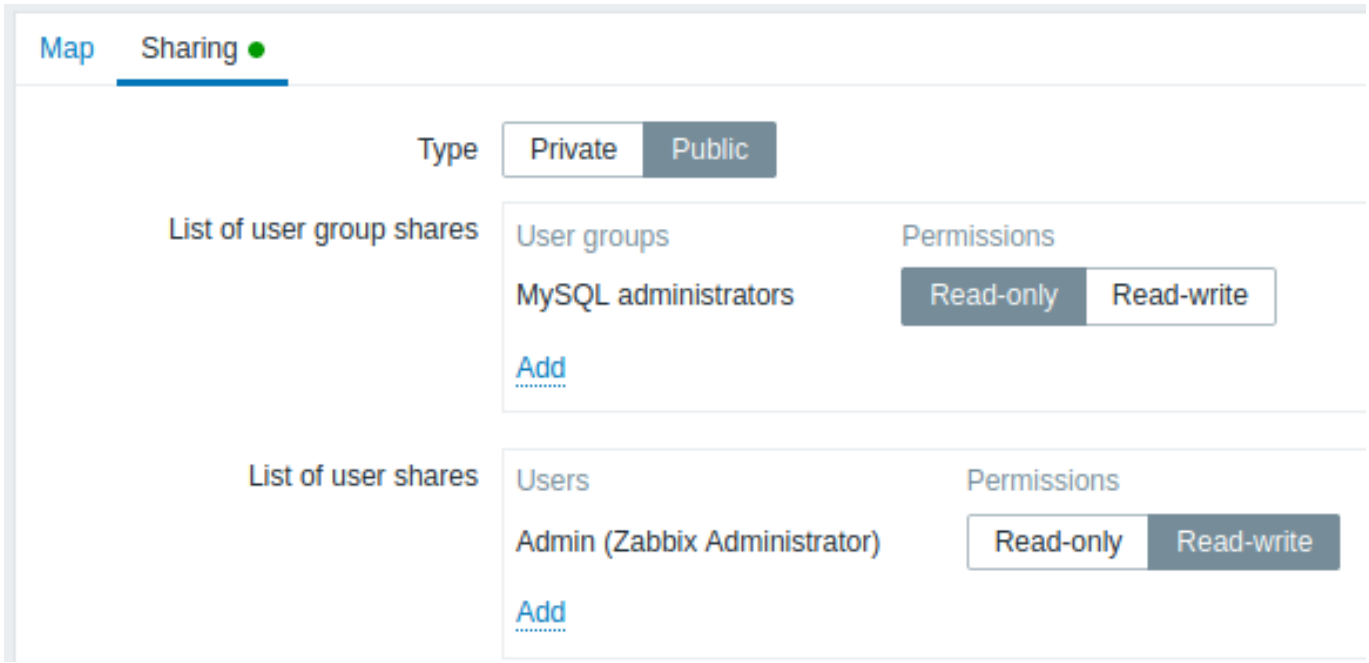
Allgemeine Kartenattribute:

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 anderes Bild verwenden, um Ihre Karte zu verbessern.
Background scale	Hintergrundskalierung verwenden: **None** - keine Skalierung; **Proportionally** - den gesamten Kartenhintergrund ausfüllen, dabei aber die Bildproportionen beibehalten (Standard).
Automatic icon mapping	Sie können festlegen, dass eine automatische Symbolzuordnung verwendet wird, die unter *Administration > General > Icon mapping* konfiguriert ist. Die Symbolzuordnung ermöglicht es, bestimmte Symbole bestimmten Feldern des Host-Inventars zuzuordnen.
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, jeweils grau bzw. orange. Siehe auch: Viewing maps

Parameter	Beschreibung
Mark elements on trigger status change	Eine kürzliche Änderung des Auslöserstatus (aktuelles Problem oder Entwarnung) wird mit Markierungen (nach innen gerichtete rote Dreiecke) an den drei Seiten des Elementsymbols hervorgehoben, die nicht vom Label belegt sind. Markierungen werden 30 Minuten lang angezeigt.
Display problems	Wählen Sie aus, wie Probleme bei 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. Als ”kritischstes” gilt ein Problem auf Grundlage des Problemschweregrads und, bei Gleichstand, der Problemereignis-ID (höhere ID bzw. späteres Problem wird zuerst angezeigt). Bei einem *trigger map element* basiert dies auf dem Problemschweregrad und, bei Gleichstand, auf der Position des Auslösers in der Auslöserliste. Bei mehreren Problemen desselben Auslösers wird das neueste angezeigt.
Advanced labels	Wenn Sie dieses Kontrollkästchen aktivieren, können Sie für verschiedene Elementtypen separate Label-Typen definieren.
Host group label type	Für das Kartenelement verwendeter Label-Typ: **Label** - Kartenelement-Label **IP address** - IP-Adresse **Element name** - Elementname (zum Beispiel Host-Name) **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 in Bezug auf das Kartenelement: **Bottom** - unter dem Kartenelement **Left** - links davon **Right** - rechts davon **Top** - darüber
Show map element labels	Wählen Sie aus, wie Kartenelement-Labels angezeigt werden: **Always** - immer anzeigen (Standard); **Auto hide** - das Label ausblenden, wenn nicht mit der Maus darübergefahren wird oder es nicht ausgewählt ist.
Show link labels	Wählen Sie aus, wie Link-Labels angezeigt werden: **Always** - immer anzeigen (Standard); **Auto hide** - das Label ausblenden, wenn nicht mit der Maus darübergefahren wird oder es nicht ausgewählt ist.
Problem display	Problemanzahl anzeigen als: **All** - die vollständige Problemanzahl wird angezeigt **Separated** - die Anzahl unbestätigter Probleme wird getrennt als Zahl der gesamten Problemanzahl angezeigt **Unacknowledged only** - nur die Anzahl unbestätigter Probleme wird angezeigt
Minimum severity	Probleme unterhalb des ausgewählten Mindestschweregrads werden auf der Karte nicht angezeigt. Wenn zum Beispiel *Warning* ausgewählt ist, werden Änderungen mit Auslösern der Stufen *Information* und *Not classified* nicht in der Karte berücksichtigt.
Show suppressed problems	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden.
URLs	Hier können URLs für jeden Elementtyp definiert werden, wobei gilt: **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 Kartenansichtsmodus auf das Element klickt. Makros können in Namen und Werten von Karten-URLs verwendet werden. Eine vollständige Liste finden Sie unter supported macros; suchen Sie dort nach ’map URL names and values’.

Freigabe

Die Registerkarte **Freigabe** enthält den Kartentyp sowie Freigabeoptionen (Benutzergruppen, Benutzer) für private Karten:



Parameter	Beschreibung
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 schreibgeschützten oder Lese-/Schreibzugriff erlauben.
List of user shares	Benutzer auswählen, für die die Karte zugänglich ist. Sie können schreibgeschützten oder Lese-/Schreibzugriff erlauben.

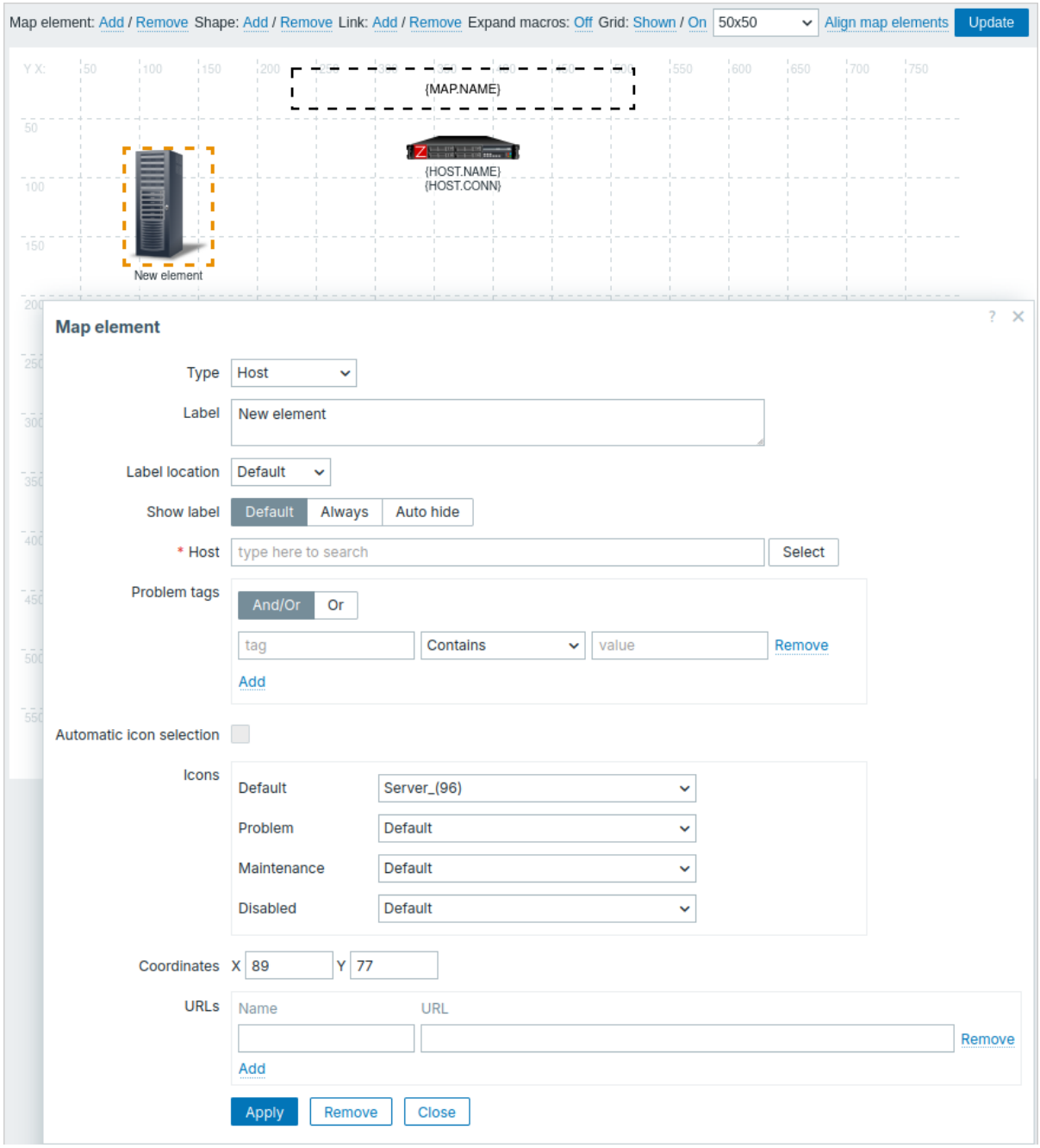
Wenn Sie auf *Add* klicken, um diese Karte zu speichern, haben Sie eine leere Karte mit einem Namen, Abmessungen und bestimmten Einstellungen erstellt. Nun müssen Sie einige Elemente hinzufügen. Klicken Sie dazu in der Kartenliste auf *Edit*, um den bearbeitbaren Bereich zu öffnen.

Elemente hinzufügen

Um ein Element hinzuzufügen, klicken Sie neben `Map element` auf *Add*. Das neue Element erscheint in der oberen linken Ecke der Karte. Ziehen Sie es per Drag-and-drop an die gewünschte Stelle.

Beachten Sie, dass bei aktivierter Rasteroption ”On” Elemente immer am Raster ausgerichtet werden (Sie können verschiedene Rastergrößen aus der Auswahlliste wählen und das Raster auch ein-/ausblenden). Wenn Sie Elemente beliebig ohne Ausrichtung platzieren möchten, setzen Sie die Option auf ”Off”. (Sie können zufällig platzierte Elemente später durch Klick auf *Align map elements* am Raster ausrichten.)

Sobald Sie einige Elemente platziert haben, möchten Sie diese möglicherweise durch Namen usw. voneinander unterscheiden. Wenn Sie auf ein 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	Beschriftung des Symbols, 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` mit Zeit als Parameter (zum Beispiel `{?avg(/host/key,1h)}`). Eine vollständige Liste der unterstützten Makros finden Sie unter supported macros; suchen Sie dort nach 'map element labels'.

Parameter	Beschreibung
Label location	Position der Beschriftung relativ zum Symbol: **Default** - Standardposition der Kartenbeschriftung; **Bottom** - unter dem Symbol; **Left** - links davon; **Right** - rechts davon; **Top** - über dem Symbol.
Show label	Wählen Sie aus, wie die Elementbeschriftung angezeigt werden soll: **Default** - verwendet die Einstellung *Show map element labels* aus der Kartenkonfiguration (Standard); **Always** - immer anzeigen (Standard); **Auto hide** - die Beschriftung ausblenden, wenn nicht mit der Maus darüber gefahren wird oder das Element nicht ausgewählt ist.
Host	Wählen Sie einen Host aus; alternativ geben Sie den Hostnamen ein. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie also beginnen, den Namen eines Hosts einzugeben, wird eine Auswahlliste mit passenden Hosts angezeigt. 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; alternativ geben Sie den Kartennamen ein. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie also beginnen, den Namen einer Karte einzugeben, wird eine Auswahlliste mit passenden Karten angezeigt. 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 beginnen, den Namen eines Auslösers einzugeben, und einen Eintrag aus der Auswahlliste passender Auslöser 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 Auslöserschweregrade. Die Auswahl mehrerer Auslöser beeinflusst auch die Auflösung des Makros {HOST.*} sowohl im Bearbeitungs- als auch im Ansichtsmodus. Im Bearbeitungsmodus werden die zuerst angezeigten Makros {HOST.*} abhängig vom ersten Auslöser in der Liste aufgelöst (basierend auf dem Auslöserschweregrad). Der Ansichtsmodus hängt vom Parameter Display problems in den allgemeinen Kartenattributen ab: - Wenn der Modus *Expand single problem* gewählt ist, werden die zuerst angezeigten Makros {HOST.*} 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 Makros {HOST.*} abhängig vom Auslöserschweregrad aufgelöst.
Host group	Wählen Sie eine Hostgruppe aus; alternativ geben Sie die Hostgruppe ein. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie also beginnen, den Namen einer Hostgruppe einzugeben, wird eine Auswahlliste mit passenden Hostgruppen angezeigt. 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 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 Dieses Feld ist nur für Elemente vom Typ *Host* und *Host group* verfügbar.
Automatic icon selection	Wenn das Kontrollkästchen markiert ist, wird die Symbolzuordnung verwendet, um zu bestimmen, welches Symbol angezeigt wird.
Icons	Sie können festlegen, dass für das Element in folgenden Fällen unterschiedliche Symbole angezeigt werden: *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 Bezeichnung für die URL; **URL** - eine URL (bis zu 2048 Zeichen). Diese werden als Links angezeigt, wenn ein Benutzer im Kartenansichtsmodus auf das Element klickt. Wenn das Element eigene URLs hat und für seinen Typ URLs auf Kartenebene definiert sind, werden diese im selben Menü zusammengeführt. Makros können in Namen und Werten von Kartenelementen verwendet werden. Eine vollständige Liste finden Sie unter supported macros; suchen Sie dort nach 'map URL names and values'.

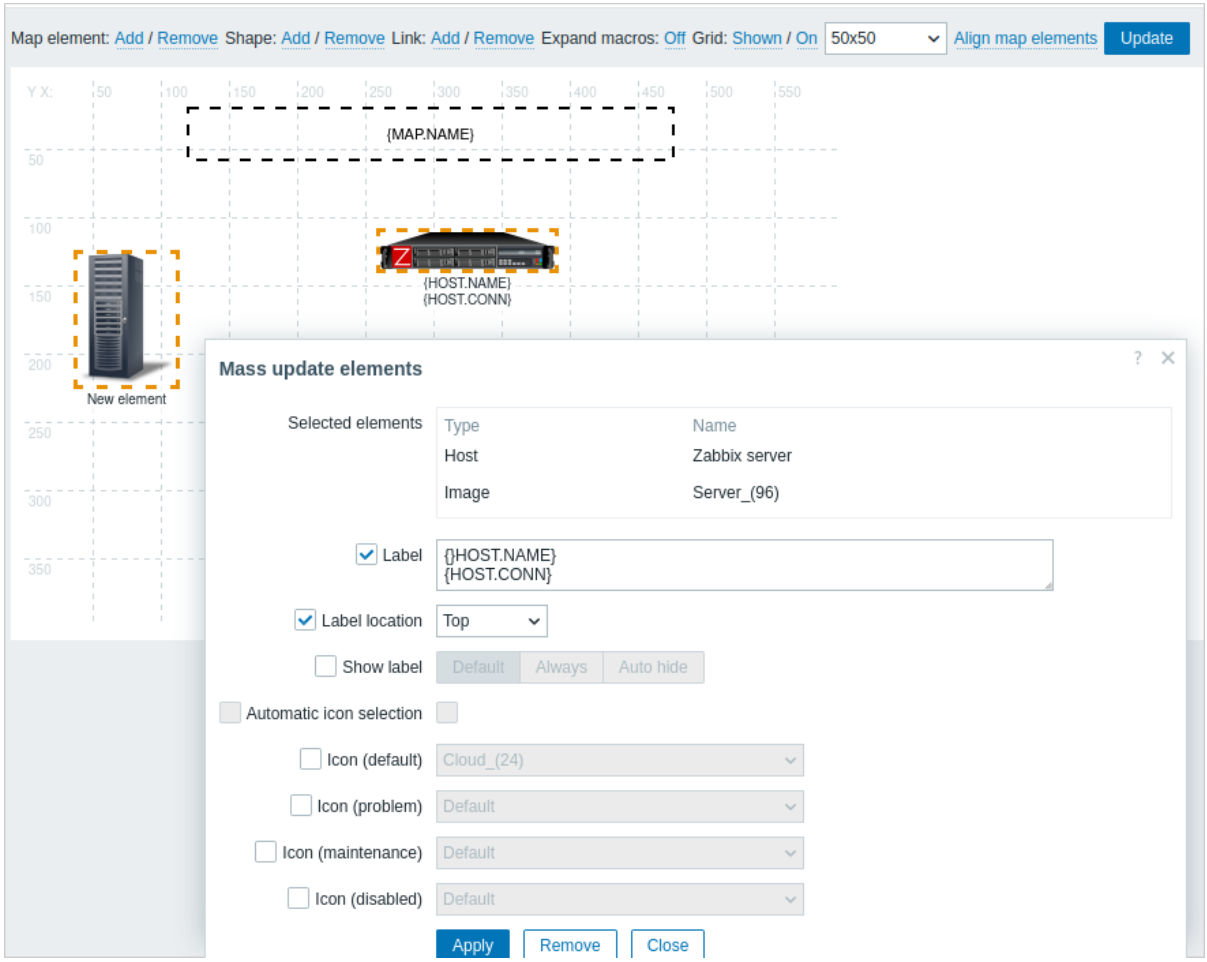
Attention:
Hinzugefügte Elemente werden nicht automatisch gespeichert. Wenn Sie die Seite verlassen, können alle Änderungen verloren gehen.
Daher empfiehlt es sich, auf die Schaltfläche **Update** in der oberen rechten Ecke zu klicken. Nach dem Klick werden die Änderungen gespeichert, unabhängig davon, 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.

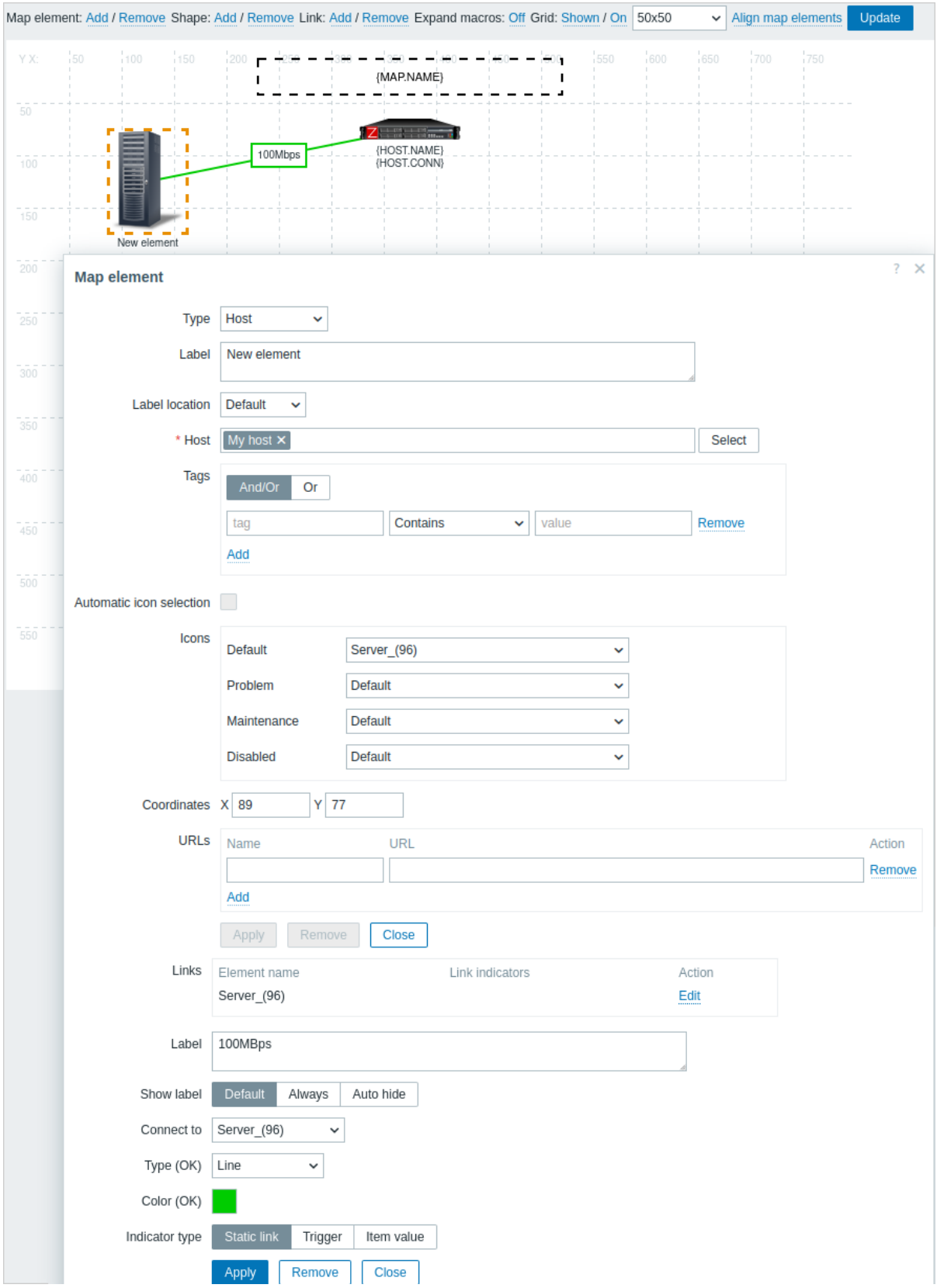
Sobald Sie mehr als ein Element auswählen, wechselt das Formular für die Elementeigenschaften in den Massenaktualisierungsmodus, sodass Sie Attribute der ausgewählten Elemente in einem Schritt ändern können. Markieren Sie dazu das Attribut mit dem Kontrollkästchen und geben Sie einen neuen Wert dafür ein. Sie können hier Makros verwenden (zum Beispiel {HOST.NAME} für die Elementbeschriftung).



Verknüpfen von Elementen

Sobald Sie einige Elemente auf der Karte platziert haben, ist es an der Zeit, sie miteinander zu verknüpfen. Um zwei Elemente zu verknüpfen, müssen Sie sie zunächst auswählen. Wenn die Elemente ausgewählt sind, klicken Sie neben Link auf *Hinzufügen*.

Nachdem eine Verknüpfung erstellt wurde, enthält das Formular für ein einzelnes Element nun einen zusätzlichen Abschnitt *Links*. Klicken Sie auf *Bearbeiten*, um die Attribute der Verknüpfung zu bearbeiten.



Attribute der Verknüpfung:

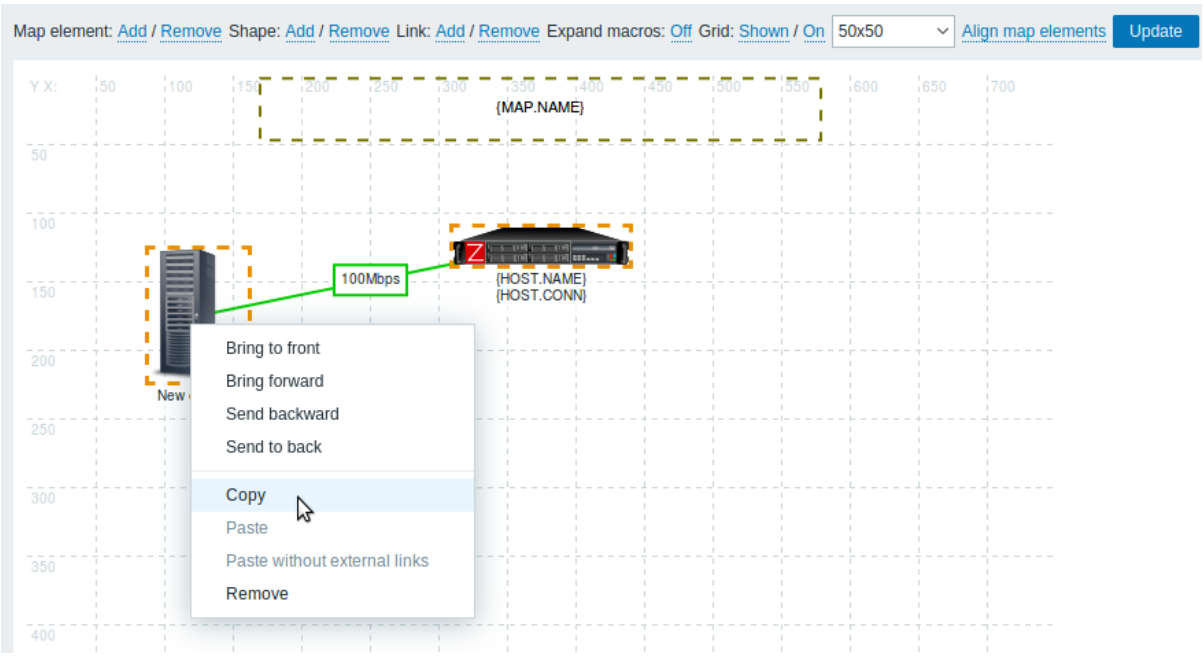
Parameter	Beschreibung
Beschriftung	Beschriftung, die oberhalb der Verknüpfung dargestellt wird. Ausdrücke mit Makros werden in diesem Feld unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max`, wobei Zeit als Parameter verwendet wird (zum Beispiel `{?avg(/host/key,1h)}`).

Parameter	Beschreibung
Beschriftung anzeigen	Wählen Sie aus, wie die Beschriftung der Verknüpfung angezeigt werden soll: **Standard** - die Einstellungen von *Show map element labels* aus der Kartenkonfiguration verwenden (Standard); **Immer** - immer anzeigen (Standard); **Automatisch ausblenden** - die Beschriftung ausblenden, wenn nicht mit der Maus darüber gefahren wird und sie nicht ausgewählt ist.
Verbinden mit	Das Element, mit dem die Verknüpfung verbunden wird.
Typ (OK)	Standardstil der Verknüpfung. Wählen Sie den Stil aus: *Linie*, *Fette Linie*, *Punkt* oder *Gestrichelte Linie*.
Farbe (OK)	Standardfarbe der Verknüpfung. Klicken Sie auf das Farbfeld, um eine andere Farbe auszuwählen.
Indikatortyp	Wählen Sie den Typ des Verknüpfungsindikators aus: **Statische Verknüpfung** - keine Indikatoren; **Auslöser** - Auslöser-basierte Verknüpfungsindikatoren zulassen; **Datenpunktwert** - Verknüpfungsindikatoren auf Basis von Datenpunktwerten zulassen.
Datenpunkt	Wählen Sie den Datenpunkt aus. Der Wert dieses Datenpunkts beeinflusst den Stil der Verknüpfung. Dieses Feld ist verfügbar, wenn für *Indikatortyp* die Option *Datenpunktwert* ausgewählt ist.
Indikatoren	Die Liste der Verknüpfungsindikatoren. Dieses Feld ist verfügbar, wenn *Indikatortyp* auf *Auslöser* oder *Datenpunktwert* gesetzt ist. Wenn *Auslöser* gesetzt 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 *Datenpunktwert* gesetzt ist und im Feld *Datenpunkt* ein Datenpunkt ausgewählt wurde, können Datenpunktwerte als Verknüpfungsindikatoren zugewiesen werden. Wenn der Datenpunktwert den angegebenen Schwellenwert erreicht (bei einem numerischen Datenpunkt) oder dem Muster entspricht (bei einem Text-Datenpunkt), wird sein Stil (ausgewählte Farbe und Linientyp) auf die Verknüpfung angewendet.

Elemente verschieben und kopieren/einfügen

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.

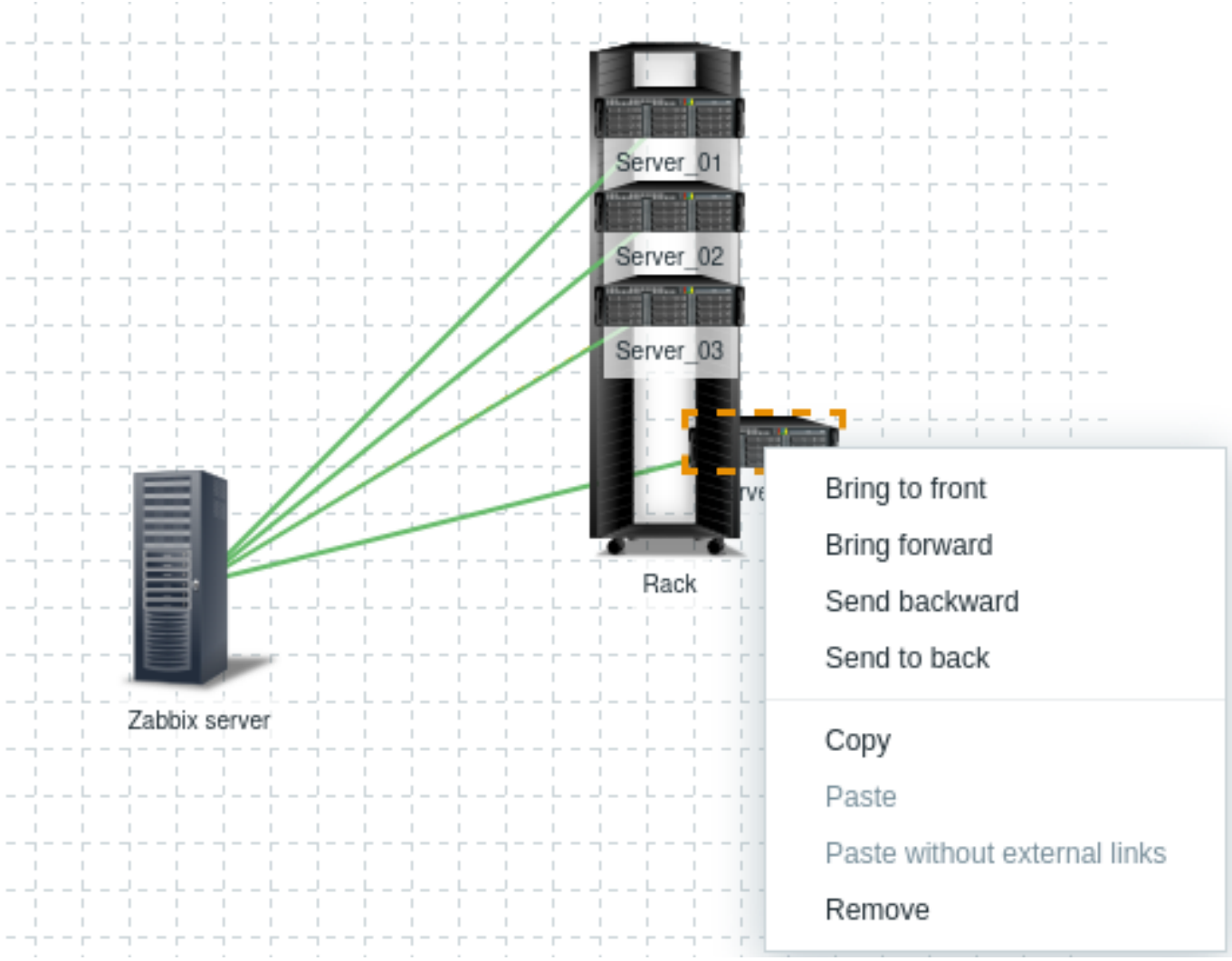


Um die Elemente einzufügen, klicken Sie mit der rechten Maustaste auf einen Bereich der Karte und wählen Sie im Menü *Paste* aus. Mit der Option *Paste without external links* werden die Elemente eingefügt, wobei nur die Verknüpfungen zwischen den ausgewählten Elementen beibehalten werden.

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/In den Vordergrund* oder *Nach hinten/In den Hintergrund*.

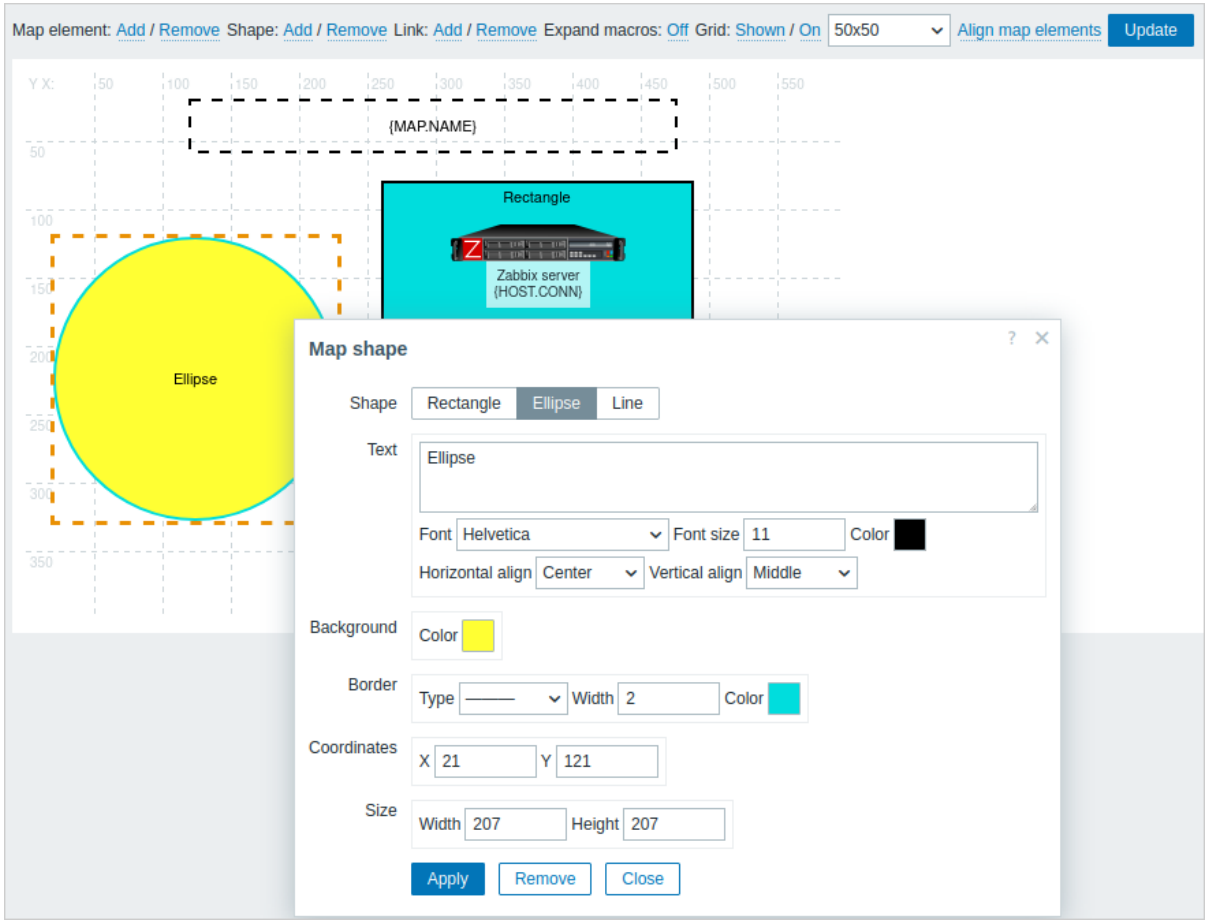


Formen hinzufügen

Zusätzlich zu Kartenelementen können auch einige Formen hinzugefügt werden. Formen sind keine Kartenelemente; sie dienen nur der visuellen Darstellung. So kann beispielsweise eine rechteckige Form als Hintergrund verwendet werden, um einige Hosts zu gruppieren. Es können rechteckige und elliptische Formen hinzugefügt werden.

Um eine Form hinzuzufügen, klicken Sie neben Form auf *Hinzufügen*. Die neue Form erscheint in der oberen linken Ecke der Karte. Ziehen Sie sie per Drag-and-drop an die gewünschte Stelle.

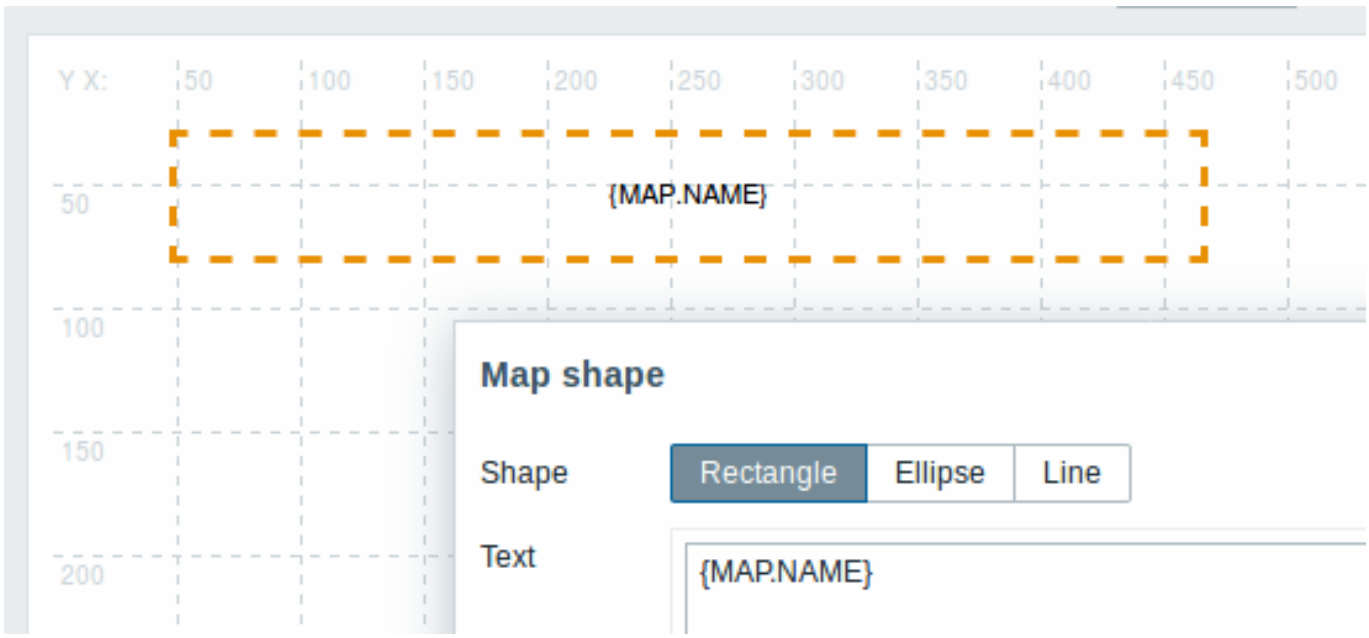
Eine neue Form wird mit Standardfarben hinzugefügt. Wenn Sie auf die Form klicken, wird ein Formular angezeigt, in dem Sie das Aussehen der Form anpassen, Text hinzufügen usw. können.



Um Formen auszuwählen, wählen Sie eine Form aus und halten dann *Strg* gedrückt, um die anderen auszuwählen. Wenn mehrere Formen ausgewählt sind, können gemeinsame Eigenschaften ähnlich wie bei Elementen per Massenaktualisierung geändert werden.

Text kann in die Formen eingefügt werden. Ausdrucks- Makros werden im Text unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max`, wobei Zeit als Parameter verwendet wird (zum Beispiel `{?avg(/host/key,1h)}`).

Um nur Text anzuzeigen, kann die Form unsichtbar gemacht werden, indem der Formrand entfernt wird (wählen Sie 'None' im Feld *Border* aus). Beachten Sie zum Beispiel, dass das Makro {MAP.NAME}, das im obigen Screenshot sichtbar ist, eigentlich eine rechteckige Form mit Text ist, was beim Klicken auf das Makro sichtbar wird:



{MAP.NAME} wird beim Anzeigen der Karte in den konfigurierten Kartennamen aufgelöst.

Wenn Hyperlinks im Text verwendet werden, werden sie beim Anzeigen der Karte anklickbar.

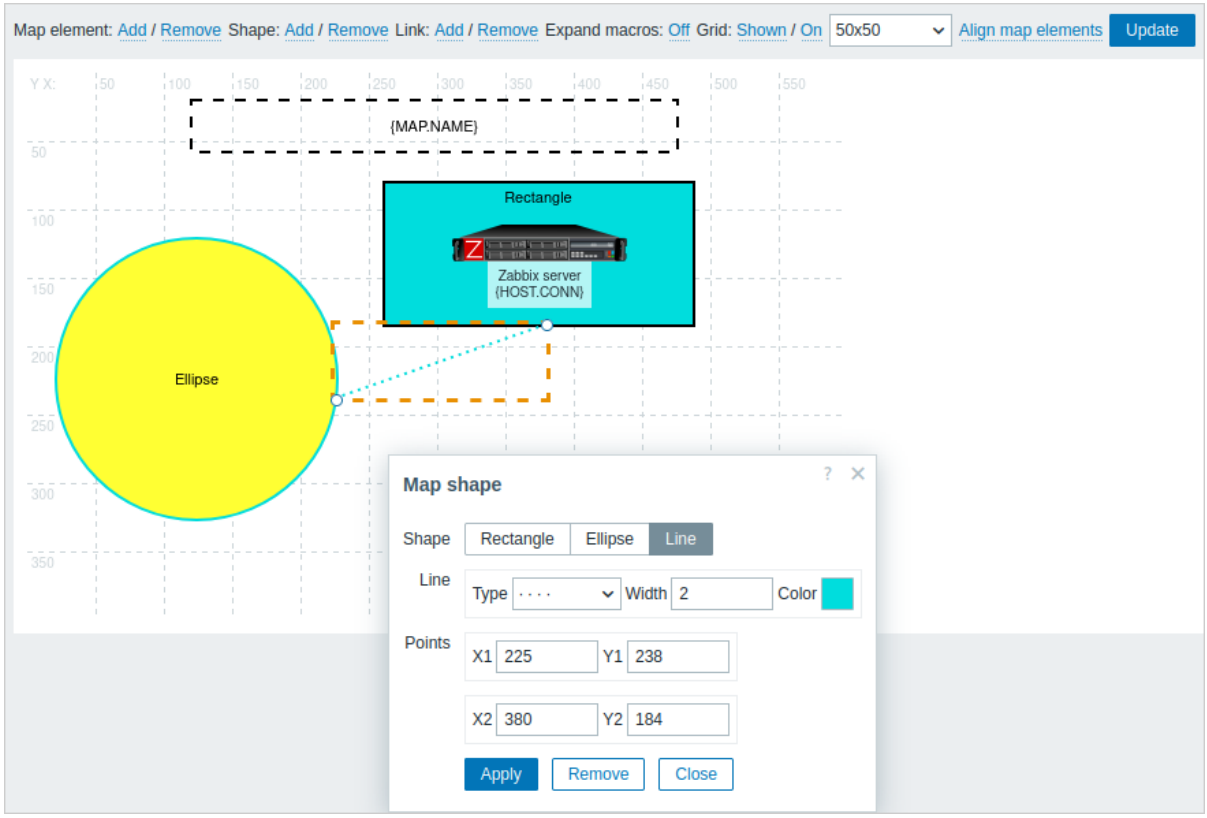
Der Zeilenumbruch für Text ist innerhalb von Formen immer aktiviert. Innerhalb einer Ellipse werden die Zeilen jedoch so um-

brochen, als wäre die Ellipse ein Rechteck. Ein Wortumbruch ist nicht implementiert, daher werden lange Wörter (Wörter, die nicht in die Form passen) nicht umbrochen, sondern maskiert (auf der Kartenbearbeitungsseite) oder abgeschnitten (auf anderen Seiten mit Karten).

Linien hinzufügen

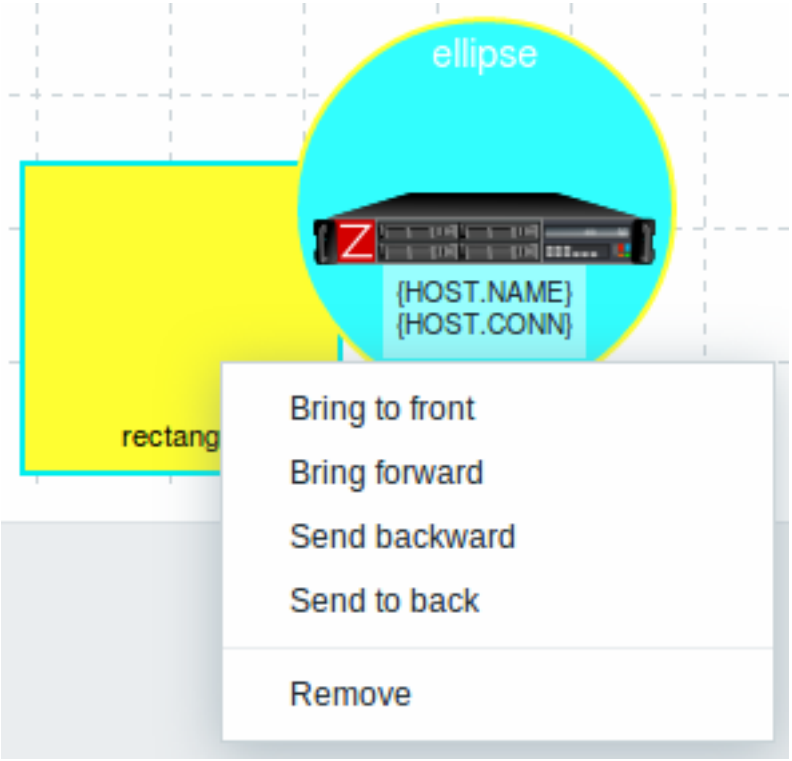
Zusätzlich zu Formen ist es auch möglich, einige Linien hinzuzufügen. Linien können verwendet werden, um Elemente oder Formen in einer Karte zu verknüpfen.

Um eine Linie hinzuzufügen, klicken Sie neben Form auf *Hinzufügen*. Eine neue Form erscheint in der oberen linken Ecke der Karte. Wählen Sie sie aus und klicken Sie im Bearbeitungsformular auf *Linie*, um die Form in eine Linie umzuwandeln. Passen Sie dann die Linieneigenschaften an, z. B. Linientyp, Breite, Farbe usw.



Formen und Linien anordnen

Um eine Form vor eine andere zu bringen (oder umgekehrt), klicken Sie mit der rechten Maustaste auf die Form, um das Menü für Kartenformen zu öffnen.

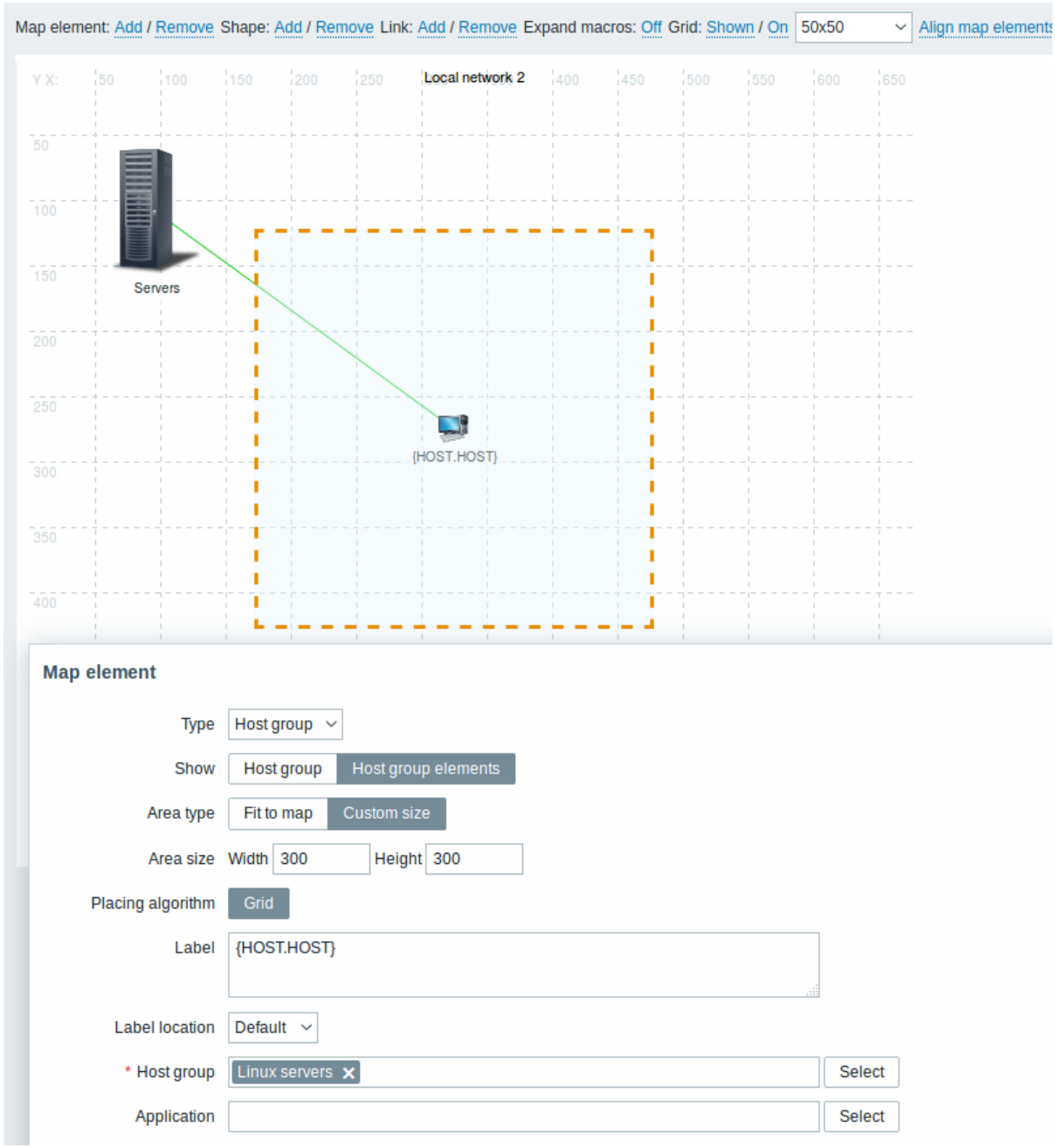


2 Hostgruppen-Elemente

Übersicht

In diesem Abschnitt wird erläutert, wie beim Konfigurieren einer Netzwerkkarte ein Element vom Typ *Host-Gruppe* hinzugefügt wird.

Konfiguration



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

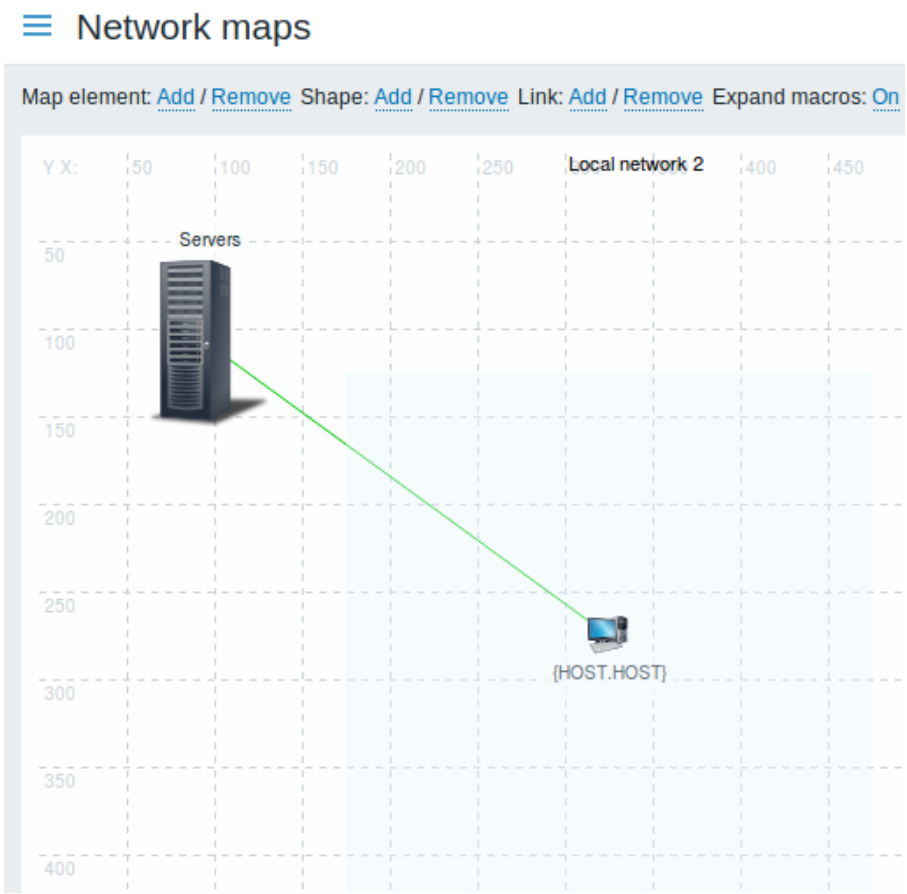
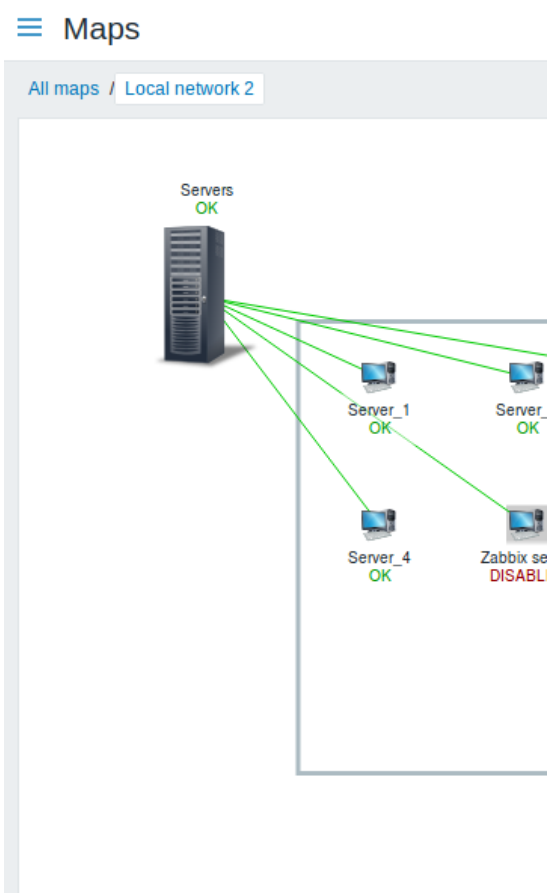
Diese Tabelle enthält Parameter, die typisch für den Elementtyp *Host-Gruppe* sind:

Parameter	Beschreibung
Type	Wählen Sie den Typ des Elements aus: **Host group** - Symbol, das den Status aller Auslöser aller Hosts darstellt, die zur ausgewählten Gruppe gehören.
Show	Anzeigeoptionen: **Host group** - Die Auswahl dieser Option führt zu einem einzelnen Symbol, das entsprechende Informationen über die jeweilige Host-Gruppe anzeigt. **Host group elements** - Die Auswahl dieser Option führt zu mehreren Symbolen, die entsprechende Informationen über jedes einzelne Element (Host) der jeweiligen Host-Gruppe anzeigen.
Area type	Diese Einstellung ist verfügbar, wenn *Show* auf *Host group elements* gesetzt ist: **Fit to map** - Alle Elemente der Host-Gruppe werden gleichmäßig innerhalb der Karte platziert. **Custom size** - Manuelle Festlegung des Kartenbereichs, in dem alle Elemente der Host-Gruppe angezeigt werden.
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 Host-Gruppe.
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 zum Beispiel das Makro {HOST.IP} verwendet wird, zeigt die Kartenbearbeitungsansicht nur das Makro {HOST.IP} selbst an, während die Kartenansicht die eindeutige IP-Adresse jedes Hosts enthält und anzeigt.

Anzeigen von Hostgruppenelementen

Diese Option ist verfügbar, wenn die Anzeigeoption *Hostgruppenelemente* ausgewählt ist.
Wenn Sie *Hostgruppenelemente* als Option *Anzeigen* 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 jeweiligen Hostgruppe enthält:

Bearbeitungsansicht der Karte	Kartenansicht
	

Beachten Sie, wie das Makro {HOST.HOST} verwendet wird.
Bei der Kartenbearbeitung wird der Makroname nicht aufgelöst, während in der Kartenansicht alle eindeutigen Namen der Hosts angezeigt werden.

3 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 ein Datenpunktwert:
  - einen Schwellenwert erreicht (bei numerischen Datenpunkten);
  - mit einem regulären Ausdruck übereinstimmt (bei Text-Datenpunkten).

Wenn Sie eine Verbindung konfigurieren, legen Sie den Standardtyp und die Standardfarbe der Verbindung fest. Durch das Zuweisen von Indikatoren zu einer Verbindung wird es möglich, den Verbindungsstil und die Verbindungsfarbe 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).

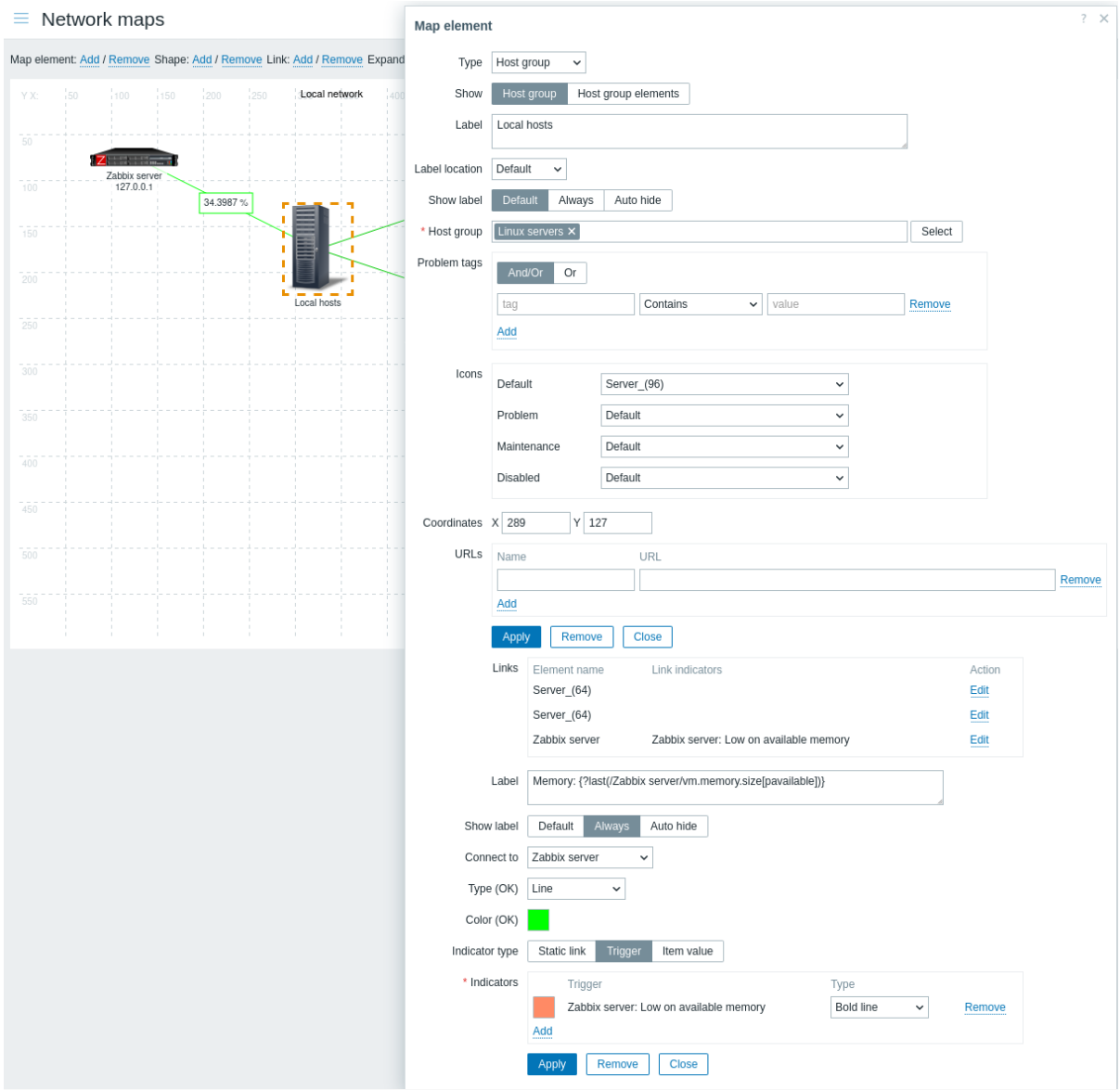
Ebenso kann der Verbindungsstil dies widerspiegeln, wenn ein Datenpunktwert einen angegebenen Schwellenwert erreicht oder mit einem angegebenen regulären Ausdruck übereinstimmt.

Konfiguration

Auslöser

Um Auslöser als Link-Indikatoren zuzuweisen, gehen Sie wie folgt vor:

- Wählen Sie ein Kartenelement aus.
- Klicken Sie im Abschnitt *Links* beim entsprechenden Link auf *Bearbeiten*.
- Wählen Sie *Auslöser* als Indikatortyp aus.
- Klicken Sie im Block *Indikatoren* auf *Hinzufügen* und wählen Sie einen oder mehrere Auslöser aus.

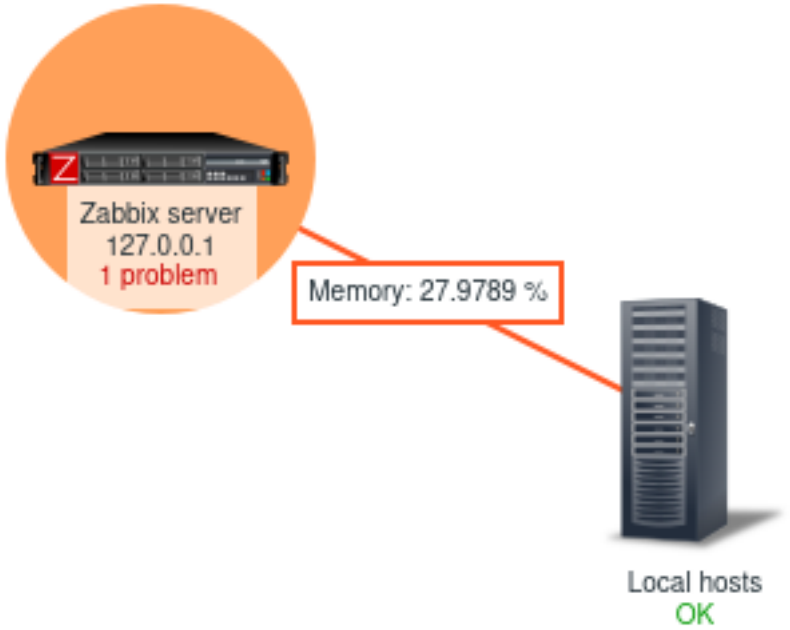


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Hinzugefügte Auslöser sind in der Liste *Indikatoren* sichtbar.

Sie können den Linktyp und die Farbe für jeden Indikator direkt in der Liste festlegen. Klicken Sie anschließend auf *Anwenden*, schließen Sie das Formular und klicken Sie auf *Aktualisieren*, um die Änderungen an der Karte zu speichern.

Unter *Monitoring* → *Karten* werden der jeweilige Linktyp und die jeweilige Farbe angezeigt, wenn der **Auslöser** in einen Problemzustand wechselt.



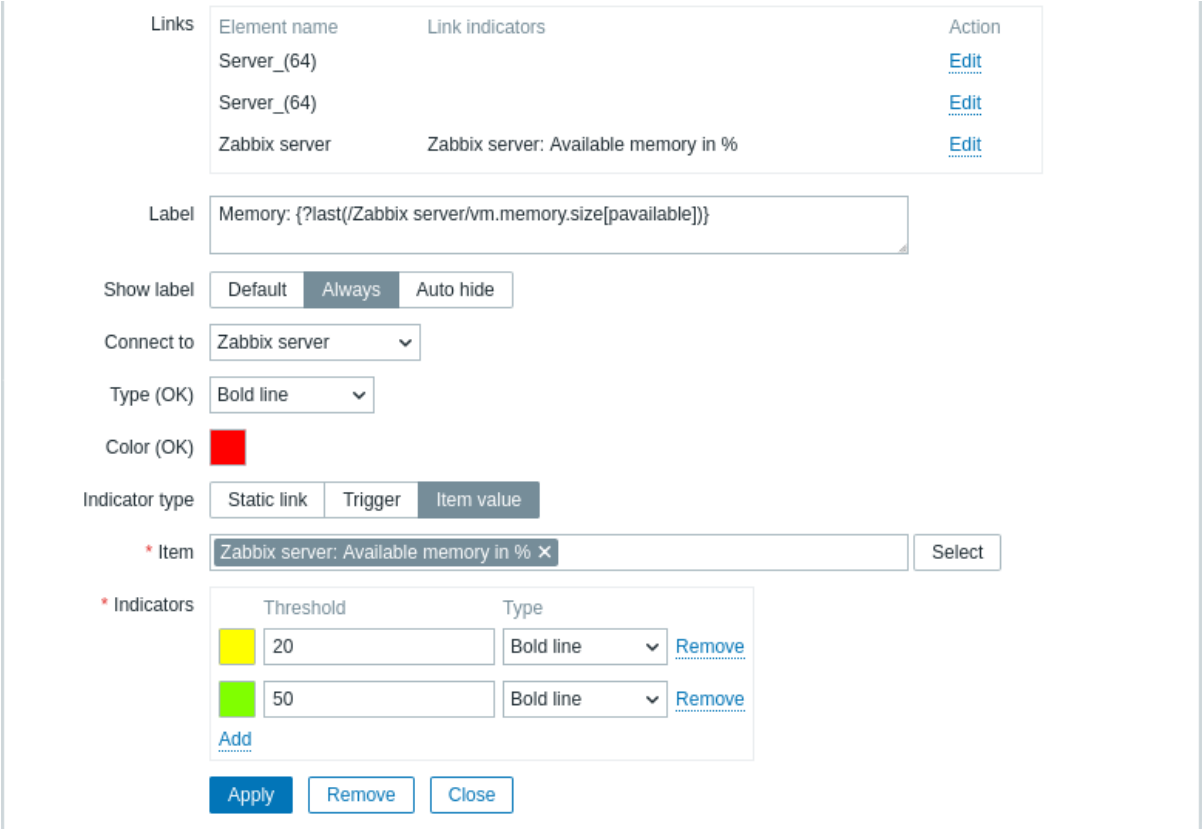
Note:
Wenn mehrere Auslöser in einen Problemzustand wechseln, bestimmt das Problem mit dem höchsten Schweregrad den Linkstil und die Farbe. Wenn mehrere Auslöser mit demselben Schweregrad demselben Karten-Link zugewiesen sind, hat derjenige mit der niedrigsten ID Vorrang. Beachten Sie außerdem Folgendes:

1. Die Einstellungen *Minimale Auslöser-Schwere* und *Unterdrücktes Problem anzeigen* aus der Kartenkonfiguration beeinflussen, welche Probleme berücksichtigt werden.
2. Bei Auslösern mit mehreren Problemen (mehrfache Problemgenerierung) kann jedes Problem einen vom Auslöser-Schweregrad abweichenden Schweregrad haben (manuell geändert), unterschiedliche Tags aufweisen (aufgrund von Makros) und unterdrückt sein.

Datenpunktwerte

Um Datenpunktwerte als Link-Indikatoren zuzuweisen, gehen Sie wie folgt vor:

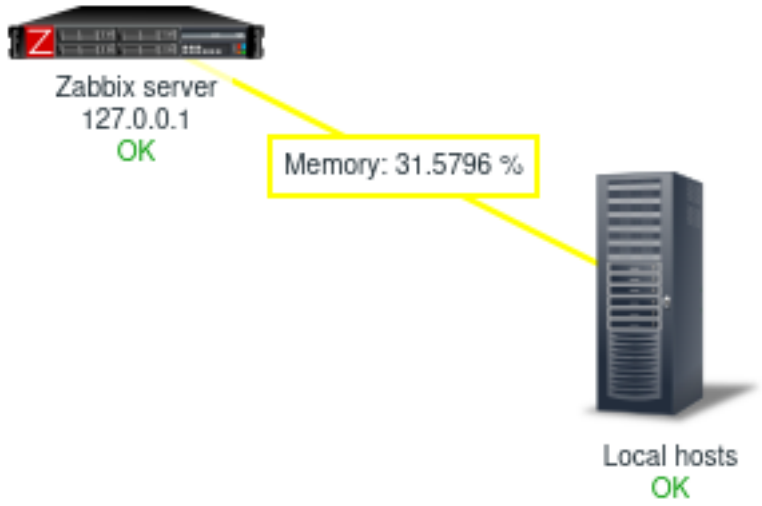
- Wählen Sie ein Kartenelement aus.
- Klicken Sie im Abschnitt *Links* beim entsprechenden Link auf *Bearbeiten*.
- Wählen Sie *Datenpunktwert* als Indikatortyp aus.
- Wählen Sie den Datenpunkt aus.
- Fügen Sie im Block *Indikatoren* einen oder mehrere Schwellenwerte oder Muster für Datenpunktwerte hinzu.



Hinzugefügte Datenpunkt-Schwellenwerte/-Muster sind in der Liste *Indikatoren* sichtbar.

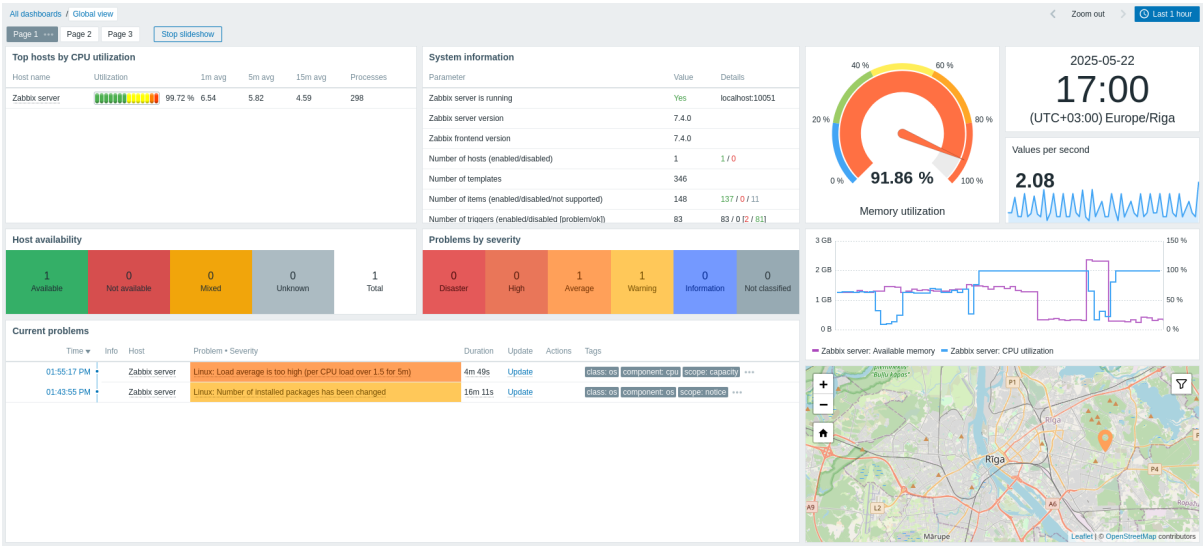
Sie können den Linktyp und die Farbe für jeden Indikator direkt in der Liste festlegen. Klicken Sie anschließend auf *Anwenden*, schließen Sie das Formular und klicken Sie auf *Aktualisieren*, um die Änderungen an der Karte zu speichern.

Unter *Monitoring* → *Karten* werden der jeweilige Linktyp und die jeweilige Farbe angezeigt, wenn der **Datenpunktwert** den festgelegten Schwellenwert erreicht (bei numerischen Datentypen) oder dem Muster eines regulären Ausdrucks entspricht (bei Textdatentypen).



3 Dashboards

Dashboards – sowohl globale Dashboards als auch Host-Dashboards – bieten eine leistungsstarke Plattform zur Visualisierung mit Widgets und Werkzeugen wie modernen Graphen, Karten, Diashows und mehr.



8 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.

9 Vorlagen sofort einsatzbereit

Übersicht

Zabbix bietet eine wachsende Anzahl vorkonfigurierter Vorlagen, um die Einrichtung von Überwachungszielen zu vereinfachen und zu beschleunigen.

Alle standardmäßig enthaltenen Vorlagen sind unter *Datensammlung* > *Vorlagen* verfügbar.

Beim Upgrade von Zabbix werden vorhandene Vorlagen nicht automatisch aktualisiert, um das Überschreiben benutzerdefinierter Änderungen zu vermeiden. Informationen zum Upgrade vorhandener Vorlagen oder zum Hinzufügen neuer Vorlagen finden Sie unter Upgrade von Vorlagen.

Bitte verwenden Sie die Seitenleiste, um Informationen zu bestimmten Vorlagentypen und Betriebsanforderungen aufzurufen.

Siehe auch:

- Verknüpfen einer Vorlage
- Bekannte Probleme bei Vorlagen

Upgrade von Vorlagen

So aktualisieren Sie eine einzelne Vorlage oder fügen nach einem Zabbix-Upgrade eine neue hinzu:

1. Gehen Sie zum Zabbix Git repository.
2. Wählen Sie in der Branch-Auswahl (unter der Bezeichnung *Source*) den Branch aus, der zu Ihrer Zabbix-Version passt (z. B. *master* für die neueste Entwicklungsversion).
3. Öffnen 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 im Zabbix Frontend zu *Data collection* > *Templates* und importieren Sie die lokale Datei.

Sie können auch alle Vorlagen aktualisieren und neue gleichzeitig hinzufügen:

1. Laden Sie das Vorlagenpaket abhängig von Ihrem Webserver herunter: `templates_1M_8.0.7z` (Nginx) oder `templates_2M_8.0.7z` (Apache).
2. Entpacken Sie das Paket. Es enthält YAML-Dateien (jeweils mit mehreren Vorlagen) und einen Index (`index-1M_80.md`), der die in jeder YAML-Datei enthaltenen Vorlagen auflistet.
3. Gehen Sie im Zabbix Frontend zu *Data collection* > *Templates* und importieren Sie die YAML-Dateien nacheinander.

Alternativ können Sie alle YAML-Dateien auf einmal mit dem Skript `import_templates.sh` importieren. Das Skript erfordert die Dienstprogramme `curl` und `jq` und verwendet die Zabbix-API-Methode `configuration.import`. Beim Ausführen des Skripts geben Sie das Verzeichnis mit den YAML-Dateien sowie Ihre Zabbix-API-URL und Ihr Authentifizierungstoken an.

Warning:
Das Skript `import_templates.sh` überschreibt alle vorhandenen Vorlagen, ohne Möglichkeit, Vorlagenänderungen zu überprüfen oder rückgängig zu machen.

Beispiel:

```
./import_templates.sh /tmp/templates_2M_8.0 https://example.com/zabbix/api_jsonrpc.php d8d6b5c78ee2a8333db
```

Kompatibilität von Vorlagen mit dem Host Wizard

Der Host Wizard ist mit allen sofort einsatzbereiten Vorlagen kompatibel.

Sie können auch Ihre benutzerdefinierten Vorlagen mit dem Host Wizard kompatibel machen:

1. Gehen Sie im Zabbix Frontend zu *Datenerfassung* > *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 optional je nach Bedarf das Vorlagenelement `readme` oder `config` hinzu.
3. Gehen Sie im Zabbix Frontend zu *Datenerfassung* > *Vorlagen* und importieren Sie die aktualisierte Vorlagendatei.

1 Betrieb von Zabbix-Agent-Vorlagen

Schritte, um den korrekten Betrieb von Vorlagen sicherzustellen, die Metriken mit Zabbix agent erfassen:

1. Stellen Sie sicher, dass Zabbix agent auf dem Host installiert ist. Stellen Sie bei aktiven Prüfungen außerdem sicher, dass die Adresse des Zabbix Server/Proxy zum Parameter 'ServerActive' in der Konfigurationsdatei des Agent hinzugefügt wurde.
2. Verknüpfen Sie die Vorlage mit einem Ziel-Host (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 Vorlagen-Makros 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 (zugänglich durch Klicken auf einen Vorlagennamen).

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
- 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
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

2 Betrieb von Zabbix-Agent-2-Vorlagen

Schritte zur Sicherstellung des korrekten Betriebs von Vorlagen, die Metriken mit Zabbix agent 2 erfassen:

1. Stellen Sie sicher, dass Agent 2 auf dem Host installiert ist und dass die installierte Version das erforderliche Plugin enthält. In einigen Fällen müssen Sie Agent 2 möglicherweise 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 Vorlagenmakros an. Beachten Sie, dass Benutzermakros verwendet werden können, um Konfigurationsparameter zu überschreiben.

4. Konfigurieren Sie die überwachte Instanz so, dass sie die gemeinsame Nutzung von Daten mit Zabbix erlaubt.

Attention:
Zabbix-Agent-2-Vorlagen arbeiten in Verbindung mit den Plugins. Während die grundlegende Konfiguration einfach durch Anpassen von Benutzermakros vorgenommen werden kann, lässt sich eine weitergehende Anpassung durch Konfiguration des Plugins selbst erreichen. Wenn ein Plugin beispielsweise benannte Sitzungen unterstützt, ist es möglich, mehrere gleichartige Entitäten (z. B. MySQL1 und MySQL2) zu überwachen, indem für jede Entität in der Konfigurationsdatei eine benannte Sitzung mit eigener URI, eigenem Benutzernamen und eigenem 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

3 Betrieb von HTTP-Vorlagen

Schritte, um den korrekten Betrieb von Vorlagen sicherzustellen, die Metriken mit dem HTTP-Agenten erfassen:

1. Erstellen Sie in Zabbix einen Host, der dem Überwachungsziel entspricht. Geben Sie die IP-Adresse/den DNS-Namen des Überwachungsziels als Haupt-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 (zugänglich durch Klicken auf einen Vorlagennamen).

Die folgenden Vorlagen sind verfügbar:

- Acronis Cyber Protect Cloud über HTTP
- Apache über HTTP
- Asterisk über HTTP
- AWS über HTTP
- AWS Cost Explorer über HTTP
- AWS EC2 über HTTP
- AWS ECS Cluster über HTTP
- AWS ECS Serverless Cluster über HTTP
- AWS ELB Application Load Balancer über HTTP
- AWS ELB Network Load Balancer über HTTP
- AWS Lambda über HTTP
- AWS RDS instance über HTTP
- AWS S3 bucket über HTTP
- Azure über HTTP
- Cisco Meraki organization über HTTP
- Cisco SD-WAN über HTTP
- Cisco Secure Firewall Threat Defense über HTTP
- ClickHouse über HTTP
- Cloudflare über HTTP
- CockroachDB über HTTP
- Control-M enterprise manager über HTTP
- Control-M server über HTTP
- Cradlepoint NCM v2 über HTTP
- DELL PowerEdge R720 über HTTP
- DELL PowerEdge R740 über HTTP
- DELL PowerEdge R820 über HTTP
- DELL PowerEdge R840 über HTTP
- Domain RDAP über HTTP
- Elasticsearch Cluster über HTTP
- Envoy Proxy über HTTP
- Etcd über HTTP
- FortiGate über HTTP
- GitHub repository über HTTP
- GitLab über HTTP
- Google Cloud Platform (GCP) über HTTP
- Hadoop über HTTP
- HAProxy über HTTP
- HashiCorp Consul Cluster über HTTP
- HashiCorp Consul Node über HTTP
- HashiCorp Nomad über HTTP
- HashiCorp Vault über HTTP
- Hikvision camera über HTTP
- HPE iLO über HTTP
- HPE MSA 2040 Storage über HTTP
- HPE MSA 2060 Storage über HTTP
- HPE Primera über HTTP
- HPE Synergy über HTTP

- InfluxDB über HTTP
- Jenkins über HTTP
- Kubernetes API server über HTTP
- Kubernetes cluster state über HTTP
- Kubernetes Controller manager über HTTP
- Kubernetes kubelet über HTTP
- Kubernetes nodes über HTTP
- Kubernetes Scheduler über HTTP
- MantisBT über HTTP
- Microsoft 365 reports über HTTP
- Microsoft SharePoint über HTTP
- NetApp AFF A700 über HTTP
- Nextcloud über HTTP
- NGINX über HTTP
- NGINX Plus über HTTP
- Nutanix Prism Element über HTTP
- OpenAI Platform über HTTP
- OpenStack über HTTP
- OpenWeatherMap über HTTP
- Oracle Cloud über HTTP
- Palo Alto PA-440 über HTTP
- PHP-FPM über HTTP
- Proxmox VE über HTTP
- Pure Storage FlashArray v1 and v2 über HTTP
- RabbitMQ cluster über HTTP
- Ribbon SBC Edge über HTTP
- Ribbon SBC SWe CE über HTTP
- Ribbon SBC SWe Core über HTTP
- TiDB über HTTP
- TiDB PD über HTTP
- TiDB TiKV über HTTP
- Travis CI über HTTP
- Veeam Backup Enterprise Manager über HTTP
- Veeam Backup and Replication über HTTP
- VeloCloud SD-WAN über HTTP
- VeloCloud SD-WAN Edge über HTTP
- YugabyteDB über HTTP
- ZooKeeper über HTTP

4 Betrieb von IPMI-Vorlagen

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, ist in der README-Datei der Vorlage verfügbar (zugänglich durch Klicken auf einen Vorlagennamen).

Verfügbare Vorlage:

- Chassis by IPMI

5 Betrieb von JMX-Vorlagen

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. Für den Host sollte die JMX-Schnittstelle eingerichtet sein. 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 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 (zugänglich durch Klicken auf einen 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

6 Betrieb von ODBC-Vorlagen

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 Vorlagen-Makros 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 (zugänglich durch Klicken auf einen Vorlagennamen).

Die folgenden Vorlagen sind verfügbar:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

7 Standardisierte Vorlagen für Netzwerkgeräte

Übersicht

Um das Monitoring von Netzwerkgeräten wie Switches und Routern bereitzustellen, haben wir zwei sogenannte Modelle erstellt: für das Netzwerkgerät selbst (im Wesentlichen sein Chassis) und für die Netzwerkschnittstelle.

Vorlagen für viele Familien von Netzwerkgeräten werden bereitgestellt. Alle Vorlagen decken (sofern es möglich ist, diese Datenpunkte vom Gerät abzurufen) Folgendes ab:

- Überwachung von Chassis-Fehlern (Netzteile, Lüfter und Temperatur, Gesamtstatus)
- Überwachung der Chassis-Performance (CPU- und Speicher-Datenpunkte)
- Erfassung des Chassis-Inventars (Seriennummern, Modellname, Firmware- Version)
- Überwachung von Netzwerkschnittstellen mit IF-MIB und EtherLike-MIB (Schnittstellenstatus, Schnittstellen-Verkehrslast, Duplex-Status für Ethernet)

Falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren.

Wenn Sie die neuen sofort einsatzbereiten Vorlagen importieren, sollten Sie möglicherweise auch den globalen regulären Ausdruck `@Network interfaces for discovery` auf Folgendes aktualisieren:

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

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 Genera-tion Firewall	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Zertifiziert
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

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
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
HP Enterprise Switch SNMP	HP	HP Enter-prise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Zertifiziert
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500V5/5600/6800/18500/18800 with 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
Mellanox SNMP	Mellanox	Mellanox-Infiniband-Geräte	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
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-Wireless-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

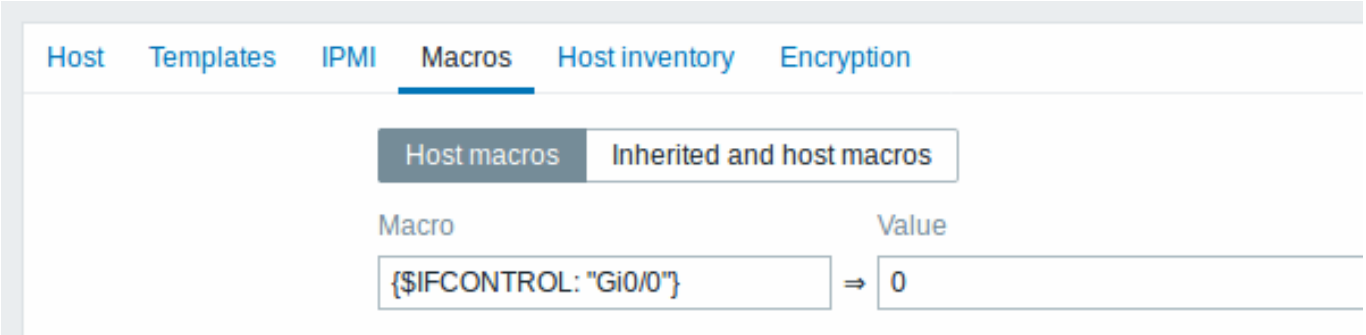
Vorlagen wurden unter Berücksichtigung der folgenden Punkte entwickelt:

- 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:

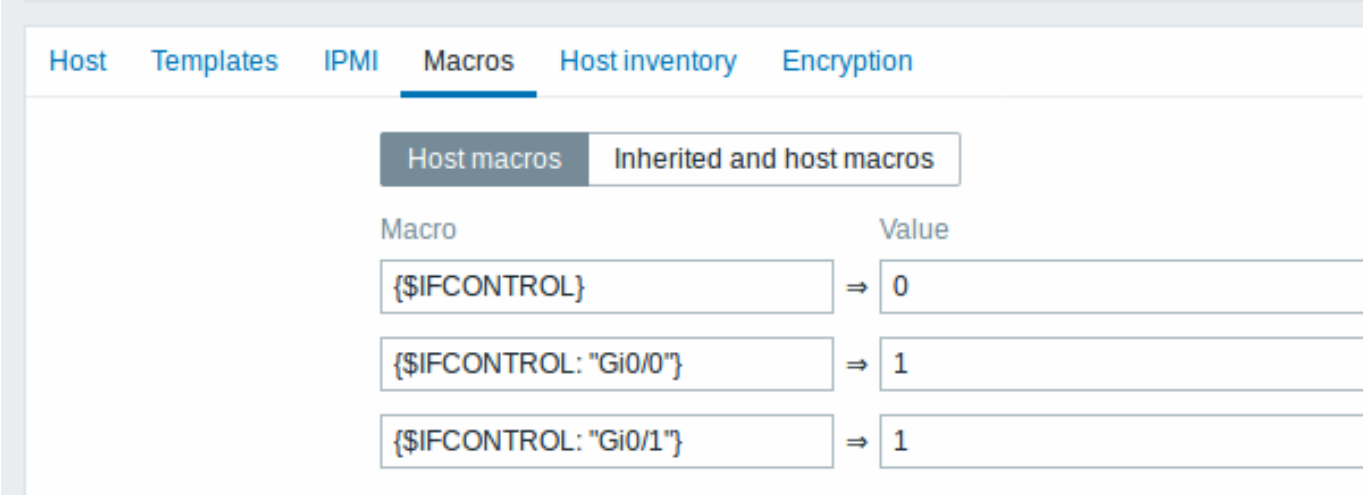
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 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.

8 Betrieb der VMware-Vorlage

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

10 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.

1 Medientypen

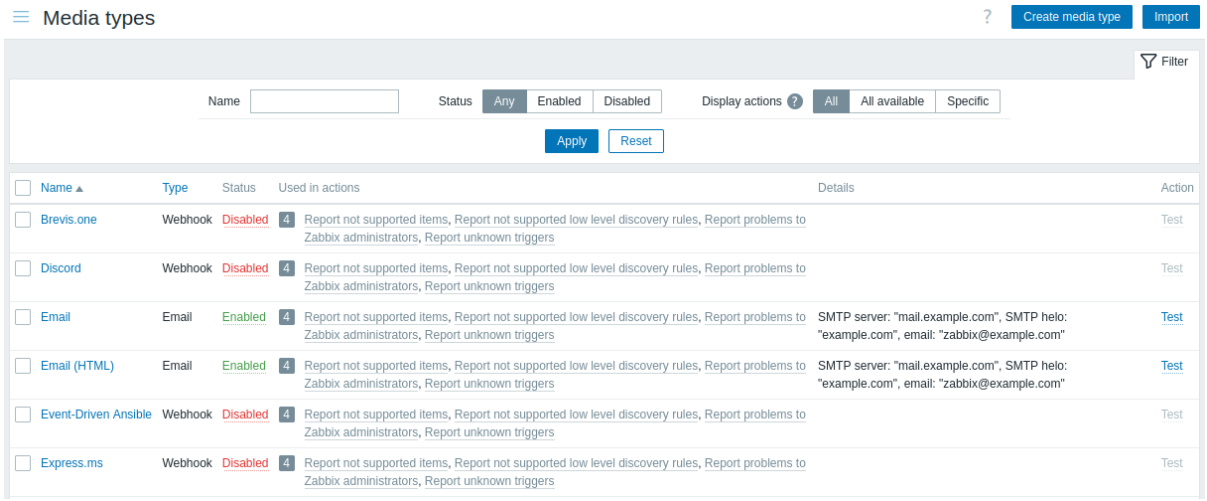
Übersicht

Medientypen sind die Übertragungskanäle, die zum Senden von Benachrichtigungen und Warnmeldungen aus Zabbix verwendet werden.

Medientypen unterstützen die folgenden Zustellmethoden:

- E-Mail
- SMS
- Benutzerdefiniertes Skript
- webhook

Medientypen werden unter *Warnmeldungen* > *Medientypen* verwaltet. Einige Medientypen sind im Standarddatensatz bereits vordefiniert.
Sie müssen nur deren Parameter feinabstimmen, damit sie funktionieren.



Um zu sehen, wie Medientypen in den Benachrichtigungsprozess passen, betrachten wir die drei Voraussetzungen für die Zustellung von Benachrichtigungen aus Zabbix an Endbenutzer:

1. Es muss eine Aktions-Operation definiert sein, die Benachrichtigungen sendet
2. Es muss ein funktionierender Medientyp definiert sein (z. B. *E-Mail*, die Warnmeldungen über SMTP sendet)
3. Zustellungsdetails auf Benutzerebene (wie E-Mail-Adressen, Telefonnummern usw.) müssen in Benutzermedien definiert sein

Testen von Medientypen

Um zu testen, ob ein konfigurierter Medientyp funktioniert, klicken Sie in der Medientypliste 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 darauf, dass der Server die Ergebnisse zurückgibt. Das Testen von Medientypen hat standardmäßig ein Timeout von 65 Sekunden (konfigurierbar unter *Administration* > *General* > *Timeouts*).

Siehe auch Testdetails für:

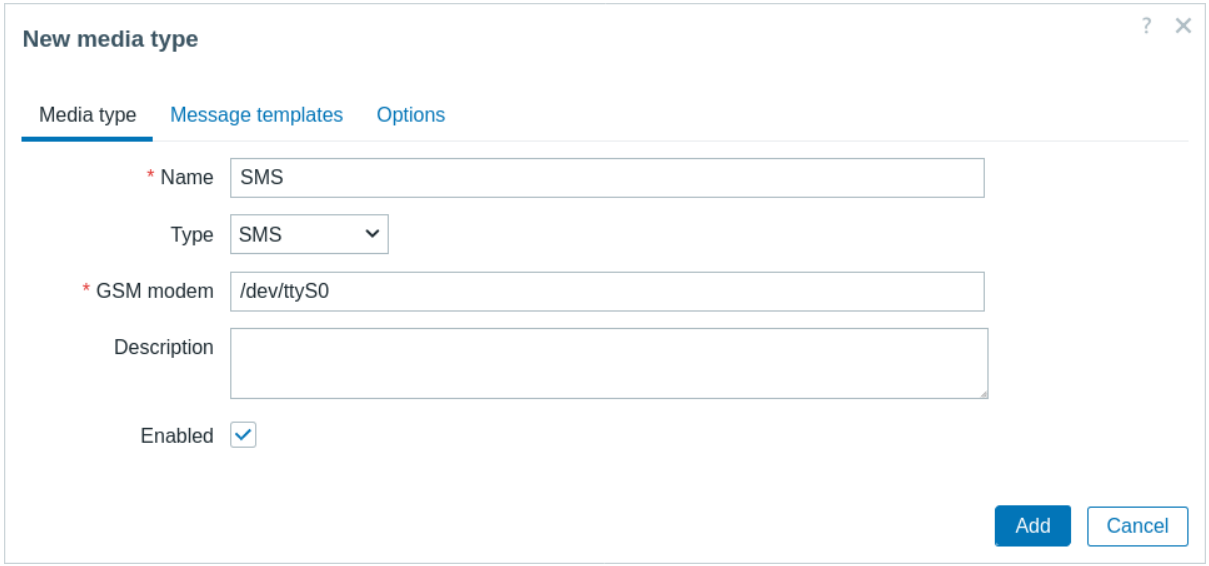
- E-Mail
- webhook
- Skript

Konfiguration

So erstellen Sie einen Medientyp im Zabbix Frontend:

- Gehen Sie zu: *Benachrichtigungen* > *Medientypen*
- Klicken Sie auf *Medientyp erstellen*
- Geben Sie die Parameter des Medientyps im Formular ein

Einige Parameter sind für alle Zustellungsmethoden **gemeinsam**.

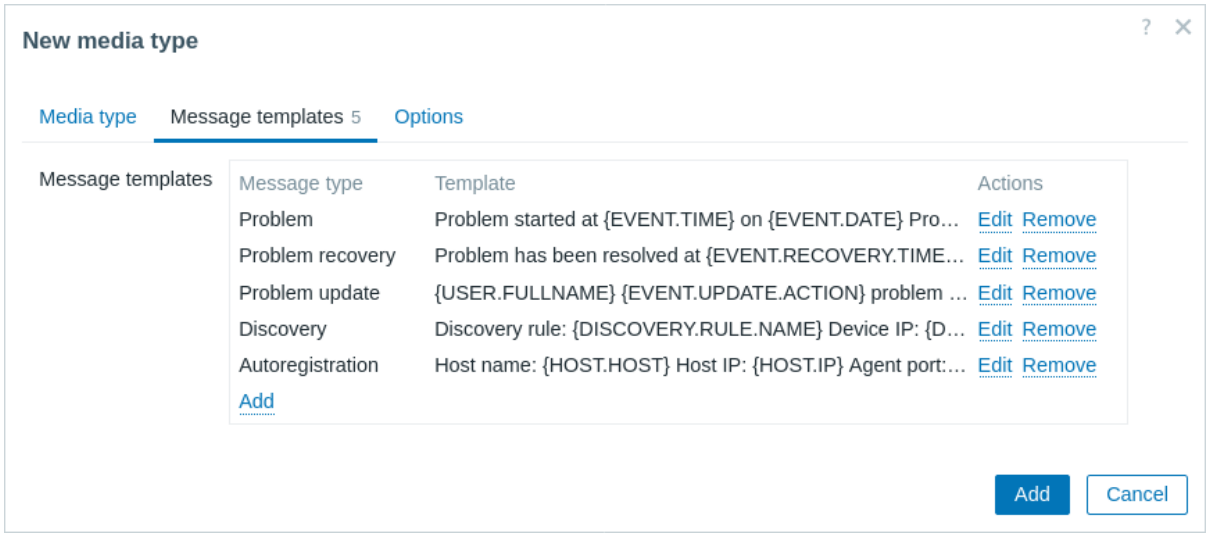


Parameter	Beschreibung
Name	Name des Medientyps.
Type	Wählen Sie die Zustellungsmethode für den Medientyp aus.
Description	Geben Sie eine Beschreibung für den Medientyp ein.
Enabled	Aktivieren Sie das Kontrollkästchen, um den Medientyp zu aktivieren.

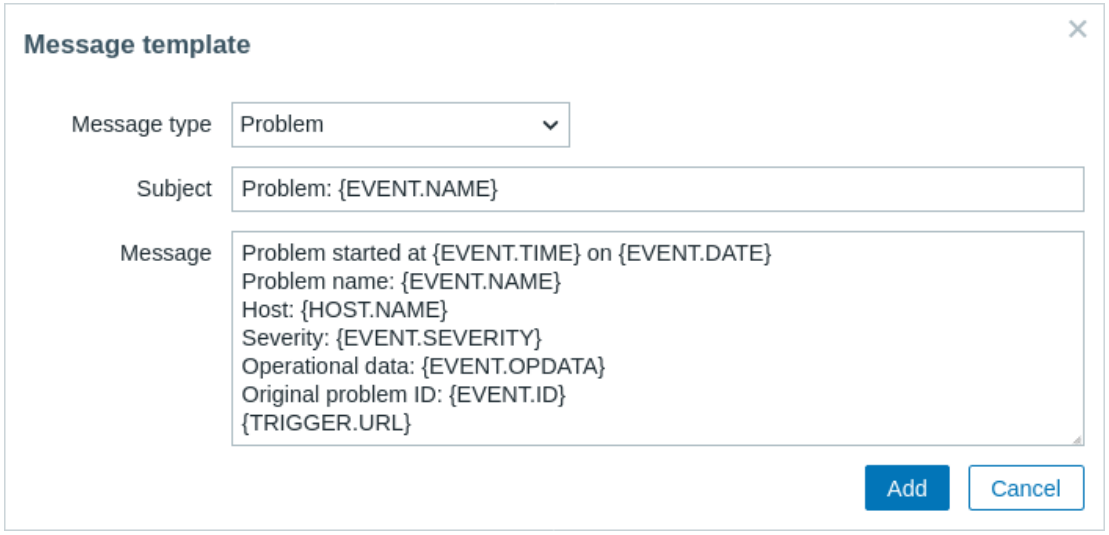
Methodenspezifische Parameter finden Sie auf den Seiten E-Mail, SMS, benutzerdefiniertes Alert-Skript oder webhook.

Nachrichtenvorlagen

Die Registerkarte **Nachrichtenvorlagen** enthält Standardnachrichten pro Ereignistyp (Problem, Problembehebung, Discovery 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 Medientyp SMS nicht verfügbar.
Nachricht	Die Standardnachricht. Sie ist abhängig vom Datenbanktyp auf eine bestimmte Anzahl von Zeichen begrenzt (weitere Informationen finden Sie unter Nachrichten senden). Die Nachricht kann unterstützte Makros enthalten. In Problem- und Problemaktualisierungsnachrichten werden Ausdrucksmakros unterstützt (zum Beispiel `{?avg(/host/key,1h)}`).

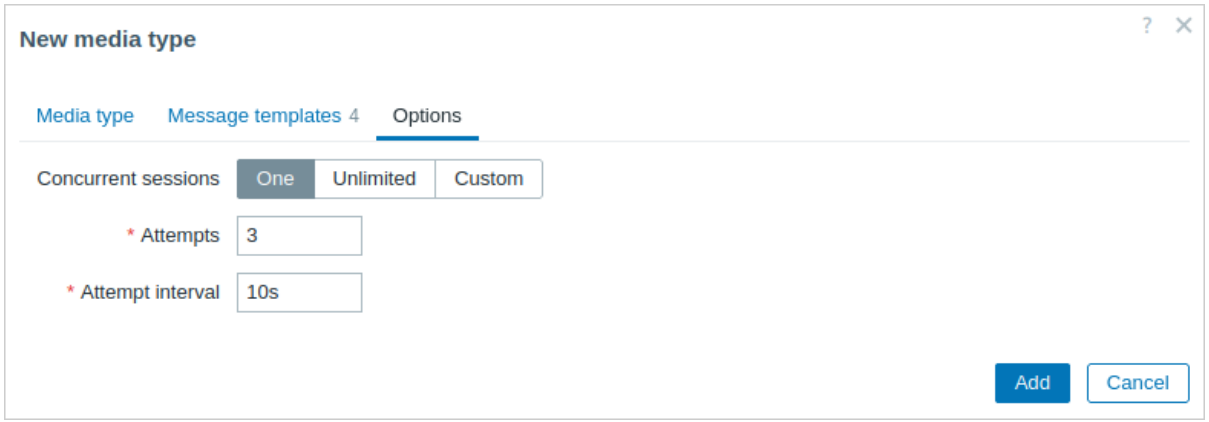
Beachten Sie, dass Standardnachrichten durch benutzerdefinierte Nachrichten überschrieben werden, falls diese in Aktionsoperationen definiert sind.

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“ beim Senden von Problembenachrichtigungen fehl, wenn die Problemnachricht für den Pushover-webhook nicht definiert ist.

Optionen

Die Registerkarte **Optionen** enthält Einstellungen für die Alarmverarbeitung. Dieselbe Gruppe von Optionen kann für jeden Medientyp konfiguriert werden.

Alle Medientypen werden parallel verarbeitet. Während die maximale Anzahl gleichzeitiger Sitzungen pro Medientyp konfigurierbar ist, kann die Gesamtzahl der Alerter-Prozesse auf dem Server nur durch den StartAlerters-Parameter begrenzt werden. Von einem Auslöser erzeugte Alarme werden sequenziell verarbeitet. Daher können mehrere Benachrichtigungen nur dann gleichzeitig verarbeitet werden, wenn sie von mehreren Auslösern erzeugt werden.



Parameter	Beschreibung
Gleichzeitige Sitzungen	Wählen Sie die Anzahl paralleler Alerter-Sitzungen für den Medientyp aus: **Eine** - eine Sitzung **Unbegrenzt** - unbegrenzte Anzahl von Sitzungen **Benutzerdefiniert** - wählen Sie eine benutzerdefinierte Anzahl von Sitzungen aus Unbegrenzte/hohe Werte bedeuten mehr parallele Sitzungen und eine höhere Kapazität für das 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 verfügbar sind, werden die verbleibenden Benachrichtigungen in eine Warteschlange gestellt; sie gehen nicht verloren.
Versuche	Anzahl der Versuche zum Senden einer Benachrichtigung. Es können bis zu 100 Versuche angegeben werden; der Standardwert ist „3“. Wenn „1“ angegeben ist, sendet Zabbix die Benachrichtigung nur einmal und versucht es bei einem Sendefehler nicht erneut.
Intervall zwischen Versuchen	Häufigkeit in Sekunden (0-3600), mit der erneut versucht wird, eine Benachrichtigung zu senden, falls das Senden fehlgeschlagen ist. Wenn „0“ angegeben ist, versucht Zabbix es sofort erneut. Zeitsuffixe werden unterstützt, z. B. 5s, 3m, 1h.

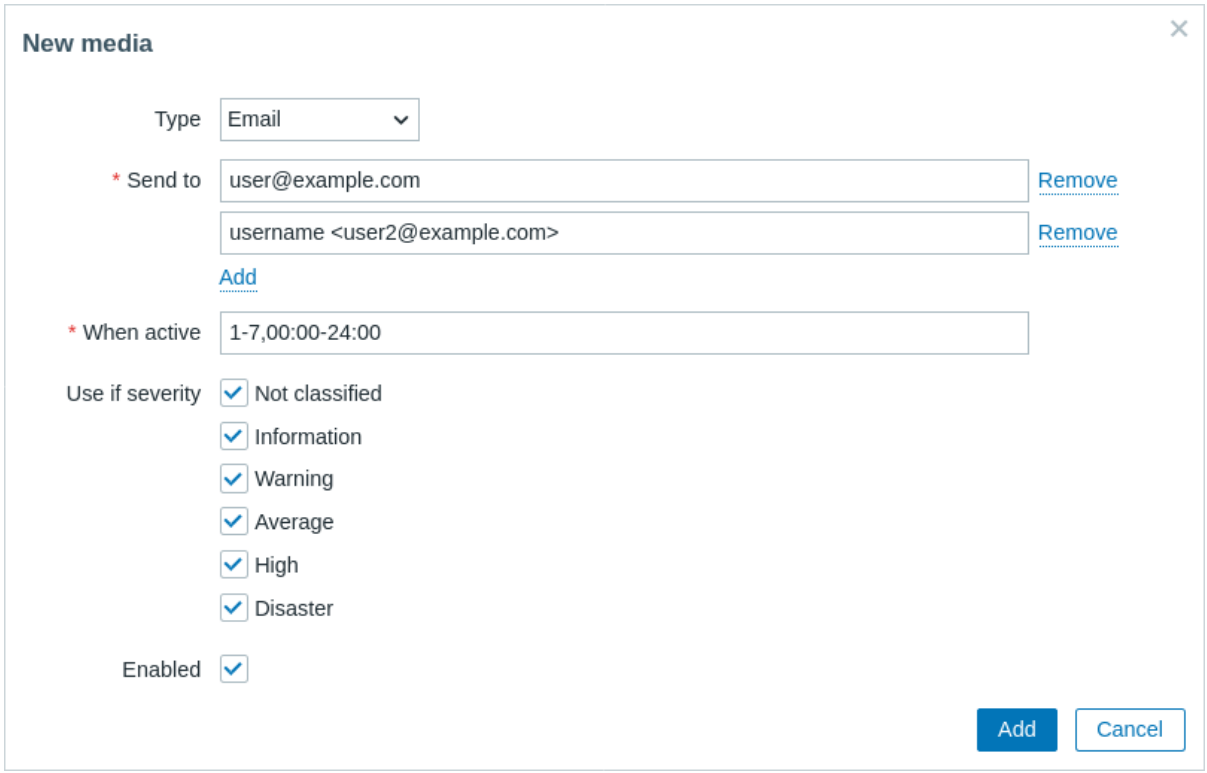
Benutzermedien

Während Medientypen definieren, **wie** eine Benachrichtigung gesendet wird, definieren Benutzermedien, **wohin** die Benachrichtigung gesendet werden muss.

Benutzermedien (z. B. E-Mail-Adresse, webhook-Benutzer-ID usw.) müssen unabhängig von der Zustellmethode im Benutzerprofil definiert werden. Eine Aktion, die Nachrichten mit webhook *X* an den Benutzer *Admin* sendet, kann nicht zugestellt werden, wenn die Zustellungsdetails für webhook *X* nicht im Benutzerprofil von Admin definiert sind.

So definieren Sie Benutzermedien:

- Gehen Sie zu *Benutzer* > *Benutzer* und öffnen Sie das Formular mit den Benutzereigenschaften (oder gehen Sie in Ihrem eigenen Benutzerprofil zu *Benutzereinstellungen* > *Benachrichtigungen*)
- Klicken Sie auf der Registerkarte „Medien“ auf *Hinzufügen*



Parameter	Beschreibung
Type	Die Dropdown-Liste enthält die Namen der aktivierten Medientypen. Beachten Sie, dass beim Bearbeiten eines Mediums eines deaktivierten Medientyps der Typ rot angezeigt wird.

Parameter	Beschreibung
Send to	Geben Sie die Kontaktinformationen ein, an die Nachrichten gesendet werden sollen. Für den Medientyp E-Mail können durch Klicken auf die Schaltfläche Add unter dem Adressfeld mehrere Adressen hinzugefügt werden. In diesem Fall werden Benachrichtigungen an alle aufgeführten Adressen gesendet. Adressbeispiele finden Sie in der Beschreibung des Parameters *Email* für den Medientyp email.
When active	Sie können den Zeitraum einschränken, in dem Nachrichten gesendet werden, z. B. nur auf Arbeitstage (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 sich von der Systemzeitzone unterscheidet, muss diese Einschränkung möglicherweise entsprechend angepasst werden, damit keine wichtigen Nachrichten verpasst werden. Eine Beschreibung des Formats finden Sie auf der Seite Zeitraumspezifikation. Benutzermakros werden unterstützt.
Use if severity	Markieren Sie die Kontrollkästchen der Auslöser-Schweregrade, für die Sie Benachrichtigungen erhalten möchten. Beachten Sie, dass der Standardschweregrad („Nicht klassifiziert“) **aktiviert sein muss**, wenn Sie Benachrichtigungen für Nicht-Auslöser-Ereignisse erhalten möchten. Nach dem Speichern werden die ausgewählten Auslöser-Schweregrade in den entsprechenden Schweregradfarben angezeigt, während nicht ausgewählte ausgegraut dargestellt werden.
Status	Status des Benutzermediums. **Aktiviert** - wird verwendet. **Deaktiviert** - wird nicht verwendet.

1 E-Mail

Übersicht

Um E-Mail als Zustellkanal für Nachrichten zu konfigurieren, müssen Sie E-Mail als Medientyp konfigurieren und Benutzern bestimmte Adressen zuweisen.

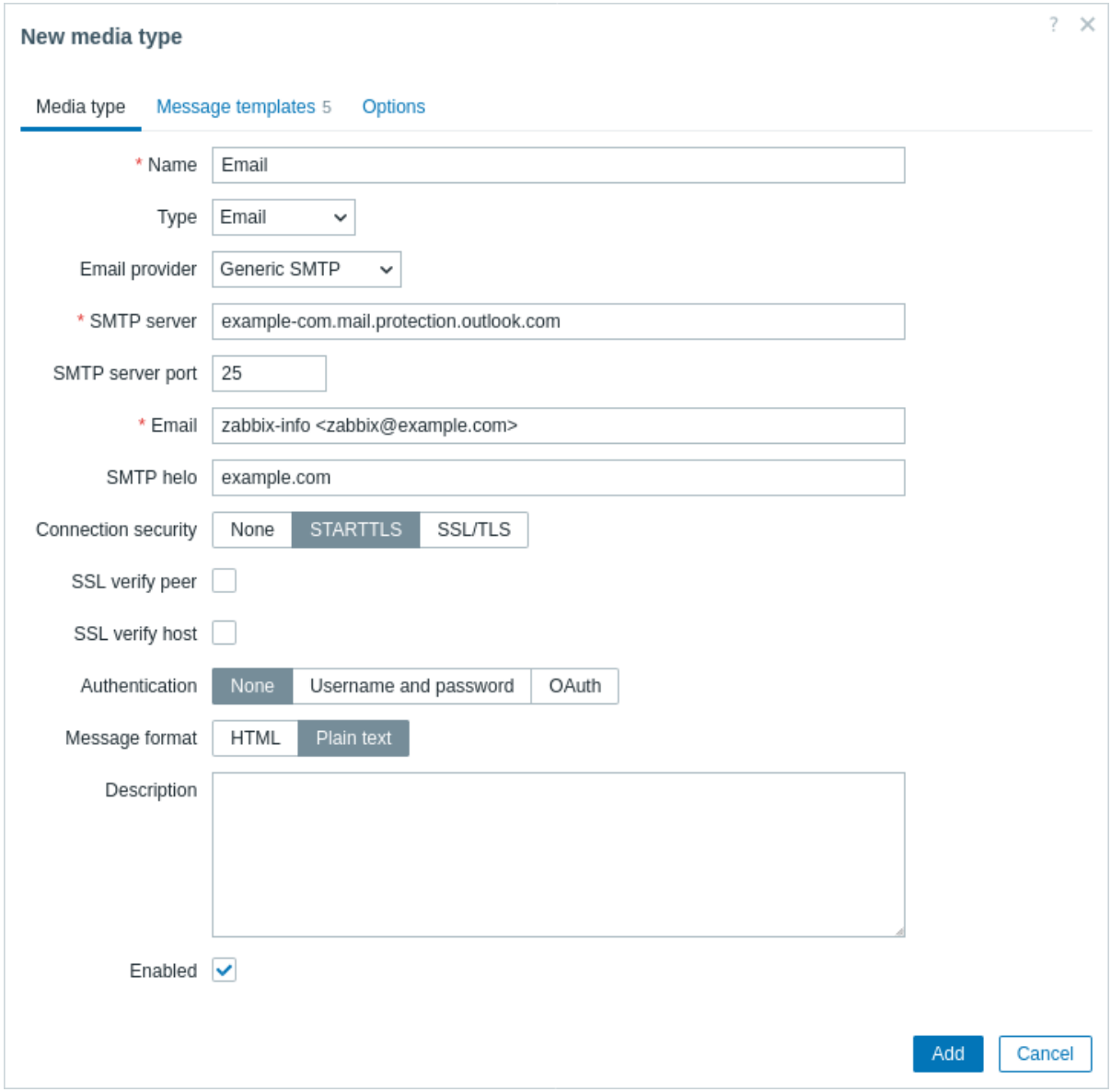
> **Note:**
> Mehrere Benachrichtigungen für ein einzelnes Ereignis werden im selben E-Mail-Thread zusammengefasst.

Konfiguration

So konfigurieren Sie E-Mail als Medientyp:

1. Gehen Sie zu *Benachrichtigungen* > *Medientypen*.
2. Klicken Sie auf *Medientyp erstellen* (oder klicken Sie in der Liste der vordefinierten Medientypen auf *E-Mail*).

Die Registerkarte **Medientyp** enthält allgemeine Attribute des Medientyps:



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Warning:
Das Passwort wird beim Exportieren der E-Mail-Medientyp-Konfiguration im Klartext exportiert.

Die folgenden Parameter sind spezifisch für den E-Mail-Medientyp:

Parameter	Beschreibung
E-Mail-Anbieter	Wählen Sie den E-Mail-Anbieter aus: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* oder *Office365 relay*. Wenn Sie die Gmail-/Office365-bezogenen Optionen auswählen, müssen Sie nur die Absender-E-Mail-Adresse und das Passwort angeben; Optionen wie *SMTP-Server*, *SMTP-Server-Port*, *SMTP helo* und *Verbindungssicherheit* werden von Zabbix automatisch ausgefüllt. Siehe auch: Automatisierte Gmail-/Office365-Medientypen.
SMTP-Server	Legen Sie einen SMTP-Server für die Verarbeitung ausgehender Nachrichten fest. Dieses Feld ist verfügbar, wenn *Generic SMTP* als E-Mail-Anbieter ausgewählt ist.
SMTP-Server-Port	Legen Sie den Port des SMTP-Servers für die Verarbeitung ausgehender Nachrichten fest. Dieses Feld ist verfügbar, wenn *Generic SMTP* als E-Mail-Anbieter ausgewählt ist.

Parameter	Beschreibung
E-Mail	Die hier eingegebene Adresse wird als **Von**-Adresse für die gesendeten Nachrichten verwendet. Das Hinzufügen eines Anzeigenamens des Absenders (wie „Zabbix_info“ in *Zabbix_info <zabbix@company.com>* im Screenshot oben) zusammen mit der eigentlichen E-Mail-Adresse wird unterstützt. Für Anzeigenamen in Zabbix-E-Mails gelten im Vergleich zu den durch RFC 5322 erlaubten Werten einige Einschränkungen, wie die folgenden Beispiele zeigen: Gültige Beispiele: *zabbix@company.com* (nur E-Mail-Adresse, spitze Klammern sind nicht erforderlich) *Zabbix_info <zabbix@company.com>* (Anzeigename und E-Mail-Adresse in spitzen Klammern) *∑Ω-monitoring <zabbix@company.com>* (UTF-8-Zeichen im Anzeigenamen) Ungültige Beispiele: *Zabbix HQ zabbix@company.com* (Anzeigename vorhanden, aber keine spitzen Klammern um die E-Mail-Adresse) *”Zabbix\@\<H(comment)Q\>” <zabbix@company.com>* (obwohl gemäß RFC 5322 gültig, werden maskierte Zeichenpaare und Kommentare in Zabbix-E-Mails nicht unterstützt)
SMTP helo	Legen Sie einen korrekten SMTP-helo-Wert fest, normalerweise einen Domainnamen. Wenn leer, wird der Domainname der E-Mail gesendet (d. h. der Teil nach @ im Feld *E-Mail*). Wenn der Domainname nicht ermittelt werden kann, wird eine Warnung auf Debug-Ebene protokolliert und der Hostname des Servers wird als Domain für den HELO-Befehl gesendet. Dieses Feld ist verfügbar, wenn *Generic SMTP* als E-Mail-Anbieter ausgewählt ist.
Verbindungssicherheit	Wählen Sie die Stufe der Verbindungssicherheit aus: **Keine** - die Option CURLOPT_USE_SSL wird nicht verwendet **STARTTLS** - die Option CURLOPT_USE_SSL wird mit dem Wert CURLUSESSL_ALL verwendet **SSL/TLS** - die Verwendung von CURLOPT_USE_SSL ist optional
SSL Peer verifizieren	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des SMTP-Servers zu verifizieren. Der Wert der Server-Konfigurationsdirektive ”SSLCALocation” sollte zur Zertifikatsvalidierung in CURLOPT_CAPATH eingetragen werden. Dies setzt die cURL-Option CURLOPT_SSL_VERIFYPEER.
SSL Host verifizieren	Aktivieren Sie das Kontrollkästchen, um zu verifizieren, dass das Feld *Common Name* oder das Feld *Subject Alternate Name* des SMTP-Serverzertifikats übereinstimmt. Dies setzt die cURL-Option CURLOPT_SSL_VERIFYHOST.
Authentifizierung	Wählen Sie die Stufe der Authentifizierung aus: **Keine** - es werden keine cURL-Optionen gesetzt **Benutzername und Passwort** - impliziert ”AUTH=*” und überlässt die Wahl des Authentifizierungsmechanismus cURL **OAuth** - OAuth-Authentifizierung OAuth-Authentifizierung wird für den E-Mail-Anbieter *Office365 relay* nicht unterstützt.
Benutzername	Benutzername für die Authentifizierung. Dies setzt den Wert von CURLOPT_USERNAME. Benutzermakros werden unterstützt.
Passwort	Passwort für die Authentifizierung. Dies setzt den Wert von CURLOPT_PASSWORD. Benutzermakros werden unterstützt. Der hier eingegebene Wert wird beim Exportieren der Medientyp-Konfiguration unverändert exportiert. Um zu vermeiden, dass vertrauliche Zugangsdaten in exportierten Dateien offengelegt werden, verwenden Sie statt eines Klartext-Passworts ein Benutzermakro (zum Beispiel `{$EMAIL_SMTP_PASSWORD}`) — beim Export wird dann der Makroverweis exportiert, während der geheime Wert auf dem Zielsystem (erneut) konfiguriert werden sollte.
OAuth-Tokens	Klicken Sie auf *Konfigurieren*, um Parameter zum Abrufen von OAuth-Tokens in einem neuen Fenster einzurichten. Dieses Feld ist nur verfügbar, wenn im Feld *Authentifizierung* „OAuth“ ausgewählt ist.
Nachrichtenformat	Wählen Sie das Nachrichtenformat aus: **HTML** - als HTML senden **Klartext** - als Klartext senden

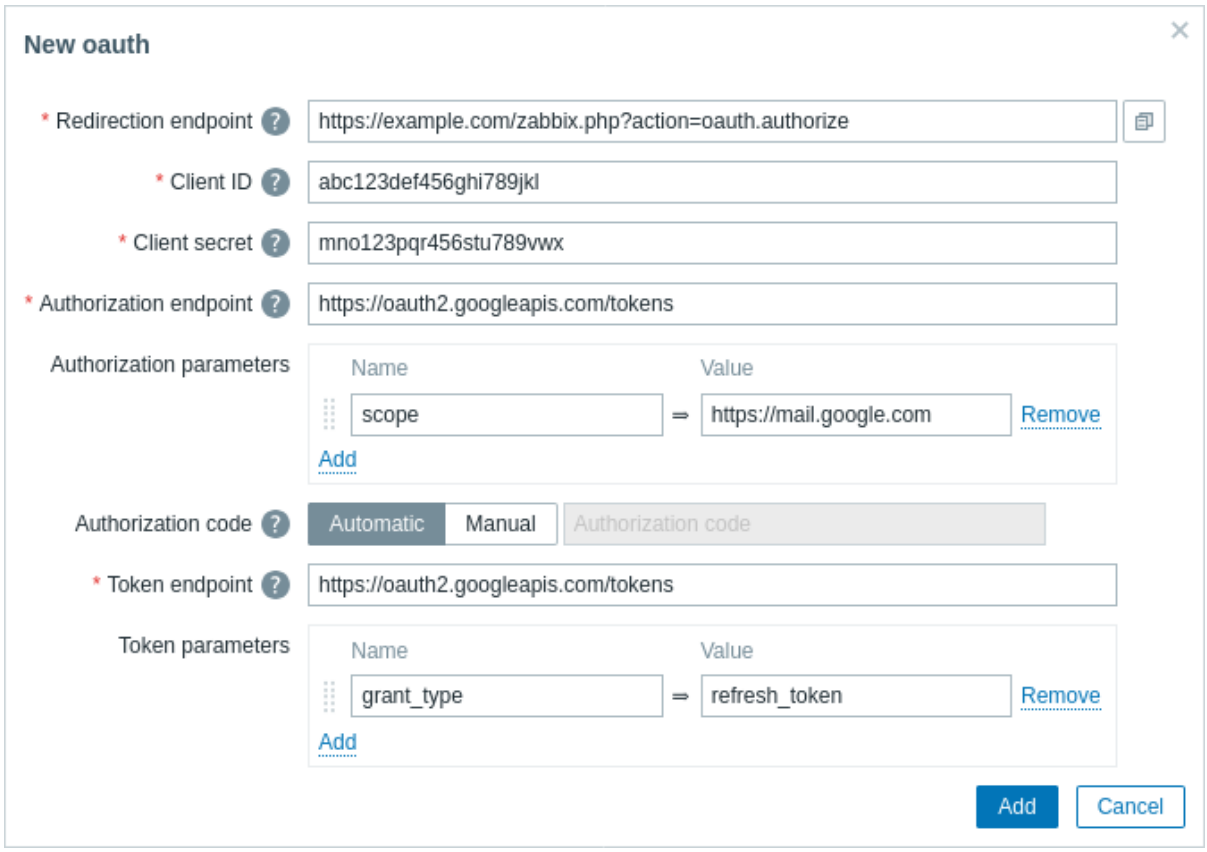
Attention:
Um SMTP-Authentifizierung zu aktivieren, muss der Zabbix Server mit der Kompilierungs-Option `--with-libcurl` kompiliert werden, die libcurl erfordert (Versionsdetails finden Sie in den Anforderungen für server).

Siehe auch allgemeine Medientyp-Parameter für Details zur Konfiguration von Standardnachrichten und Optionen zur Alarmverar-

beitung.

OAuth-Token

Die folgenden Parameter sind zum Abrufen von OAuth-Token erforderlich:



Parameter	Beschreibung
Umleitungsendpunkt	Geben Sie die URL des Zabbix Frontends ein, zu der der OAuth-Dienst nach der OAuth-Autorisierung zurückleitet (im Format `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Bei einem neuen E-Mail-Medientyp wird dieser automatisch gesetzt, wenn die Frontend-URL definiert ist.
Client-ID	Geben Sie die eindeutige Kennung der Client-Anwendung ein, die im OAuth-Autorisierungsserver registriert ist.
Client-Secret	Geben Sie das private Geheimnis der Client-Anwendung ein, die im OAuth-Autorisierungsserver registriert ist.
Autorisierungsendpunkt	Geben Sie die URL des OAuth-Autorisierungsservers für die Anforderung der Benutzerautorisierung ein.
Autorisierungsparameter	Geben Sie Parameter für den Autorisierungsendpunkt ein.
Autorisierungscode	Geben Sie den Autorisierungscode ein: **Automatisch** - der Code wird automatisch über eine Umleitungsseite abgerufen **Manuell** - geben Sie den Code manuell ein, wenn der automatische Abruf fehlschlägt
Token-Endpunkt	Geben Sie die URL des OAuth-Autorisierungsservers ein, um den Autorisierungscode gegen Zugriffs- und Aktualisierungs-Token auszutauschen.
Token-Parameter	Geben Sie Parameter für das Zugriffstoken ein.

Note:
Das Abrufen von OAuth-Token verfügt für die E-Mail-Anbieter **Gmail**, **Gmail relay** und **Office365** über automatisierte Funktionen. Es ist nur erforderlich, Werte für die Parameter *Umleitungsendpunkt*, *Client-ID* und *Client-Secret* anzugeben. Zabbix füllt die anderen erforderlichen Werte automatisch aus (siehe Standardwerte für OAuth-URLs 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.

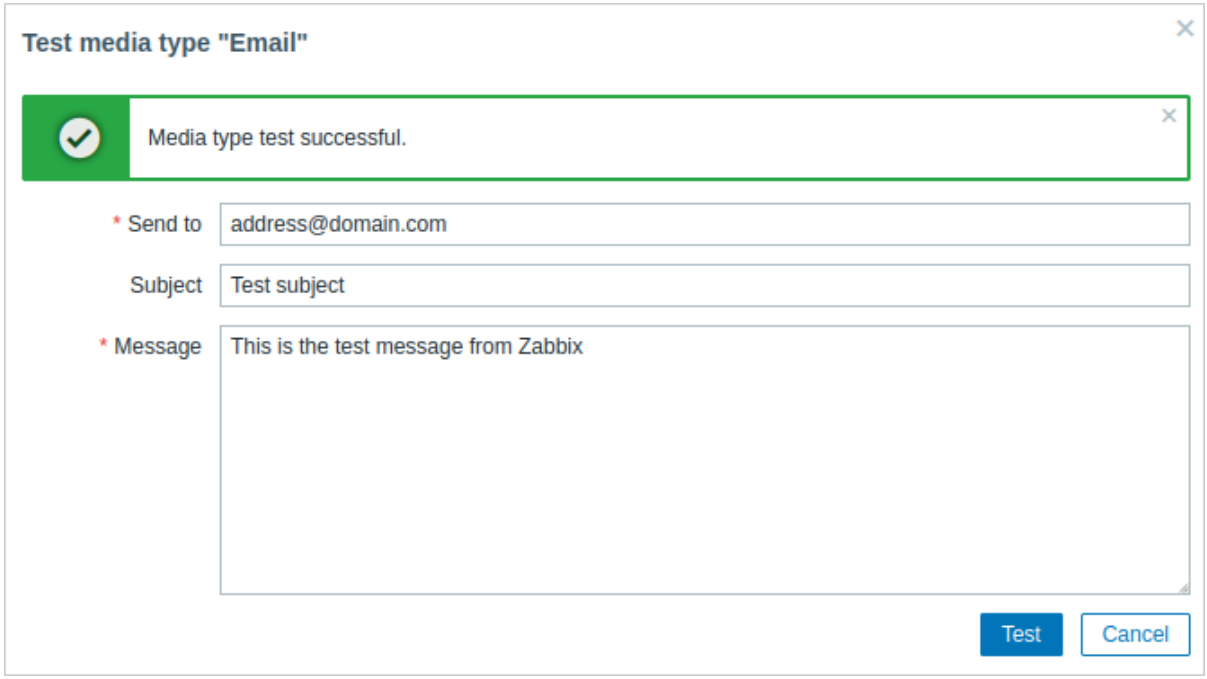
2. Der Benutzer autorisiert Zabbix im OAuth-Dienst.
3. Der OAuth-Dienst leitet den Benutzer zur Zabbix-Frontend-Aktion `oauth.authorize` mit dem Autorisierungscode und dem Scope-Wert weiter.
4. Als Antwort sendet Zabbix eine Anfrage an den *Token endpoint*, um den Autorisierungscode gegen Zugriffs- und Aktualisierungs-Token auszutauschen.

Testen

So testen Sie, ob ein konfigurierter E-Mail-Medientyp korrekt funktioniert:

1. Suchen Sie die entsprechende 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 in *Send to*, den Nachrichtentext und optional einen Betreff ein.
4. Klicken Sie auf *Test*, um eine Testnachricht zu senden.

Eine Meldung über Erfolg oder Fehlschlag des Tests wird im selben Fenster angezeigt:



Benutzermedien

Sobald der E-Mail-Medientyp konfiguriert ist, gehen Sie zum Abschnitt *Benutzer* > *Benutzer* und bearbeiten Sie das Benutzerprofil, um dem Benutzer E-Mail-Medien zuzuweisen. Die Schritte zum Einrichten von Benutzermedien, die für alle Medientypen gleich sind, werden auf der Seite Medientypen beschrieben.

1 Automatisierte Gmail-/Office365-Medientypen

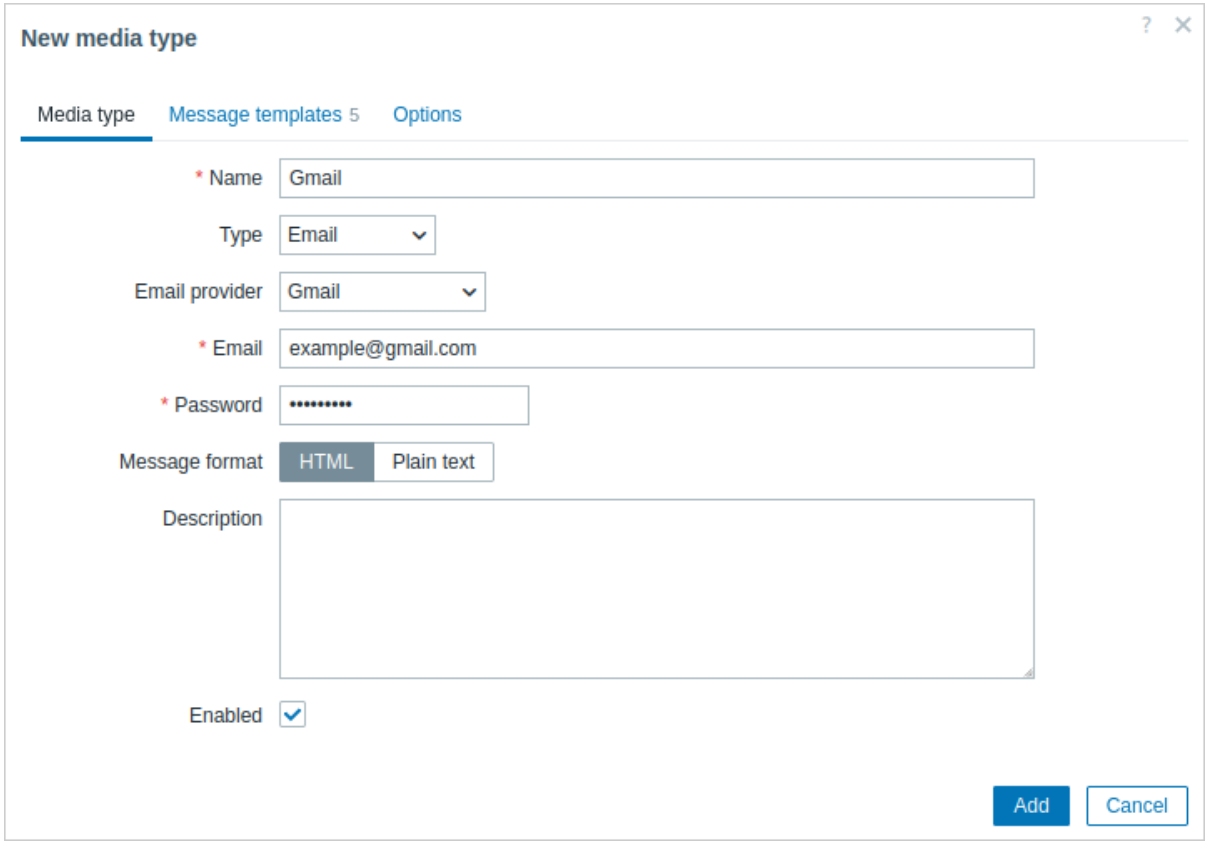
Übersicht

Gmail- oder Office365-Benutzer können von automatisierten Funktionen in der Medientyp-Konfiguration profitieren.

Konfiguration

Das Feld *E-Mail-Anbieter* in der Konfiguration des E-Mail-Medientyps ermöglicht die Auswahl vorkonfigurierter Optionen für Gmail und Office 365.

Bei Auswahl der zu Gmail/Office365 gehörenden Optionen müssen nur die E-Mail-Adresse und das Passwort des Absenders angegeben werden, um einen funktionsfähigen Medientyp zu erstellen.



Sobald E-Mail-Adresse und Passwort angegeben wurden, kann Zabbix automatisch alle erforderlichen Einstellungen für Gmail/Office365-Medientypen mit den tatsächlichen/empfohlenen Werten ausfüllen, d. h. *SMTP-Server*, *SMTP-Server-Port*, *SMTP helo* und *Verbindungssicherheit*. Aufgrund dieser Automatisierung werden diese Felder gar nicht angezeigt. Es ist jedoch möglich, die SMTP-Server- und E-Mail-Details in der Medientyp-Liste einzusehen (siehe die Spalte *Details*).

Beachten Sie außerdem:

- Für die Relay-Optionen ist kein Passwort erforderlich.
- Bei Office365-Relay wird der Domainname 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/authori`	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.

2 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 Zustellungskanal für Nachrichten zu konfigurieren, müssen Sie außerdem SMS als Medientyp konfigurieren und die entsprechenden Telefonnummern für die Benutzer eingeben.

Konfiguration

So konfigurieren Sie SMS als Medientyp:

- Gehen Sie zu *Benachrichtigungen* → *Medientypen*
- Klicken Sie auf *Medientyp erstellen* (oder klicken Sie in der Liste der vordefinierten Medientypen auf *SMS*).

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 Server-Parameters SMSDevices validiert (falls angegeben).

Weitere Informationen zur Konfiguration von Standardnachrichten und Optionen für die Alarmverarbeitung finden Sie unter allgemeine Parameter für Medientypen. Beachten Sie, dass die parallele Verarbeitung beim Versand von SMS-Benachrichtigungen nicht möglich ist.

Benutzermedien

Sobald der Medientyp SMS konfiguriert ist, gehen Sie zum Abschnitt *Benutzer* → *Benutzer* und bearbeiten Sie das Benutzerprofil, um dem Benutzer SMS als Medium zuzuweisen. Die Schritte zum Einrichten von Benutzermedien, die für alle Medientypen gleich sind, werden auf der Seite Medientypen beschrieben.

3 Benutzerdefinierte Alarmierungsskripte

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 Benachrichtigungsskripte 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 Benachrichtigungsskript:

```
#####!/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 sich von **0** unterscheidet, wird als Fehler bei der Befehlsausführung betrachtet. In solchen Fällen versucht Zabbix, die fehlgeschlagene Ausführung zu wiederholen.

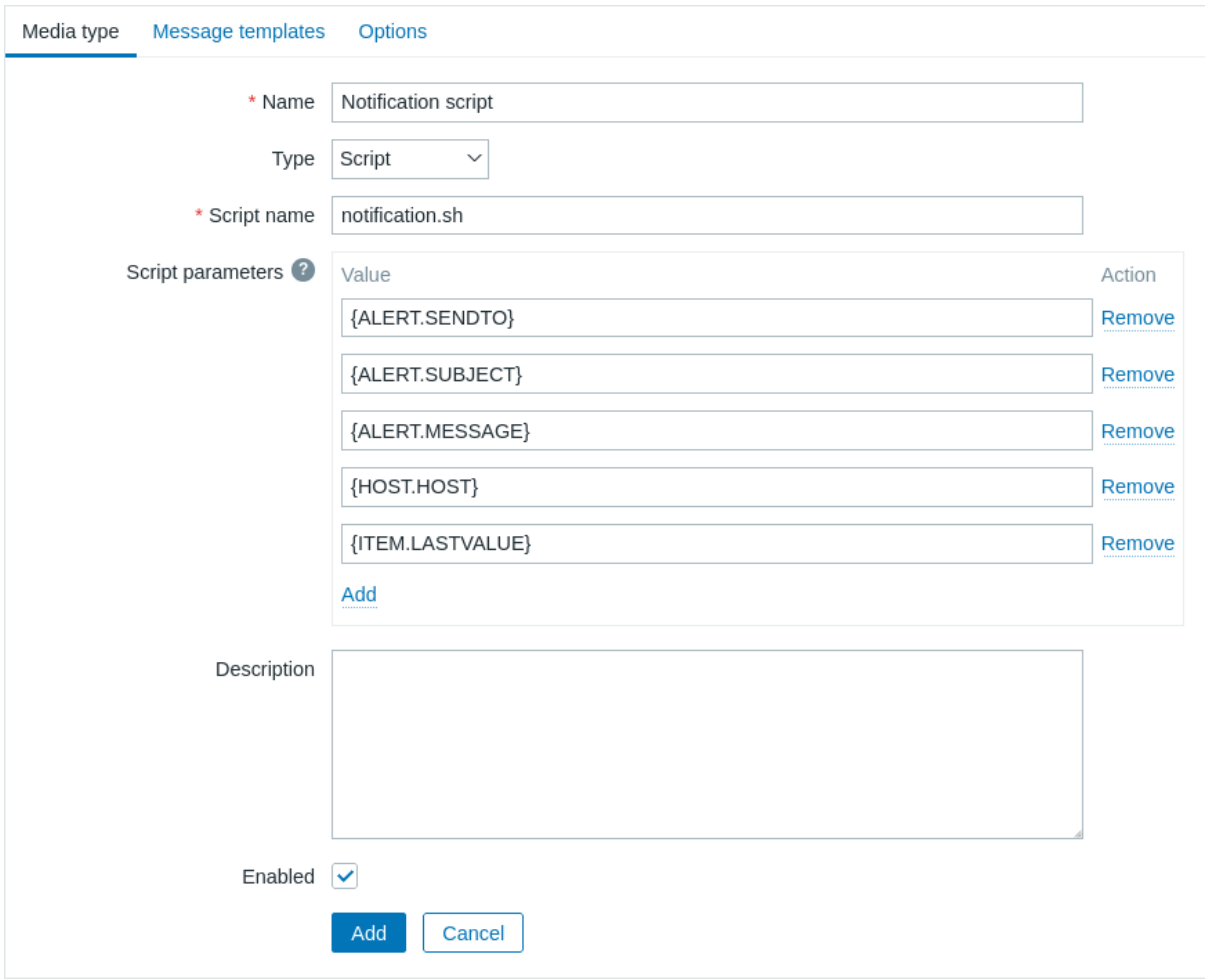
Umgebungsvariablen werden für das Skript weder beibehalten noch erstellt und sollten daher explizit behandelt werden.

Konfiguration

So konfigurieren Sie benutzerdefinierte Alarm-Skripte als Medientyp:

1. Gehen Sie zu *Benachrichtigungen* → *Medientypen*.
2. Klicken Sie auf *Medientyp erstellen*.

Die Registerkarte **Medientyp** enthält allgemeine Attribute des Medientyps:



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Die folgenden Parameter sind spezifisch für den Skript-Medientyp:

Parameter	Beschreibung
Skriptname	Geben Sie den Namen der Skriptdatei ein (z. B. notification.sh), die sich in dem Verzeichnis befindet, das im Server-Konfigurationsparameter AlertScriptsPath angegeben ist.
Skriptparameter	Fügen Sie optionale Skriptparameter hinzu, die in der Reihenfolge, in der sie definiert sind, 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, ebenso wie Benutzermakros.

Weitere Informationen zur Konfiguration von Standardnachrichten und Optionen für die Alarmverarbeitung finden Sie unter Allgemeine Medientyp-Parameter.

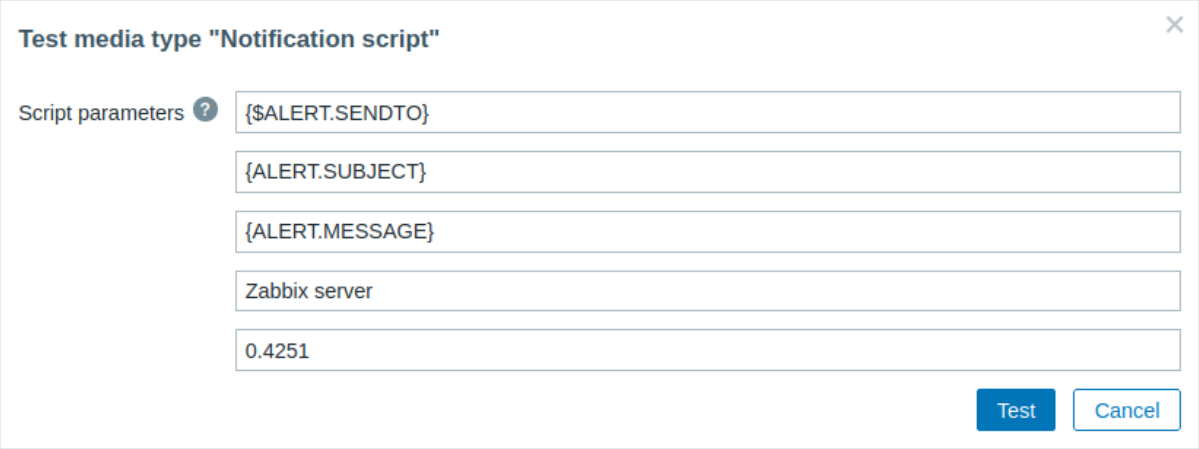
Warning:
Auch wenn ein Alarm-Skript keine Standardnachrichten verwendet, müssen die Nachrichtenvorlagen für die von diesem Medientyp verwendeten Vorgangsarten dennoch definiert werden. Andernfalls wird keine Benachrichtigung gesendet.

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 Server-Konfigurationsdatei begrenzt.

Testen

So testen Sie einen konfigurierten Skript-Medientyp:

1. Suchen Sie das entsprechende Skript in der Liste der Medientypen.
2. Klicken Sie in der letzten Spalte der Liste auf *Test*; ein Testformular wird in einem Pop-up-Fenster geöffnet. Das Testformular enthält dieselbe Anzahl an Parametern, die für den Skript-Medientyp konfiguriert sind.
3. Bearbeiten Sie bei Bedarf die Werte der Skriptparameter. 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 Skript-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 es während des Tests kein zugehöriges Ereignis gibt, aus dem die Details abgerufen werden könnten. 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.

Benutzermedien

Sobald der Medientyp konfiguriert ist, wechseln Sie zum Abschnitt *Benutzer* → *Benutzer* und bearbeiten Sie ein Benutzerprofil, indem Sie diesem Benutzer diesen Medientyp zuweisen. Die Schritte zum Einrichten von Benutzermedien, die für alle Medientypen gleich sind, werden auf der Seite Medientypen beschrieben.

Beachten Sie, dass beim Definieren der Benutzermedien das Feld *Senden an* nicht leer sein darf. Wenn dieses Feld im Warnskript nicht verwendet wird, geben Sie eine beliebige Kombination unterstützter Zeichen ein, um die Validierungsanforderungen zu umgehen.

4 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 und 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
- 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 auch in **Spiceworks** integriert werden (es ist kein webhook erforderlich). Um Zabbix-Benachrichtigungen in Spiceworks-Tickets umzuwandeln, erstellen Sie einen E-Mail-Medientyp und geben Sie die Spiceworks-Helpdesk-E-Mail-Adresse (z. B. help@zabbix.on.spiceworks.com) in den Profileinstellungen eines dafür vorgesehenen Zabbix-Benutzers ein.

Konfiguration

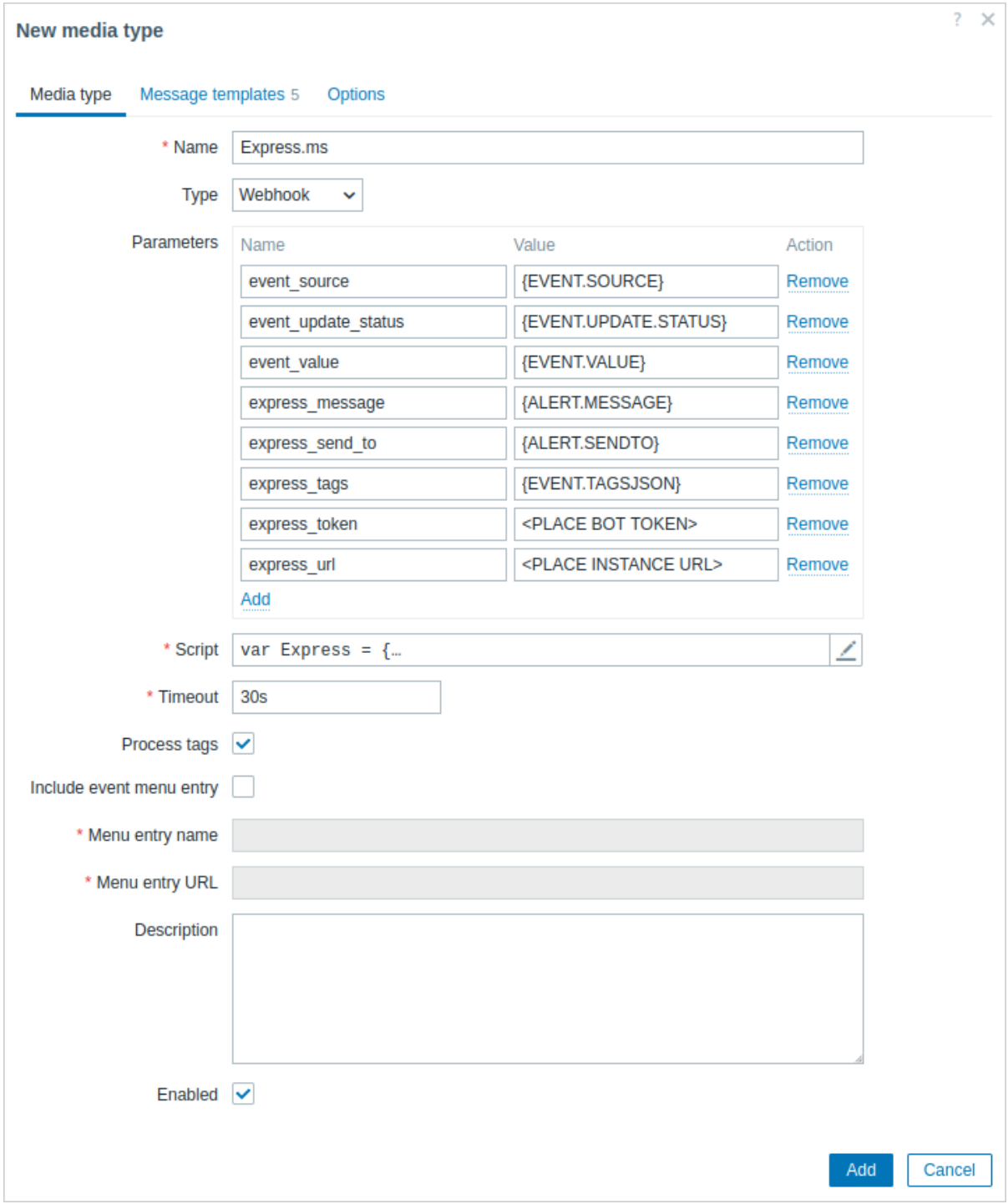
So beginnen Sie mit der Verwendung einer webhook-Integration:

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. Der webhook wird in der Liste der Medientypen angezeigt.
3. Konfigurieren Sie den webhook gemäß den Anweisungen in der Datei *Readme.md* (Sie können oben auf den Namen eines webhook klicken, um schnell auf *Readme.md* zuzugreifen).

So erstellen Sie einen benutzerdefinierten webhook von Grund auf:

1. Gehen Sie zu *Benachrichtigungen* > *Medientypen*.
2. Klicken Sie auf *Medientyp erstellen*.

Die Registerkarte **Medientyp** enthält verschiedene Attribute, die für diesen Medientyp spezifisch sind:



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Die folgenden Parameter sind spezifisch für den webhook-Medientyp:

Parameter	Beschreibung
Parameter	Geben Sie die webhook-Variablen als Attribut-Wert-Paare an. Bei vorkonfigurierten webhooks variiert die Liste der Parameter je nach Dienst. Prüfen Sie die Datei *Readme.md* des webhook auf die Parameterbeschreibung. 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]://` vorangestellt werden, um anzugeben, welche Art von Proxy verwendet wird (z. B. https, socks4, socks5; siehe documentation).
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 mit der Methode JSON.parse() in JSON-Objekte umgewandelt werden, zum Beispiel: `var params = JSON.parse(value);`. Der Code hat Zugriff auf alle Parameter; er kann HTTP-GET-, POST-, PUT- und DELETE-Anfragen ausführen, zusätzliche Methoden wie CONNECT, PATCH, HEAD, OPTIONS und TRACE unterstützen sowie HTTP-Header und den Anfrage-Body 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 die Warnung erstellt wurde; daher werden von einem webhook-Skript zurückgegebene Tags erst nach der anfänglichen Erstellung der Warnung 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, um sicherzustellen, dass jedes Skript mit seinen eigenen Daten arbeitet, und um Kollisionen zwischen gleichzeitigen Aufrufen zu vermeiden (siehe known issues). Siehe auch: Richtlinien für die webhook-Entwicklung, Beispiele für webhook-Skripte, Zusätzliche JavaScript-Objekte.
Timeout	Zeitüberschreitung für die JavaScript-Ausführung (1-60s, Standard 30s). 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 der webhook bei Verwendung von webhook tags 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
Eintrag im Ereignismenü einschließen	Aktivieren Sie das Kontrollkästchen, um einen Eintrag im Ereignismenü einzuschließen, der auf ein erstelltes externes Ticket verweist. Für jeden webhook, der aktiviert ist und bei dem dieses Kontrollkästchen markiert ist, wird ein Eintrag eingefügt. Beachten Sie, dass ein Eintrag nur dann eingefügt wird, wenn die Parameter *Name des Menüeintrags* und *URL des Menüeintrags* Makros vom Typ {EVENT.TAGS.<tag name>} enthalten und diese Makros aufgelöst werden können (d. h. das Ereignis hat diese Tags definiert). Wenn diese Option markiert ist, sollte der webhook nicht zum Senden von Benachrichtigungen an verschiedene Benutzer verwendet werden (erwägen Sie stattdessen die Erstellung eines dedicated user) und nicht in mehreren Warnaktionen 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 dann obligatorisch, wenn *Eintrag im Ereignismenü einschließen* markiert 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 dann obligatorisch, wenn *Eintrag im Ereignismenü einschließen* markiert ist.

Siehe allgemeine Medientyp-Parameter für Details zur Konfiguration von Standardmeldungen und Optionen zur Warnungsverarbeitung.

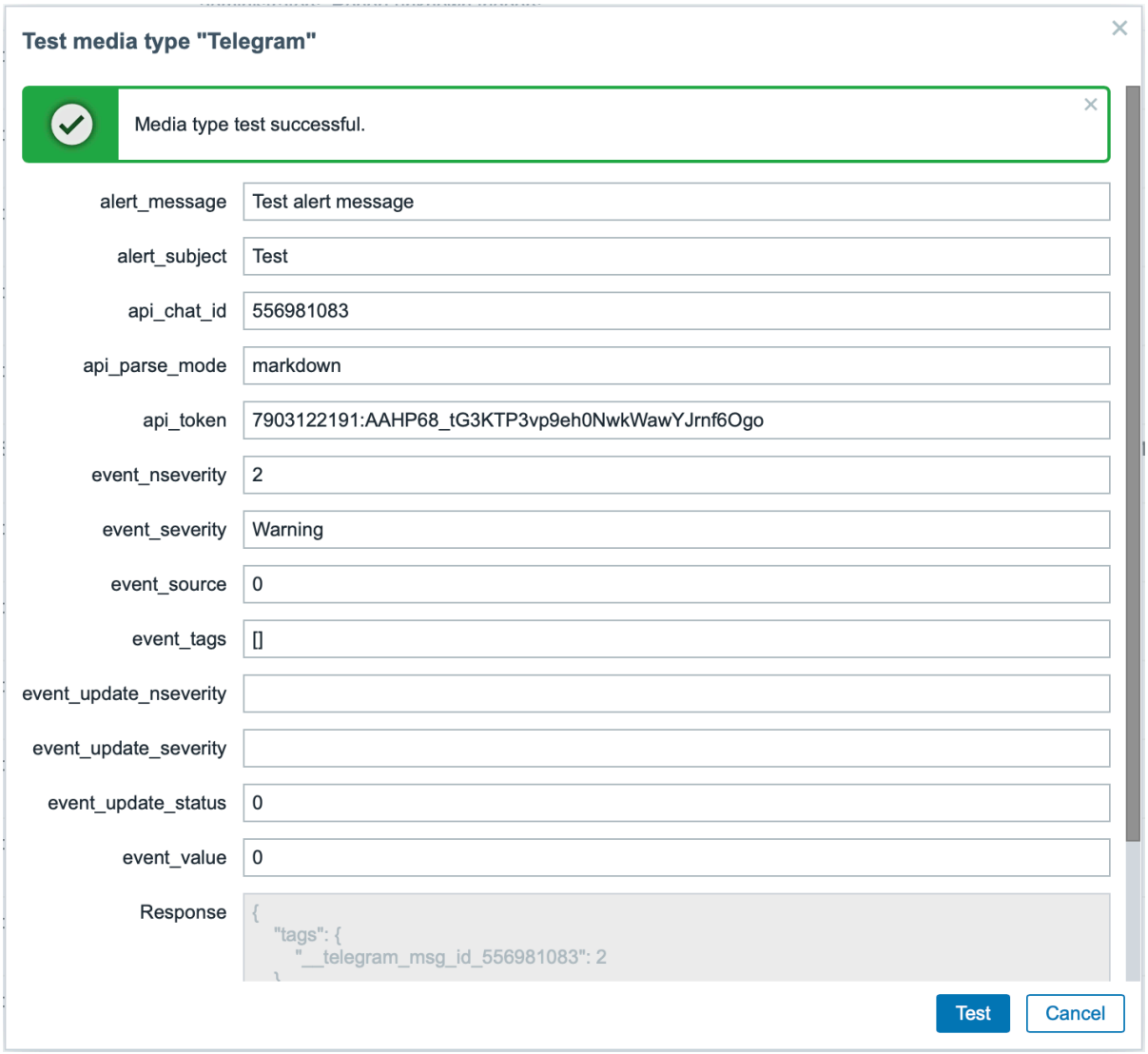
Warning:
Auch wenn ein webhook keine Standardmeldungen verwendet, müssen Nachrichtenvorlagen für die von diesem webhook verwendeten Operationstypen dennoch definiert werden.

Testen

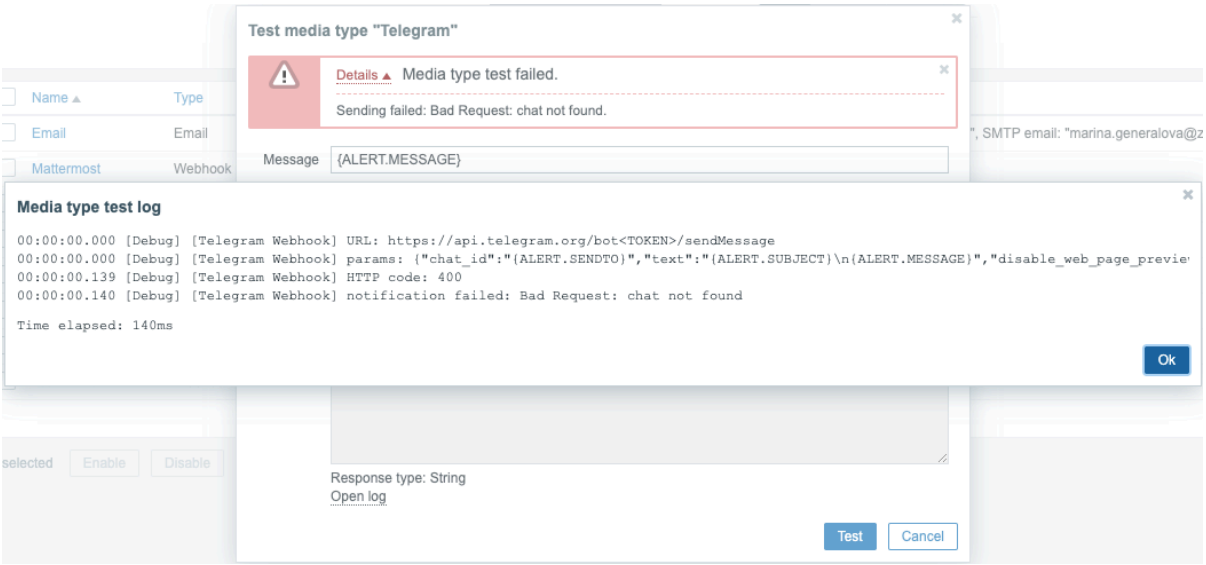
So testen Sie einen konfigurierten webhook-Medientyp:

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 webhook-Parameterwerte nach Bedarf. Ersetzen Sie Makros durch Beispielwerte; andernfalls werden 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 webhook bleiben unverändert.



Um Testprotokolleinträge des Medientyps anzuzeigen, ohne das Testfenster zu verlassen, klicken Sie auf *Open log* (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

Sobald der Medientyp konfiguriert ist, gehen Sie zum Abschnitt *Benutzer* > *Benutzer* und weisen Sie das webhook-Medium einem bestehenden Benutzer zu oder erstellen Sie einen neuen Benutzer, der das webhook repräsentiert. Die Schritte zum Einrichten von Benutzermedien für einen bestehenden Benutzer, die für alle Medientypen gleich sind, werden auf der Seite Medientypen beschrieben.

Wenn ein webhook Tags verwendet, um die Ticket\Nachrichten-ID zu speichern, vermeiden Sie es, dasselbe webhook als Medium verschiedenen Benutzern zuzuweisen, da dies zu webhook-Fehlern führen kann (gilt für die meisten webhooks, die die Option *Ereignismenüeintrag einbeziehen* verwenden). In diesem Fall empfiehlt es sich, einen dedizierten Benutzer zu erstellen, der das webhook repräsentiert:

1. Nachdem Sie den webhook-Medientyp konfiguriert haben, gehen 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 den Medien können auf ihren Standardwerten belassen werden, da sich dieser Benutzer nicht bei Zabbix anmelden wird.
2. Gehen Sie im Benutzerprofil auf die Registerkarte *Medien* und fügen Sie ein webhook hinzu und geben Sie die erforderlichen Kontaktinformationen an. Wenn das webhook kein Feld *Senden an* verwendet, geben Sie eine beliebige Kombination unterstützter Zeichen ein, um die Validierungsanforderungen zu umgehen.
3. Gewähren Sie diesem Benutzer mindestens Lese-Berechtigungen für alle Hosts, für die er die Benachrichtigungen senden soll.

Fügen Sie bei der Konfiguration der Aktionsbenachrichtigung diesen Benutzer im Feld *An Benutzer senden* in den Vorgangsdetails hinzu – dadurch weist Zabbix an, das webhook für Benachrichtigungen aus dieser Aktion zu verwenden.

Konfigurieren von Alarmierungsaktionen

Aktionen bestimmen, welche Benachrichtigungen über den webhook gesendet werden sollen. Die Schritte zum Konfigurieren von Aktionen, die webhooks betreffen, sind dieselben wie bei allen anderen Medientypen, mit folgenden Ausnahmen:

- Wenn ein webhook webhook-Tags verwendet, um Ticket\Nachrichten-ID zu speichern und Aktualisierungs\Lösungs-Operationen zu verarbeiten, vermeiden Sie die Verwendung desselben webhook in mehreren Alarmierungsaktionen für ein einzelnes Problemereignis. Wenn {EVENT.TAGS.<tag name>} existiert und im webhook aktualisiert wird, ist sein resultierender 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 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 sowie in verschiedenen Aktionen verwendet werden kann, 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 festlegen. Andernfalls wird keine Benachrichtigung gesendet.

1 Beispiele für webhook-Skripte

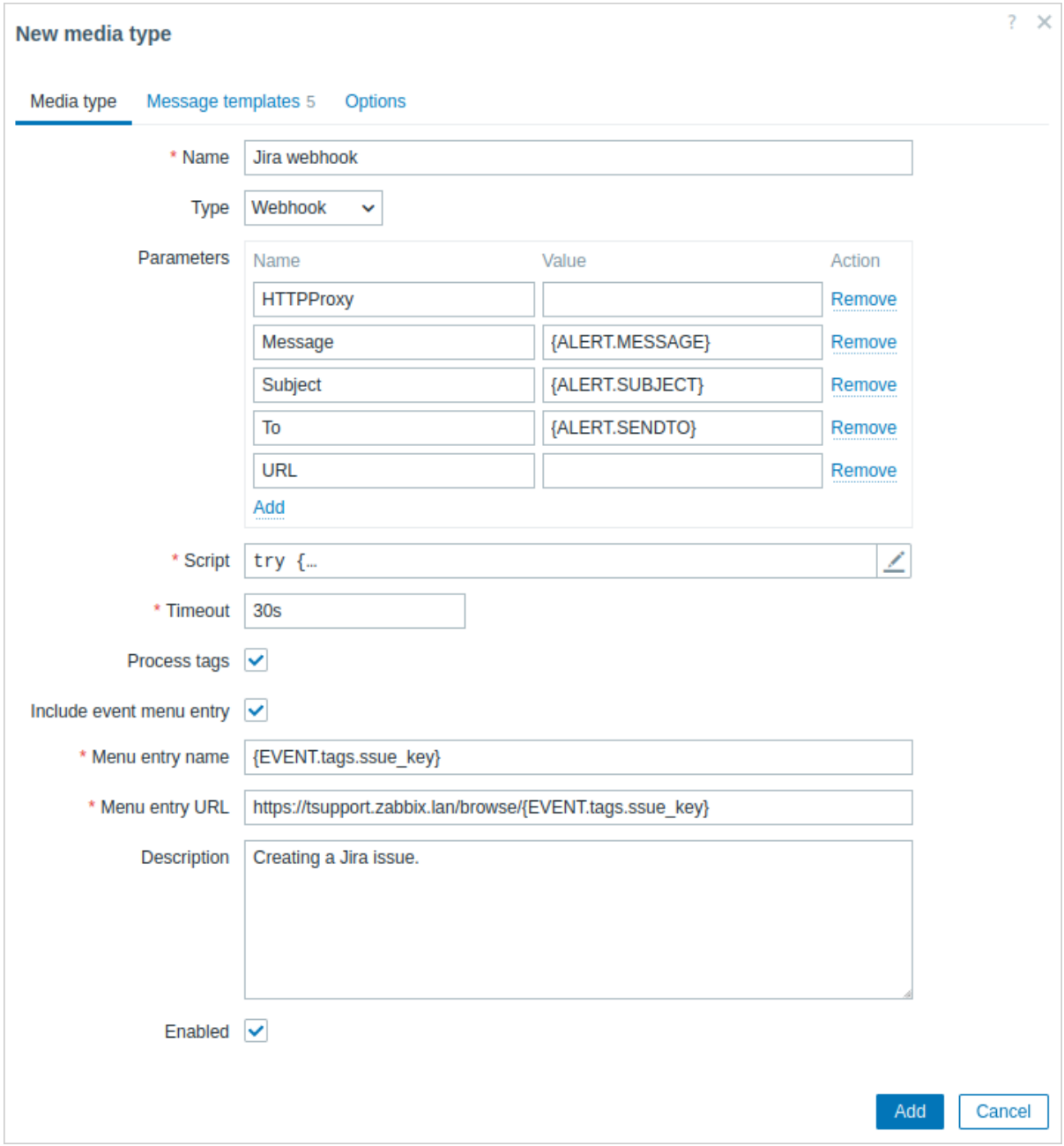
Übersicht

Obwohl Zabbix eine große Anzahl sofort verfügbarer webhook-Integrationen bietet, möchten Sie möglicherweise stattdessen eigene webhooks erstellen. Dieser Abschnitt enthält Beispiele für benutzerdefinierte webhook-Skripte (verwendet im Parameter *Script*). Eine Beschreibung weiterer webhook-Parameter finden Sie im Abschnitt webhook.

> **Attention:**
> Verwenden Sie in der JavaScript-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 Tags des gelösten Ereignisses (einschließlich Tags, die von Vorlagen, Hosts und Auslösern geerbt wurden). Webhook-Skripte werden ausgeführt, nachdem die Warnung erstellt wurde; daher werden von einem webhook-Skript zurückgegebene Tags erst nach der anfänglichen Erstellung der Warnung angewendet und sind möglicherweise in der Warnung, die 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 Ereignis-Tags vom Server ab (zum Beispiel über die Event-API) oder speichern Sie von einem webhook erzeugte Tags in einem externen persistenten Speicher und korrelieren Sie sie dort.

Jira webhook (benutzerdefiniert)



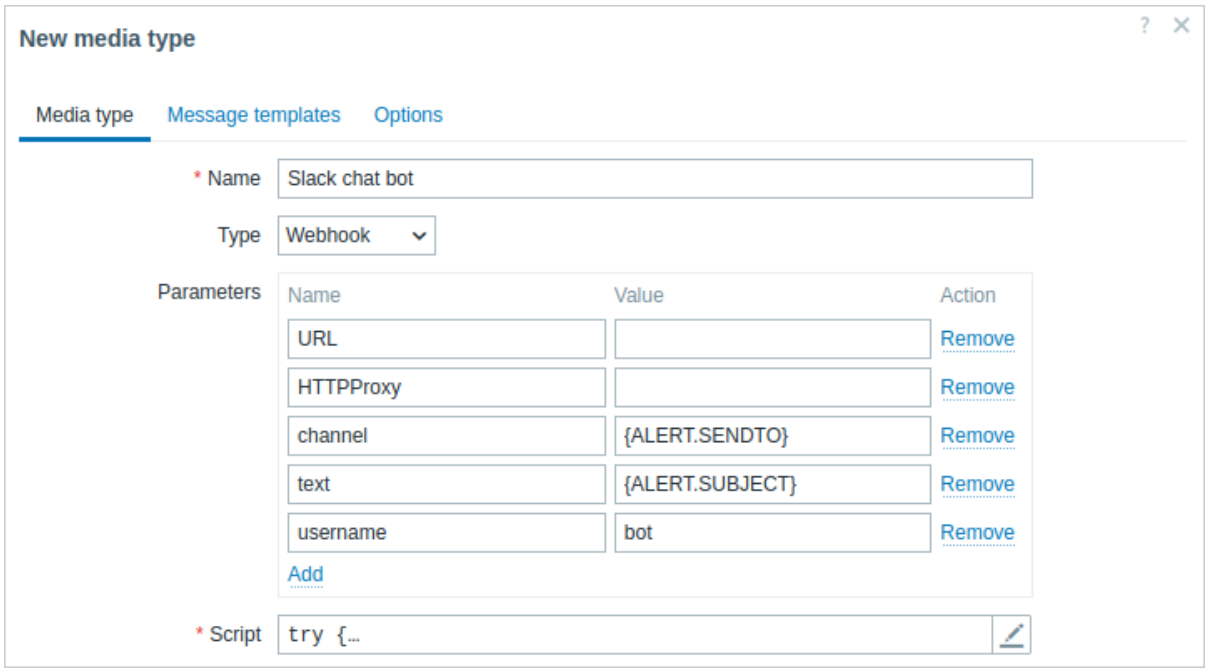
Dieses Skript erstellt ein JIRA-Ticket und gibt einige Informationen zum erstellten Ticket zurück.

```
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 (benutzerdefiniert)

Dieser 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.hook_url, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

2 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:

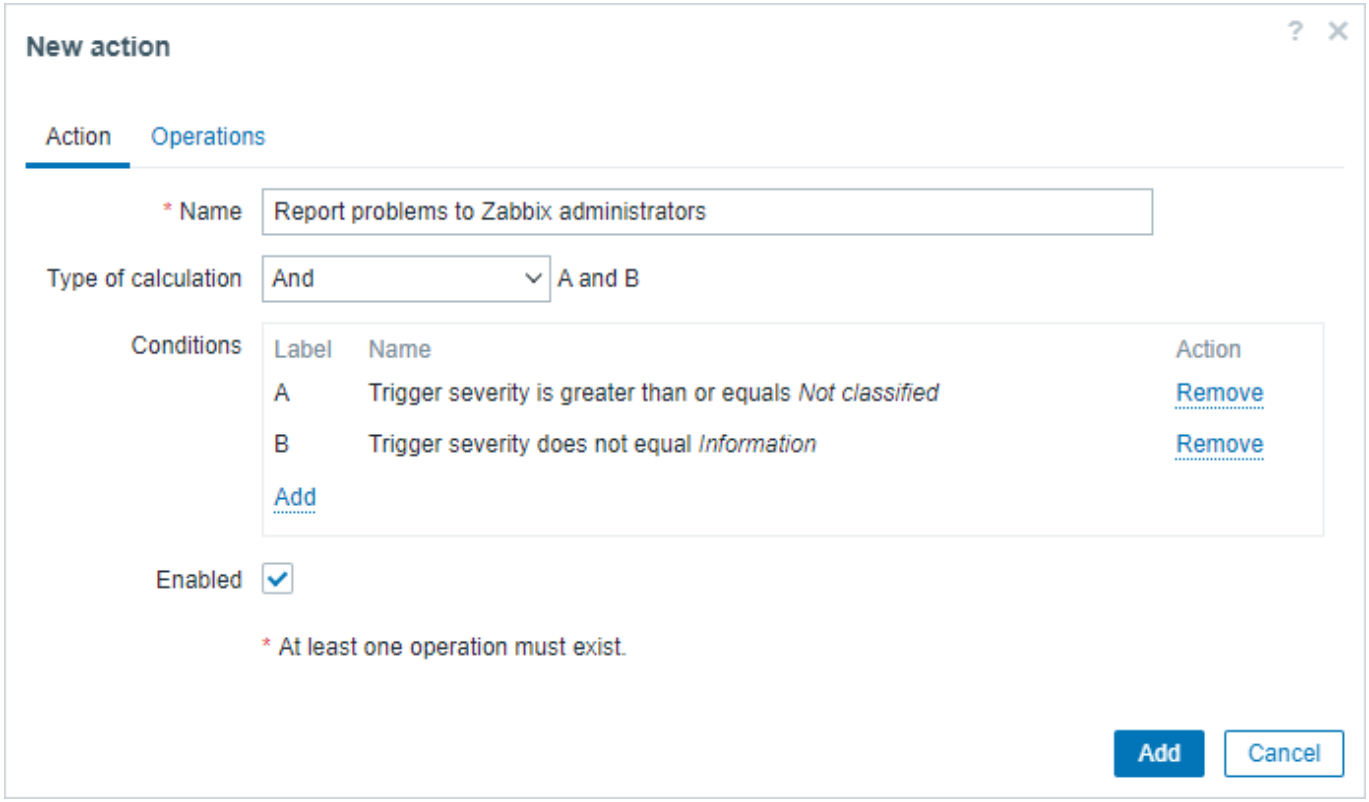
- Der Benutzerzugriff auf Service-Aktionen hängt von den Zugriffsrechten auf Services ab, die durch die Rolle des Benutzers gewährt werden
- Service-Aktionen unterstützen einen anderen Satz von Bedingungen

Konfigurieren einer Aktion

Gehen Sie wie folgt vor, um eine Aktion zu konfigurieren:

- Gehen Sie zu *Alerts* > *Actions* und wählen Sie im Untermenü den erforderlichen Aktionstyp aus (Sie können später über das Dropdown im Titel zu einem anderen Typ wechseln).
- Klicken Sie auf *Create action*.
- Benennen Sie die Aktion.
- Wählen Sie die Bedingungen, unter denen Operationen ausgeführt werden.
- Wählen Sie die Operationen, die ausgeführt werden sollen.

Allgemeine Aktionsattribute:



Alle erforderlichen Eingabefelder 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 beiden: AND bei unterschiedlichen Bedingungstypen und OR bei demselben 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 konfigurierten Aktionstyp entspricht.
Enabled	Aktivieren Sie das Kontrollkästchen, um die Aktion zu aktivieren. Andernfalls ist sie deaktiviert.

1 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 unterscheidet zwischen Groß- und Kleinschreibung.

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

Die folgenden Bedingungen können in Service-Aktionen verwendet werden:

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Service	gleich ungleich	Geben Sie einen Service oder einen auszuschließenden Service an. **gleich** - das Ereignis gehört zu diesem Service. **ungleich** - das Ereignis gehört nicht zu diesem Service. Die Angabe eines übergeordneten Service wählt implizit auch alle untergeordneten Services aus. Um nur den übergeordneten Service anzugeben, müssen alle verschachtelten Services zusätzlich mit dem Operator **ungleich** festgelegt werden.
Service-Name	enthält enthält nicht	Geben Sie eine Zeichenfolge im Service-Namen oder eine auszuschließende Zeichenfolge an. **enthält** - das Ereignis wird von einem Service erzeugt, dessen Name diese Zeichenfolge enthält. **enthält nicht** - diese Zeichenfolge kann im Service-Namen nicht gefunden werden.
Service-Tag-Name	gleich ungleich enthält enthält nicht	Geben Sie ein Ereignis-Tag oder ein auszuschließendes Ereignis-Tag an. Service-Ereignis-Tags können im Abschnitt *Tags* der Service-Konfiguration definiert werden. **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.
Service-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. Service-Ereignis-Tags können im Abschnitt *Tags* der Service-Konfiguration definiert werden. **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.

Attention:
Stellen Sie sicher, dass Sie Nachrichtenvorlagen für Service-Aktionen im Menü *Benachrichtigungen* → *Medientypen* definieren. Andernfalls werden die Benachrichtigungen nicht gesendet.

Discovery-Aktionen

Die folgenden Bedingungen können in Discovery-basierten Ereignissen verwendet werden:

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Host-IP	gleich ungleich	Geben Sie einen IP-Adressbereich oder einen auszuschließenden Bereich für einen erkannten Host an. **gleich** - die Host-IP liegt im Bereich. **ungleich** - die Host-IP liegt 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.
Diensttyp	gleich ungleich	Geben Sie einen Diensttyp eines erkannten Dienstes oder einen auszuschließenden Diensttyp an. **gleich** - entspricht dem erkannten Dienst. **ungleich** - entspricht nicht dem erkannten Dienst. Verfügbare Diensttypen: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix Agent, SNMPv1-Agent, SNMPv2-Agent, SNMPv3-Agent, ICMP-Ping, telnet.
Dienst-Port	gleich ungleich	Geben Sie einen TCP-Portbereich eines erkannten Dienstes oder einen auszuschließenden Bereich an. **gleich** - der Dienst-Port liegt im Bereich. **ungleich** - der Dienst-Port liegt nicht im Bereich.
Discovery-Regel	gleich ungleich	Geben Sie eine Discovery-Regel oder eine auszuschließende Discovery-Regel an. **gleich** - verwendet diese Discovery-Regel. **ungleich** - verwendet eine beliebige andere Discovery-Regel außer dieser.
Discovery-Prüfung	gleich ungleich	Geben Sie eine Discovery-Prüfung oder eine auszuschließende Discovery-Prüfung an. **gleich** - verwendet diese Discovery-Prüfung. **ungleich** - verwendet eine beliebige andere Discovery-Prüfung außer dieser.
Discovery-Objekt	gleich	Geben Sie das erkannte Objekt an. **gleich** - entspricht dem erkannten Objekt (ein Gerät oder ein Dienst).
Discovery-Status	gleich	**Up** - entspricht Ereignissen vom Typ ’Host Up’ und ’Service Up’. **Down** - entspricht Ereignissen vom Typ ’Host Down’ und ’Service Down’. **Discovered** - entspricht Ereignissen vom Typ ’Host Discovered’ und ’Service Discovered’. **Lost** - entspricht Ereignissen vom Typ ’Host Lost’ und ’Service Lost’.
Uptime/Downtime	ist größer oder gleich ist kleiner oder gleich	Uptime für Ereignisse vom Typ ’Host Up’ und ’Service Up’. Downtime für Ereignisse vom Typ ’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	gleich ungleich 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. Zeichenkettenvergleich. Wenn für eine Regel mehrere Zabbix-Agent- oder SNMP-Prüfungen konfiguriert sind, werden die empfangenen Werte für jede von ihnen geprüft (jede Prüfung erzeugt ein neues Ereignis, das mit allen Bedingungen abgeglichen wird). **gleich** - entspricht dem Wert. **ungleich** - 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 Teilzeichenkette. Der Parameter wird als Zeichenkette angegeben. **enthält nicht** - enthält die Teilzeichenkette nicht. Der Parameter wird als Zeichenkette angegeben.

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Proxy	gleich ungleich	Geben Sie einen Proxy oder einen auszuschließenden Proxy an. **gleich** - verwendet diesen Proxy. **ungleich** - verwendet einen beliebigen anderen Proxy außer diesem.

Note:
Dienstprüfungen in einer Discovery-Regel, die zu Discovery-Ereignissen führen, finden nicht gleichzeitig statt. Wenn daher **mehrere** Werte für die Bedingungen `Service type`, `Service port` oder `Received value` in der Aktion konfiguriert sind, werden sie jeweils mit einem Discovery-Ereignis gleichzeitig verglichen, aber **nicht** mit mehreren Ereignissen gleichzeitig. Daher werden Aktionen mit mehreren Werten für dieselben Prüfungstypen möglicherweise nicht korrekt ausgeführt.

Aktionen zur Autoregistrierung

Die folgenden Bedingungen können in Aktionen verwendet werden, die auf der aktiven Agent-Autoregistrierung 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** - verwendet diesen Proxy. **ist nicht gleich** - verwendet einen beliebigen anderen Proxy außer diesem.

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.

Bedingungstyp	Unterstützte Operatoren	Beschreibung
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 werden. 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 beidem: UND bei unterschiedlichen Bedingungstypen und ODER bei demselben Bedingungstyp, zum Beispiel:

Host-Gruppe entspricht Oracle-Servern
Host-Gruppe entspricht MySQL-Servern
Ereignisname enthält ’Database is down’
Ereignisname enthält ’Database is unavailable’

wird ausgewertet als

(Host-Gruppe entspricht Oracle-Servern **or** Host-Gruppe entspricht MySQL-Servern**) 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, sind in einem benutzerdefinierten Ausdruck auch mehrere andere Berechnungsarten möglich:

(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.

2 Betrieb

Übersicht

Sie können die folgenden Operationen für alle Ereignisse definieren:

- Eine Nachricht senden
- Einen Remote-Befehl ausführen

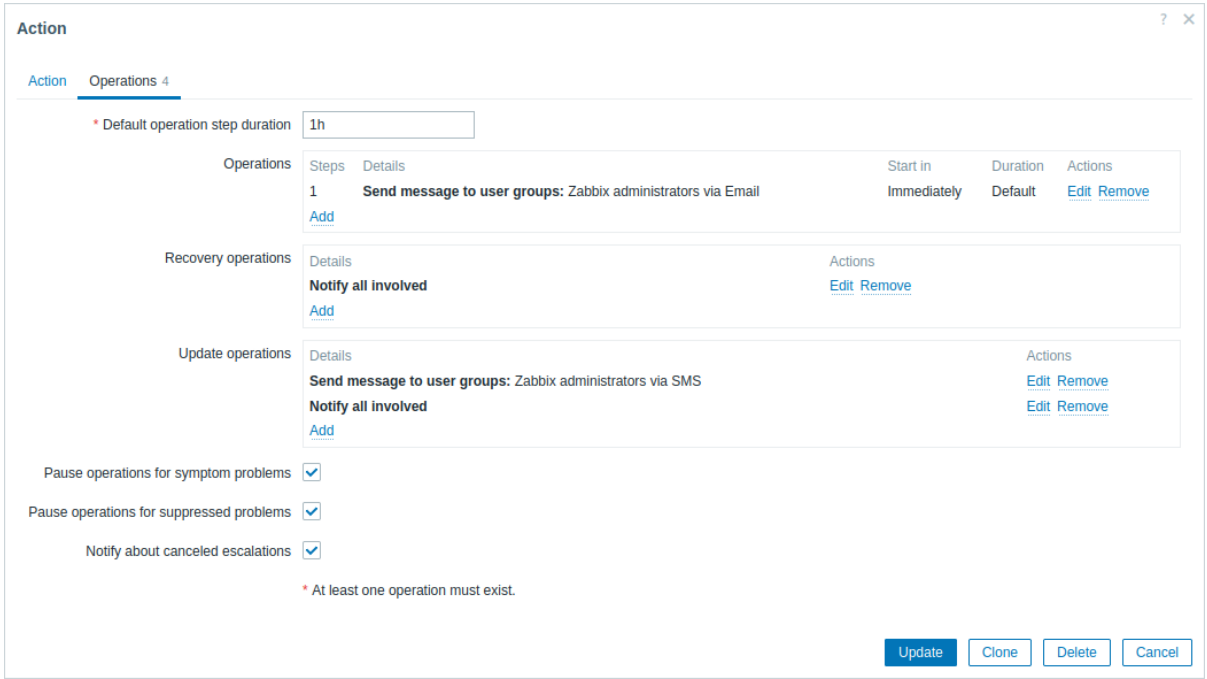
Attention:
Der Zabbix Server erstellt keine Benachrichtigungen, wenn der Zugriff auf den Host für den als Empfänger der Aktionsoperation definierten Benutzer ausdrücklich „verweigert“ ist oder wenn für den Benutzer überhaupt keine Rechte für den Host definiert sind.

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 mit Vorlage aufheben
- Hostinventarmodus festlegen

Konfigurieren einer Operation

Um eine Operation zu konfigurieren, wechseln Sie zur Registerkarte *Operations* in der Konfiguration von action.



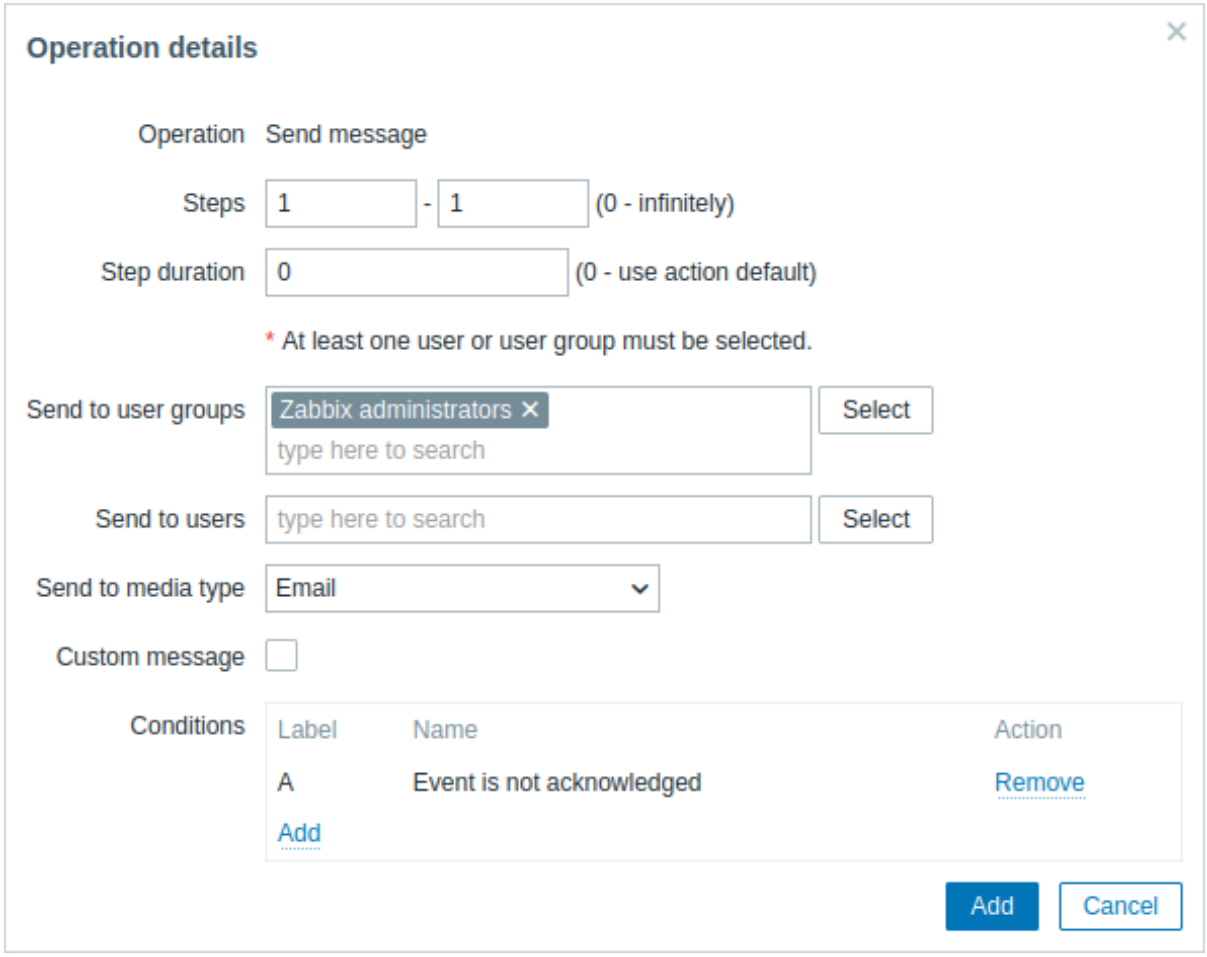
Allgemeine Operationsattribute:

Parameter	Beschreibung
Default operation step duration	Standarddauer eines Operationsschritts (60 Sekunden bis 1 Woche). Beispielsweise bedeutet eine Schrittdauer von einer Stunde, dass nach der Ausführung einer Operation eine Stunde vergeht, bevor der nächste Schritt erfolgt. Zeitsuffixe werden unterstützt, z. B. 60s, 1m, 2h, 1d. Benutzermakros werden unterstützt.
Operations	Action-Operationen (falls vorhanden) werden mit folgenden Details angezeigt: **Steps** - Eskalationsschritt(e), denen die Operation zugewiesen ist. **Details** - Typ der Operation und ihr Empfänger/Ziel. In der Operationsliste werden außerdem der verwendete Medientyp (E-Mail, SMS oder Skript) sowie der Vor- und Nachname (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers angezeigt. **Start in** - wie lange nach einem Ereignis die Operation ausgeführt wird. **Duration (sec)** - die Schrittdauer wird angezeigt. *Default* wird angezeigt, wenn der Schritt die Standarddauer verwendet, und eine Zeitangabe wird angezeigt, wenn eine benutzerdefinierte Dauer verwendet wird. **Actions** - Links zum Bearbeiten und Entfernen einer Operation werden angezeigt.
Recovery operations	Action-Operationen (falls vorhanden) werden mit folgenden Details angezeigt: **Details** - Typ der Operation und ihr Empfänger/Ziel. In der Operationsliste werden außerdem der verwendete Medientyp (E-Mail, SMS oder Skript) sowie der Vor- und Nachname (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers angezeigt. **Actions** - Links zum Bearbeiten und Entfernen einer Operation werden angezeigt.
Update operations	Action-Operationen (falls vorhanden) werden mit folgenden Details angezeigt: **Details** - Typ der Operation und ihr Empfänger/Ziel. In der Operationsliste werden außerdem der verwendete Medientyp (E-Mail, SMS oder Skript) sowie der Vor- und Nachname (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers angezeigt. **Actions** - Links zum Bearbeiten und Entfernen einer Operation werden angezeigt.
Pause operations for symptom problems	Aktivieren Sie dieses Kontrollkästchen, um Operationen (nach der ersten Operation) für Symptomprobleme anzuhalten. Beachten Sie, dass diese Einstellung nur Problemeskalationen betrifft; Recovery- und Update-Operationen sind davon nicht betroffen. Diese Option ist nur für *Trigger actions* verfügbar.
Pause operations for suppressed problems	Aktivieren Sie dieses Kontrollkästchen, um den Start von Operationen für die Dauer eines Wartungszeitraums zu verzögern. Wenn die Operationen nach der Wartung gestartet werden, werden alle Operationen ausgeführt, einschließlich derjenigen für Ereignisse während der Wartung. Beachten Sie, dass diese Einstellung nur Problemeskalationen betrifft; Recovery- und Update-Operationen sind davon nicht betroffen. Wenn Sie dieses Kontrollkästchen deaktivieren, werden Operationen auch während eines Wartungszeitraums ohne Verzögerung ausgeführt. Diese Option ist für *Service actions* nicht verfügbar.
Notify about canceled escalations	Deaktivieren Sie dieses Kontrollkästchen, um Benachrichtigungen über abgebrochene Eskalationen zu deaktivieren (wenn Host, Datenpunkt, Auslöser oder Action deaktiviert ist).

Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Um die Details einer neuen Operation zu konfigurieren, klicken Sie auf Add im Block *Operations*. Um eine bestehende Operation zu bearbeiten, klicken Sie auf Edit neben der Operation. Es wird ein Pop-up-Fenster geöffnet, in dem Sie die Details des Operationsschritts bearbeiten können.

Details zur Operation



Parameter	Beschreibung
Operation	Wählen Sie die Operation aus: **Nachricht senden** - Nachricht an Benutzer senden. **<Name des Remote-Befehls>** - einen Remote-Befehl ausführen. Befehle stehen zur Ausführung zur Verfügung, wenn sie zuvor in globalen Skripten mit dem als Geltungsbereich ausgewählten Wert *Action operation* definiert wurden. Für auf Discovery und Autoregistrierung basierende Ereignisse sind weitere Operationen verfügbar (siehe oben).
Steps	Wählen Sie die Schritte aus, denen die Operation in einem Eskalations-Zeitplan zugewiesen werden soll: **Von** - ab diesem Schritt ausführen. **Bis** - bis zu diesem Schritt ausführen (0=unendlich, die Ausführung wird nicht begrenzt).
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 Schrittdauern definiert sind, wird die kürzeste berücksichtigt und auf den Schritt angewendet.
Operationstyp: Nachricht senden	
An Benutzergruppen senden	Wählen Sie Benutzergruppen aus, an die die Nachricht gesendet werden soll. Die Benutzergruppe muss mindestens über „Lesen“- Berechtigungen für den Host verfügen, um benachrichtigt zu werden.

Parameter	Beschreibung
An Benutzer senden	Wählen Sie Benutzer aus, an die die Nachricht gesendet werden soll. Der Benutzer muss mindestens über „Lesen“- Berechtigungen für den Host verfügen, um benachrichtigt zu werden.
An Medientyp senden	Nachricht an alle verfügbaren (konfigurierten und aktivierten) Medientypen oder nur an einen bestimmten senden.
Benutzerdefinierte Nachricht	Falls ausgewählt, kann die benutzerdefinierte Nachricht konfiguriert werden. Für Benachrichtigungen über interne Ereignisse per webhooks ist eine benutzerdefinierte Nachricht obligatorisch.
Betreff	Betreff der benutzerdefinierten Nachricht. Der Betreff kann Makros enthalten. Er ist auf 255 Zeichen begrenzt.
Nachricht	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 Nachricht senden).
Operationstyp: Remote-Befehl	
Zielliste	Wählen Sie 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** - Host(s) auswählen, auf denen der Befehl ausgeführt werden soll. **Host-Gruppe** - Host-Gruppe(n) auswählen, auf denen der Befehl ausgeführt werden soll. Die Angabe einer übergeordneten Host-Gruppe wählt implizit alle verschachtelten Host-Gruppen aus. Somit wird der Remote-Befehl auch auf Hosts aus verschachtelten Gruppen ausgeführt. Ein Befehl auf einem Host wird nur einmal ausgeführt, auch wenn der Host mehrfach übereinstimmt (z. B. aus mehreren Host-Gruppen; einzeln und aus einer Host-Gruppe). Die Zielliste ist bedeutungslos, wenn ein benutzerdefiniertes Skript auf dem Zabbix Server ausgeführt wird. Die Auswahl weiterer Ziele führt in diesem Fall nur dazu, dass das Skript auf dem Server mehrfach ausgeführt wird. Beachten Sie, dass bei globalen Skripten die Zielauswahl auch von der Einstellung *Host group* in der globalen Skript-Konfiguration abhängt. Die Option *Zielliste* ist für *Service actions* nicht verfügbar, da in diesem Fall Remote-Befehle immer auf dem Zabbix Server ausgeführt werden.
Conditions	Bedingung für die Ausführung der Operation: **Ereignis ist nicht bestätigt** - nur wenn das Ereignis nicht bestätigt ist. **Ereignis ist bestätigt** - nur wenn das Ereignis bestätigt ist. Die Option *Conditions* ist nur für *Trigger actions* verfügbar.

Wenn Sie fertig sind, klicken Sie auf *Add*, um die Operation zur Liste der *Operations* hinzuzufügen.

1 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:

- die Medien definieren, an die eine Nachricht gesendet werden soll

Wenn die Operation außerhalb des Zeitraums When active stattfindet, der für das ausgewählte Medium in der Benutzerkonfiguration definiert ist, wird die Nachricht nicht gesendet.

Die Standard-Auslöser-Schwere (’Not classified’) **muss** in der Benutzer-Medien- Konfiguration aktiviert sein, wenn Sie Benachrichtigungen für Nicht-Auslöser-Ereignisse wie Discovery, aktive Agent-Autoregistrierung oder interne Ereignisse erhalten möchten.

- eine Aktionsoperation konfigurieren, die eine Nachricht an eines der definierten Medien sendet

Attention:
Zabbix sendet Benachrichtigungen nur an diejenigen Benutzer, die mindestens Leseberechtigungen für den Host haben, der das Ereignis erzeugt hat. Mindestens ein Host eines Auslöser-Ausdrucks muss zugänglich sein.

Sie können benutzerdefinierte Szenarien für das Senden von Nachrichten mit Escalations konfigurieren.

Um E-Mails von Zabbix erfolgreich zu empfangen und zu lesen, müssen E-Mail- Server/-Clients das Standardformat ’SMTP/MIME email’ unterstützen, da Zabbix UTF-8-Daten sendet (wenn der Betreff nur ASCII-Zeichen enthält, wird er nicht in UTF-8 kodiert). Der Betreff und der Nachrichtentext sind base64-kodiert, um dem Standardformat ’SMTP/MIME email’ zu entsprechen.

Das Nachrichtenlimit nach der vollständigen Makro-Erweiterung ist dasselbe wie das Nachrichtenlimit für Remote commands.

Nachrichtenverfolgung

Sie können den Status der gesendeten Nachrichten unter *Monitoring* → *Probleme* einsehen.

In der Spalte *Aktionen* sehen Sie zusammengefasste Informationen über die durchgeführten Aktionen. Dabei stehen grüne Zahlen für gesendete Nachrichten, rote für fehlgeschlagene Nachrichten. *In Bearbeitung* zeigt an, dass eine Aktion initiiert wurde. *Fehlgeschlagen* informiert darüber, dass keine Aktion erfolgreich ausgeführt wurde.

Wenn Sie auf die Ereigniszeit klicken, um die Ereignisdetails anzuzeigen, können Sie im Block *Aktionen* Details zu den aufgrund des Ereignisses gesendeten (oder nicht gesendeten) Nachrichten sehen.

Unter *Berichte* → *Aktionsprotokoll* sehen Sie Details zu allen ausgeführten Aktionen für diejenigen Ereignisse, für die eine Aktion konfiguriert ist.

2 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.

Dadurch sind Remote-Befehle ein leistungsstarker Mechanismus für eine intelligente, proaktive Überwachung.

In den naheliegendsten Anwendungsfällen können Sie beispielsweise Folgendes versuchen:

- 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 er auf Anfragen nicht antwortet
- Automatisch Speicherplatz freigeben (ältere Dateien entfernen, /tmp bereinigen), wenn der Festplattenspeicher knapp wird
- Eine VM abhängig von der CPU-Auslastung von einem physischen System auf ein anderes migrieren
- Neue Knoten zu einer Cloud-Umgebung hinzufügen, wenn CPU-, Festplatten-, Arbeitsspeicher- oder andere Ressourcen nicht ausreichen

Die Konfiguration einer Aktion für Remote-Befehle ist ähnlich wie die für das Senden einer Nachricht; der einzige Unterschied besteht darin, dass Zabbix einen Befehl ausführt, anstatt eine Nachricht zu senden.

Remote-Befehle können durch Zabbix Server, Proxy oder Agent ausgeführt werden. Remote- Befehle auf dem Zabbix Agent können direkt durch Zabbix Server oder über Zabbix Proxy ausgeführt werden. Sowohl auf dem Zabbix Agent als auch auf dem Zabbix Proxy sind Remote-Befehle standardmäßig deaktiviert. Sie können aktiviert werden durch:

- Hinzufügen des Parameters `AllowKey=system.run[*]` in der Agent-Konfiguration;
- Setzen des Parameters EnableRemoteCommands auf „1“ in der Proxy- Konfiguration.

Von Zabbix Server ausgeführte Remote-Befehle werden wie unter Command execution beschrieben ausgeführt, einschließlich der Prüfung des Exit-Codes.

Remote-Befehle werden auch dann ausgeführt, wenn sich der Ziel-Host in Wartung befindet.

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.

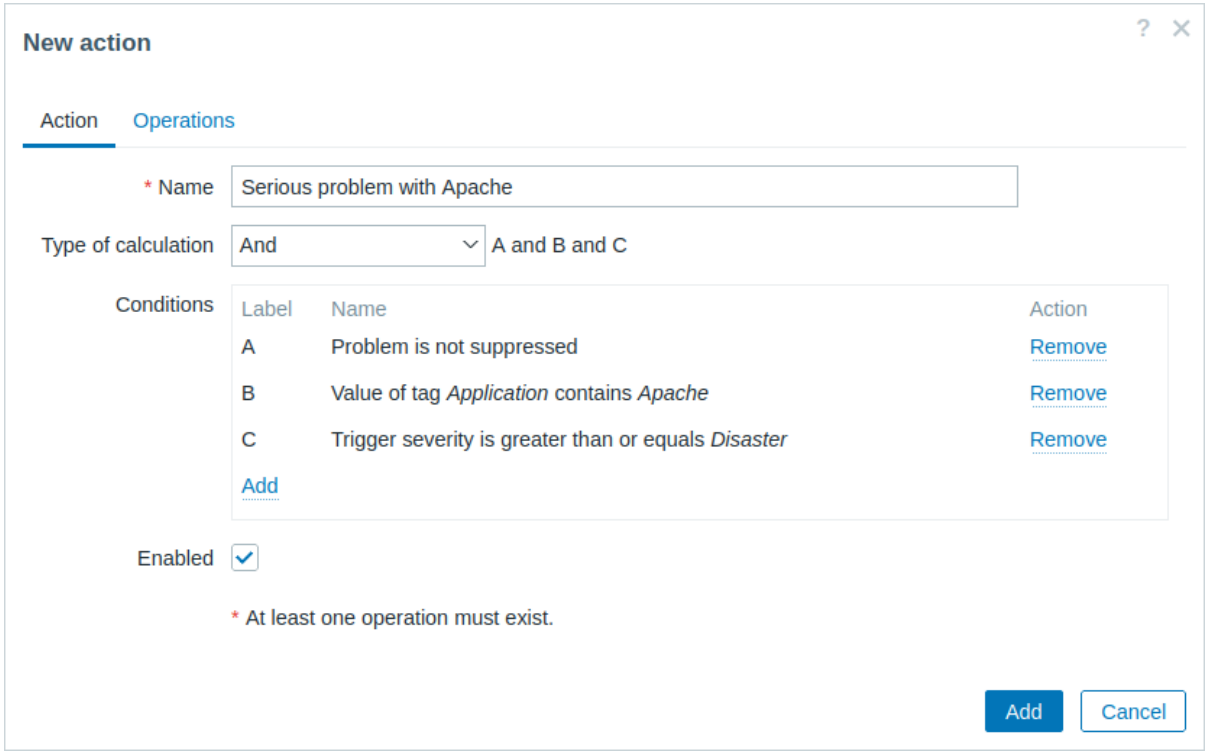
Konfiguration

Die Remote-Befehle, die auf dem Zabbix Agent ausgeführt werden (benutzerdefinierte Skripte), müssen zunächst in der Konfiguration des Agent aktiviert werden.

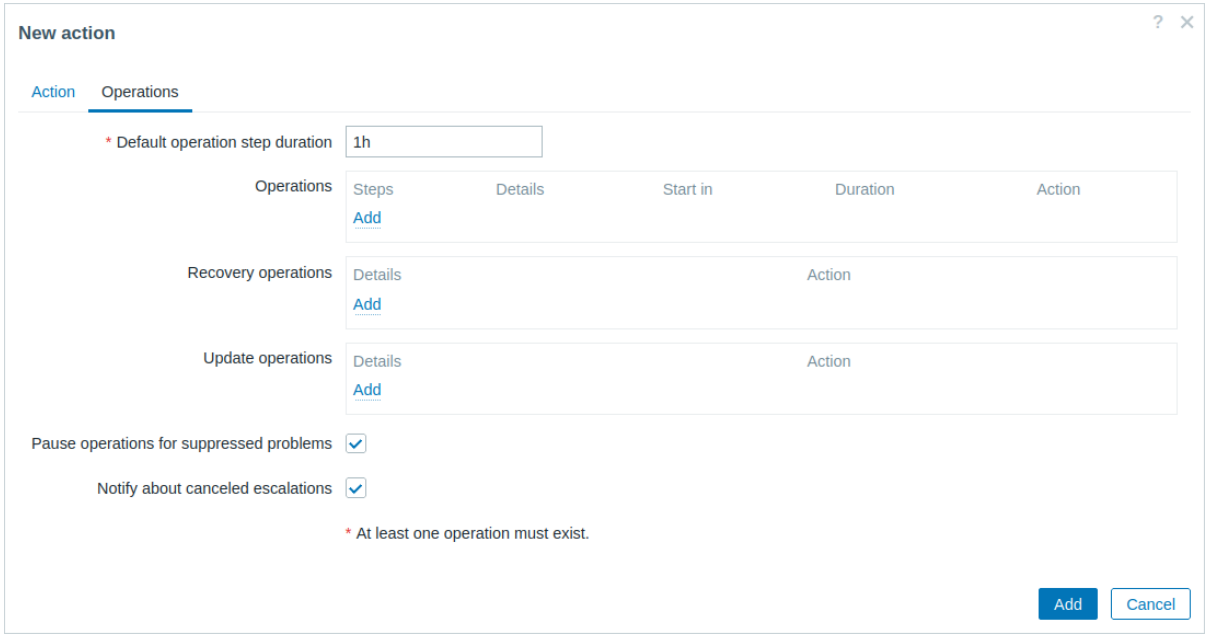
Stellen Sie sicher, dass der Parameter `AllowKey=system.run[<command>,\*]` für jeden erlaubten Befehl in der Agent-Konfiguration hinzugefügt wird, um einen bestimmten Befehl im nowait-Modus zuzulassen. Starten Sie den Agent-Daemon neu, wenn Sie diesen Parameter ändern.

Gehen Sie dann beim Konfigurieren einer neuen Aktion unter *Benachrichtigungen → Aktionen → Auslöser-Aktionen* wie folgt vor:

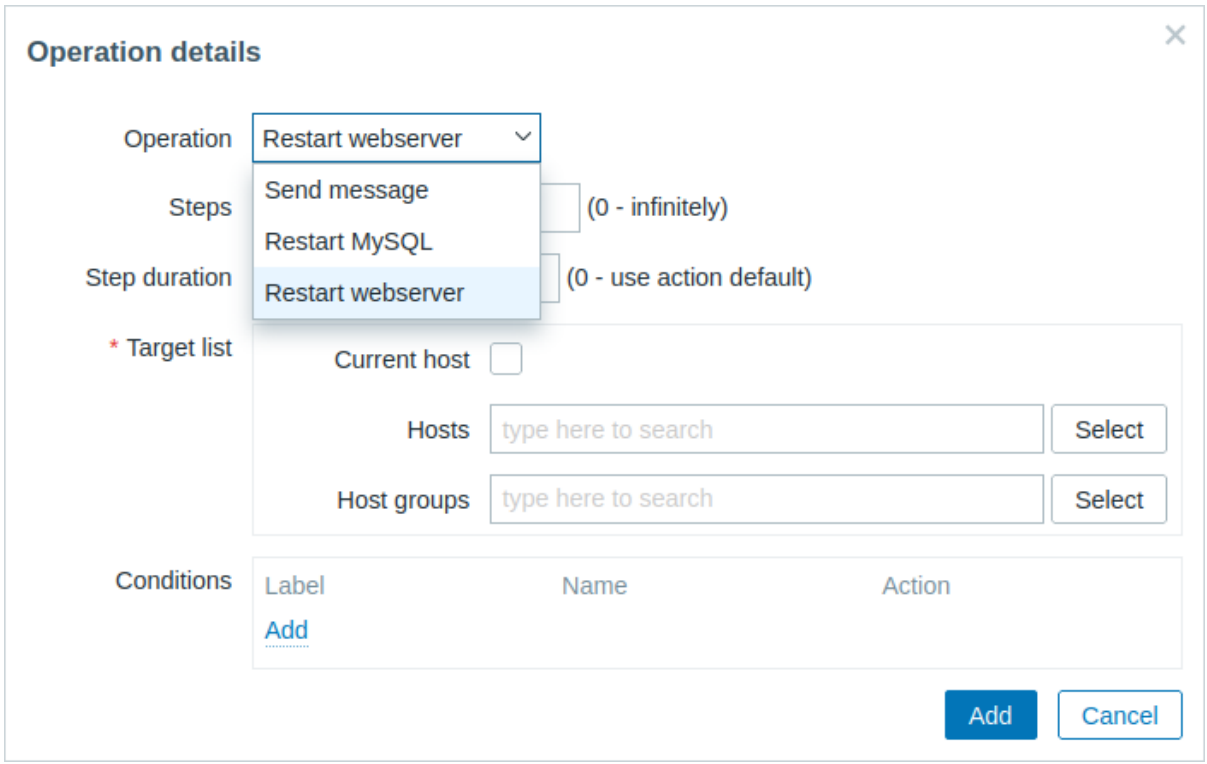
1. Definieren Sie die entsprechenden Bedingungen, zum Beispiel, dass die Aktion bei allen Katastrophenproblemen einer der Apache-Anwendungen aktiviert wird.



2. Klicken Sie auf der Registerkarte *Operationen* im Block *Operationen*, *Wiederherstellungsoperationen* oder *Aktualisierungsoperationen* auf *Hinzufügen*.



3. Wählen Sie in der Dropdown-Liste *Operation* eines der vordefinierten Skripte aus und legen Sie die *Zielliste* für das Skript fest.



Vordefinierte Skripte

Skripte, die für Aktionsoperationen (webhook, Skript, SSH, Telnet, IPMI) verfügbar sind, werden in den globalen Skripten 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 bei *Ausführen auf* auf die Schaltfläche *Zabbix agent*).

Attention:
Beachten Sie die Verwendung von **sudo** – der Zabbix-Benutzer verfügt standardmäßig nicht über die Berechtigung, Systemdienste neu zu starten. Hinweise zur Konfiguration von **sudo** finden Sie weiter unten.

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 Remote-Host laufen und führt die Befehle im Hintergrund aus.

Remote-Befehle auf dem Zabbix Agent werden ohne Zeitüberschreitung über den Schlüssel system.run[,nowait] ausgeführt und ihre Ausführungsergebnisse werden nicht geprüft.
Auf Zabbix Server und Zabbix Proxy werden Remote-Befehle mit der Zeitüberschreitung ausgeführt, die im Parameter TrapperTimeout der Datei zabbix_server.conf bzw. zabbix_proxy.conf festgelegt ist, und ihre Ausführungsergebnisse werden geprüft.
Weitere Informationen finden Sie unter *Skript-Zeitüberschreitung*.

Zugriffsberechtigungen

Stellen Sie sicher, dass der Benutzer 'zabbix' Ausführungsberechtigungen für die konfigurierten Befehle hat. Es kann sinnvoll sein, **sudo** zu verwenden, um Zugriff auf privilegierte Befehle zu gewähren. Um den Zugriff zu konfigurieren, führen Sie als root Folgendes 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 einer bestimmten Bedingung.

Um Windows bei einem von Zabbix erkannten Problem automatisch neu zu starten, definieren Sie das folgende Skript:

Skriptparameter	Wert
Bereich	'Aktionsoperation'
Typ	'Skript'
Befehl	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

3 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.

4 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
```

3 Wiederherstellungsoperationen

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:

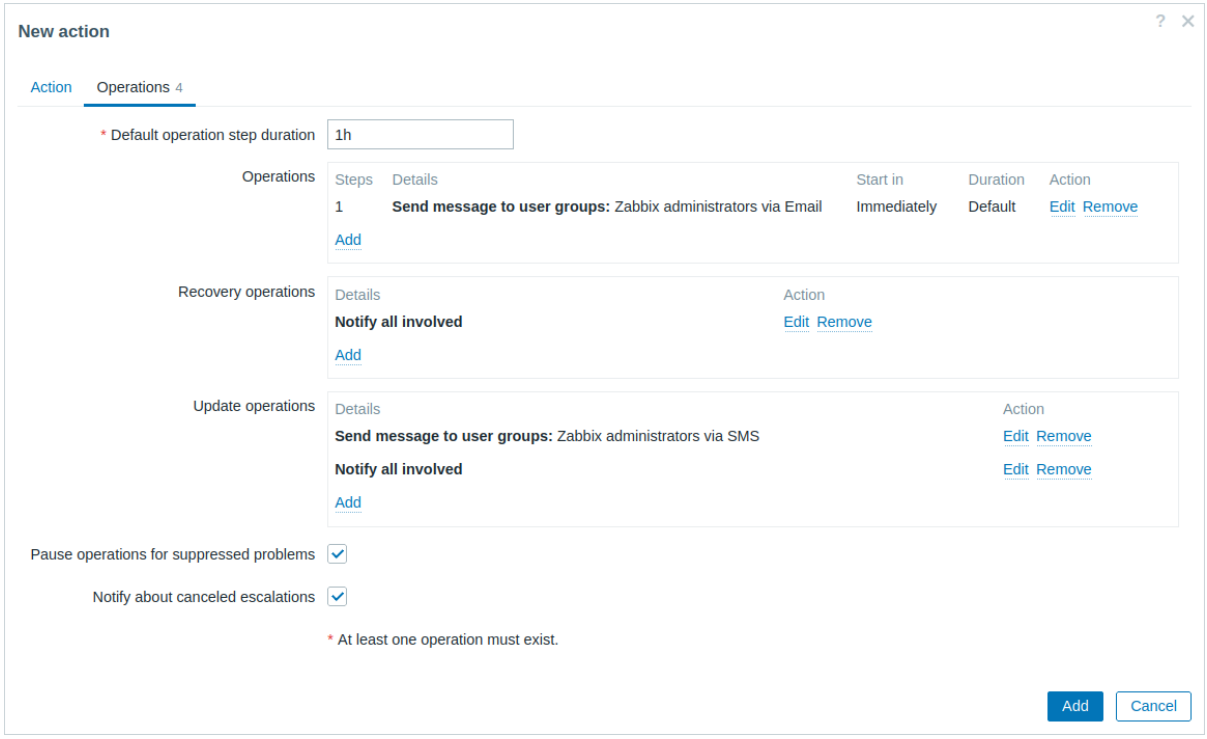
1. Bei einer Wiederherstellung alle Benutzer benachrichtigen, die über das Problem benachrichtigt wurden:
   - Wählen Sie *Alle Beteiligten benachrichtigen* als Operationstyp aus.
2. Mehrere Operationen bei einer Wiederherstellung ausführen: eine Benachrichtigung senden und einen Remote-Befehl ausführen:
   - 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

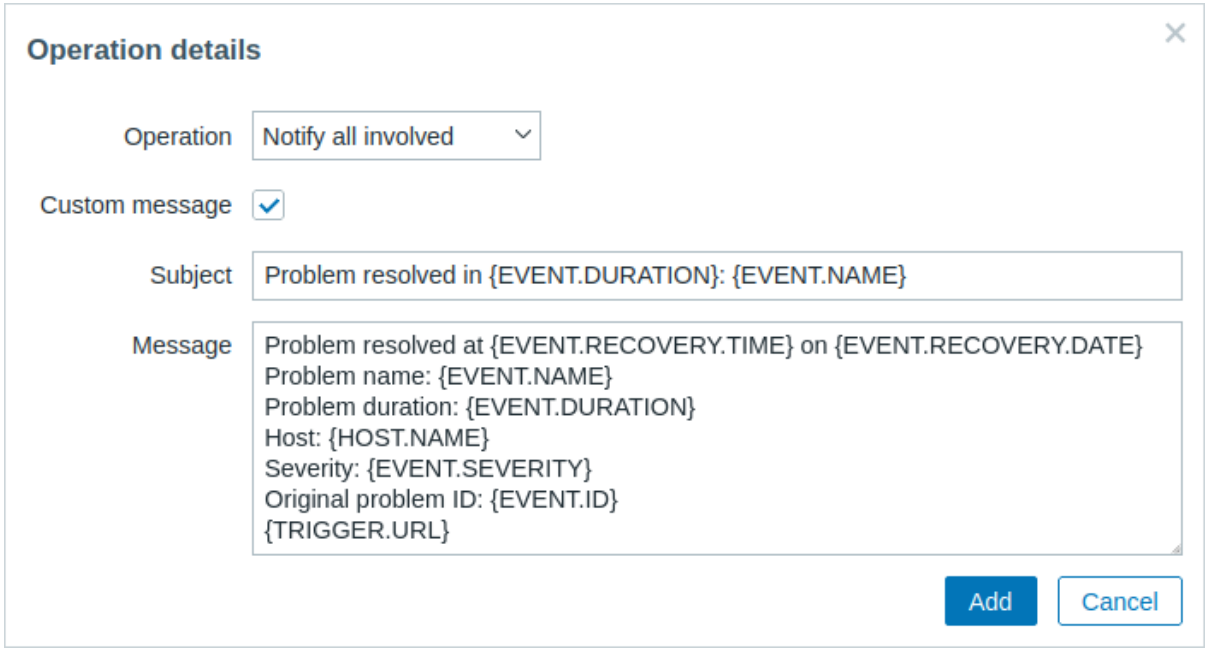
Gehen Sie wie folgt vor, um eine Wiederherstellungsoperation zu konfigurieren:

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 können ausgeführt werden, wenn sie zuvor in globalen Skripten definiert wurden und *Aktionsoperation* als Geltungsbereich ausgewählt ist.

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 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 sich mehrere Hosts im Auslöser befinden. **Host** - wählen Sie den/die Host(s) aus, auf dem/denen der Befehl ausgeführt werden soll. **Host-Gruppe** - wählen Sie die Host-Gruppe(n) aus, auf der/denen der Befehl ausgeführt werden soll. Wenn eine übergeordnete Host-Gruppe angegeben wird, werden implizit alle untergeordneten Host-Gruppen ausgewählt. Daher wird der Remote-Befehl auch auf Hosts aus untergeordneten Gruppen ausgeführt. Ein Befehl auf einem Host wird nur einmal ausgeführt, auch wenn der Host mehr als einmal übereinstimmt (z. B. aus mehreren Host-Gruppen; einzeln und aus einer Host-Gruppe). Die Zielliste ist bedeutungslos, wenn der Befehl auf dem Zabbix Server ausgeführt wird. Wenn in diesem Fall weitere Ziele ausgewählt werden, führt dies nur dazu, dass der Befehl auf dem Server mehrfach ausgeführt wird. Beachten Sie, dass bei globalen Skripten die Zielauswahl auch von der Einstellung *Host-Gruppe* 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.

4 Aktualisierungsoperationen

Übersicht

Aktualisierungsoperationen sind in Aktionen mit den folgenden Ereignisquellen verfügbar:

- *Auslöser* - wenn Probleme von anderen Benutzern aktualisiert werden, d. h. kommentiert, bestätigt, der Schweregrad geändert oder (manuell) geschlossen wurde;

- *Services* - wenn sich der Schweregrad eines Service geändert hat, der Service aber noch nicht wiederhergestellt ist.

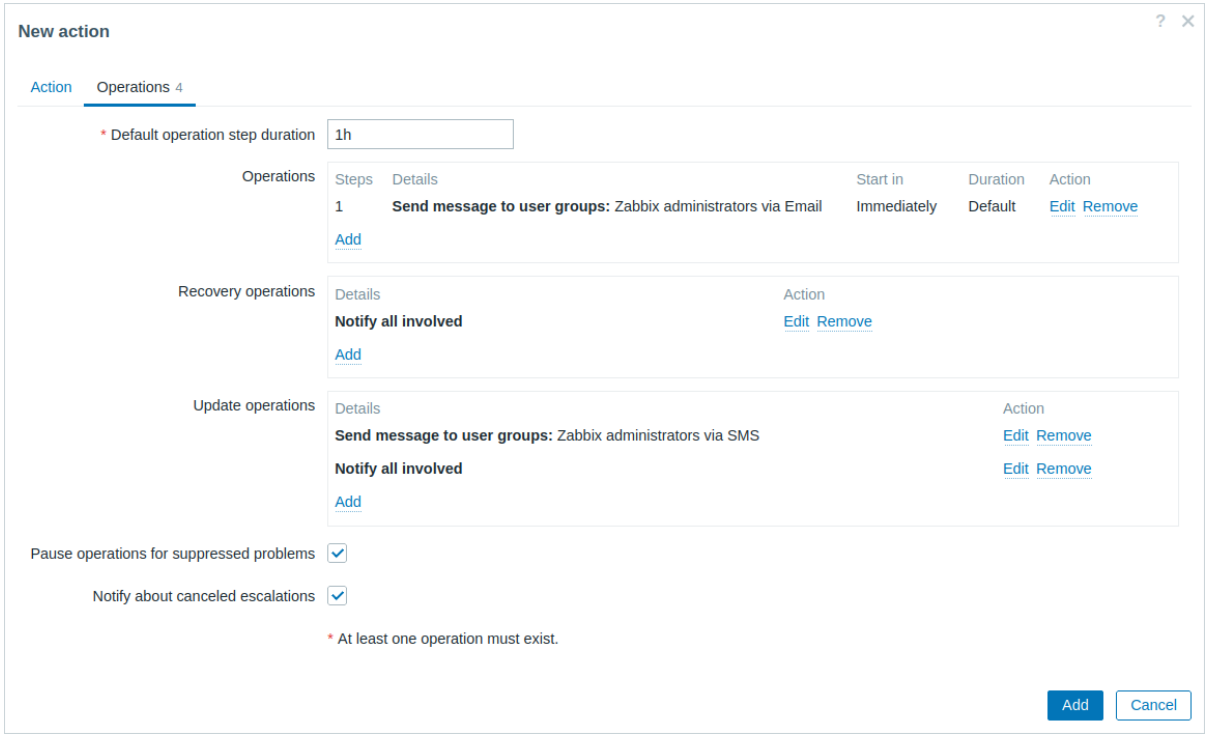
Bitte beachten Sie, dass Benutzer keine Benachrichtigungen über ihre eigenen Aktualisierungen erhalten.

Sowohl Nachrichten als auch Remote-Befehle werden in Aktualisierungsoperationen unterstützt. Es können zwar mehrere Operationen hinzugefügt werden, eine Eskalation wird jedoch nicht unterstützt - alle Operationen werden einem einzelnen Schritt zugewiesen und daher gleichzeitig ausgeführt.

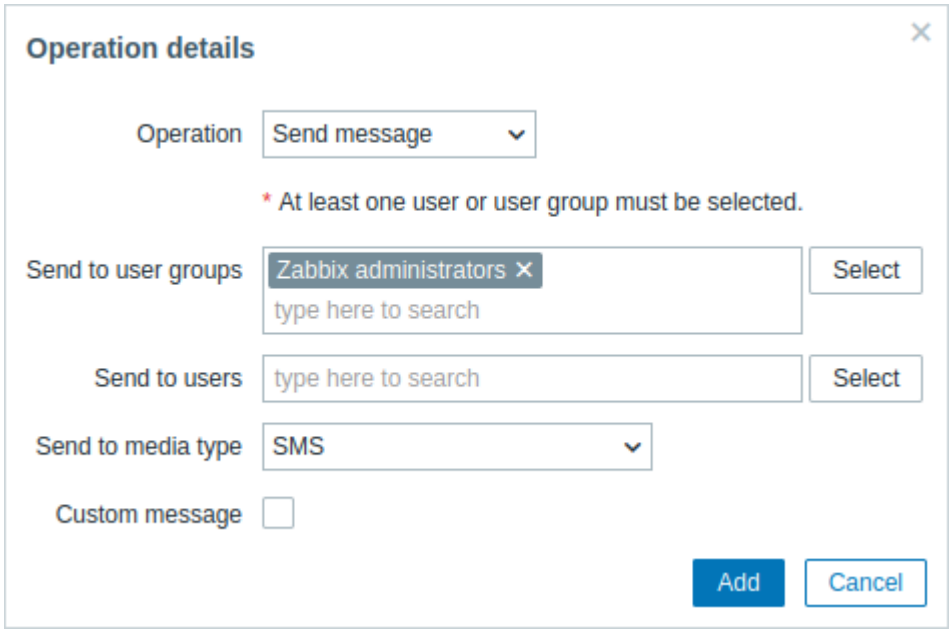
Konfigurieren einer Aktualisierungsoperation

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.

5 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 anhand des **Eskalationsschritts** 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 ab jedem Schritt 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 die Wiederherstellung.

Verschiedene Aspekte des Eskalationsverhaltens

Betrachten wir, was unter verschiedenen Umständen passiert, wenn eine Aktion mehrere Eskalationsschritte enthält.

Situation	Verhalten
Der betreffende Host geht in die Wartung, nachdem die erste Problembenachrichtigung gesendet wurde	Abhängig von der Einstellung *Operationen für unterdrückte Probleme pausieren* in der Aktions-Konfiguration werden alle verbleibenden Eskalationsschritte entweder mit einer durch den Wartungszeitraum verursachten Verzögerung oder ohne Verzögerung ausgeführt. Ein Wartungszeitraum bricht Operationen nicht ab.
*Der in der Aktionsbedingung **Zeitperiode** definierte Zeitraum endet, nachdem die erste Benachrichtigung gesendet wurde*	Alle verbleibenden Eskalationsschritte werden ausgeführt. Die Bedingung *Zeitperiode* kann Operationen nicht stoppen; sie wirkt sich darauf aus, wann Aktionen gestartet bzw. nicht gestartet werden, nicht auf Operationen.
Ein Problem beginnt während der Wartung und besteht nach dem Ende der Wartung weiter (wird nicht behoben)	Abhängig von der Einstellung *Operationen für unterdrückte Probleme pausieren* in der Aktions-Konfiguration werden alle Eskalationsschritte entweder ab dem Zeitpunkt ausgeführt, an dem die Wartung endet, oder sofort.
Ein Problem beginnt während einer Keine-Daten-Wartung und besteht nach dem Ende der Wartung weiter (wird nicht behoben)	Es muss gewartet werden, bis der Auslöser auslöst, bevor alle Eskalationsschritte ausgeführt werden.
Verschiedene Eskalationen folgen in kurzer Folge aufeinander und überlappen sich	Die Ausführung jeder neuen Eskalation ersetzt die vorherige Eskalation, jedoch wird bei der vorherigen Eskalation immer mindestens ein Eskalationsschritt ausgeführt. Dieses Verhalten ist relevant bei Aktionen für Ereignisse, die bei JEDER Problemauswertung des Auslösers erstellt werden.

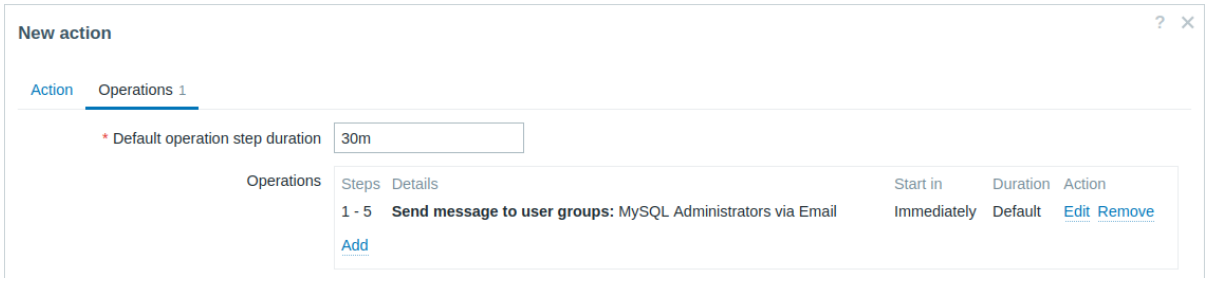
Situation	Verhalten
Während einer laufenden Eskalation (z. B. beim Senden einer Nachricht), basierend auf einem beliebigen Ereignistyp: - die Aktion wird deaktiviert Basierend auf einem Auslöser-Ereignis: - der Auslöser wird deaktiviert - der Host oder der Datenpunkt wird deaktiviert Basierend auf einem internen Ereignis über Auslöser: - der Auslöser wird deaktiviert Basierend auf einem internen Ereignis über Datenpunkte/Low-Level-Discovery-Regeln: - der Datenpunkt wird deaktiviert - der Host wird deaktiviert	Die gerade laufende Nachricht wird gesendet, und danach wird noch eine weitere Nachricht in der Eskalation gesendet. Die Folgenachricht enthält am Anfang des Nachrichtentextes den Abbruchtext (*HINWEIS: Eskalation abgebrochen*) mit Angabe des Grundes (zum Beispiel *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 Grund für den Abbruch wird auch in die Server-Logdatei geschrieben (ab Debug Level 3=Warning). Beachten Sie, dass die Nachricht *Eskalation abgebrochen* auch gesendet wird, wenn die Operationen abgeschlossen sind, aber Wiederherstellungsoperationen konfiguriert sind und noch nicht ausgeführt wurden.
Während einer laufenden Eskalation (z. B. beim Senden einer Nachricht) wird die Aktion gelöscht	Es werden keine weiteren Nachrichten gesendet. Die Information wird in die Server-Logdatei geschrieben (ab Debug Level 3=Warning), 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).

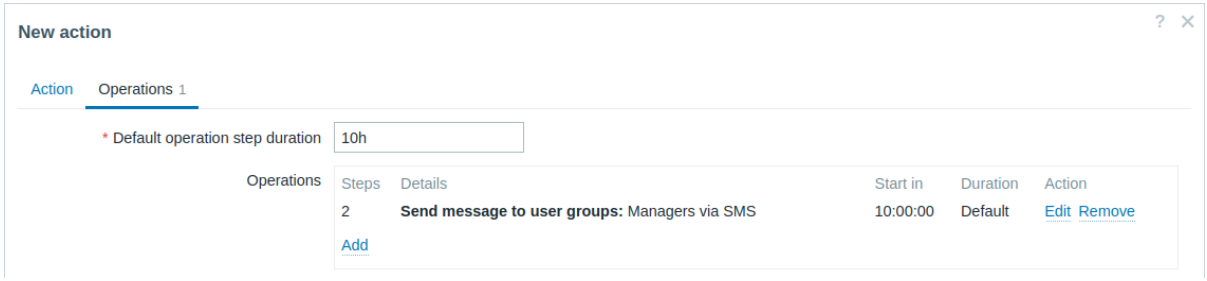
Wenn das Problem behoben wird und eine Wiederherstellungsnachricht konfiguriert ist, wird sie an diejenigen gesendet, die innerhalb dieses Eskalationsszenarios mindestens eine Problembenachrichtigung erhalten haben.

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".



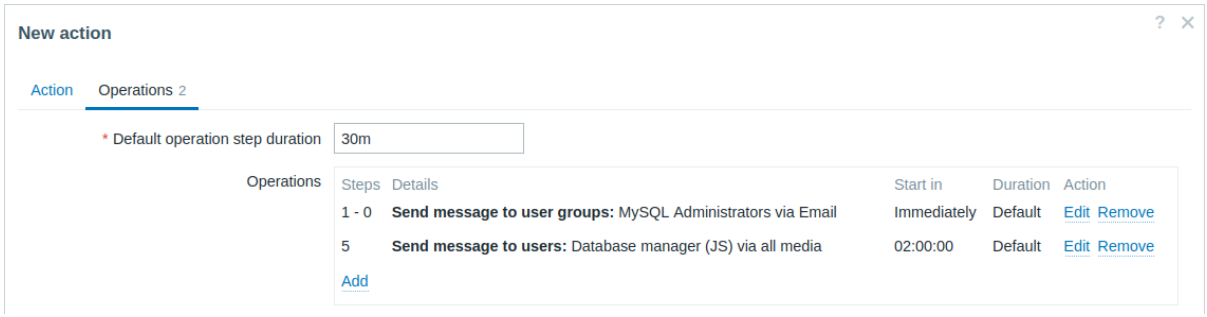
Eine Benachrichtigung wird nur in Schritt 2 des Eskalationsszenarios gesendet, also 10 Stunden nach Beginn des Problems.

Sie können den Nachrichtentext beispielsweise in „Das Problem besteht seit mehr als 10 Stunden“ ändern.

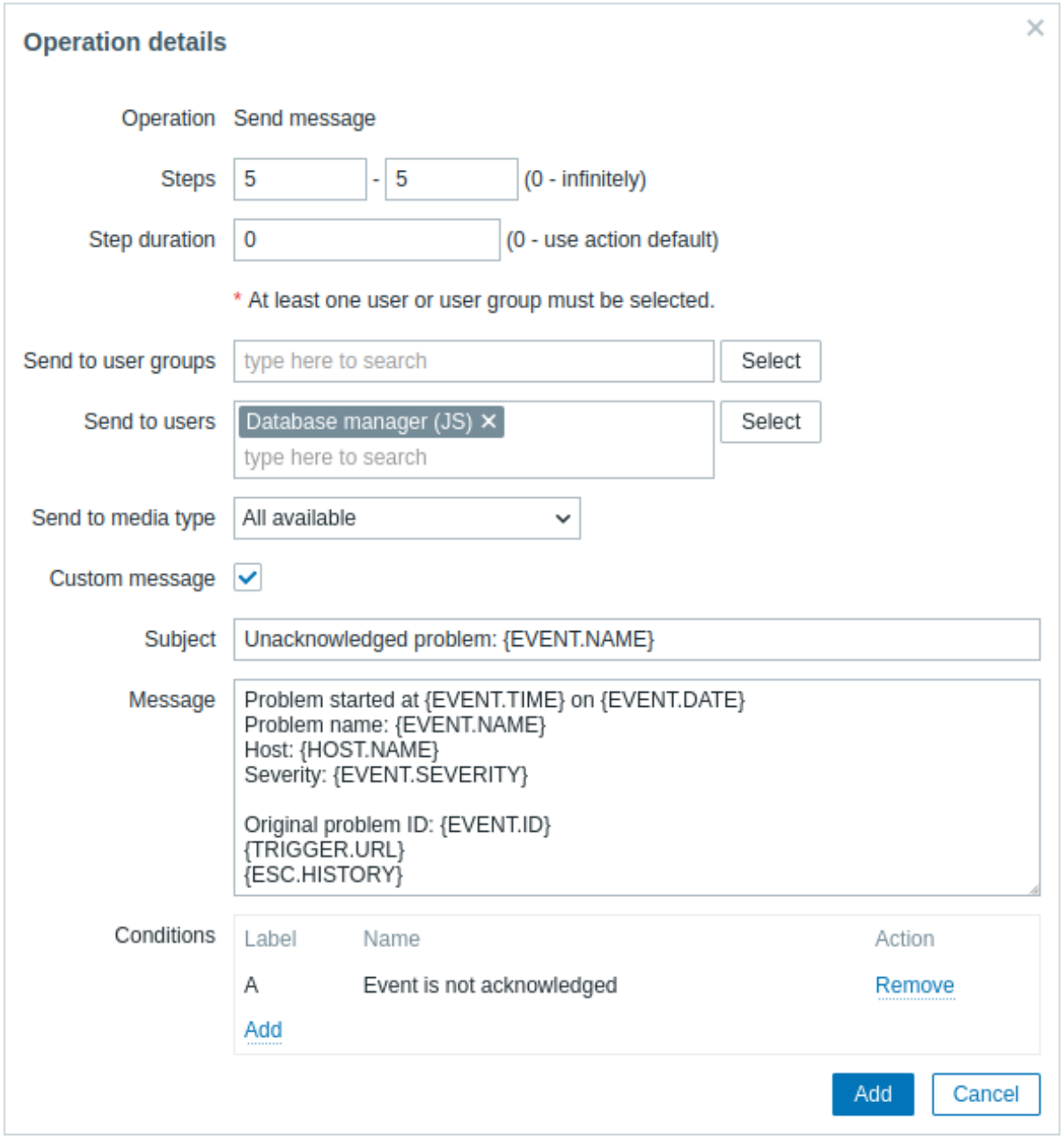
Beispiel 3

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:



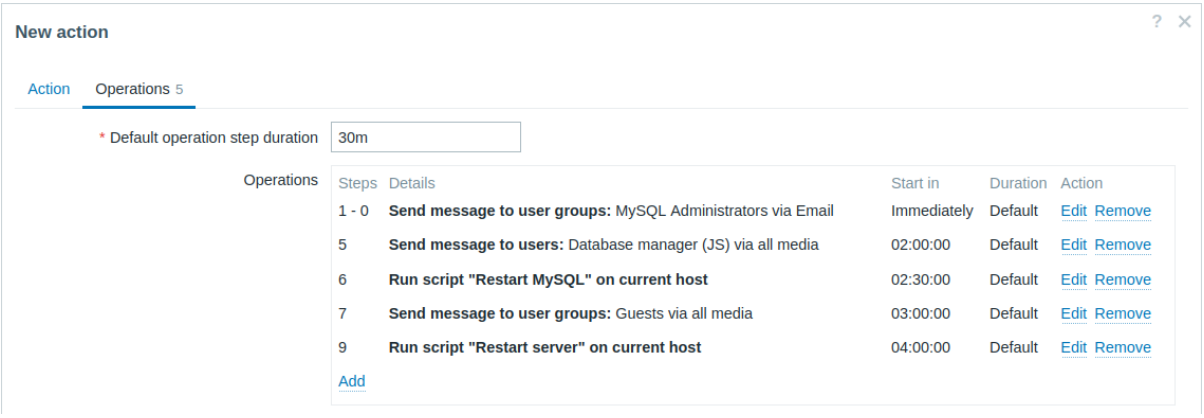
Beachten Sie die Verwendung des Makros {ESC.HISTORY} in der angepassten Nachricht. Das Makro enthält Informationen über alle zuvor ausgeführten Schritte dieser Eskalation, wie gesendete Benachrichtigungen und ausgeführte Befehle.

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.

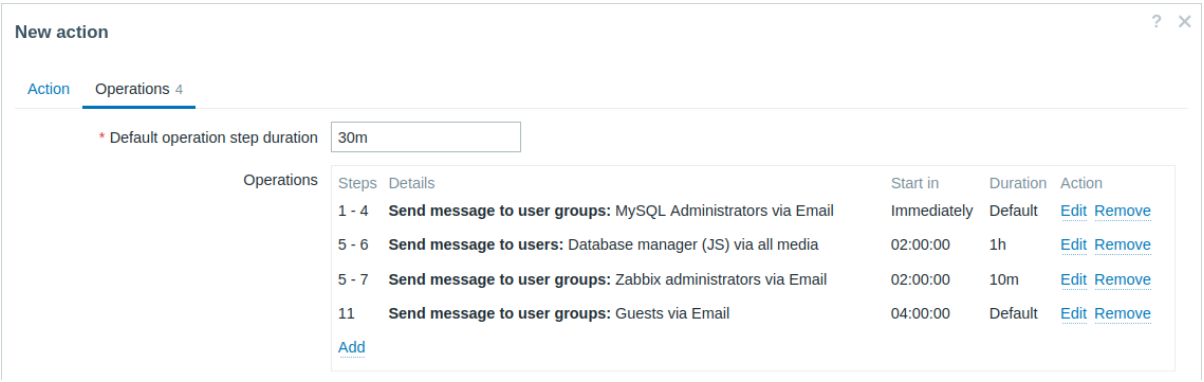
Wenn das Problem weiterhin besteht, sendet Zabbix nach weiteren 30 Minuten eine Nachricht an alle Gastbenutzer.

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).

3 Benachrichtigung bei nicht unterstützten Datenpunkten erhalten

Übersicht

Es ist möglich, Benachrichtigungen über nicht unterstützte Datenpunkte in Zabbix zu erhalten.

Dies ist Teil des Konzepts der internen Ereignisse in Zabbix, das es Benutzern ermöglicht, bei diesen Gelegenheiten benachrichtigt zu werden. Interne Ereignisse spiegeln eine Zustandsänderung wider:

- wenn Datenpunkte von „normal“ zu „nicht unterstützt“ wechseln (und zurück);
- wenn Auslöser von „normal“ zu „unbekannt“ wechseln (und zurück);
- wenn Low-Level-Discovery-Regeln von „normal“ zu „nicht unterstützt“ wechseln (und zurück).

Dieser Abschnitt zeigt eine Anleitung zum **Empfangen von Benachrichtigungen**, wenn ein Datenpunkt nicht mehr 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 die Standard- Schwere (’Not classified’) verwendet. Lassen Sie diese daher bei der Konfiguration von Benutzermedien aktiviert, wenn Sie Benachrichtigungen für interne Ereignisse erhalten möchten.

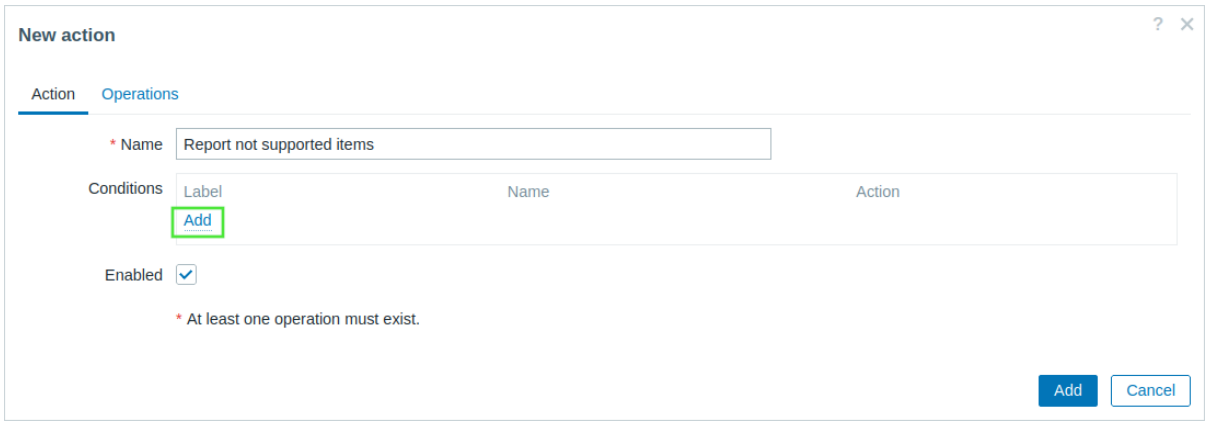
Schritt 2

Gehen Sie zu *Benachrichtigungen* → *Aktionen* → *Interne Aktionen*.

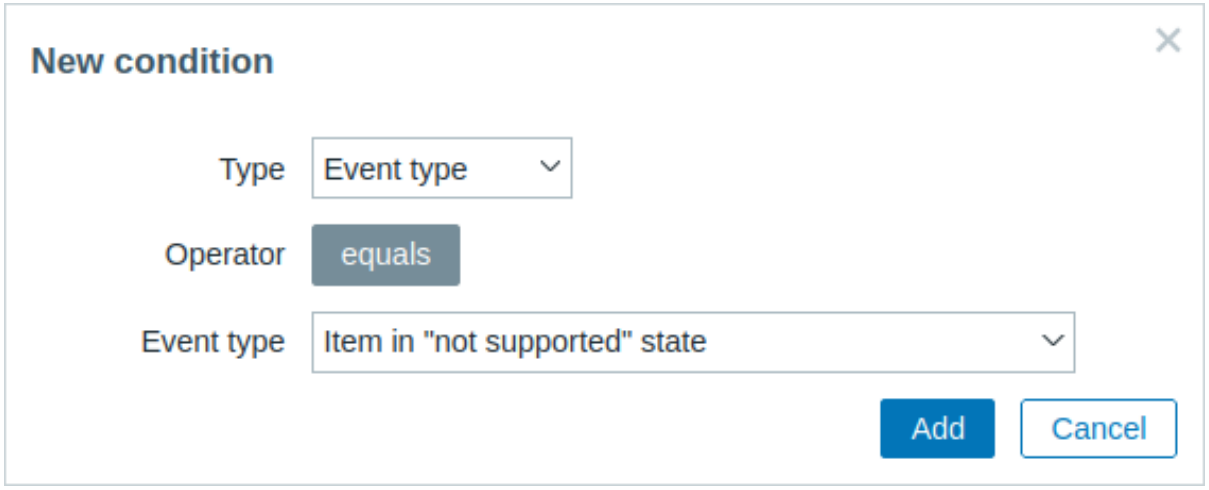
Klicken Sie oben rechts auf der Seite auf *Aktion erstellen*, um ein Formular zur Aktionskonfiguration zu öffnen.

Schritt 3

Geben Sie auf der Registerkarte *Aktion* einen Namen für die Aktion ein. Klicken Sie dann im Block *Bedingungen* auf *Hinzufügen*, um eine neue Bedingung hinzuzufügen.



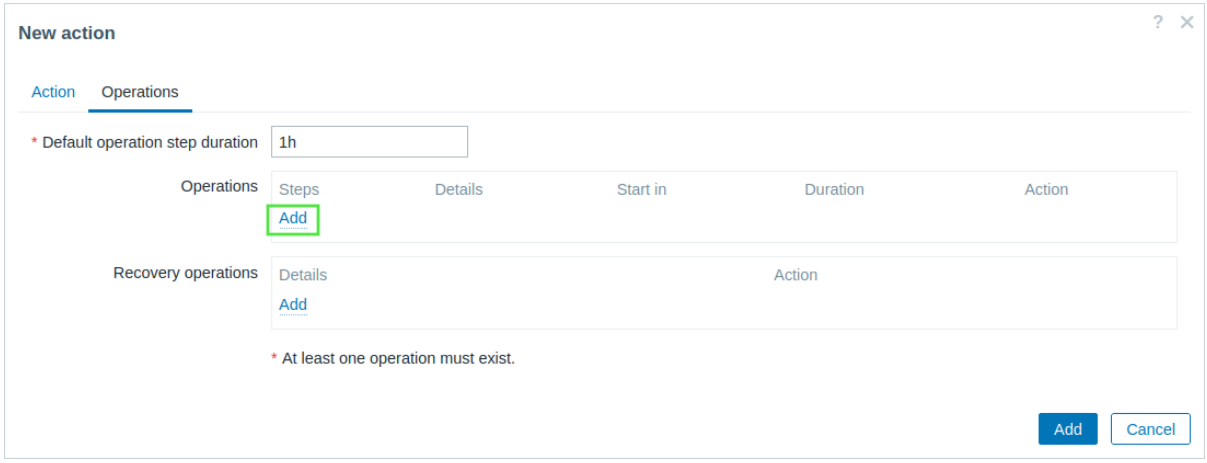
Wählen Sie im Pop-up-Fenster *Neue Bedingung* als Bedingungstyp „Ereignistyp“ aus und wählen Sie dann als Ereignistyp „Datenpunkt im Zustand ‚nicht unterstützt‘ “ aus.



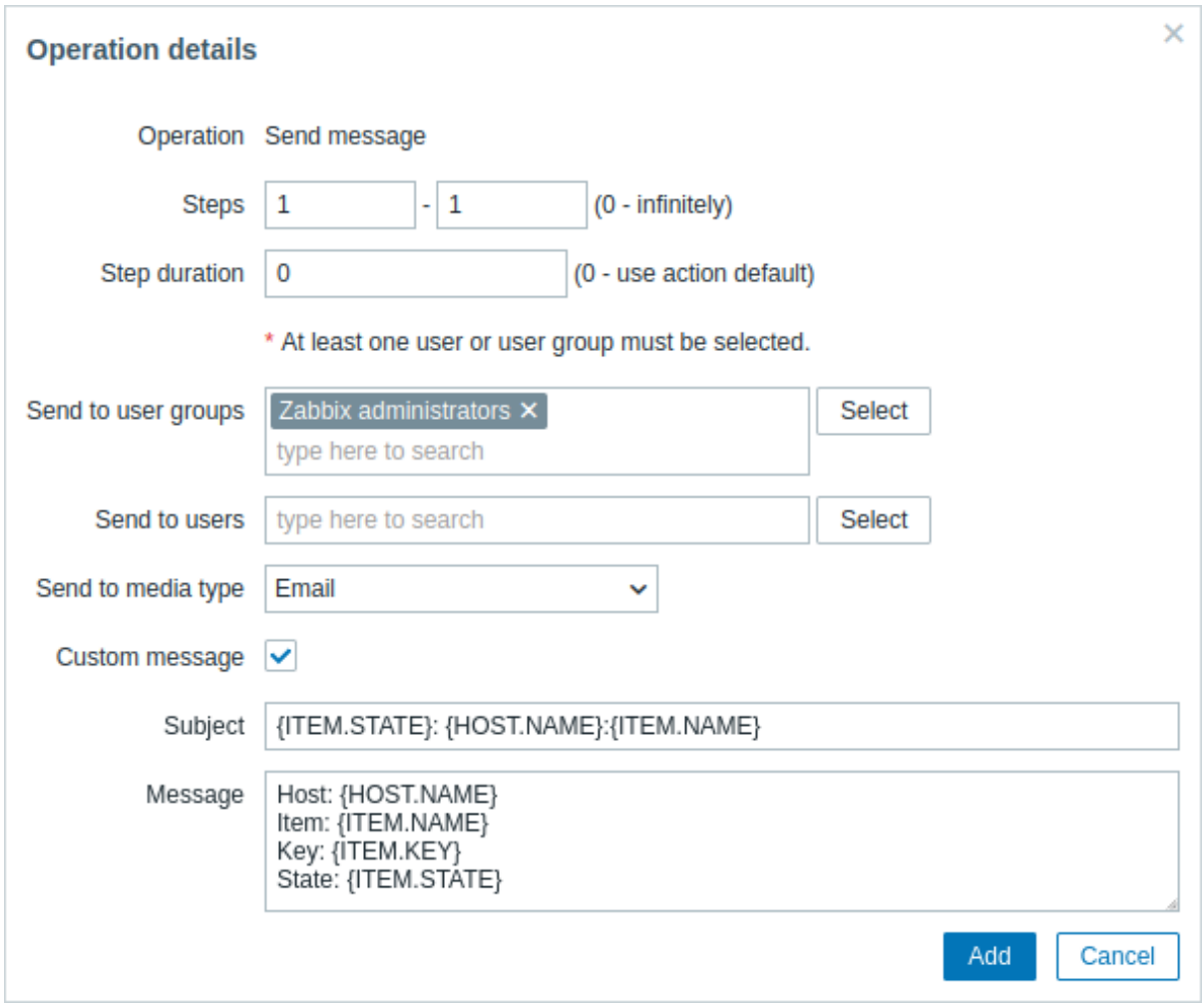
Vergessen Sie nicht, auf *Hinzufügen* zu klicken, damit die Bedingung tatsächlich im Block *Bedingungen* aufgeführt wird.

Schritt 4

Klicken Sie auf der Registerkarte *Operations* im Block *Operations* auf *Add*, um eine neue Operation hinzuzufügen.



Wählen Sie einige Empfänger der Nachricht (Benutzergruppen/Benutzer) sowie den Medientyp (oder „All“) für die Zustellung aus. Aktivieren Sie das Kontrollkästchen *Custom message*, wenn Sie einen benutzerdefinierten Betreff/Inhalt der Problemnachricht eingeben möchten.

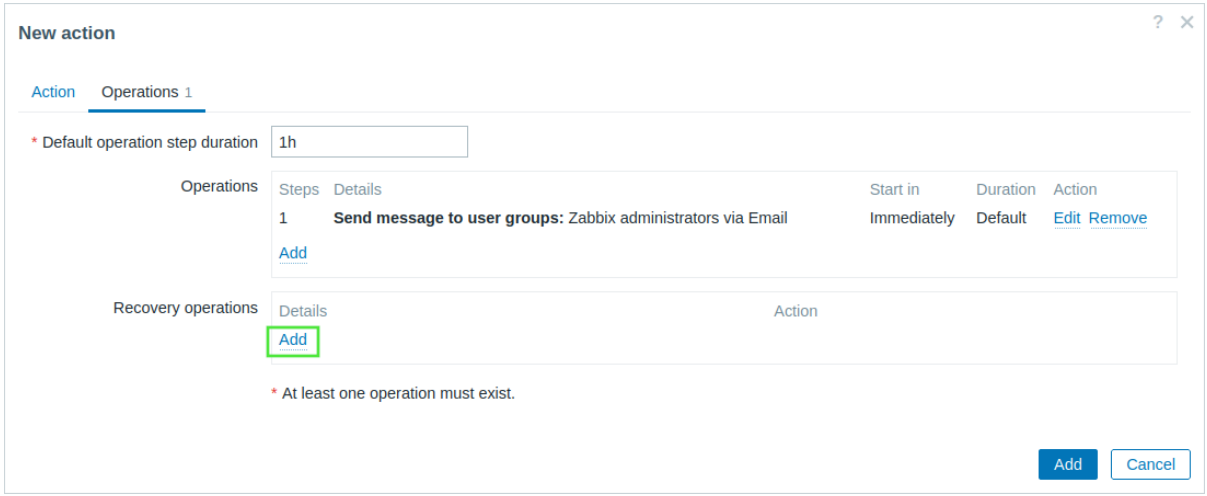


Klicken Sie auf *Add*, um die Operation tatsächlich im Block *Operations* aufzulisten.

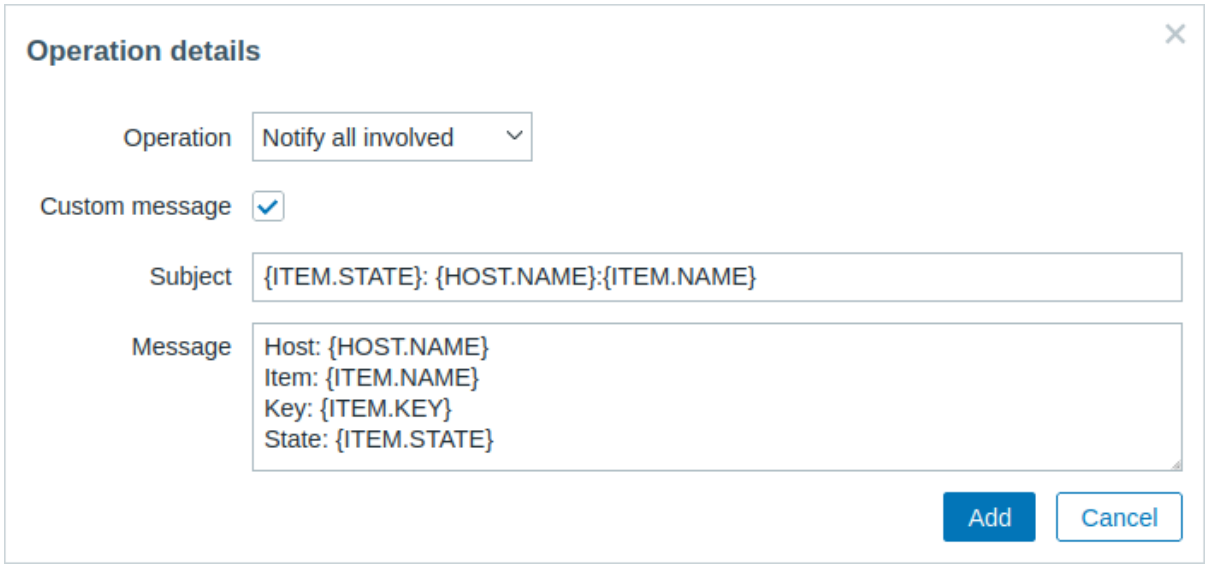
Wenn Sie mehr als eine Benachrichtigung erhalten möchten, legen Sie die Schrittdauer der Operation (Intervall zwischen gesendeten Nachrichten) fest und fügen Sie einen weiteren Schritt hinzu.

Schritt 5

Der Block *Wiederherstellungsaktionen* ermöglicht die Konfiguration einer Wiederherstellungsbenachrichtigung, wenn ein Datenpunkt in den normalen Zustand zurückkehrt. Klicken Sie im Block *Wiederherstellungsaktionen* auf *Hinzufügen*, um eine neue Wiederherstellungsaktion hinzuzufügen.



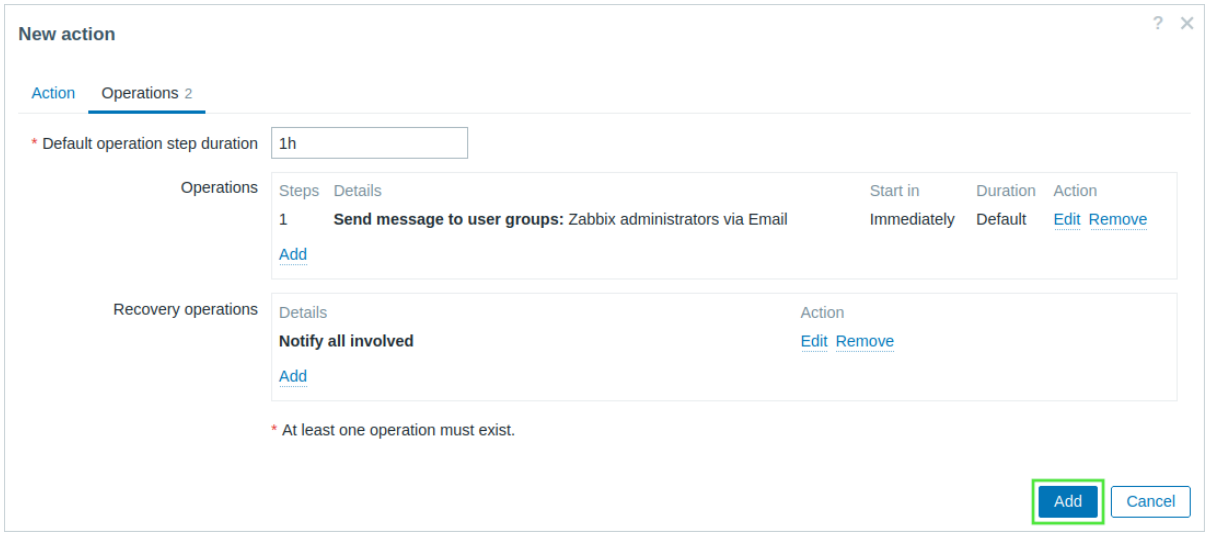
Wählen Sie den Aktionstyp „Alle Beteiligten benachrichtigen“ aus. Aktivieren Sie das Kontrollkästchen *Benutzerdefinierte Nachricht*, wenn Sie einen benutzerdefinierten Betreff/Inhalt der Wiederherstellungsnachricht eingeben möchten.



Klicken Sie im Pop-up-Fenster *Aktionsdetails* auf *Hinzufügen*, damit die Aktion tatsächlich im Block *Wiederherstellungsaktionen* aufgeführt wird.

Schritt 6

Wenn Sie fertig sind, klicken Sie unten im Formular auf die Schaltfläche *Hinzufügen*.



Und das war’s, Sie sind fertig! Jetzt können Sie sich darauf freuen, Ihre erste Benachrichtigung von Zabbix zu erhalten, wenn ein Datenpunkt in den Status „Nicht unterstützt“ wechselt.

11 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 spart Zeit und macht die Zabbix- Konfiguration transparenter.

In einem typischen Anwendungsfall kann ein Makro in einer Vorlage verwendet werden. So kann ein Auslöser in einer Vorlage beispielsweise „Prozessorlast ist auf {HOST.NAME} zu hoch“ heißen. Wenn die Vorlage auf den Host angewendet wird, z. B. auf den Zabbix Server, wird der Name zu „Prozessorlast ist auf Zabbix server zu hoch“ aufgelöst, wenn der Auslöser im Abschnitt „Monitoring“ angezeigt wird.

Makros können in Datenpunkt-Schlüsselparametern verwendet werden. Ein Makro kann auch nur für einen Teil des Parameters verwendet werden, zum Beispiel `item.key[server_{HOST.HOST}_local]`. Es ist nicht erforderlich, den Parameter in doppelte Anführungszeichen zu setzen, da Zabbix sich um mehrdeutige Sonderzeichen kümmert, falls solche im aufgelösten Makro enthalten sind.

Es gibt in Zabbix noch weitere Arten von Makros.

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 Low-Level-Discovery
- `{?EXPRESSION}` - Ausdrucksmakro

1 Makrofunktionen

Übersicht

Makrofunktionen bieten die Möglichkeit, Werte von Makros anzupassen (zum Beispiel bestimmte Teilzeichenfolgen zu kürzen oder zu extrahieren), sodass sie sich leichter verwenden lassen.

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
- Low-level-discovery-Makros
- 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 Filtern von Low-level-discovery-Regeln).

Pro Makro wird nur eine einzelne Funktion unterstützt; mehrere verkettete Makrofunktionen werden nicht unterstützt.

Note:
Bitte beachten Sie die Escaping-Beispiele für Fälle, in denen Makrofunktionen in anderen Kontexten verwendet werden (Funktion, Datenpunktschlüssel, anderes Makro usw.).

Unterstützte Funktionen

Die Funktionen sind ohne zusätzliche Informationen aufgelistet. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
btoa	Kodierung des Makrowerts in Base64-Kodierung.
fmtnum	Zahlenformatierung zur Steuerung der Anzahl der nach dem Dezimalpunkt ausgegebenen Ziffern.
fmttime	Zeitformatierung.
htmldecode	Dekodierung des Makrowerts aus der HTML-Kodierung.
htmlencode	Kodierung des Makrowerts in HTML-Kodierung.
iregsub	Extraktion einer Teilzeichenfolge anhand einer Übereinstimmung mit einem regulären Ausdruck (Groß-/Kleinschreibung wird nicht beachtet).
lowercase	Umwandlung der Zeichen des Makrowerts in Kleinbuchstaben.
regrepl	Ersetzung eines Zeichens/einer Teilzeichenfolge im Makrowert.
regsub	Extraktion einer Teilzeichenfolge anhand einer Übereinstimmung mit einem regulären Ausdruck (Groß-/Kleinschreibung wird beachtet).
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.

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 zu einem Wert in einem der folgenden Zeitformate aufgelöst werden:

- `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; sie sollte mit `-<N><time_unit>` oder `+<N><time_unit>` beginnen, wobei:
  - `N` - die Anzahl der Zeiteinheiten, die addiert oder subtrahiert werden sollen;
  - `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 auf den Anfang der Zeiteinheit zu verschieben (`/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 ohne Prioritäten von links nach rechts berechnet. Zum Beispiel wird `-1M/d+1h/w` als `((-1M/d)+1h)/w` geparst.

Beispiele:

```
{{TIME}.fmttime(%B)} - gibt "October" für den empfangenen Wert "1633098961" zurück
{{TIME}.fmttime(%d %B,-1M/M)} - gibt "1 September" für den empfangenen Wert "1633098961" zurück
```

htmldecode

Dekodieren eines Makrowerts aus der HTML-Kodierung.

Die folgenden Zeichen werden unterstützt:

Wert	Dekodierter Wert
&amp;	&
&lt;	<
&gt;	>
&quot;	"
'	'
'	'

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
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	'

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
```

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 abzugleichende reguläre Ausdruck;
- **replacement** - die Ersetzungszeichenfolge. **\1 - \9** Platzhalter werden in Ersetzungszeichenfolgen für Erfassungsgruppen unterstützt.

Kommentare:

- Die Muster und Ersetzungen werden sequenziell 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 (zum Beispiel,
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - ersetzt mehrere Teile im Datenpunkt-Wert
```

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"
```

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:

Makrofunktion	Empfangener Wert	Ausgabe
{{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}.

2 Benutzermakros

Übersicht

Benutzermakros werden in Zabbix zusätzlich zu den standardmäßig unterstützten Makros unterstützt, um mehr Flexibilität zu bieten.

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 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 Vorlagen der zweiten Ebene des Hosts definiert sind, sortiert nach Vorlagen-ID
4. Makros, die für Vorlagen der dritten Ebene des Hosts definiert sind, sortiert nach Vorlagen-ID usw.
5. globale Makros (werden zuletzt geprüft)

Mit anderen Worten: Wenn ein Makro für einen Host nicht existiert, versucht Zabbix, es in den Host-Vorlagen mit zunehmender Tiefe zu finden. Wird es weiterhin nicht gefunden, wird ein globales Makro verwendet, falls vorhanden.

Warning:
Wenn ein Makro mit **demselben Namen** in mehreren verknüpften Vorlagen derselben Ebene vorhanden ist, wird das Makro aus der Vorlage mit der niedrigsten ID verwendet. Daher stellen 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 Abschnitt „Konfiguration“ (zum Beispiel in der Auslöserliste) absichtlich nicht aufgelöst, um komplexe Konfigurationen transparenter zu machen.

Benutzermakros können verwendet werden in:

- Datenpunkt-Name
- Datenpunkt-Schlüsselparameter
- Datenpunkt-Aktualisierungsintervallen und flexiblen Intervallen
- Auslösername und -beschreibung
- Auslöserausdrucksparametern und Konstanten (siehe Beispiele)
- vielen anderen Stellen – 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.

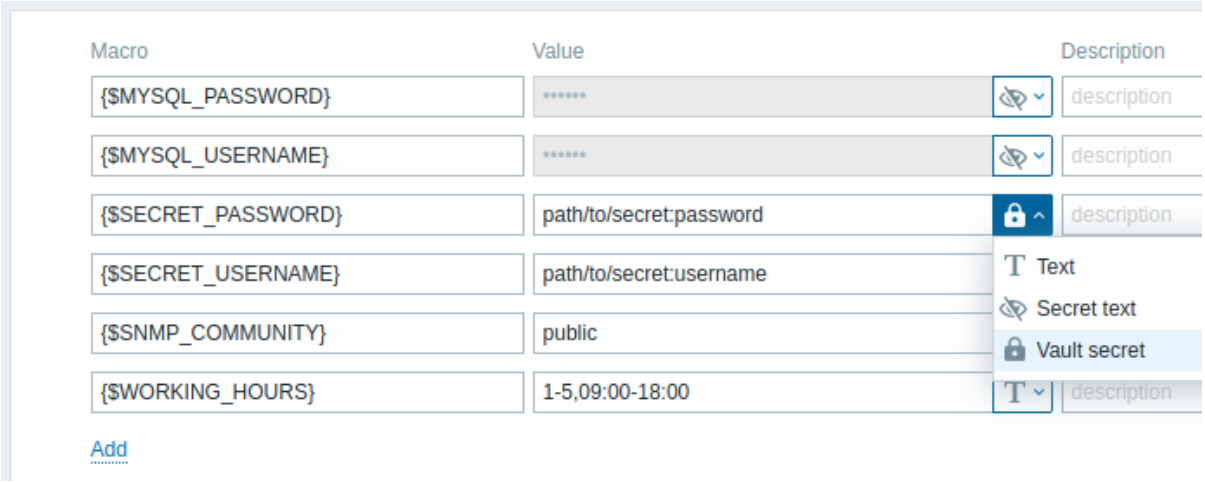
Note:
Es wird empfohlen, Host-Makros anstelle von globalen Makros zu verwenden, da das Hinzufügen, Aktualisieren oder Löschen globaler Makros ein inkrementelles Konfigurations-Update für alle Hosts erzwingt. Weitere Informationen finden Sie unter Passive und aktive Agent-Prüfungen.

Konfiguration

Um Benutzermakros zu definieren, gehen Sie im entsprechenden Bereich des Frontends wie folgt vor:

- für globale Makros öffnen Sie *Administration* → *Makros*
- für Makros auf Host- und Vorlagenebene öffnen Sie die Eigenschaften des Hosts oder der Vorlage und wechseln zur Registerkarte *Makros*

Ein Benutzermakro hat die folgenden Attribute:



Parameter	Beschreibung
Makro	Makroname. Der Name muss in geschweifte Klammern eingeschlossen sein und mit einem Dollarzeichen beginnen. Beispiel: {$FRONTEND_URL}. Die folgenden Zeichen sind in Makronamen zulässig: **A-Z** (nur Großbuchstaben), **0-9**, _, **.**
Wert	Makrowert. Es werden drei Werttypen unterstützt: **Text** (Standard) - Klartextwert **Geheimer Text** - der Wert wird mit Sternchen maskiert **Vault-Geheimnis** - der Wert enthält einen Pfad/eine Abfrage zu einem Vault-Geheimnis. Um den Werttyp zu ändern, klicken Sie auf die Schaltfläche am Ende des Werteingabefelds. Die maximale Länge eines Benutzermakrowerts beträgt 2048 Zeichen.

Parameter	Beschreibung
Beschreibung	Textfeld zur Bereitstellung weiterer Informationen zu diesem Makro.

Beachten Sie bei der Konfiguration von Benutzermakros die folgenden kontextspezifischen Verhaltensweisen:

- wenn Benutzermakros in Vorlagen-Datenpunkten oder Auslösern verwendet werden, sollten Sie diese Makros auch zur Vorlage hinzufügen (selbst 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 geheimer Makros werden nicht exportiert)
- wenn Benutzermakros in Auslöserausdrücken 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 Datenpunktschlüssel, eine Funktion, einen Operator oder einen anderen Auslöserausdruck verweisen (geheime Makros können nicht in Auslöserausdrücken verwendet werden)
- wenn Benutzermakros auf einem Host verwendet werden, der eine Low-Level-Discovery-Regel mit Host-Prototypen hat, übernehmen entdeckte Hosts alle auf diesem Host definierten Benutzermakros

Beispiele

Beispiel 1

Verwendung eines Makros auf Host-Ebene im Datenpunkt-Schlüssel „Status of SSH daemon“:

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Dieser Datenpunkt kann mehreren Hosts zugewiesen werden, vorausgesetzt, dass der Wert von **{$SSH_PORT}** auf diesen Hosts definiert ist.

Beispiel 2

Verwendung eines Makros auf Host-Ebene 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.

Note:
Wenn Sie die Anzahl der Werte als Funktionsparameter verwenden möchten (zum Beispiel **max(/host/key,#3)**), schließen Sie die Raute in die Makrodefinition wie folgt ein: SOME_PERIOD => #3

Beispiel 3

Verwendung von zwei Makros im Auslöser „CPU load is too high“:

```
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 die Funktion **min()**.

Beispiel 4

Synchronisieren Sie die Bedingung für die Nichtverfügbarkeit des Agent mit dem Aktualisierungsintervall des Datenpunkts:

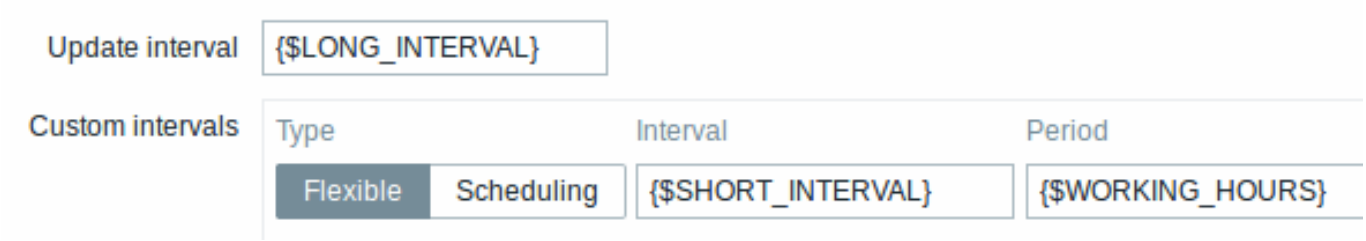
- Definieren Sie das Makro {$INTERVAL} und verwenden Sie es im Aktualisierungsintervall des Datenpunkts;
- verwenden Sie {$INTERVAL} als Parameter des Auslösers für die Nichtverfügbarkeit des Agent:

```
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 → General → GUI*.
- Verwenden Sie es im Feld *Aktiv wenn* unter *Users → Users*, Registerkarte *Media* eines Benutzers.
- Verwenden Sie es, um während der Arbeitszeiten ein häufigeres Abfragen von Datenpunkten einzurichten:

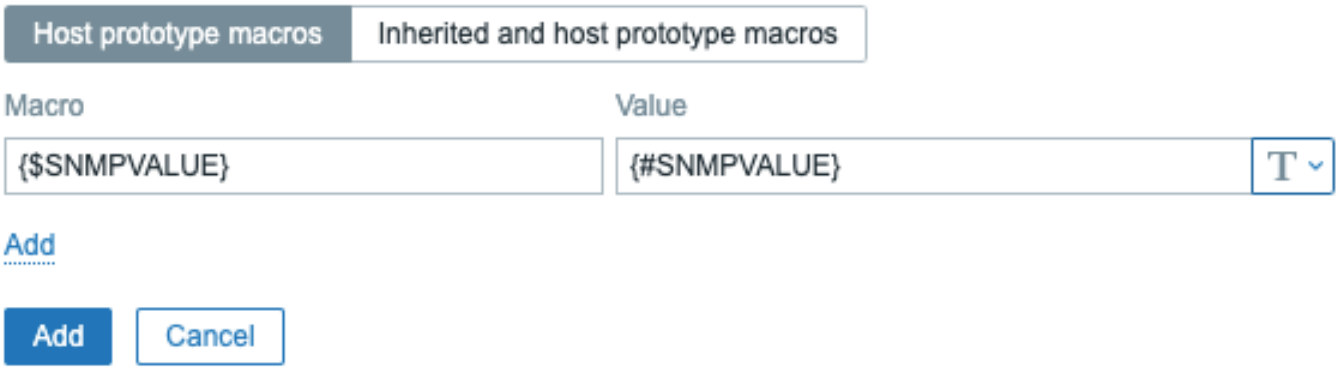


- Verwenden Sie es in der Aktionsbedingung *Zeitraum*.
- Passen Sie die Arbeitszeit bei Bedarf unter *Administration* → *Macros* an.

Beispiel 6

Verwenden Sie ein Host-Prototyp-Makro, um Datenpunkte für erkannte Hosts zu konfigurieren:

- Definieren Sie in einem Host-Prototyp das Benutzermakro {$SNMPVALUE} mit dem Makro {#SNMPVALUE} der Low-Level-Discovery als Wert:



- Weisen Sie dem Host-Prototyp die Vorlage *Generic SNMPv2* zu;
- verwenden Sie {$SNMPVALUE} im Feld *SNMP OID* der Datenpunkte der Vorlage *Generic SNMPv2*.

Kontext von Benutzermakros

Siehe Benutzermakros mit Kontext.

3 Benutzermakros mit Kontext

Übersicht

Ein optionaler Kontext kann in Benutzer- Makros 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 Kontext als regulärer Ausdruck nur in der Konfiguration von Benutzermakros definiert werden kann. Wenn das Präfix `regex:` an anderer Stelle als Kontext für ein Benutzermakro verwendet wird, wie z. B. in einem Auslöser-Ausdruck, wird es als statischer Kontext behandelt.

Die Anführungszeichen für den Kontext sind 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 mit dem Präfix /var/log/.

Anwendungsfälle

Benutzermakros mit Kontext können definiert werden, um flexiblere Schwellenwerte in Auslöserausdrücken zu ermöglichen (basierend auf den Werten, die durch Low-Level-Discovery abgerufen werden). Zum Beispiel können Sie 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 Makrokontext in einem Auslöserprototyp für die Discovery eingehängter Dateisysteme verwendet werden:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Nach der Discovery gelten in Auslösern unterschiedliche Schwellenwerte für wenig freien Speicherplatz, abhängig von den erkannten Einhängepunkten oder Dateisystemtypen. Problemereignisse werden erzeugt, wenn:

- der Ordner /home weniger als 20 % freien Festplattenspeicher hat
- Ordner, die dem Regexp-Muster entsprechen (wie /etc, /tmp oder /var), weniger als 30 % freien Festplattenspeicher haben
- Ordner, die dem Regexp-Muster nicht entsprechen und nicht /home sind, weniger als 10 % freien Festplattenspeicher 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.

4 Geheime Benutzermakros

Übersicht

Zabbix bietet zwei Optionen zum Schutz sensibler Informationen in Benutzer-Makrowerten:

- Geheimer Text
- Vault-Geheimnis

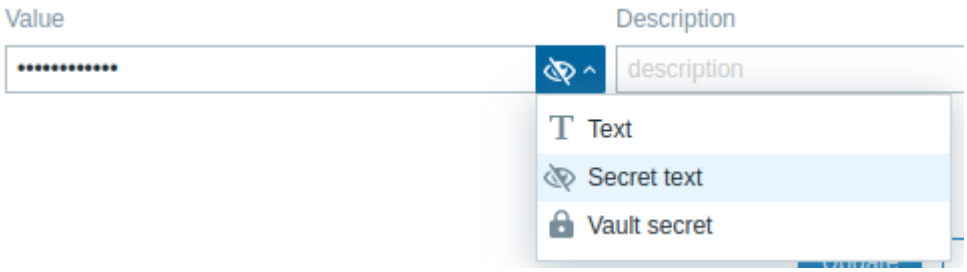
Note:
Obwohl der Wert eines geheimen Makros ausgeblendet ist, kann er durch die Verwendung in Datenpunkten offengelegt werden. Beispielsweise kann in einem externen Skript eine echo-Anweisung, die auf ein geheimes Makro verweist, verwendet werden, um den Makrowert im Frontend offenzulegen, da der Zabbix Server Zugriff auf den tatsächlichen Makrowert hat. Siehe Orte, an denen geheime Makrowerte eingeblendet werden.

Geheime Makros können nicht in Auslöser-Ausdrücken verwendet werden.

Geheimer Text

Bei Makros vom Typ „Geheimer Text" wird der Makrowert mit Sternchen maskiert.

Um einen Makrowert geheim zu machen, klicken Sie auf die Schaltfläche am Ende des Feldes *Wert* und wählen Sie die Option *Geheimer Text* aus:



Sobald die Konfiguration gespeichert ist, kann der Wert nicht mehr angezeigt werden.

Um den Makrowert zu ändern, bewegen Sie den Mauszeiger über das Feld *Wert* und klicken Sie auf die Schaltfläche *Neuen Wert festlegen* (erscheint beim Darüberfahren):

Wenn Sie auf die Schaltfläche *Neuen Wert festlegen* klicken (oder den Typ des Makrowerts ändern), wird der aktuelle Wert gelöscht.

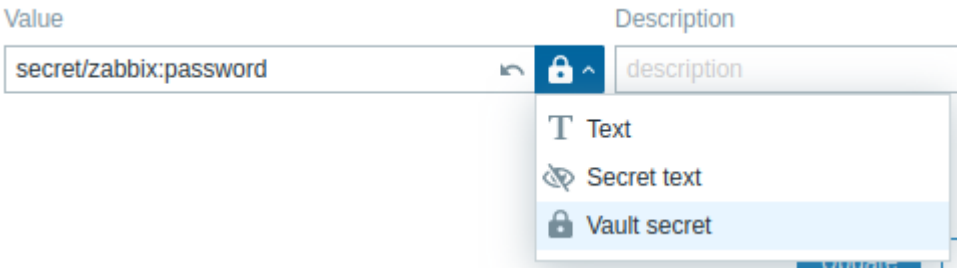
Sie können den ursprünglichen Wert wiederherstellen, indem Sie auf den Pfeil am Ende des Feldes *Wert* klicken (nur verfügbar, bevor die neue Konfiguration gespeichert wird). Beachten Sie, dass durch das Wiederherstellen des ursprünglichen Werts dieser nicht offengelegt wird.

Note:
URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das Makro darin als "******" aufgelöst wird.

Vault-Geheimnis

Bei Vault-Geheimnismakros wird der Makrowert in einer externen Software zur Verwaltung von Geheimnissen (Vault) gespeichert.

Um ein Vault-Geheimnismakro zu konfigurieren, klicken Sie auf die Schaltfläche am Ende des Feldes *Wert* und wählen Sie die Option *Vault-Geheimnis* aus:



Der Makrowert muss auf ein Vault-Geheimnis verweisen. Das Eingabeformat hängt vom Vault-Anbieter ab. Anbieterspezifische Konfigurationsbeispiele finden Sie unter:

- HashiCorp
- CyberArk

Werte von Vault-Geheimnis-Makros werden vom Vault durch den Zabbix Server abgerufen (und durch den Zabbix Proxy, wenn *Resolve secret vault macros by* auf *Zabbix server and proxy* gesetzt ist), und zwar bei jeder Aktualisierung der Konfigurationsdaten; anschließend werden sie im Konfigurations-Cache gespeichert. Zabbix Server und Zabbix Proxy können unterschiedliche Vaults verwenden.

Wenn *Resolve secret vault macros by* auf *Zabbix server* gesetzt ist, dann werden Vault-Geheimnisse nur vom Server abgerufen, und der Zabbix Proxy erhält die Werte der Vault-Geheimnis-Makros 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.

Nicht maskierte Speicherorte

Diese Liste enthält Speicherorte von Parametern, an denen Werte geheimer Makros nicht maskiert werden.

Note:
Werte geheimer Makros bleiben an den unten aufgeführten Speicherorten maskiert, wenn sie indirekt referenziert werden. Zum Beispiel werden {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} integrierte Makros, die in Medientypen (Skript- oder Webhook-Parameter) verwendet werden, zu Datenpunktschlüsseln aufgelöst, die maskierte geheime Makros enthalten, wie z. B. `net.tcp.port[******,******]` anstelle von `net.tcp.port[192.0.2.0,80]`.

Kontext		Parameter
Datenpunkte, Datenpunktprototypen, LLD-Regeln		
	Datenpunkt	*Datenpunktschlüsselparameter*
	Datenpunktprototyp	*Datenpunktprototyp-Schlüsselparameter*
	Low-level-Discovery-Regel	*Schlüsselparameter des Discovery-Datenpu*
	SNMP-Agent	*SNMP-Community*
		Kontextname (SNMPv3)
		Sicherheitsname (SNMPv3)
		Authentifizierungs-Passphrase (SNMPv3)
		Privacy-Passphrase (SNMPv3)
	HTTP-Agent	*URL*
		Abfragefelder
		Request-Body
		Header
		Benutzername
		Passwort
		SSL-Schlüsselpasswort
	Skript	*Parameter*
		Skript
	Browser	*Parameter*
		Skript
	Datenbankmonitor	*SQL-Abfrage*
	TELNET-Agent	*Skript*
		Benutzername
		Passwort
	SSH-Agent	*Skript*
		Benutzername
		Passwort
	Einfacher Check	*Benutzername*
		Passwort
	JMX-Agent	*Benutzername*
		Passwort
Vorverarbeitung von Datenpunktwerten		
	JavaScript-Vorverarbeitungsschritt	*Skript*
Webszenarien		
	Webszenario	*Variablenwert*
		Header-Wert
		URL
		Abfragefeldwert
		Post-Feldwert
		Roh-Postdaten
	Webszenario-Authentifizierung	*Benutzer*
		Passwort
		SSL-Schlüsselpasswort
Konnektoren		
	Konnektor	*URL*
		Benutzername
		Passwort

Kontext		Parameter
		Token *HTTP-Proxy* *SSL-Zertifikatsdatei* *SSL-Schlüsseldatei* *SSL-Schlüsselpasswort*
Netzwerkerkennung		
	SNMP	*SNMP-Community* *Kontextname* (SNMPv3) *Sicherheitsname* (SNMPv3) *Authentifizierungs-Passphrase* (SNMPv3) *Privacy-Passphrase* (SNMPv3)
Globale Skripte		
	Webhook	*JavaScript-Skript* *JavaScript-Skriptparameterwert*
	Telnet	*Benutzername* *Passwort*
	SSH	*Benutzername* *Passwort*
	Skript	*Skript*
Medientypen		
	Skript	*Skriptparameter*
	Webhook	*Parameter*
IPMI-Verwaltung		
	Host	*Benutzername* *Passwort*

5 Makros für Low-Level-Discovery

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.

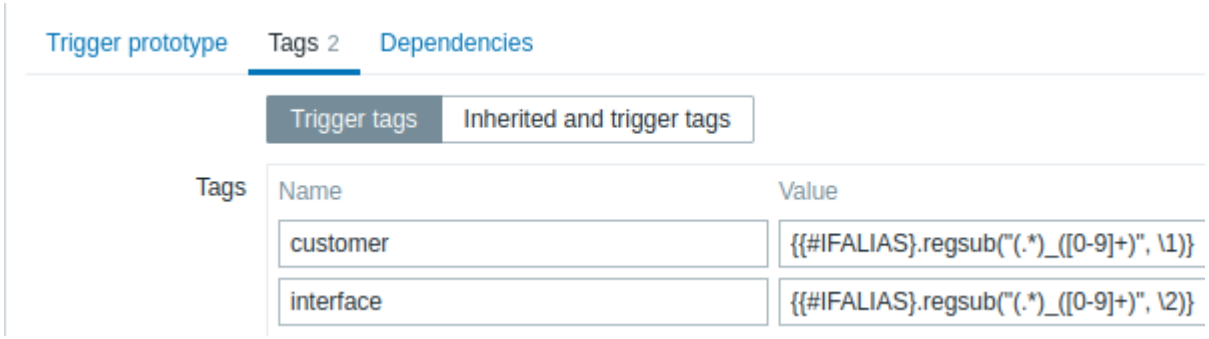
Verwendung von Makrofunktionen

Makrofunktionen werden mit Low-Level-Discovery-Makros unterstützt (außer im Filter der Low-Level-Discovery-Regel) und ermöglichen es, mithilfe eines regulären Ausdrucks einen bestimmten Teil des Makrowerts zu extrahieren.

Beispielsweise möchten Sie möglicherweise für die Ereignis-Tagging-Zwecke den Kundennamen und die Schnittstellennummer aus dem folgenden LLD-Makro extrahieren:

```
{#IFALIAS}=customername_1
```

Dazu kann die Makrofunktion `regsub` zusammen mit dem Makro im Feld für den Ereignis-Tag-Wert eines Auslöserprototyps verwendet werden:



Beachten Sie, dass Kommas in nicht in Anführungszeichen gesetzten Datenpunkt-Schlüsselparametern nicht 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 in Low-Level-Discovery-Makros seit Zabbix 4.0 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.

6 Ausdrucks-Makros

Übersicht

Ausdrucksmakros ermöglichen es Ihnen, Berechnungen in Feldern durchzuführen.

Ihr Wert wird berechnet, indem zunächst alle inneren Makros aufgelöst und anschließend der resultierende Ausdruck ausgewertet wird.

Syntax:

```
{?EXPRESSION}
```

`EXPRESSION` verwendet dieselbe Syntax und unterstützt dieselben Funktionen wie Auslöser-Ausdrücke.

Beispiel:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

An Stellen, die Ausdrucksmakros unterstützen, suchen Sie in der Tabelle unterstützte Makros nach "{?EXPRESSION}".

Hinweise zur Verwendung:

- Verwenden Sie Makros vom Typ {FUNCTION.*}, um auf Funktionswerte von Auslöser-Ausdrücken/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 beim ersten Host den Host ganz weg – zum Beispiel `{?avg(//item1,1h)}` – anstelle von Vorlagennamen, 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 Auslöser-Ausdrücke für ein Beispiel zur Verwendung eines Ausdrucksmakros in einem Ereignisnamen.

12 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, es sei denn, sie wurden zusätzlich entsprechend unter UNIX eingerichtet. Die Kommunikation zwischen dem Webserver und dem Browser des Benutzers kann mit SSL geschützt werden.

Mit einem flexiblen Berechtigungsschema für Benutzer können Sie Rechte einschränken und unterscheiden für:

- den Zugriff auf administrative Funktionen des Zabbix-Frontend
- die Ausführung bestimmter Aktionen im Frontend
- den Zugriff auf überwachte Hosts in Hostgruppen
- die Verwendung bestimmter API-Methoden

1 Konfigurieren eines Benutzers

Übersicht

Die anfängliche Zabbix-Installation verfügt über zwei vordefinierte Benutzer:

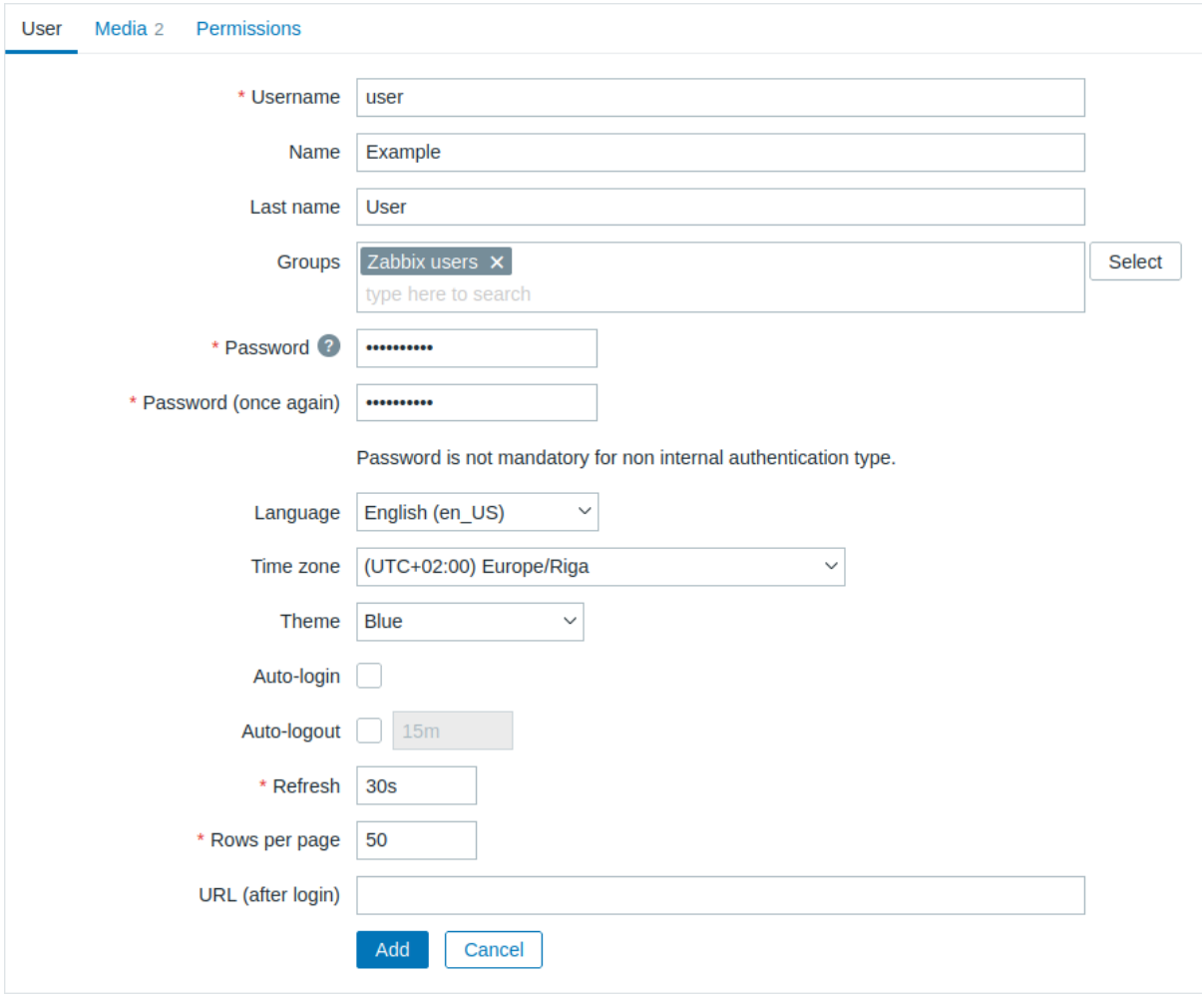
- *Admin* – ein Zabbix- Superuser mit vollständigen Berechtigungen.
- *guest* – ein spezieller Zabbix- Benutzer. Der Benutzer „guest“ ist standardmäßig deaktiviert. Wenn Sie ihn zur Benutzergruppe „Guests“ hinzufügen, können Sie sich mit diesem Benutzer anmelden und auf Monitoring-Seiten in Zabbix zugreifen. Beachten Sie, dass „guest“ standardmäßig keine Berechtigungen für Zabbix-Objekte hat.

So konfigurieren Sie einen Benutzer:

- Gehen Sie zu *Benutzer* → *Benutzer*.
- Klicken Sie auf *Benutzer erstellen* (oder auf einen Benutzernamen, um einen vorhandenen Benutzer zu bearbeiten).
- Bearbeiten Sie die Benutzerattribute im Formular.

Allgemeine Attribute

Die Registerkarte *Benutzer* enthält allgemeine Benutzerattribute:



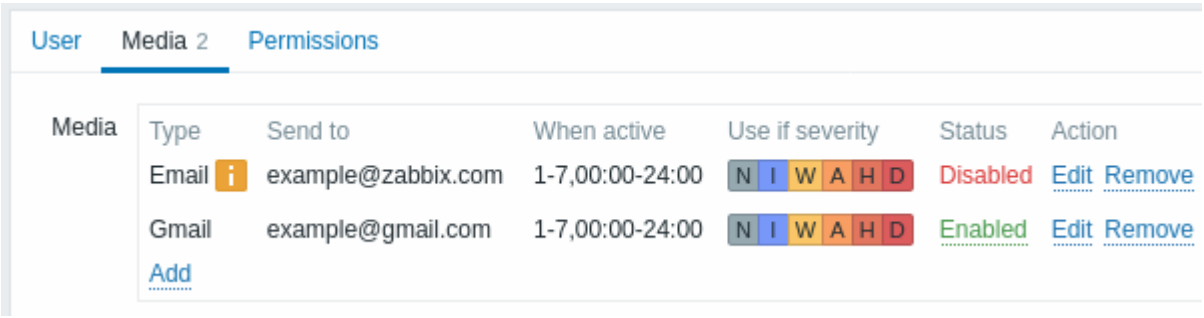
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Benutzername	Eindeutiger Benutzername, der als Anmeldename verwendet wird.
Vorname	Vorname des Benutzers (optional). Falls nicht leer, in Bestätigungsinformationen und Informationen zum Benachrichtigungsempfänger sichtbar.
Nachname	Nachname des Benutzers (optional). Falls nicht leer, in Bestätigungsinformationen und Informationen zum Benachrichtigungsempfänger sichtbar.
Gruppen	Wählen Sie die Benutzergruppen aus, denen der Benutzer angehört. Dieses Feld unterstützt Auto-Vervollständigung; wenn Sie beginnen, den Namen einer Benutzergruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Scrollen Sie nach unten, um eine Auswahl zu treffen. Alternativ klicken Sie auf *Auswählen*, um Gruppen hinzuzufügen. Klicken Sie auf „x“, um die ausgewählte Gruppe zu entfernen. Die Zugehörigkeit zu Benutzergruppen bestimmt, auf welche Host-Gruppen und Hosts der Benutzer Zugriff hat.
Passwort	Zwei Felder zur Eingabe des Benutzerpassworts oder eine Schaltfläche *Passwort ändern*, falls der Benutzer bereits existiert. Durch Klicken auf die Schaltfläche *Passwort ändern* werden zwei Felder zur Eingabe eines neuen Passworts geöffnet. Wenn ein Benutzer mit der Rolle *Super admin role* sein eigenes Passwort ändert, öffnet ein Klick auf die Schaltfläche *Passwort ändern* zusätzlich ein Feld zur Eingabe des aktuellen (alten) Passworts. Nach einer erfolgreichen Passwortänderung wird der Benutzer, für den das Passwort geändert wurde, aus allen aktiven Sitzungen abgemeldet. Beachten Sie, dass das Passwort nur für Benutzer geändert werden kann, die die interne Zabbix-Authentifizierung verwenden.
Sprache	Sprache des Zabbix Frontends. Damit die Übersetzungen funktionieren, ist die php-gettext-Erweiterung erforderlich.

Parameter	Beschreibung
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	Definiert das Erscheinungsbild des Frontends: **Systemstandard** - Standard-Systemeinstellungen verwenden **Blau** - standardmäßiges blaues Design **Dunkel** - alternatives dunkles Design **Hell mit hohem Kontrast** - helles Design mit hohem Kontrast **Dunkel mit hohem Kontrast** - dunkles Design mit hohem Kontrast
Automatische Anmeldung	Aktivieren Sie dieses Kontrollkästchen, damit Zabbix sich den Benutzer merkt und ihn 30 Tage lang automatisch anmeldet. Bei der Anmeldung mit *30 Tage merken*: - Die automatische Abmeldung wird zurückgesetzt (die Sitzung bleibt 30 Tage bestehen). - Die automatische Anmeldung wird für eine nahtlose erneute Authentifizierung aktiviert. Bei der Anmeldung ohne *30 Tage merken*: - Die automatische Anmeldung wird für eine nahtlose erneute Authentifizierung aktiviert. - Die automatische Abmeldung bleibt durch die standardmäßige Timeout-Einstellung geregelt. 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, maximal 1 Tag). Beachten Sie, dass diese Einstellung überschrieben wird, wenn *30 Tage merken* 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 im Menü „Monitoring“ Informationen im Hintergrund aktualisieren. * Wenn bei der Anmeldung die Option *30 Tage merken* aktiviert ist.
Aktualisierung	Legen Sie die Aktualisierungsrate fest, die für Diagramme, Klartextdaten usw. verwendet wird. Kann auf 0 gesetzt werden, um sie 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 Anmeldung)	Sie können Zabbix so konfigurieren, dass der Benutzer nach erfolgreicher Anmeldung zu einer bestimmten URL weitergeleitet wird, zum Beispiel 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 Medien zuzuweisen.

Wenn der Medientyp deaktiviert wurde:

- Nach dem Namen wird ein gelbes Info-Symbol angezeigt.
- In der Spalte Status wird *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*)

Weitere Informationen zur Konfiguration von Benutzermedien finden Sie im Abschnitt Medientypen.

Berechtigungen

Die Registerkarte *Berechtigungen* enthält Informationen zu den folgenden Elementen:

- Benutzerrolle (obligatorisch für jeden neu erstellten Benutzer), die nur von einem Benutzer des Typs *Super admin* geändert werden kann.

> **Attention:**
> Benutzer können nicht ohne eine Benutzerrolle erstellt werden (außer mit der Zabbix-User API). Bereits zuvor erstellte Benutzer, die keine Rolle haben, können weiterhin bearbeitet werden, ohne ihnen eine Rolle zuzuweisen. Sobald jedoch eine Rolle zugewiesen wurde, kann sie nur geändert, nicht entfernt werden.

 Beachten Sie, dass sich Benutzer ohne Rolle nur über LDAP- oder SAML-Authentifizierung bei Zabbix anmelden können, sofern ihre LDAP-/SAML-Informationen mit den in Zabbix konfigurierten Benutzergruppen-Zuordnungen übereinstimmen.

- Benutzertyp (*User*, *Admin*, *Super admin*), der in der Konfiguration der Benutzerrolle definiert ist.
- Host- und Vorlagengruppen, auf die der Benutzer Zugriff hat.
  - Benutzer des Typs *User* und *Admin* haben standardmäßig keinen Zugriff auf Gruppen, Vorlagen und Hosts. Um solchen Zugriff zu gewähren, müssen Benutzer in Benutzergruppen aufgenommen werden, die mit Berechtigungen für die entsprechenden Entitäten konfiguriert sind.
- Zugriffsrechte auf Abschnitte und Elemente des Zabbix Frontend, Module und API-Methoden.
  - Elemente mit erlaubtem Zugriff werden grün angezeigt, während Elemente mit verweigertem Zugriff hellgrau dargestellt werden.
- Rechte zum Ausführen bestimmter Aktionen.
  - Aktionen, die der Benutzer ausführen darf, werden grün angezeigt, während verweigerte Aktionen hellgrau dargestellt werden.

Weitere Informationen finden Sie auf der Seite Berechtigungen.

2 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

Berechtigungen in Zabbix hängen in erster Linie vom Benutzertyp ab:

- *Benutzer* - hat eingeschränkte Zugriffsrechte auf Menübereiche (siehe unten) und standardmäßig keinen Zugriff auf irgendwelche Ressourcen. Alle Berechtigungen für Host- oder Vorlagengruppen müssen explizit zugewiesen werden;
- *Admin* - hat unvollständige Zugriffsrechte auf Menübereiche (siehe unten). Der Benutzer hat standardmäßig keinen Zugriff auf Host-Gruppen. Alle Berechtigungen für Host- oder Vorlagengruppen müssen explizit vergeben werden;
- *Super-Admin* - hat Zugriff auf alle Menübereiche. Der Benutzer hat Lese-/Schreibzugriff auf alle Host- und Vorlagengruppen. Berechtigungen können nicht entzogen werden, indem der Zugriff auf bestimmte Gruppen verweigert wird.

Menüzugriff

> **Note:**
> Der eingeschränkte Zugriff auf einige UI-Elemente verhindert nur 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 Zabbix-Menübereiche je Benutzertyp:

Menübereich		Benutzer	Admin	Super-Admin
Dashboards		+	+	+
Monitoring		+	+	+
	Probleme	+	+	+
	Hosts	+	+	+
	Letzte Daten	+	+	+
	Karten	+	+	+
	Discovery		+	+
Services		+	+	+
	Services	+	+	+

Menübereich		Benutzer	Admin	Super-Admin
	SLA		+	+
	SLA-Bericht	+	+	+
Inventar		+	+	+
	Übersicht	+	+	+
	Hosts	+	+	+
Berichte		+	+	+
	Systeminformationen			+
	Geplante Berichte		+	+
	Verfügbarkeitsbericht	+	+	+
	Top 100 Auslöser	+	+	+
	Audit-Log			+
	Aktionsprotokoll			+
	Benachrichtigungen		+	+
Datenerfassung			+	+
	Vorlagengruppen		+	+
	Host-Gruppen		+	+
	Vorlagen		+	+
	Hosts		+	+
	Wartung		+	+
	Ereigniskorrelation			+
	Discovery		+	+
Warnungen			+	+
	Auslöser-Aktionen		+	+
	Service-Aktionen		+	+
	Discovery-Aktionen		+	+
	Autoregistrierungs-Aktionen		+	+
	Interne Aktionen		+	+
	Medientypen			+
	Skripte			+
Benutzer				+
	Benutzergruppen			+
	Benutzerrollen			+
	Benutzer			+
	API-Tokens			+
	Authentifizierung			+
Administration				+
	Allgemein			+
	Audit-Log			+
	Housekeeping			+
	Proxy-Gruppen			+
	Proxies			+
	Makros			+
	Warteschlange			+

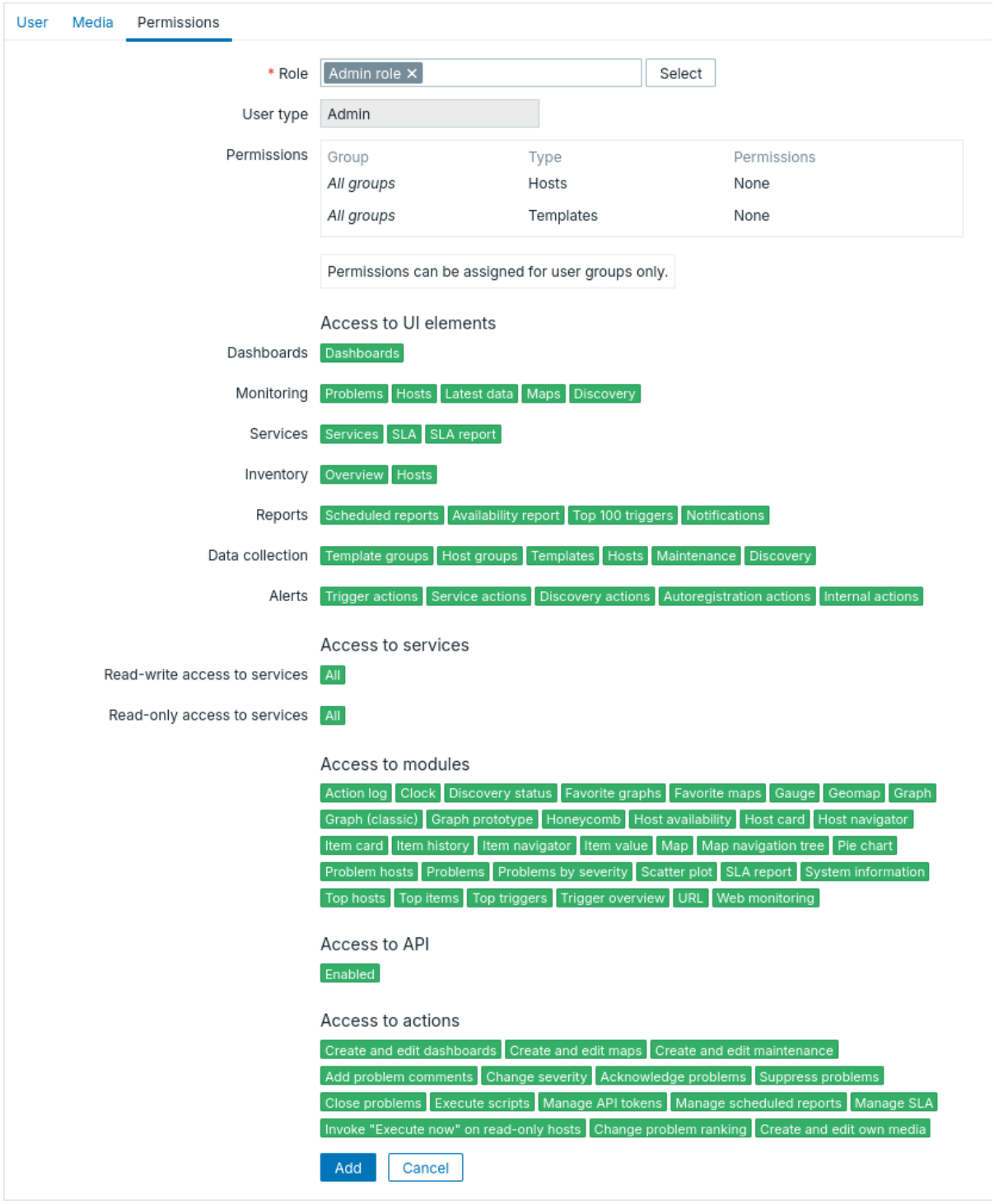
Benutzerrollen

Benutzerrollen ermöglichen benutzerdefinierte Anpassungen der durch den Benutzertyp festgelegten Berechtigungen. Es können zwar keine Berechtigungen hinzugefügt werden (da dies die des Benutzertyps überschreiten würde), einige Berechtigungen können jedoch entzogen werden.

Außerdem bestimmt eine Benutzerrolle den Zugriff nicht nur auf Menüabschnitte, sondern auch auf Services, Module, API-Methoden und verschiedene Aktionen im Frontend.

Benutzerrollen werden im Abschnitt *Benutzer* → *Benutzerrollen* von Benutzern mit der Rolle Super admin konfiguriert.

Benutzerrollen werden Benutzern im Benutzerkonfigurationsformular auf der Registerkarte *Berechtigungen* von Benutzern mit der Rolle Super admin zugewiesen.



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.

3 Benutzergruppen

Übersicht

Benutzergruppen ermöglichen es, Benutzer sowohl zu organisatorischen Zwecken als auch zur Zuweisung von Berechtigungen für Daten zu gruppieren. Berechtigungen zum Anzeigen und Konfigurieren von Daten aus Host-Gruppen und Vorlagen-Gruppen

werden Benutzergruppen zugewiesen, nicht einzelnen Benutzern.

Es ist oft sinnvoll, zu trennen, welche Informationen einer Benutzergruppe und welche einer anderen zur Verfügung stehen. Dies kann erreicht werden, indem Benutzer gruppiert und anschließend unterschiedliche Berechtigungen für Host- und Vorlagen-Gruppen zugewiesen werden.

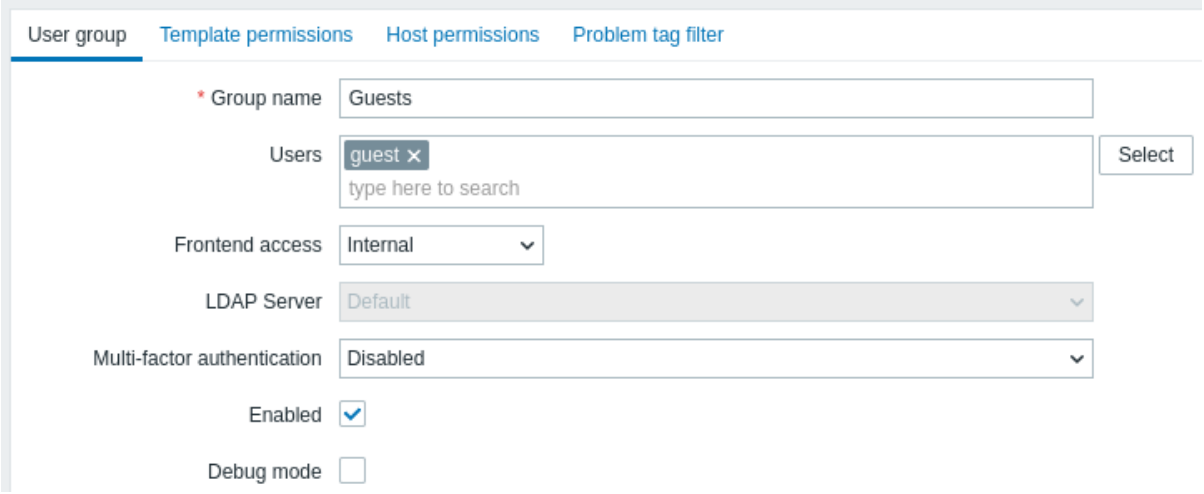
Ein Benutzer kann einer beliebigen Anzahl von Gruppen angehören.

Konfiguration

So konfigurieren Sie eine Benutzergruppe:

- Gehen Sie zu *Benutzer* > *Benutzergruppen*
- Klicken Sie auf *Benutzergruppe erstellen* (oder auf den Gruppennamen, um eine vorhandene Gruppe zu bearbeiten)
- Bearbeiten Sie die Gruppenattribute im Formular

Die Registerkarte **Benutzergruppe** enthält allgemeine Gruppenattribute:

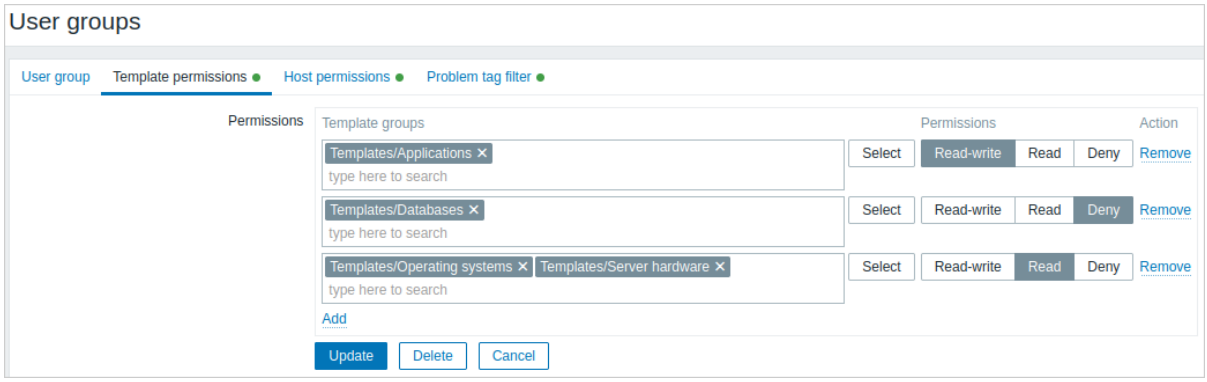


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

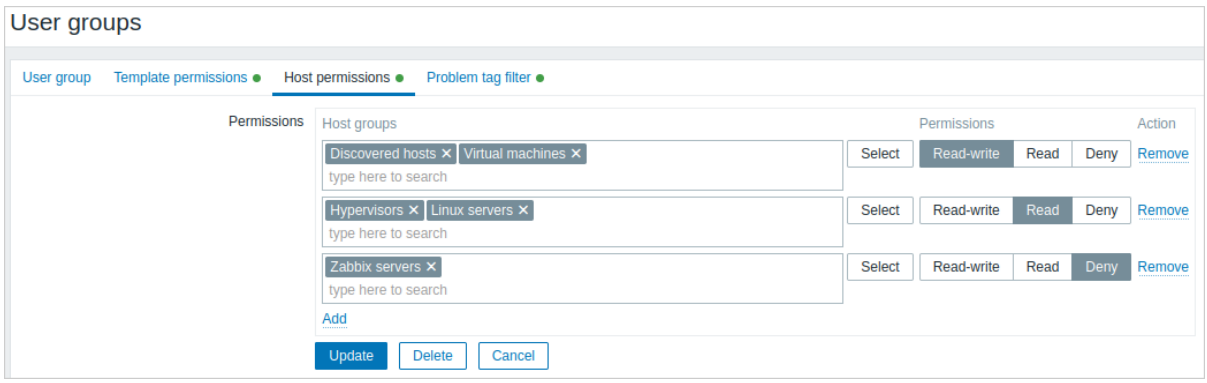
Parameter	Beschreibung
Gruppenname	Eindeutiger Gruppenname.
Benutzer	Um Benutzer zur Gruppe hinzuzufügen, beginnen Sie mit der Eingabe des Namens eines vorhandenen Benutzers. Wenn die Dropdown-Liste mit passenden Benutzernamen angezeigt wird, scrollen Sie nach unten, um einen Benutzer auszuwählen. Alternativ können Sie auf die Schaltfläche *Auswählen* klicken, um Benutzer in einem Popup auszuwählen.
Frontend-Zugriff	Wie die Benutzer der Gruppe authentifiziert werden. **Systemstandard** - die Standardauthentifizierungsmethode verwenden (global festgelegt globally) **Intern** - die interne Zabbix-Authentifizierung verwenden (auch wenn LDAP-Authentifizierung global verwendet wird). Wird ignoriert, wenn HTTP-Authentifizierung global als Standard festgelegt ist. **LDAP** - LDAP-Authentifizierung verwenden (auch wenn interne Authentifizierung global verwendet wird). Wird ignoriert, wenn HTTP-Authentifizierung global als Standard festgelegt ist. **Deaktiviert** - der Zugriff auf das Zabbix-Frontend ist für diese Gruppe verboten
LDAP-Server	Wählen Sie aus, welcher LDAP-Server zur Authentifizierung des Benutzers verwendet werden soll. Dieses Feld ist nur aktiviert, wenn *Frontend-Zugriff* auf LDAP oder Systemstandard gesetzt ist.

Parameter	Beschreibung
Multi-Faktor-Authentifizierung	Wählen Sie aus, welche Methode der Multi-Faktor-Authentifizierung zur Authentifizierung des Benutzers verwendet werden soll: **Standard** - die in der MFA-Konfiguration als Standard festgelegte Methode verwenden; diese Option ist standardmäßig für neue Benutzergruppen ausgewählt, wenn MFA aktiviert ist; **<Methodenname>** - die ausgewählte Methode verwenden (zum Beispiel „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 diese zur Authentifizierung des Benutzers verwendet; andernfalls wird die erste Methode in alphabetischer Reihenfolge zur Authentifizierung verwendet.
Aktiviert	Status der Benutzergruppe und der Gruppenmitglieder. *Aktiviert* - Benutzergruppe und Benutzer sind aktiviert *Nicht aktiviert* - Benutzergruppe und Benutzer sind deaktiviert
Debug-Modus	Aktivieren Sie dieses Kontrollkästchen, um den Debug-Modus für die Benutzer zu aktivieren.

Die Registerkarte **Vorlagenberechtigungen** ermöglicht es, den Zugriff der Benutzergruppe auf Daten von Vorlagengruppen (und damit Vorlagen) festzulegen:



Die Registerkarte **Host-Berechtigungen** ermöglicht es, den Zugriff der Benutzergruppe auf Daten von Host-Gruppen (und damit Hosts) festzulegen:



Klicken Sie auf Add, um die Vorlagen-/Host-Gruppen auszuwählen (entweder eine übergeordnete oder eine verschachtelte Gruppe) und diesen 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 dann die Optionsschaltflächen, um den ausgewählten Gruppen Berechtigungen zuzuweisen. Folgende Berechtigungen sind möglich:

- **Lesen/Schreiben** - Lese-/Schreibzugriff auf eine Gruppe
- **Lesen** - schreibgeschützter Zugriff auf eine Gruppe
- **Verweigern** - Zugriff auf eine Gruppe verweigert

Wenn dieselbe Vorlagen-/Host-Gruppe in mehreren Zeilen mit unterschiedlichen Berechtigungen hinzugefügt wird, wird die strengste Berechtigung angewendet.

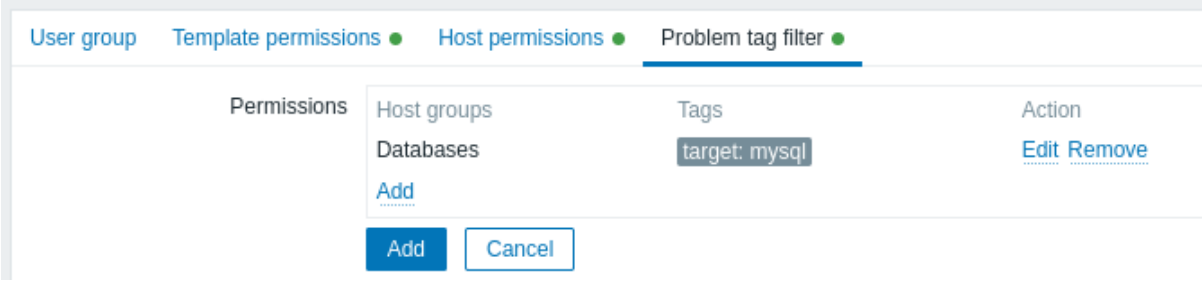
Beachten Sie, dass ein Benutzer vom Typ *Super Admin* erzwingen kann, dass verschachtelte Gruppen dieselbe Berechtigungsstufe wie die übergeordnete Gruppe haben; dies kann im Konfigurationsformular der Host-/Vorlagen-gruppe festgelegt werden.

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 und **Verweigern** oder keine Berechtigungen für eine mit diesem Host verknüpfte Vorlage hat, können die Benutzer dieser Gruppe keine aus Vorlagen stammenden Datenpunkte auf dem Host bearbeiten, und der Vorlagenname wird als *Nicht zugängliche Vorlage* angezeigt.

Die Registerkarte **Problem-Tag-Filter** ermöglicht das Festlegen tagbasierter Berechtigungen für Benutzergruppen, damit Probleme nach Tag-Name und -Wert gefiltert angezeigt werden:

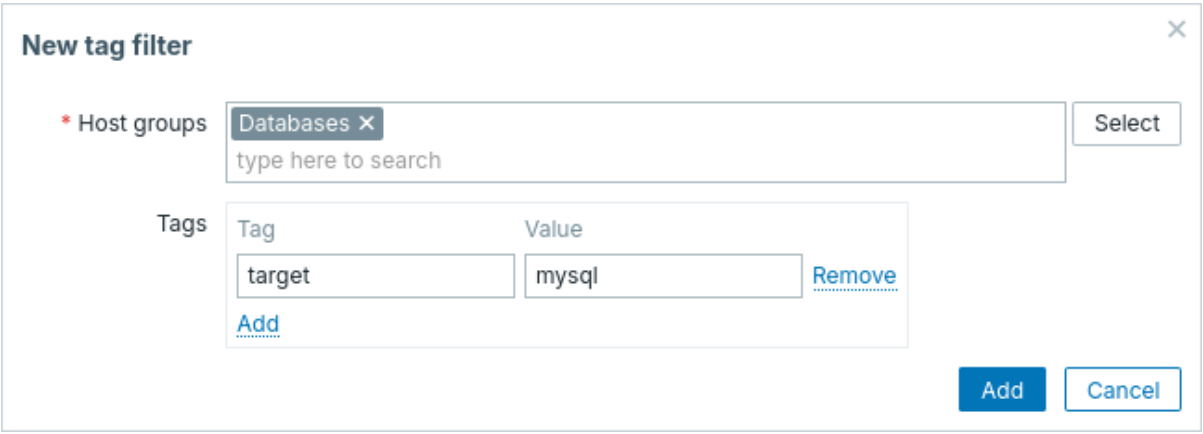


Klicken Sie auf Add, um die Host-Gruppen auszuwählen. Um eine Host-Gruppe auszuwählen, auf die ein Tag-Filter angewendet werden soll, klicken Sie auf *Auswählen*, um die vollständige Liste vorhandener Host-Gruppen anzuzeigen, oder beginnen Sie mit der Eingabe des Namens einer Host-Gruppe, um eine Dropdown-Liste mit passenden Gruppen zu erhalten. Es werden nur Host-Gruppen angezeigt, da der Problem-Tag-Filter nicht auf Vorlagengruppen angewendet werden kann.

Tag-Namen ohne Werte können hinzugefügt werden, Werte ohne Namen jedoch nicht. Im Block *Berechtigungen* werden nur die ersten drei Tags (gegebenenfalls mit Werten) angezeigt; wenn es mehr gibt, können diese durch Klicken auf oder Bewegen des Mauszeigers über das Symbol ••• angezeigt werden.

Der Tag-Filter ermöglicht es, den Zugriff auf eine Host-Gruppe von der Möglichkeit zu trennen, Probleme zu sehen.

Wenn beispielsweise ein Datenbankadministrator nur Probleme der Datenbank „MySQL“ sehen soll, muss zunächst eine Benutzergruppe für Datenbankadministratoren erstellt und dann der Tag-Name „target“ sowie der Wert „mysql“ angegeben werden.



Wenn der Tag-Name „target“ angegeben und das Wertefeld leer gelassen wird, sieht die Benutzergruppe alle Probleme mit dem Tag-Namen „target“ für die ausgewählte Host-Gruppe.

Stellen Sie sicher, dass Tag-Name und Tag-Wert korrekt angegeben sind, andernfalls sieht die Benutzergruppe keine Probleme.

Sehen wir uns ein Beispiel an, bei dem ein Benutzer Mitglied mehrerer ausgewählter Benutzergruppen ist. In diesem Fall wird bei der Filterung eine ODER-Bedingung für Tags verwendet.

Benutzergruppe A	**Benutzergruppe B**	**Sichtbares Ergebnis für einen Benutzer (Mitglied) beider Gruppen**
Tag-Filter		

Host-Gruppe	*Tag-Name*	*Tag-Wert*	*Host-Gruppe*	*Tag-Name*	*Tag-Wert*	
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.

13 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 von Folgendem verwendet werden:

- Werten von Benutzermakros
- Zugangsdaten für den Datenbankzugriff

Zabbix bietet schreibgeschützten Zugriff auf die Secrets in einem Vault, wobei davon ausgegangen wird, dass die Secrets von jemand anderem verwaltet werden.

Informationen zur Konfiguration bestimmter Vault-Anbieter finden Sie unter:

- HashiCorp-Konfiguration
- CyberArk-Konfiguration

Caching von geheimen Werten

Standardmäßig werden Vault-Geheimnismakrowerte vom Zabbix Server bei jeder Aktualisierung der Konfigurationsdaten abgerufen und anschließend im Konfigurations-Cache gespeichert. Der Zabbix Proxy empfängt die Werte der Vault-Geheimnismakros vom Zabbix Server bei jeder Konfigurationssynchronisierung und speichert sie in seinem eigenen Konfigurations-Cache.

Attention:
Die Verschlüsselung muss zwischen Zabbix Server und Proxy aktiviert sein; andernfalls wird eine Server-Warnmeldung protokolliert.

Es ist auch möglich, zu konfigurieren, dass Makrowerte vom Zabbix Server und vom Zabbix Proxy unabhängig voneinander abgerufen werden.

Um die Aktualisierung zwischengespeicherter geheimer Werte aus einem Vault manuell auszulösen, verwenden Sie die Befehlszeilen-Option 'secrets_reload'.

Für die Zwischenspeicherung der Datenbank-Zugangsdaten des Zabbix Frontend ist das Caching standardmäßig deaktiviert, kann jedoch durch Setzen der Option `$DB['VAULT_CACHE'] = true` in zabbix.conf.php aktiviert werden. Die Zugangsdaten werden in einem lokalen Cache unter Verwendung des temporären Dateisystemverzeichnisses gespeichert. Der Webserver muss das Schreiben in einen privaten temporären Ordner erlauben (zum Beispiel muss für Apache die Konfigurationsoption `PrivateTmp=True` gesetzt sein). Um zu steuern, wie oft der Daten-Cache aktualisiert/ungültig gemacht wird, verwenden Sie die Konstante ZBX_DATA_CACHE_TTL constant .

TLS-Konfiguration

Um TLS für die Kommunikation zwischen Zabbix-Komponenten und dem Vault zu konfigurieren, fügen Sie dem systemweiten Standard-CA-Speicher ein von einer Zertifizierungsstelle (CA) signiertes Zertifikat hinzu. Um einen anderen Speicherort zu verwenden, geben Sie das Verzeichnis im Zabbix-Konfigurationsparameter SSLCALocation für server/proxy an, legen Sie die Zertifikatsdatei in diesem Verzeichnis ab und führen Sie dann den CLI-Befehl aus:

```
c_rehash .
```

1 CyberArk-Konfiguration

In diesem Abschnitt wird erläutert, wie Zabbix so konfiguriert wird, dass Secrets aus CyberArk Vault CV12 abgerufen werden.

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 Secrets*.

Datenbank-Anmeldedaten

Der Zugriff auf ein Secret mit Datenbank-Anmeldedaten wird für jede Zabbix-Komponente separat konfiguriert.

Server und Proxys

Um Datenbank-Anmeldedaten aus dem Vault für den Zabbix-server oder proxy zu beziehen, 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 Datenbank-Anmeldedaten, 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 der Schlüsseldatei; das Einrichten 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 bei der Verarbeitung von Secret-Vault-Makros durch den Zabbix-Server verwendet (und durch den Zabbix-Proxy, falls konfiguriert). Zabbix-Server und Zabbix-Proxys öffnen keine Vault-Secret-Makros, die Datenbank-Anmeldedaten aus VaultDBPath enthalten.

Zabbix-Server und 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 Vault enthält die Schlüssel ”Content” und ”UserName”:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

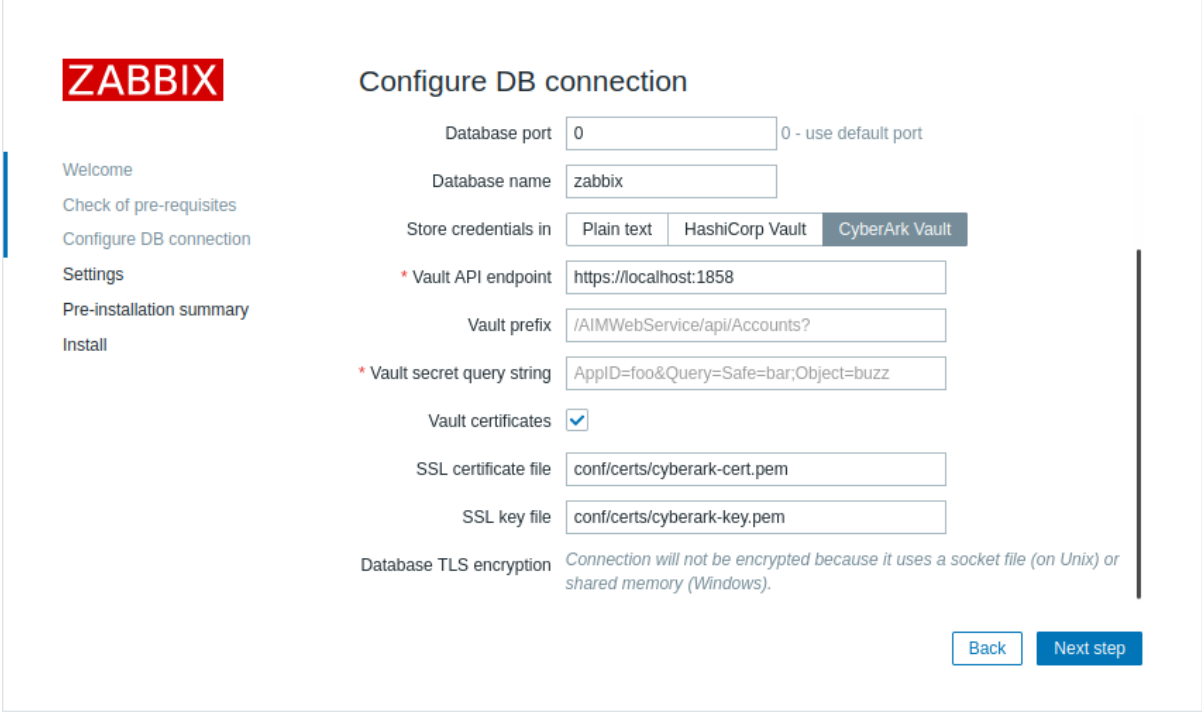
4. Daher verwendet Zabbix die folgenden Anmeldedaten für die Datenbankauthentifizierung:

- Benutzername: <username>
- Passwort: <password>

Frontend

Um Datenbank-Anmeldedaten aus dem Vault für das Zabbix-Frontend abzurufen, geben Sie während der Frontend-Installation 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	Pflichtfeld	Standardwert	Beschreibung
Vault API endpoint	ja	https://localhost:1858	Geben Sie die URL für die Verbindung zum Vault im Format `scheme://host:port` an.
Vault prefix	nein	/AIMWebService/api/Accounts?	Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an. Wenn nichts angegeben wird, wird der Standardwert verwendet.
Vault secret query string	ja		Eine Abfrage, die angibt, von wo die Datenbank-Anmeldedaten abgerufen werden sollen. Beispiel: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	nein		Nach dem Aktivieren des Kontrollkästchens werden zusätzliche Parameter angezeigt, mit denen die Client-Authentifizierung konfiguriert werden kann. Obwohl dieser Parameter optional ist, wird dringend empfohlen, ihn für die Kommunikation mit dem CyberArk Vault zu aktivieren.
SSL certificate file	nein	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 *SSL key file* leer.
SSL key file	nein	conf/certs/cyberark-key.pem	Name der SSL-Datei mit dem privaten Schlüssel, die für die Client-Authentifizierung verwendet wird. Die Datei muss im PEM-Format vorliegen.

Werte von Benutzermakros

Um CyberArk Vault zum Speichern von Benutzermakrowerten 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) benötigen Zugriff auf Makrowerte vom Typ *Vault secret* aus dem Vault. Das Zabbix Frontend benötigt keinen solchen Zugriff.

Der Makrowert sollte eine Abfrage enthalten (als `query:key`).

Ausführliche Informationen zur Verarbeitung von Makrowerten durch Zabbix finden Sie unter *Vault secret macros*.

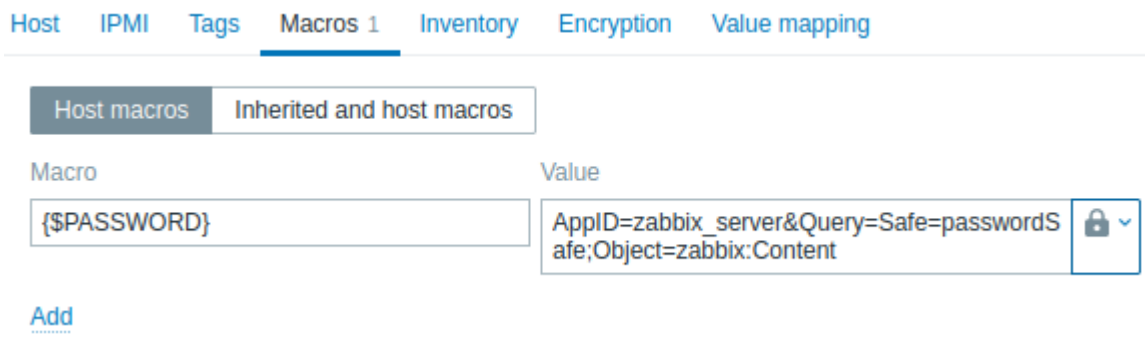
Abfragesyntax

Das Doppelpunktzeichen (":") ist für die Trennung der Abfrage vom Schlüssel reserviert.

Wenn eine Abfrage selbst einen Schrägstrich oder einen Doppelpunkt enthält, sollten diese Zeichen 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-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 Vault 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.

Bestehende Konfiguration aktualisieren

Um eine bestehende Konfiguration zum Abrufen von Geheimnissen aus einem CyberArk Vault zu aktualisieren:

1. Aktualisieren Sie die Parameter der Zabbix-Server- oder Proxy-Konfigurationsdatei wie im Abschnitt *Datenbankzugangsdaten* beschrieben.
2. Aktualisieren Sie die DB-Verbindungseinstellungen, indem Sie das Zabbix Frontend neu konfigurieren und die erforderlichen Parameter wie im Abschnitt *Frontend* beschrieben angeben. Um das Zabbix Frontend neu zu konfigurieren, öffnen Sie die Frontend-Setup-URL im Browser:

- für Apache: http://<server_ip_or_name>/zabbix/setup.php
- für Nginx: http://<server_ip_or_name>/setup.php

Alternativ können diese Parameter in der Frontend-Konfigurationsdatei (*zabbix.conf.php*) gesetzt werden:

```
$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.

Um eine bestehende Konfiguration zum Abrufen von Geheimnissen aus einem HashiCorp Vault zu aktualisieren, siehe *HashiCorp-Konfiguration*.

2 HashiCorp-Konfiguration

Übersicht

In diesem Abschnitt wird erläutert, wie Zabbix für das Abrufen von Geheimnissen aus der HashiCorp Vault KV Secrets Engine – Version 2 konfiguriert wird.

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 Datenbank-Zugangsdaten

Um ein Secret mit Datenbank-Zugangsdaten 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-Provider verwendet werden soll;
- `VaultToken` - Vault-Authentifizierungs-Token (siehe Zabbix-Server-/Proxy-Konfigurationsdatei für Details);
- `VaultURL` - HTTP[S]-URL des Vault-Servers;
- `VaultDBPath` - Pfad zum Vault-Secret, das die Datenbank-Zugangsdaten enthält (diese Option kann nur verwendet werden, wenn DBUser und DBPassword nicht angegeben sind); Zabbix Server oder Proxy ruft die Zugangsdaten über die 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.

Attention:
Die Konfigurationsparameter `Vault`, `VaultToken`, `VaultURL` und `VaultPrefix` werden auch für die Vault-Authentifizierung verwendet, wenn Secret-Vault-Makros vom Zabbix Server verarbeitet werden (und vom Zabbix Proxy, falls konfiguriert). Zabbix Server und Proxys öffnen keine Vault-Secret-Makros, die Datenbank-Zugangsdaten aus VaultDB-Path enthalten.

Die Verwendung unterschiedlicher Tokens für verschiedene Proxys wird dringend empfohlen.

Zabbix Server und Zabbix Proxy lesen die Vault-bezogenen Konfigurationsparameter beim Start aus *zabbix_server.conf* bzw. *zabbix_proxy.conf*. Zusätzlich lesen Zabbix Server und Zabbix Proxy beim Start einmalig die Umgebungsvariable `VAULT_TOKEN` und setzen sie anschließend 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:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
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 Einhängepunkt "secret/", falls er noch nicht aktiviert ist; beachten Sie, dass "ku
vault secrets enable -path=secret/ kv-v2

#### Legen Sie neue Secrets mit den Schlüsseln username und password unter dem Einhängepunkt "secret/" und
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Testen Sie, dass das Secret erfolgreich hinzugefügt wurde.
vault kv get secret/zabbix/database

#### Testen Sie abschließend mit Curl; beachten Sie, dass "data" nach dem Einhängepunkt und "/v1" vor dem
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Als Ergebnis ruft der Zabbix Server die folgenden Zugangsdaten für die Datenbankauthentifizierung ab:

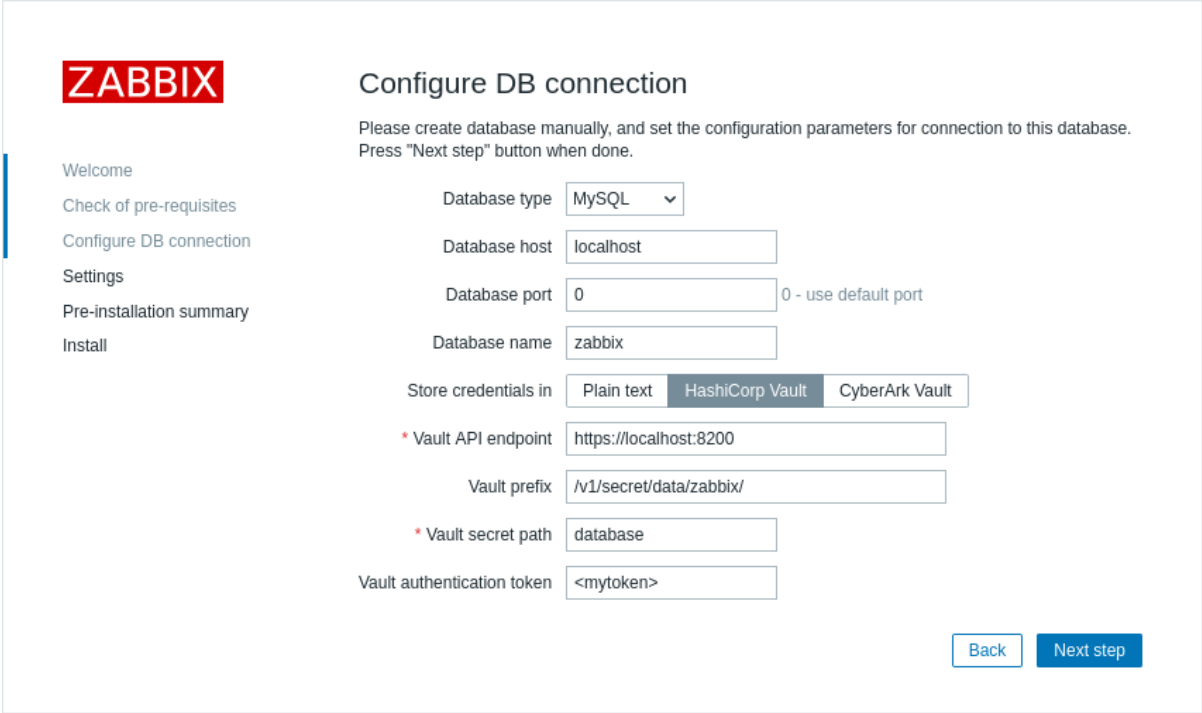
- Benutzername: zabbix
- Passwort: <password>

Frontend

Das Zabbix Frontend kann so konfiguriert werden, dass es Datenbank-Anmeldedaten aus dem Vault abruft, entweder während der Frontend-Installation oder durch Aktualisierung der Frontend-Konfigurationsdatei (`zabbix.conf.php`).

Attention:
Wenn die Vault-Anmeldedaten seit der vorherigen Frontend-Installation geändert wurden, führen Sie die Frontend-Installation erneut aus oder aktualisieren Sie `zabbix.conf.php`. Siehe auch: Vorhandene Konfiguration aktualisieren.

Während der **Frontend-Installation** müssen die Konfigurationsparameter im Schritt *Configure DB Connection* angegeben werden:



- Setzen Sie den Parameter *Store credentials in* auf „HashiCorp Vault“.
- Geben Sie die Verbindungsparameter an:

Parameter	Pflichtfeld	Standardwert	Beschreibung
Vault API endpoint	ja	https://localhost:8200	Geben Sie die URL für die Verbindung zum Vault im Format `scheme://host:port` an
Vault prefix	nein	/v1/secret/data/	Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an. Wenn nichts angegeben wird, wird der Standardwert verwendet. Beispiel: `/v1/secret/data/zabbix/`
Vault secret path	nein		Ein Pfad zum Secret, aus dem die Anmeldedaten für die Datenbank über die Schlüssel „password“ und „username“ abgerufen werden sollen. Beispiel: `database`
Vault authentication token	nein		Geben Sie ein Authentifizierungs-Token für den schreibgeschützten Zugriff auf den Secret-Pfad an. Informationen zum Erstellen von Tokens und Vault-Richtlinien finden Sie in der HashiCorp-Dokumentation.

Abrufen von Benutzermakro-Werten

Um HashiCorp Vault zum Speichern von Benutzermakro-Werten vom Typ *Vault secret* zu verwenden, stellen Sie sicher, dass:

- der Zabbix Server/Proxy konfiguriert ist, um mit HashiCorp Vault zu arbeiten;
- der Parameter *Vault provider* unter *Administration → General → Other* auf „HashiCorp Vault“ gesetzt ist (Standard);

Note:
Der Zabbix Server (und der Zabbix Proxy, falls konfiguriert) benötigen Zugriff auf Makrowerte vom Typ *Vault secret* 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 Konfiguration von Zabbix Server/Proxy angegebene Authentifizierungs-Token (über den Parameter `VaultToken`) muss schreibgeschützten Zugriff auf diesen Pfad gewähren.

Ausführliche Informationen zur Verarbeitung von Makrowerten durch Zabbix finden Sie unter *Vault secret macros*.

Pfadsyntax

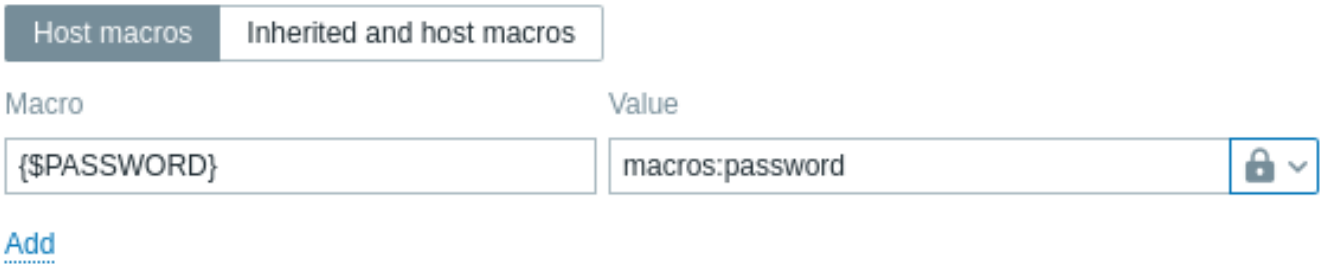
Die Symbole Schrägstrich (”/”) und Doppelpunkt (”:”) sind reserviert.

Ein Schrägstrich kann nur verwendet werden, um einen Einhängepunkt von einem Pfad zu trennen (z. B. *secret/zabbix*, wobei der Einhängepunkt „secret“ und der Pfad „zabbix“ ist). Im Fall von Vault-Makros kann ein Doppelpunkt nur verwendet werden, um einen Pfad/eine Abfrage von einem Schlüssel zu trennen.

Es ist möglich, die Symbole Schrägstrich und Doppelpunkt URL-zu-kodieren, wenn ein Einhängepunkt mit einem Namen erstellt werden muss, der durch einen Schrägstrich getrennt ist (z. B. *foo/bar/zabbix*, wobei der Einhängepunkt „foo/bar“ und der Pfad „zabbix“ ist, kann als „foo%2Fbar/zabbix“ kodiert werden), und wenn ein Einhängepunktname 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 Einhängepunkt "secret/", falls er noch nicht aktiviert ist; beachten Sie, dass "ku
vault secrets enable -path=secret/ kv-v2

#### Legen Sie ein neues Secret mit dem Schlüssel "password" unter dem Einhängepunkt "secret/" und dem Pfa
vault kv put -mount=secret zabbix/macros password=<password>

#### Testen Sie, dass das Secret erfolgreich hinzugefügt wurde.
vault kv get secret/zabbix/macros

#### Testen Sie abschließend mit Curl; beachten Sie, dass "data" nach dem Einhängepunkt und "/v1" vor dem
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. Als Ergebnis löst Zabbix das Makro {$PASSWORD} in den Wert <password> auf.

Aktualisierung einer bestehenden Konfiguration

Um eine bestehende Konfiguration zum Abrufen von Geheimnissen aus einem HashiCorp Vault zu aktualisieren:

1. Aktualisieren Sie die Parameter der Zabbix-Server- oder Proxy-Konfigurationsdatei wie im Abschnitt *Datenbankzugangsdaten* beschrieben.

2. Aktualisieren Sie die DB-Verbindungseinstellungen, indem Sie das Zabbix Frontend neu konfigurieren und die erforderlichen Parameter wie im Abschnitt *Frontend* beschrieben angeben. Um das Zabbix Frontend neu zu konfigurieren, öffnen Sie die Frontend-Setup-URL im Browser:

- für Apache: http://<server_ip_or_name>/zabbix/setup.php
- für Nginx: http://<server_ip_or_name>/setup.php

Alternativ können diese Parameter in der Frontend-Konfigurationsdatei (*zabbix.conf.php*) festgelegt werden:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

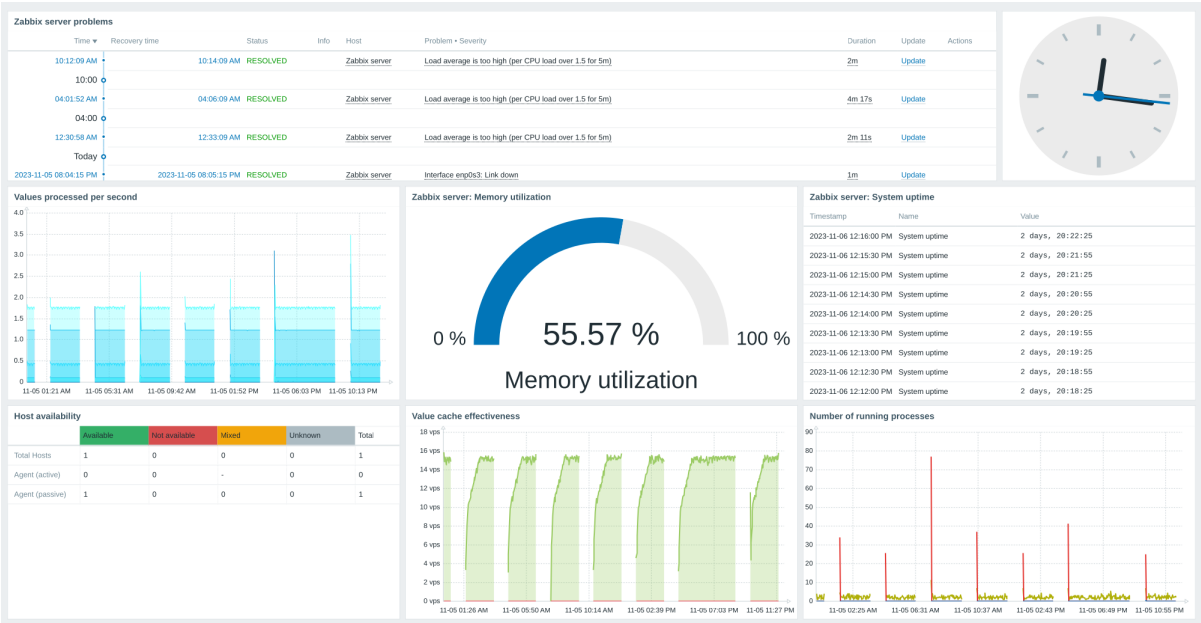
3. Konfigurieren Sie bei Bedarf Benutzermakros wie im Abschnitt *Werte von Benutzermakros* beschrieben.

Um eine bestehende Konfiguration zum Abrufen von Geheimnissen aus einem CyberArk Vault zu aktualisieren, siehe *CyberArk-Konfiguration*.

14 Geplante Berichte

Übersicht

Mit der Funktion *Geplante Berichte* können Sie festlegen, dass eine PDF-Version eines bestimmten Dashboards in wiederkehrenden Intervallen an angegebene Empfänger gesendet wird.



Voraussetzungen:

- Der Zabbix-Webservice muss installiert und korrekt konfiguriert sein, damit die Erstellung geplanter Berichte möglich ist – Anweisungen finden Sie unter Einrichtung 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)

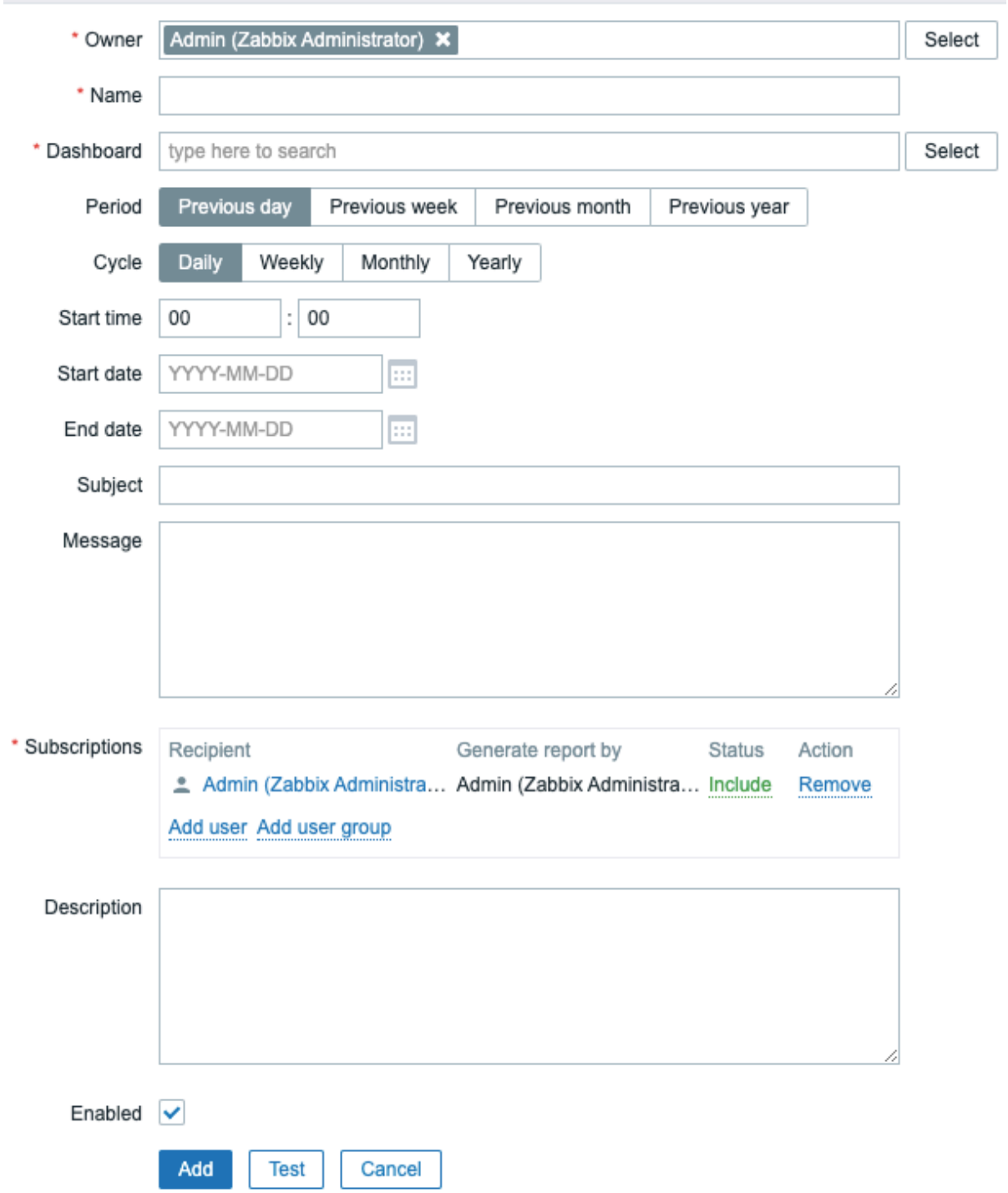
Gehen Sie wie folgt vor, um im Zabbix Frontend einen geplanten Bericht zu erstellen:

- Gehen Sie zu: *Berichte* > *Geplante Berichte*.
- Klicken Sie oben rechts auf dem Bildschirm auf *Bericht erstellen*.
- Geben Sie die Parameter des Berichts im Formular ein.

Sie können einen Bericht auch erstellen, indem Sie einen vorhandenen Bericht öffnen, auf die Schaltfläche *Klonen* klicken und ihn dann unter einem anderen Namen speichern.

Konfiguration

Die Registerkarte *Geplante Berichte* enthält allgemeine Berichtsattribute.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Owner	Benutzer, der einen Bericht erstellt. Benutzer mit der Berechtigungsstufe *Super admin* dürfen den Owner 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 passender Dashboards wird angezeigt; scrollen Sie nach unten, um eines auszuwählen. Alternativ können Sie neben dem Feld auf *Select* klicken und ein Dashboard aus der angezeigten Liste auswählen.
Period	Zeitraum, für den der Bericht erstellt wird. Wählen Sie den vorherigen Tag, die vorherige Woche, den vorherigen Monat oder das vorherige Jahr aus.

Parameter	Beschreibung
Cycle	Häufigkeit der Berichtserstellung. Die Berichte können täglich, wöchentlich, monatlich oder jährlich gesendet werden. Im Modus „Wöchentlich“ können die Wochentage ausgewählt werden, an denen der Bericht gesendet wird.
Start time	Uhrzeit des Tages im Format hh:mm, zu der der Bericht erstellt wird. Beachten Sie, dass die Zeitzone des Zabbix Server verwendet wird.
Repeat on	Wochentage, an denen der Bericht gesendet wird. Dieses Feld ist nur verfügbar, wenn *Cycle* auf „Wöchentlich“ gesetzt ist.
Start date	Datum, an dem die regelmäßige Berichtserstellung beginnen soll.
End date	Datum, an dem die regelmäßige Berichtserstellung beendet werden soll.
Subject	Betreff der Berichts-E-Mail. Unterstützte Makros: {TIME}, {TIMESTAMP}.
Message	Text der Berichts-E-Mail. Unterstützte Makros: {TIME}, {TIMESTAMP}.
Subscriptions	Liste der Berichtsempfänger. Standardmäßig enthält sie nur den Owner des Berichts. Jeder Zabbix-Benutzer mit konfiguriertem E-Mail-Medium kann als Berichtsempfänger angegeben werden. Klicken Sie auf *Add user* oder *Add user group*, um weitere Empfänger hinzuzufügen. Klicken Sie auf den Benutzernamen, um die Einstellungen zu bearbeiten: *Generate report by* – ob die Berichtsdaten auf Grundlage der Dashboard-Berechtigungen des aktuellen Benutzers oder des Empfängers erzeugt werden sollen. *Status* – wählen Sie „Include“, um den Bericht an den Benutzer zu senden, oder „Exclude“, um den Versand an diesen Benutzer zu verhindern. Mindestens ein Benutzer muss den Status „Include“ haben. Der Status „Exclude“ kann verwendet werden, um bestimmte Benutzer aus einer eingeschlossenen Benutzergruppe auszuschließen. Beachten Sie, dass Benutzer mit unzureichenden Berechtigungen (d. h. Benutzer mit einer auf dem Benutzertyp *Admin* basierenden Rolle, die nicht Mitglieder derselben Benutzergruppe wie der Empfänger oder der Owner des Berichts sind) in den Feldern *Recipient* und *Generate report by* anstelle der tatsächlichen Namen „Inaccessible user“ oder „Inaccessible user group“ sehen; die Felder *Status* und *Action* werden schreibgeschützt angezeigt.
Enabled	Berichtsstatus. Wenn dieses Kontrollkästchen deaktiviert wird, wird der Bericht deaktiviert.
Description	Eine optionale Beschreibung des Berichts. Diese Beschreibung ist für den internen Gebrauch bestimmt und wird nicht an die Berichtsempfänger gesendet.

Formularschaltflächen

Mit den Schaltflächen am unteren Rand des Formulars können mehrere Operationen ausgeführt werden.

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.

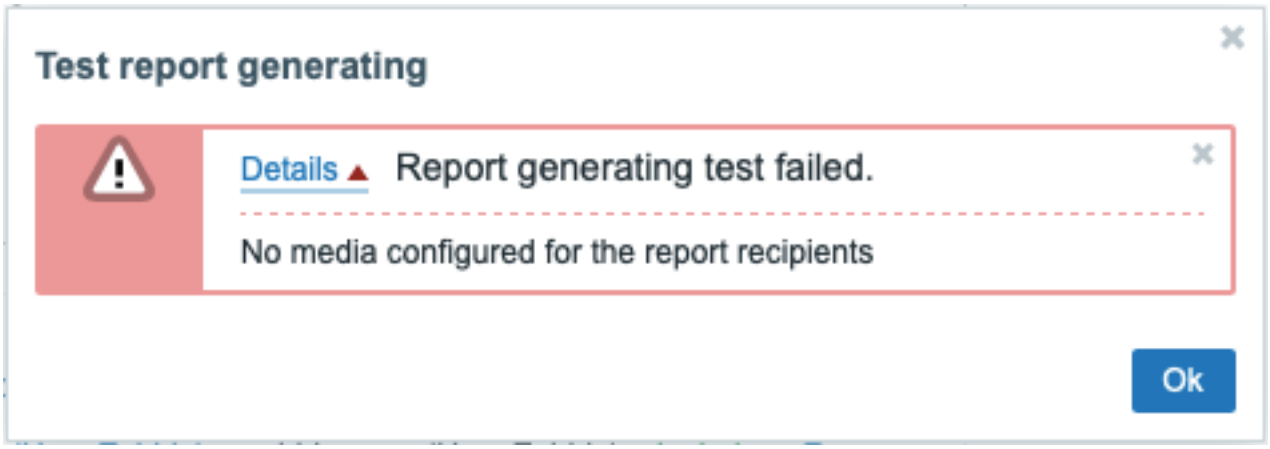
Testen

Um einen Bericht zu testen, klicken Sie unten im Berichts-Konfigurationsformular auf die Schaltfläche *Test*.

> **Note:**
> Die Schaltfläche *Test* ist nicht verfügbar, wenn das Berichts-Konfigurationsformular über das Aktionsmenü des Dashboards geöffnet wurde.

Wenn die Konfiguration korrekt ist, wird der Testbericht sofort an den aktuellen Benutzer gesendet. Bei Testberichten werden Abonnenten und die Benutzereinstellungen für *Bericht generieren durch* ignoriert.

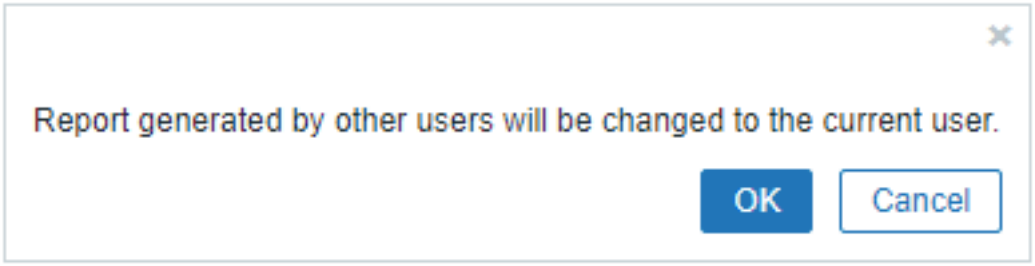
Wenn die Konfiguration nicht korrekt ist, wird eine Fehlermeldung angezeigt, die die mögliche Ursache beschreibt.



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*.

15 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)

1 Export in Dateien

Übersicht

Es ist möglich, den Echtzeit-Export von Auslöser-Ereignissen, Datenpunkt-Werten und Trends in einem durch Zeilenumbrüche getrennten JSON-Format zu konfigurieren.

Der Export erfolgt in Dateien, wobei jede Zeile der Exportdatei ein JSON-Objekt ist. Wertezuordnungen werden nicht angewendet.

Im Fehlerfall (wenn Daten nicht in die Exportdatei geschrieben werden können oder die Exportdatei nicht umbenannt werden kann oder nach dem Umbenennen keine neue Datei erstellt werden kann) wird der Dateneintrag verworfen und niemals in die Exportdatei geschrieben. Er wird nur in die Zabbix-Datenbank geschrieben. Das Schreiben von Daten in die Exportdatei wird fortgesetzt, sobald das Schreibproblem behoben ist.

Die Exportdatei wird mit Lese- und Schreibrechten nur für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen werden verweigert.

Ausführliche Informationen darüber, welche Informationen exportiert werden, finden Sie auf der Seite Exportprotokoll.

Beachten Sie, dass für Host/Datenpunkt möglicherweise keine Metadaten (Host-Gruppen, Host-Name, Datenpunkt-Name) vorhanden sind, 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, Datenpunkt-Werten 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:

- `ExportFileSize` kann verwendet werden, um die maximal zulässige Größe einer einzelnen Exportdatei festzulegen. Wenn ein Prozess in eine Datei schreiben muss, prüft er zunächst die Größe der Datei. Wenn sie die konfigurierte Größenbegrenzung überschreitet, wird die Datei umbenannt, indem `.old` an ihren Namen angehängt wird, und eine neue Datei mit dem ursprünglichen Namen wird erstellt.

Attention:
Für jeden Prozess, der Daten schreibt, wird eine eigene Datei erstellt (d. h. ungefähr 4–30 Dateien). Da die Standardgröße pro Exportdatei 1G beträgt, kann das Beibehalten großer Exportdateien den Festplattenspeicher schnell aufbrauchen.

- `ExportType` ermöglicht die Angabe, welche Entitätstypen (Ereignisse, Verlauf, Trends) exportiert werden.

2 Streaming an externe Systeme

Übersicht

Es ist möglich, Datenpunkt-Werte und Ereignisse von Zabbix über HTTP an externe Systeme zu streamen (siehe Protokolldetails).

Der Tag-Filter kann verwendet werden, um Teilmengen von Datenpunkt-Werten oder Ereignissen zu streamen.

Zwei Zabbix-Serverprozesse sind für das Daten-Streaming verantwortlich: `connector manager` und `connector worker`. Ein interner Zabbix-Datenpunkt `zabbix[connector_queue]` ermöglicht die Überwachung der Anzahl der in die Connector-Warteschlange eingereihten Werte.

Konfiguration

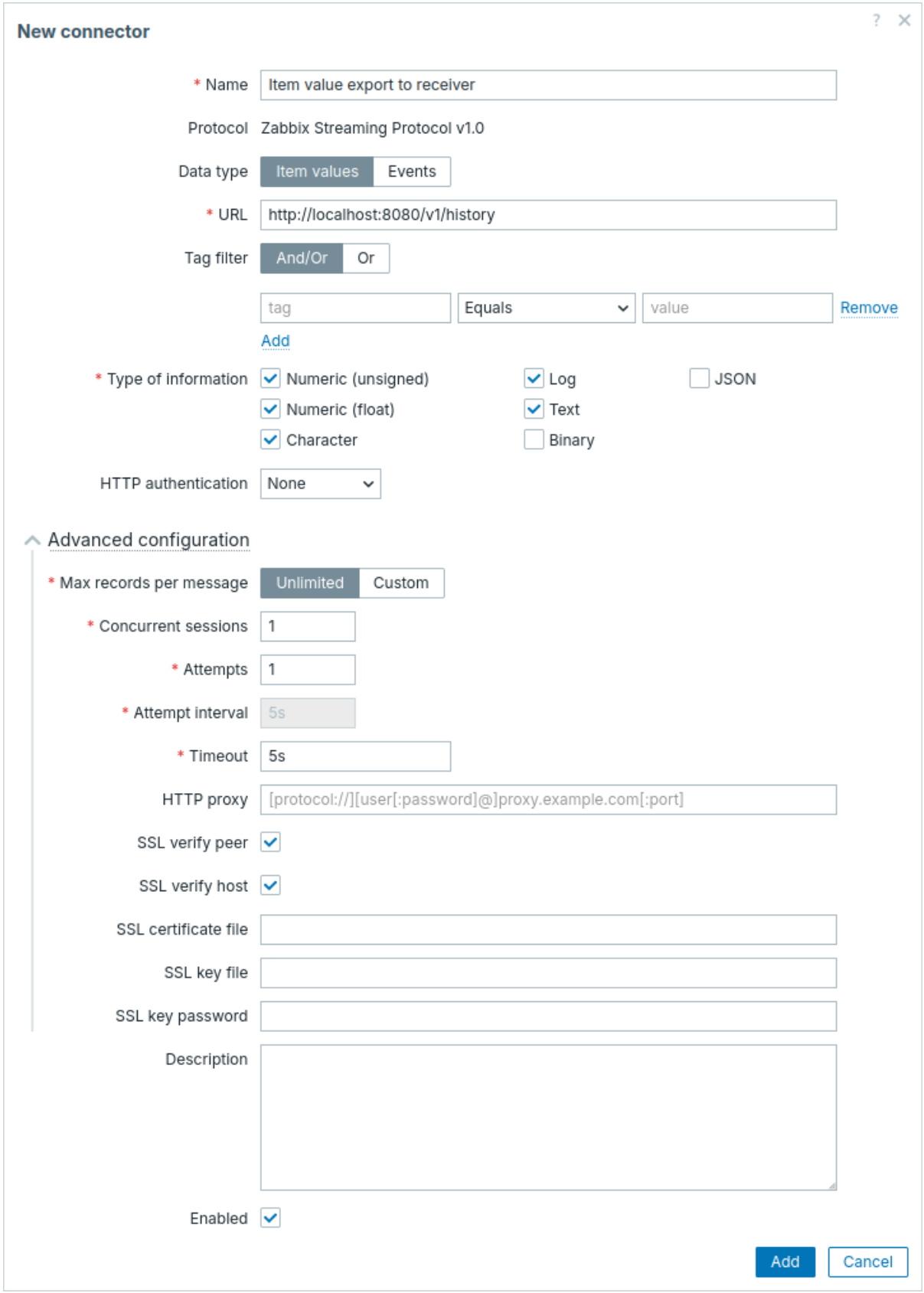
Die folgenden Schritte sind erforderlich, um das Daten-Streaming zu einem externen System zu konfigurieren:

1. Richten Sie ein entferntes System für den Empfang von Daten aus Zabbix ein. Zu diesem Zweck stehen die folgenden Werkzeuge zur Verfügung:

- Ein Beispiel für einen einfachen receiver, der die empfangenen Informationen in den Dateien `events.ndjson` und `history.ndjson` protokolliert.
- Kafka connector for Zabbix server - ein leichtgewichtiger, in Go geschriebener Server, der dafür ausgelegt ist, Datenpunkt-Werte 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 mit der im Zabbix Frontend konfigurierten Anzahl von Connectors übereinstimmen (oder diese überschreiten, wenn mehr als 1 gleichzeitige Sitzung verwendet wird). 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 markiert.

Parameter	Beschreibung
Name	Geben Sie den Namen des Connectors ein.
Data type	Wählen Sie den zu streamenden Datentyp aus: **Item values** - Datenpunkt-Werte von Zabbix an externe Systeme streamen; **Events** - Ereignisse von Zabbix an externe Systeme streamen.
URL	Geben Sie die URL des Empfängers ein. Benutzermakros werden unterstützt.

Parameter	Beschreibung
Tag filter	Exportieren Sie nur Datenpunkt-Werte oder Ereignisse, die dem Tag-Filter entsprechen. Wenn nichts festgelegt ist, wird alles exportiert. 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 (Teilstring-Abgleich, 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 (Teilstring-Abgleich, 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.
Type of information	Wählen Sie den Informationstyp (numeric (unsigned), numeric (float), character usw.) aus, nach dem die Datenpunkt-Werte 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** - es wird keine Authentifizierung verwendet; **Basic** - Basic-Authentifizierung wird verwendet; **NTLM** - NTLM-Authentifizierung (Windows NT LAN Manager) wird verwendet; **Kerberos** - Kerberos-Authentifizierung wird verwendet (siehe auch: Configuring Kerberos with Zabbix); **Digest** - Digest-Authentifizierung wird verwendet; **Bearer** - 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 Bezeichnung *Advanced configuration*, um erweiterte 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 Sender-Prozesse 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 10s angegeben werden; der Standardwert ist ”5s”. Dieses Feld ist verfügbar, wenn *Attempts* auf ”2” oder höher gesetzt ist. Erfolglose Versuche sind solche, bei denen der Verbindungsaufbau fehlgeschlagen ist oder bei denen der HTTP-Antwortcode nicht 200, 201, 202, 203 oder 204 ist. Wiederholungsversuche werden **ausgelöst**, wenn Kommunikationsfehler auftreten oder wenn der HTTP-Antwortcode nicht 200, 201, 202, 203, 204, 400, 401, 403, 404, 405 oder 415, 422 ist. Weiterleitungen werden verfolgt, daher ist 302 -> 200 eine positive Antwort; 302 -> 503 hingegen löst einen Wiederholungsversuch aus.
Timeout	Geben Sie das Nachrichten-Timeout an (1-60 Sekunden, Standard - 5 Sekunden). Zeitsuffixe 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 Protokollpräfixe 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 proxy-bezogene Umgebungsvariablen wie `http_proxy`, `HTTPS_PROXY`. Wenn nichts angegeben ist, überschreibt der Proxy keine proxy-bezogenen Umgebungsvariablen. Der eingegebene Wert wird unverändert weitergegeben; es findet keine Plausibilitätsprüfung statt. Sie können auch eine SOCKS-Proxy-Adresse eingeben. Wenn Sie das falsche Protokoll angeben, schlägt die Verbindung fehl und der Datenpunkt wird nicht unterstützt. Beachten Sie, dass mit HTTP-Proxy nur einfache Authentifizierung unterstützt wird.
SSL verify peer	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des Webservers zu überprüfen. Das Serverzertifikat wird automatisch aus dem systemweiten Speicherort der Zertifizierungsstelle (CA) übernommen. Sie können den Speicherort der CA-Dateien mit dem Konfigurationsparameter `SSLCALocation` von Zabbix Server oder Proxy überschreiben.
SSL verify host	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserver-Zertifikats übereinstimmt. Dadurch wird die cURL-Option `CURLOPT_SSL_VERIFYHOST` gesetzt.
SSL certificate file	Name der SSL-Zertifikatsdatei, die für die Client-Authentifizierung 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 Konfigurationsparameter `SSLCertLocation` von Zabbix Server oder Proxy angegeben.
SSL key file	Name der SSL-Datei mit dem privaten Schlüssel, die für die Client-Authentifizierung 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 Konfigurationsparameter `SSLKeyLocation` von Zabbix Server oder Proxy angegeben.
SSL key password	Passwort der Datei mit dem privaten SSL-Schlüssel. 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-Broker-Adressen konfiguriert ist (zum Beispiel `Kafka.URL=kafka1.example.com:9093,kafka2.example.com:9093`), verbindet sich der Kafka-Client mit dem/den Broker(n), die zuerst antworten, und verwendet deren Cluster-Metadaten. Wenn die Liste Adressen aus verschiedenen Kafka-Clustern enthält, wird nur das Cluster mit der schnellsten Antwort verwendet, und andere Adressen werden als nicht verfügbar protokolliert; infolgedessen 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 (private Netzwerke, Container-Netzwerke oder nicht standardmäßige DNS-/Hosts-Konfigurationen) können Hostnamen oder IPs zu Loopback-Adressen (zum Beispiel `127.0.0.1/localhost`) aufgelöst oder vom Client normalisiert werden, was solche Warnungen irreführend machen kann. 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 zur von den Brokern angekündigten Adresse aufgelöst werden.

Protokoll

Die Kommunikation zwischen dem Server und dem Empfänger erfolgt über HTTP unter Verwendung der REST-API, NDJSON, „Content-Type: application/x-ndjson“.

Weitere Einzelheiten 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, andere Codes stehen für fehlgeschlagene Anfragen.

Beispiel für einen Erfolg:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

Beispiel mit Fehlern:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 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`)

6 Service-Überwachung

Übersicht Service-Monitoring ist ein Monitoring auf Geschäftsebene, das verwendet werden kann, um einen Überblick über den gesamten Service-Baum der IT-Infrastruktur zu erhalten, Schwachstellen der Infrastruktur zu identifizieren, SLA verschiedener IT-Services zu berechnen und weitere Informationen auf einer höheren Ebene zu prüfen. Service-Monitoring konzentriert sich auf die Gesamtverfügbarkeit eines Service anstelle von Details auf niedriger Ebene, wie z. B. fehlendem Festplattenspeicher, hoher Prozessorlast usw. Service-Monitoring bietet außerdem Funktionen, um die Grundursache eines Problems zu finden, wenn ein Service nicht wie erwartet funktioniert.

Service-Monitoring ermöglicht es, eine hierarchische Darstellung überwachter Daten zu erstellen.

Eine sehr einfache Servicestruktur kann wie folgt aussehen:

```
Service
|
|-Arbeitsstationen
| |
| |-Arbeitsstation1
| |
| |-Arbeitsstation2
|
|-Server
```

Jeder Knoten der Struktur hat das Attribut Status. Der Status wird gemäß dem ausgewählten Algorithmus berechnet und an die oberen Ebenen weitergegeben. Der Status einzelner Knoten wird durch den Status der zugeordneten Probleme beeinflusst. Die Problemzuordnung erfolgt mithilfe von Tagging.

Zabbix kann Benachrichtigungen senden oder automatisch ein Skript auf dem Zabbix Server ausführen, wenn eine Änderung des Service-Status erkannt wird. Es ist möglich, flexible Regeln dafür zu definieren, ob ein übergeordneter Service in einen „Problemzustand“ wechseln soll, basierend auf den Statuswerten untergeordneter Services. Problemdaten von Services können dann verwendet werden, um SLA zu berechnen und SLA-Berichte auf Grundlage eines flexiblen Satzes von Bedingungen zu senden.

Service-Monitoring wird im Menü **Services** konfiguriert, das aus den folgenden Abschnitten besteht:

- Services

Im Abschnitt **Services** können Sie eine Hierarchie Ihrer überwachten Infrastruktur aufbauen, indem Sie übergeordnete Services und anschließend untergeordnete Services zu den übergeordneten Services hinzufügen.

Zusätzlich zur Konfiguration des Service-Baums bietet dieser Abschnitt einen Überblick über die gesamte Infrastruktur und ermöglicht es, schnell die Probleme zu identifizieren, die zu einer Änderung des Service-Status geführt haben.

- SLA

In diesem Abschnitt können Sie Service-Level-Agreements definieren und Service-Level-Ziele für bestimmte Services festlegen.

- SLA report

In diesem Abschnitt können Sie SLA-Berichte anzeigen.

Service-Aktionen

Sie können auch Service-Aktionen konfigurieren.

Service-Aktionen sind optional und ermöglichen Folgendes:

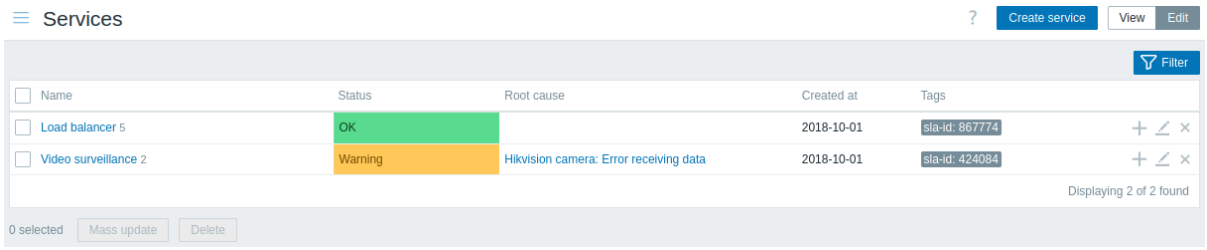
- eine Benachrichtigung senden, dass ein Service ausgefallen ist
- bei einer Änderung des Service-Status einen Remote-Befehl auf dem Zabbix Server ausführen
- eine Wiederherstellungsbenachrichtigung senden, wenn ein Service wieder verfügbar ist.

Siehe auch:

- Beispiel für die Konfiguration des SLA-Monitorings
- Hinweise zum Upgrade von Services aus Zabbix-Versionen unter 6.0

1 Service-Baum

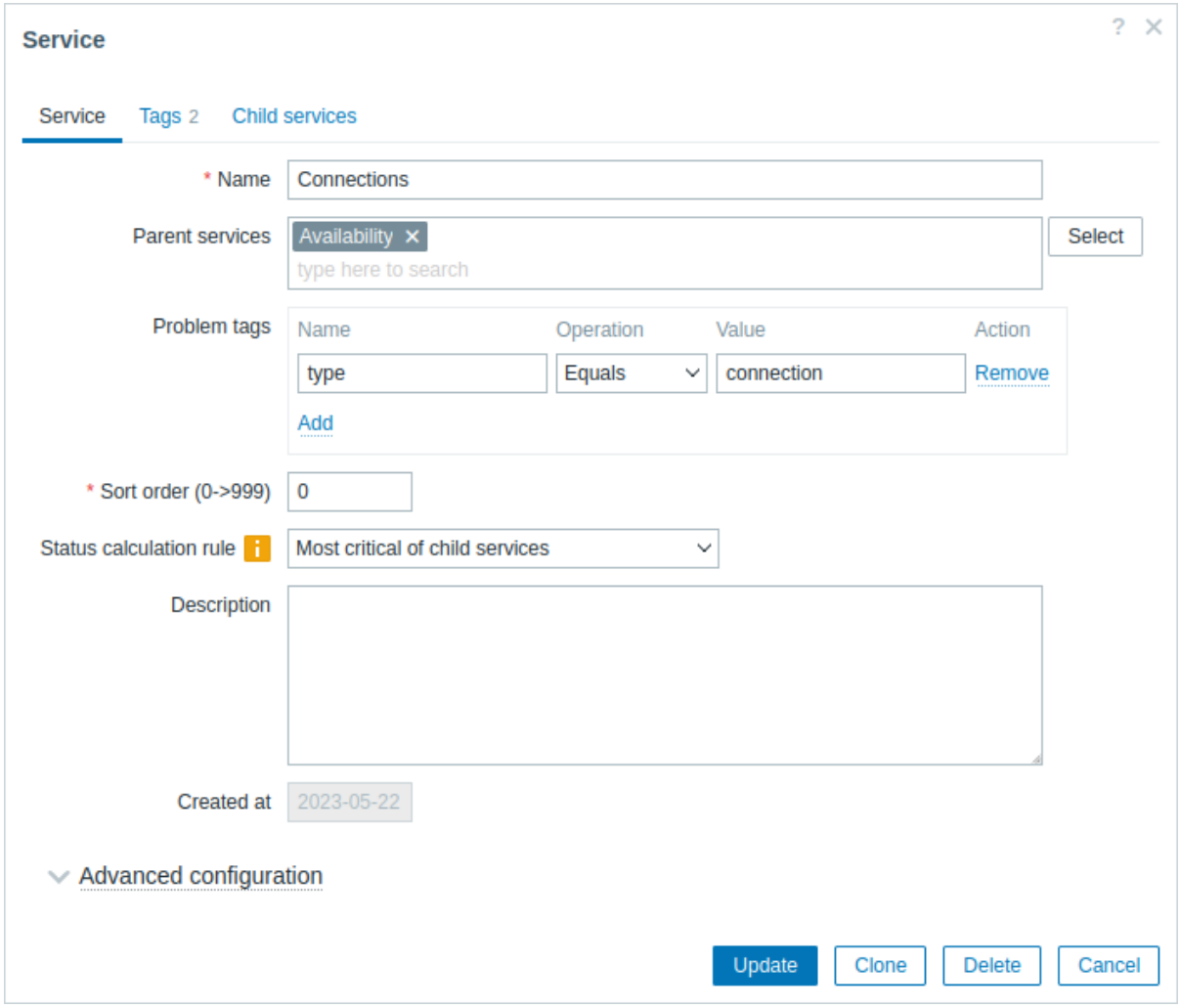
Der Service-Baum wird im Menüabschnitt *Services -> Services* konfiguriert. Wechseln Sie in der oberen rechten Ecke von der Ansicht in den Bearbeitungsmodus.



Um einen neuen Service zu konfigurieren, klicken Sie auf die Schaltfläche *Create service* in der oberen rechten Ecke.

Um schnell einen untergeordneten Service hinzuzufügen, können Sie alternativ auf ein Plus-Symbol neben dem übergeordneten Service klicken. Dadurch wird dasselbe Service-Konfigurationsformular geöffnet, aber der Parameter *Parent services* ist bereits vorausgefüllt.

Service-Konfiguration Geben Sie auf der Registerkarte **Service** die erforderlichen Service-Parameter an:

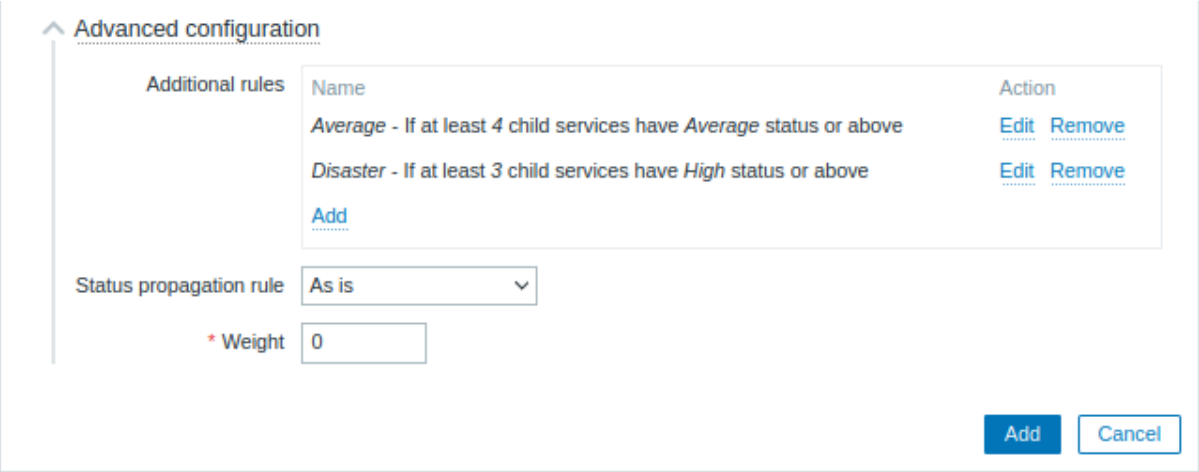


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Service-Name.
Übergeordnete Services	Übergeordnete Services, zu denen der Service gehört. Lassen Sie dieses Feld leer, wenn Sie den Service der höchsten 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. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen zwei Operatoren zur Verfügung: **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, groß-/kleinschreibungssensitiv)
Sortierreihenfolge	Sortierreihenfolge für die Anzeige; der niedrigste Wert wird zuerst angezeigt.
Regel zur Statusberechnung	Regel zur Berechnung des Service-Status: **Am kritischsten, wenn alle untergeordneten Services Probleme haben** - das kritischste Problem in den untergeordneten Services wird verwendet, um den Service-Status einzufärben, wenn alle untergeordneten Services Probleme haben **Am kritischsten der untergeordneten Services** - das kritischste Problem in den untergeordneten Services wird verwendet, um den Service-Status einzufärben **Status auf OK setzen** - den Service-Status nicht berechnen Zusätzliche Regeln zur Statusberechnung können in den Optionen der erweiterten Konfiguration konfiguriert werden.
Beschreibung	Service-Beschreibung.
Erstellt am	Der Zeitpunkt, zu dem der Service erstellt wurde; wird beim Bearbeiten eines vorhandenen Service angezeigt.

Parameter	Beschreibung
Erweiterte Konfiguration	Klicken Sie auf die Bezeichnung *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

Erweiterte Konfiguration



Parameter	Beschreibung
Zusätzliche Regeln	Klicken Sie auf *Add*, um zusätzliche Regeln für die Statusberechnung zu konfigurieren.
Status setzen auf	Setzen Sie den Service-Status bei Erfüllung einer Bedingung auf *OK* (Standard), *Nicht klassifiziert*, *Information*, *Warnung*, *Durchschnitt*, *Hoch* oder *Katastrophe*.
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 untergeordneter Services mit dem Status (Status) oder höher mindestens (W) beträgt** **wenn das Gewicht untergeordneter Services mit dem Status (Status) oder höher mindestens (N%) beträgt** **wenn das Gewicht untergeordneter Services mit dem Status (Status) oder niedriger weniger als (W) beträgt** **wenn das Gewicht untergeordneter 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 der höchste Schweregrad gesetzt.
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), *Nicht klassifiziert*, *Information*, *Warnung*, *Durchschnitt*, *Hoch* oder *Katastrophe*.
Regel für Statusweitergabe	Regel für die Weitergabe des Service-Status an den übergeordneten Service: **Wie er ist** - der Status wird unverändert weitergegeben **Erhöhen um** - Sie können den weitergegebenen Status um 1 bis 5 Schweregrade erhöhen **Verringern um** - Sie können den weitergegebenen Status um 1 bis 5 Schweregrade verringern **Diesen Service ignorieren** - der Status wird überhaupt nicht an den übergeordneten Service weitergegeben **Fester Status** - der Status wird statisch weitergegeben, d. h. immer derselbe
Gewicht	Gewicht des Service (Ganzzahl im Bereich von 0 (Standard) bis 1000000).

Note:
Zusätzliche Regeln für die Statusberechnung können nur verwendet werden, um den Schweregrad über das gemäß dem Hauptparameter *Regel für Statusberechnung* berechnete Niveau hinaus zu erhöhen. Wenn der Status gemäß den zusätzlichen Regeln *Warnung* sein sollte, gemäß der *Regel für Statusberechnung* jedoch *Katastrophe* ist, dann hat der Service den Status *Katastrophe*.

Die Registerkarte **Tags** enthält Tags auf Service-Ebene. Tags auf Service-Ebene 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).

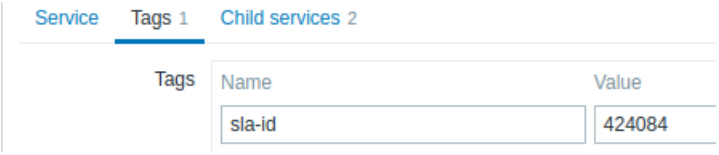
Die Registerkarte **Untergeordnete Services** ermöglicht es, abhängige Services anzugeben. Klicken Sie auf *Hinzufügen*, um einen Service aus der Liste der vorhandenen Services hinzuzufügen. Wenn Sie einen neuen untergeordneten Service hinzufügen möchten, speichern Sie diesen Service zuerst und klicken Sie dann auf das Plus-Symbol neben dem Service, den Sie gerade erstellt haben.

Tags Es gibt zwei verschiedene Arten von Tags in Services:

- Service-Tags
- Problem-Tags

Service-Tags

Service-Tags werden verwendet, um Services mit Service-Aktionen und SLAs abzugleichen. Diese Tags werden im Service-Konfigurationsreiter *Tags* angegeben. Für die Zuordnung von SLAs wird eine *ODER*-Logik verwendet: Ein Service wird einem SLA zugeordnet, wenn er mindestens ein übereinstimmendes Tag hat. Bei Service-Aktionen sind die Zuordnungsregeln konfigurierbar und können entweder *UND*-, *ODER*- oder *UND/ODER*-Logik verwenden.

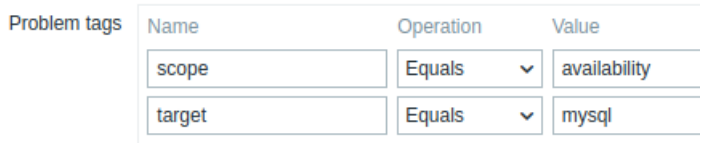


Problem-Tags

Problem-Tags werden verwendet, um Probleme und Services abzugleichen. Diese Tags werden auf der primären Registerkarte der Service-Konfiguration angegeben.

Nur untergeordnete Services der niedrigsten Hierarchieebene dürfen definierte Problem-Tags haben und direkt mit Problemen korreliert werden. Wenn die Problem-Tags übereinstimmen, ändert sich der Service-Status auf denselben Status wie das Problem. Bei mehreren Problemen hat ein Service den Status des Problems mit dem höchsten Schweregrad. Der Status eines übergeordneten Service wird dann basierend auf den Statuswerten der untergeordneten Services gemäß den Regeln zur Statusberechnung berechnet.

Wenn mehrere Tags angegeben sind, wird eine *UND*-Logik verwendet: Ein Problem muss alle in der Service-Konfiguration angegebenen Tags haben, um dem Service zugeordnet zu werden.



Note:
Ein Problem in Zabbix erbt Tags aus der gesamten Kette von Vorlagen, Hosts, Datenpunkten, Webszenarien und Auslösern. Jedes 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 *Warnung*

Der Service **Web camera 3** hat nur den angegebenen Problem-Tag: `name:webcam-3`

Der Service-Status ändert sich von *OK* zu *Warnung*, 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, da die Bedingungen nur teilweise erfüllt sind.

Vorhandene Services ändern

> **Note:**
> Die unten beschriebenen Schaltflächen sind nur sichtbar, wenn sich der Abschnitt *Services* im Bearbeitu

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 in der Service-Hierarchie um eine Ebene nach oben verschoben (Services der 1. Ebene erhalten dieselbe Ebene wie der gelöschte übergeordnete Service).

Zwei Schaltflächen unterhalb der Service-Liste bieten einige Optionen zur Massenbearbeitung:

- *Mass update* - Service-Eigenschaften gesammelt aktualisieren
- *Delete* - die Services löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den entsprechenden Services und klicken Sie dann auf die gewünschte Schaltfläche.

2 SLA

Übersicht Sobald die Services erstellt wurden, können Sie mit der Überwachung beginnen, ob ihre Leistung den Service Level Agreements (SLA) entspricht.

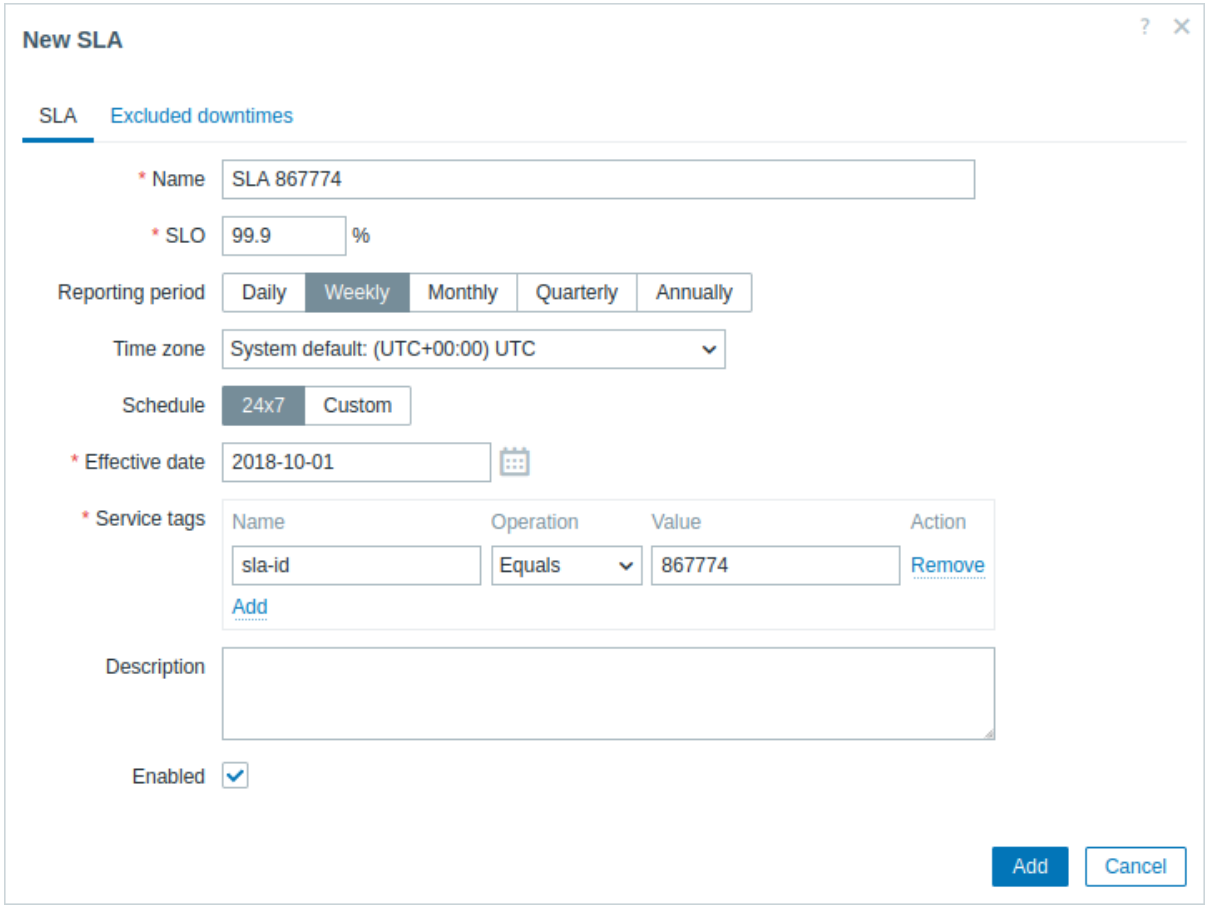
Im Menübereich *Services->SLA* können SLAs für verschiedene Services konfiguriert werden. Ein SLA in Zabbix definiert das Service Level Objective (SLO), den erwarteten Uptime-Zeitplan und geplante Ausfallzeiten.

SLAs und Services werden über Service-Tags 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 sein – die Daten für jedes dieser SLAs werden separat angezeigt.

In SLA-Berichten stellt Zabbix Daten zum Service Level Indicator (SLI) bereit, der die tatsächliche Serviceverfügbarkeit misst. Ob ein Service die SLA-Ziele erfüllt, wird durch den Vergleich von SLO (erwartete Verfügbarkeit in %) mit SLI (tatsächliche Verfügbarkeit in %) bestimmt.

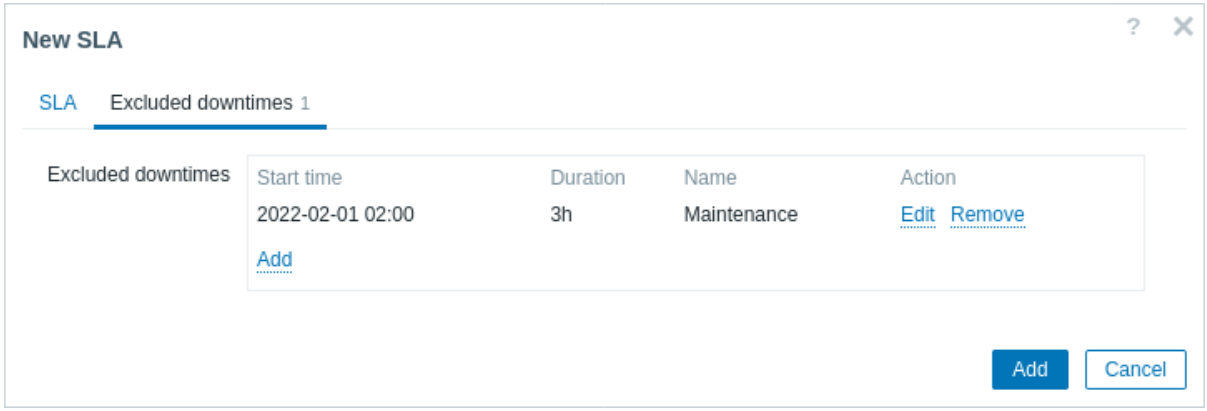
Konfiguration Um ein neues SLA zu erstellen, klicken Sie auf die Schaltfläche *SLA erstellen*.

Auf der Registerkarte **SLA** können allgemeine SLA-Parameter festgelegt werden.



Parameter	Beschreibung
Name	Geben Sie den SLA-Namen ein.
SLO	Geben Sie das Service Level Objective (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–Mär), Q2 (Apr–Jun), Q3 (Jul–Sep), Q4 (Okt–Dez). **Jährlich** - jedes Kalenderjahr, vom 1. Januar 00:00:00 bis zum 31. Dezember 23:59:59.
Zeitzone	Wählen Sie die SLA-Zeitzone aus.
Zeitplan	Wählen Sie den SLA-Zeitplan aus - 24x7 oder benutzerdefiniert.
Gültig ab	Wählen Sie das Datum aus, ab dem die SLA-Berechnung beginnt.
Service-Tags	Fügen Sie Service-Tags hinzu, um die Services zu identifizieren, auf die dieses SLA angewendet werden soll. **Name** - Name des Service-Tags, muss exakt übereinstimmen, Groß-/Kleinschreibung wird beachtet. **Operation** - wählen Sie *Gleich* aus, wenn der Tag-Wert exakt übereinstimmen muss (Groß-/Kleinschreibung wird beachtet), oder *Enthält*, wenn ein Teil des Tag-Werts übereinstimmen muss (Groß-/Kleinschreibung wird nicht beachtet). **Wert** - Wert des Service-Tags, nach dem entsprechend der ausgewählten Operation gesucht werden soll. Das SLA wird auf einen Service angewendet, wenn mindestens ein Service-Tag übereinstimmt.
Beschreibung	Fügen Sie eine Beschreibung für das SLA hinzu.
Aktiviert	Aktivieren Sie das Kontrollkästchen, um die SLA-Berechnung zu aktivieren.

Auf der Registerkarte **Ausgeschlossene Ausfallzeiten** können Ausfallzeiten festgelegt werden, die von der SLA-Berechnung ausgeschlossen sind.



Klicken Sie auf *Hinzufügen*, um ausgeschlossene Ausfallzeiten zu konfigurieren, und geben Sie dann 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* durch Klicken auf den Hyperlink des SLA-Berichts;
- im Abschnitt *Services* durch Klicken auf den SLA-Namen auf der Registerkarte „Info“;
- im Dashboard-Widget *SLA report*.

Sobald ein SLA konfiguriert ist, zeigt die Registerkarte *Info* im Abschnitt „Services“ ebenfalls einige Informationen zur Service-Performance an.

3 Einrichtungsbeispiel

Übersicht Dieser Abschnitt beschreibt eine einfache Einrichtung zur Überwachung eines Zabbix-Hochverfügbarkeitsclusters als Dienst.

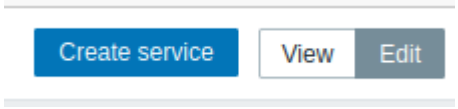
Voraussetzungen Bevor Sie die Service-Überwachung konfigurieren, müssen die Hosts konfiguriert sein:

- *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`

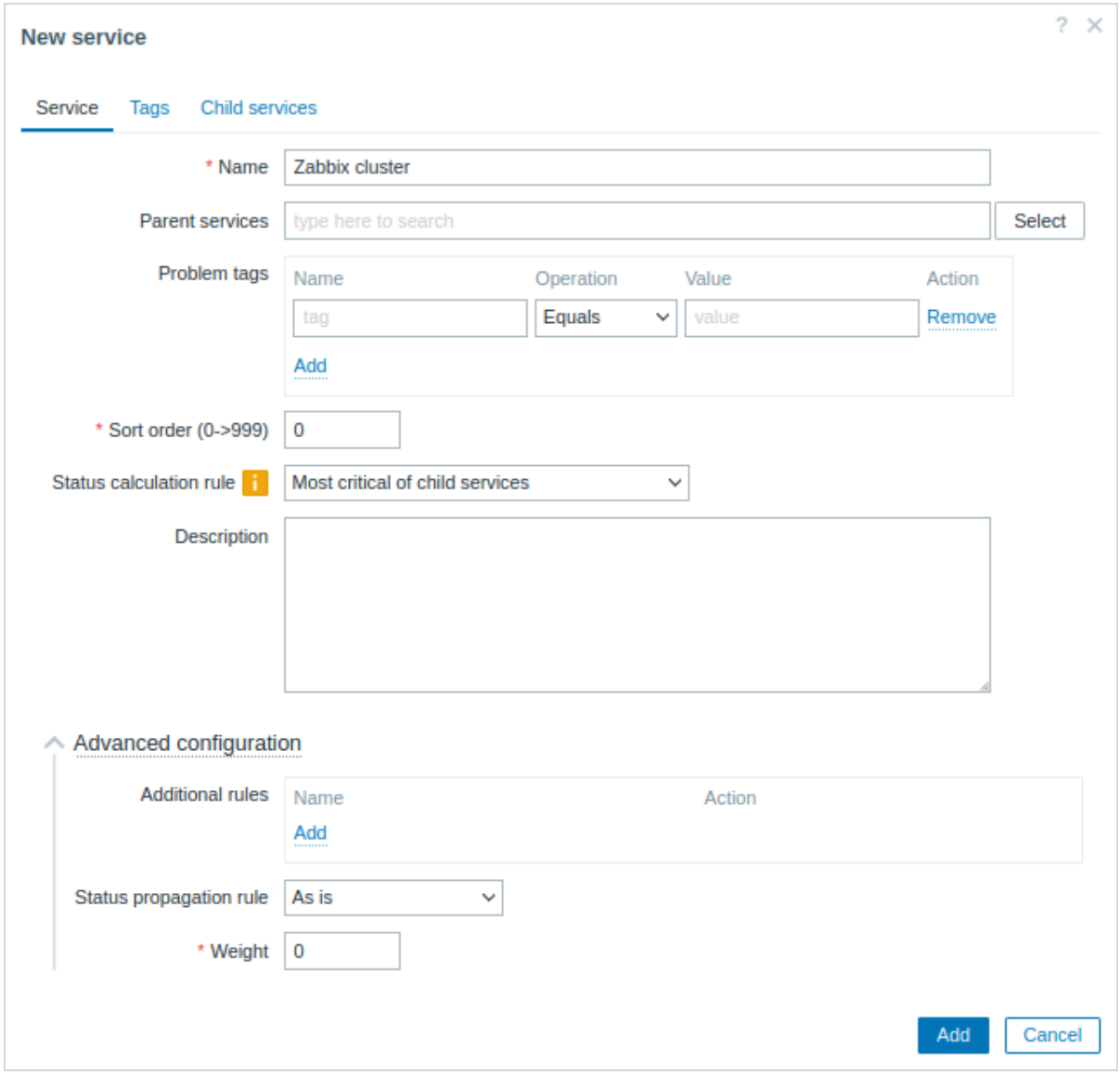
Service-Baum Der nächste Schritt besteht darin, den Service-Baum zu erstellen. In diesem Beispiel ist die Infrastruktur sehr einfach und besteht aus drei Services: *Zabbix cluster* (übergeordnet) und zwei untergeordneten Services *Zabbix server node 1* und *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

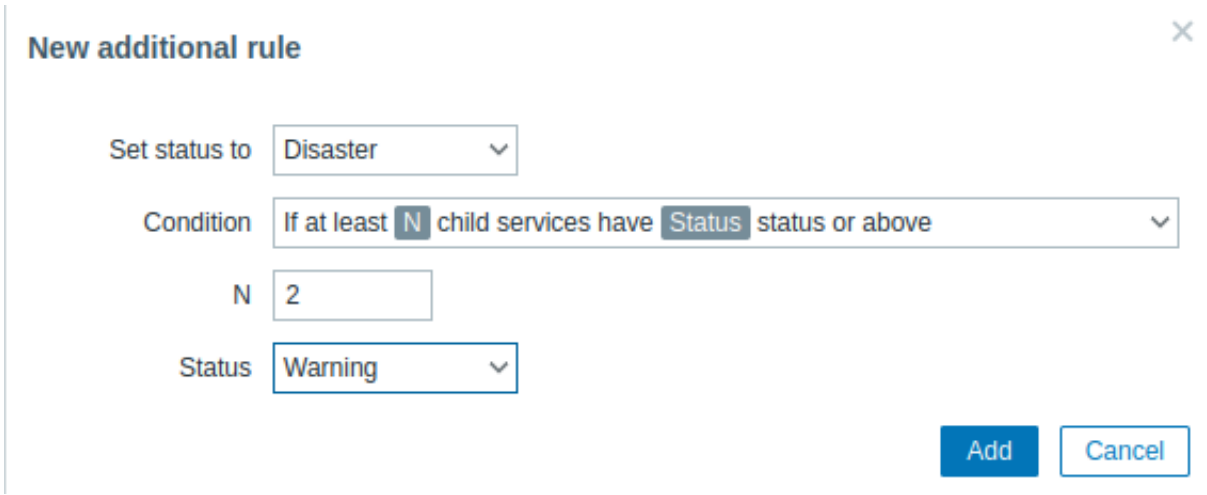
Aktivieren Sie auf der Seite **Services** den *Edit*-Modus und klicken Sie auf **Create service**:



Geben Sie im Service-Konfigurationsfenster den Namen *Zabbix cluster* ein und klicken Sie auf die Beschriftung *Advanced configuration*, um die erweiterten Konfigurationsoptionen anzuzeigen.



Konfigurieren Sie eine zusätzliche Regel:

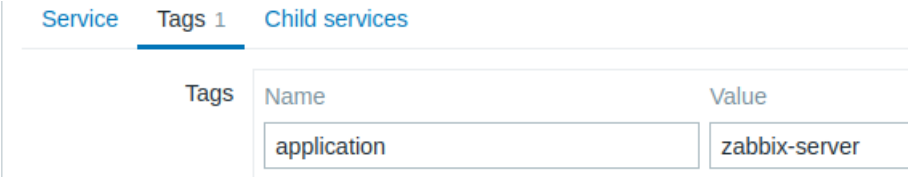


Zabbix cluster wird zwei untergeordnete Services haben – einen für jeden der HA-Knoten. Wenn beide HA-Knoten Probleme mit mindestens dem Status *Warning* haben, sollte der Status des übergeordneten Service auf *Disaster* gesetzt werden. Um dies zu erreichen, sollte die zusätzliche Regel wie folgt konfiguriert werden:

- Set status to: Disaster
- Condition: If at least `N` child services have `Status` status or above
- N: 2
- Status: Warning

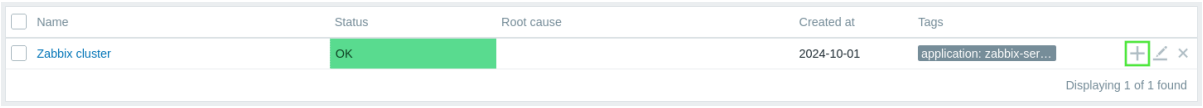
Wechseln Sie zur Registerkarte *Tags* und fügen Sie das Tag `application:zabbix-server` hinzu. Dieses Tag wird später für

Service-Aktionen und SLA-Berichte verwendet.



Speichern Sie den neuen Service.

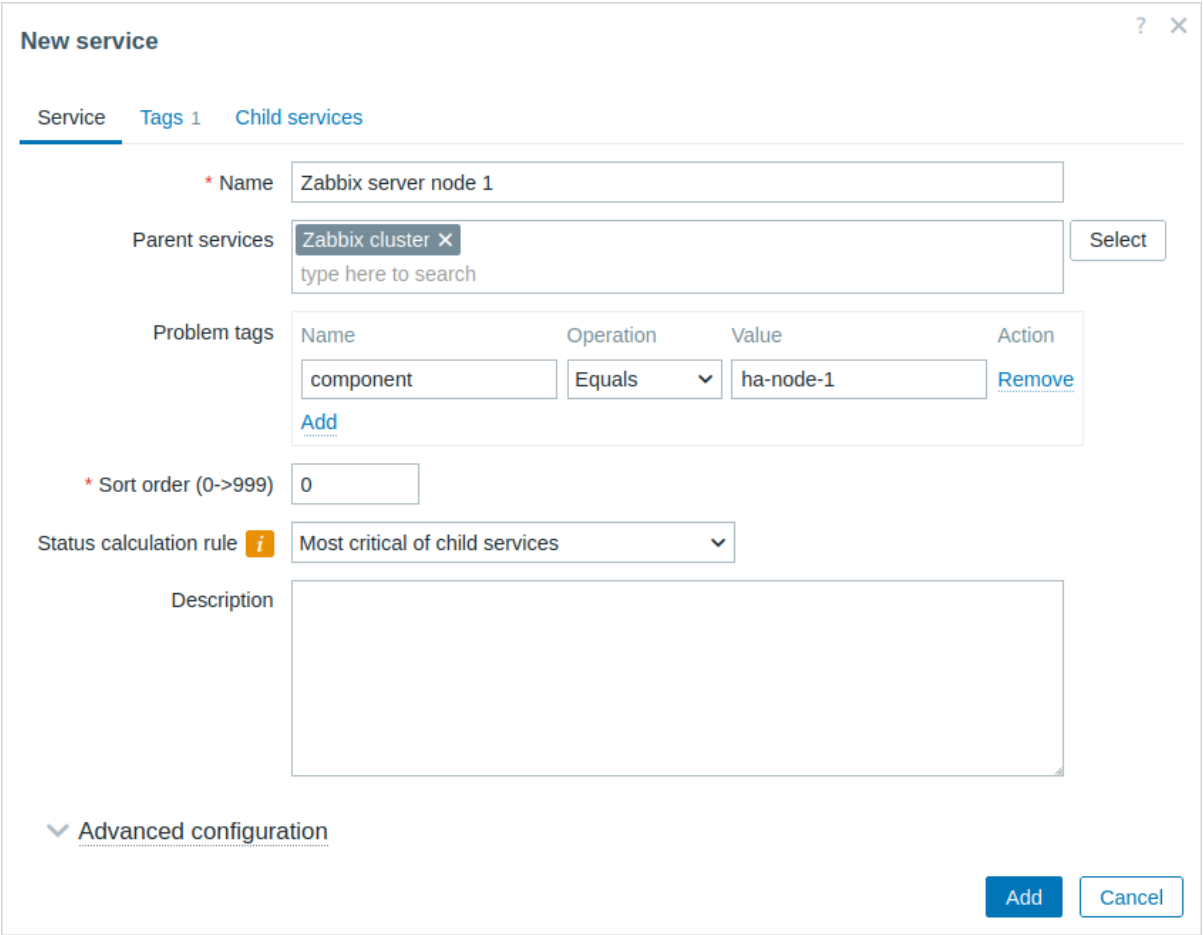
Um einen untergeordneten Service hinzuzufügen, klicken Sie auf das Plus-Symbol neben dem Service *Zabbix cluster* (das Symbol ist nur im Modus *Bearbeiten* sichtbar).



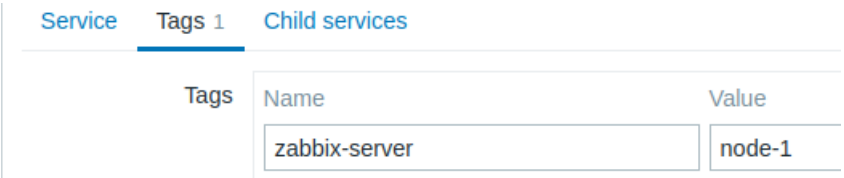
Geben Sie im Service-Konfigurationsfenster den Namen *Zabbix server node 1* ein. Beachten Sie, dass der Parameter Parent services bereits mit *Zabbix cluster* vorausgefüllt ist.

Die Verfügbarkeit dieses Service wird durch Probleme auf dem Host *HA node 1* beeinflusst, die mit dem Problem-Tag `component:ha-node-1` gekennzeichnet sind. Geben Sie im Parameter Problem tags Folgendes ein:

- Name: component
- Operation: Equals
- Value: 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.

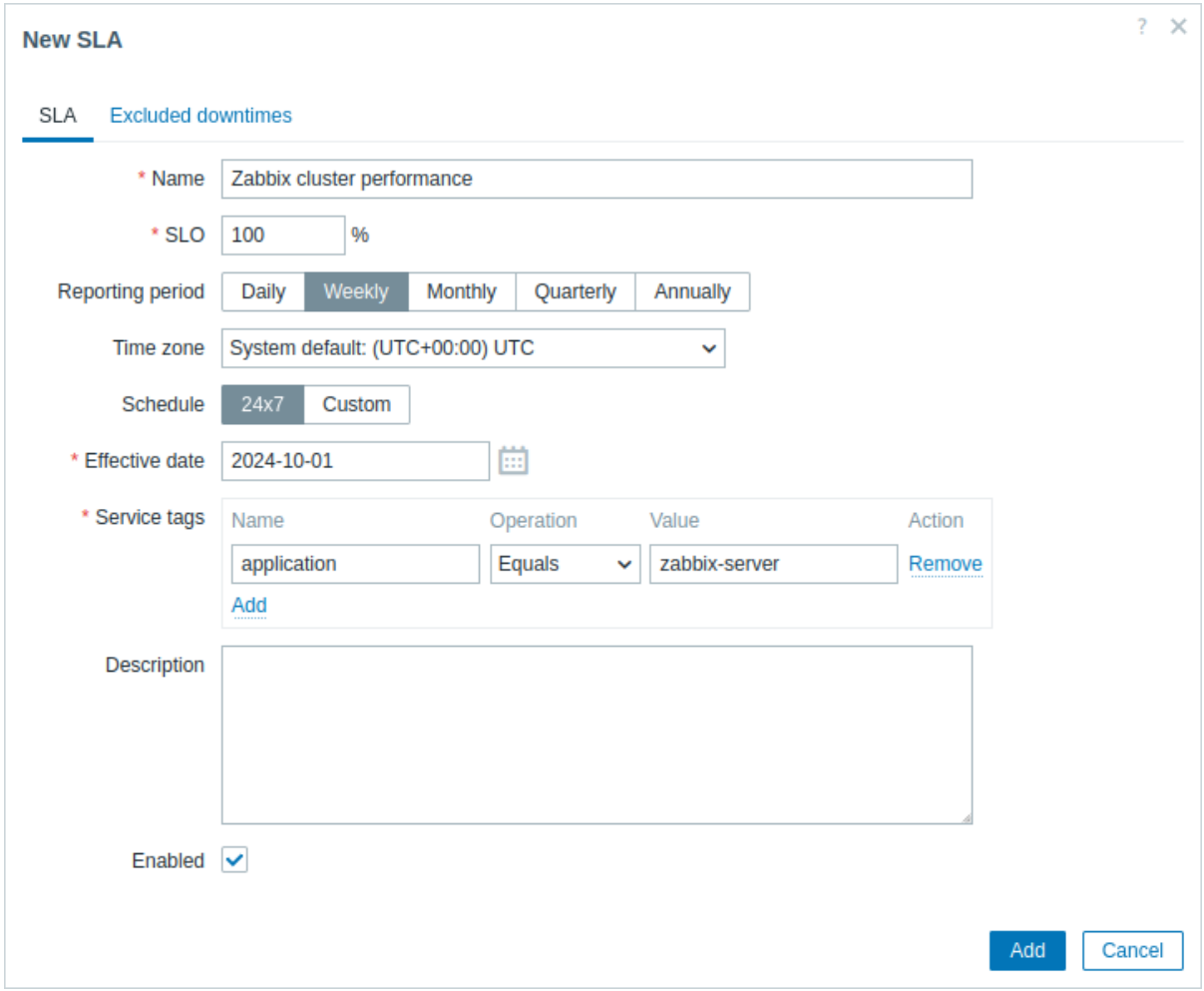
SLA In diesem Beispiel beträgt die erwartete Leistung des Zabbix-Clusters 100 %, wobei eine halbjährliche Wartungszeit von einer Stunde ausgenommen ist.

Zuerst müssen Sie eine neue Service-Level-Vereinbarung hinzufügen.

Gehen Sie zum Menüabschnitt *Services->SLA* und klicken Sie auf SLA erstellen. Geben Sie den Namen *Zabbix cluster performance* ein und setzen Sie das SLO auf 100 %.

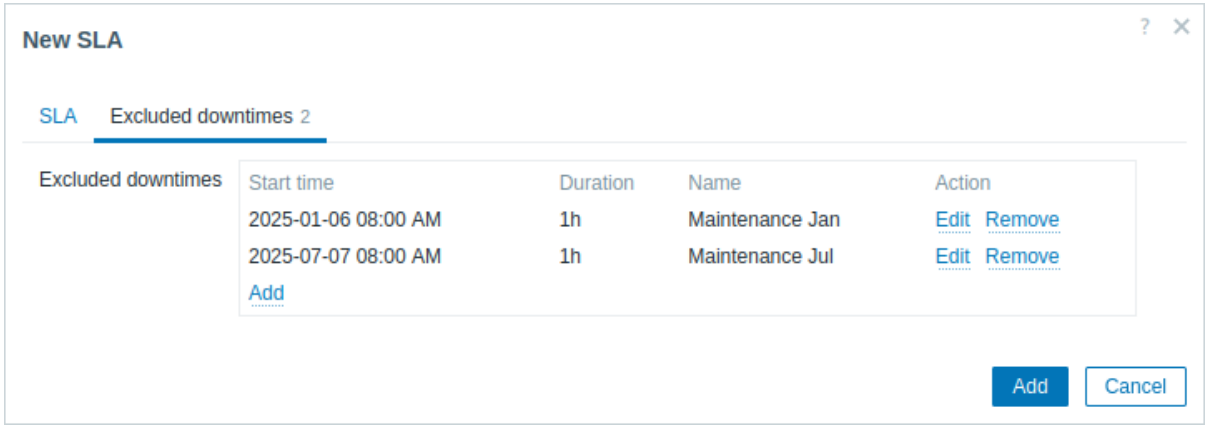
Der Service Zabbix cluster hat das Service-Tag `application:zabbix-server`. Um diese SLA zur Messung der Leistung des Zabbix-Clusters zu verwenden, geben Sie im Parameter *Service tags* Folgendes an:

- Name: application
- Operation: Equals
- Value: zabbix-server



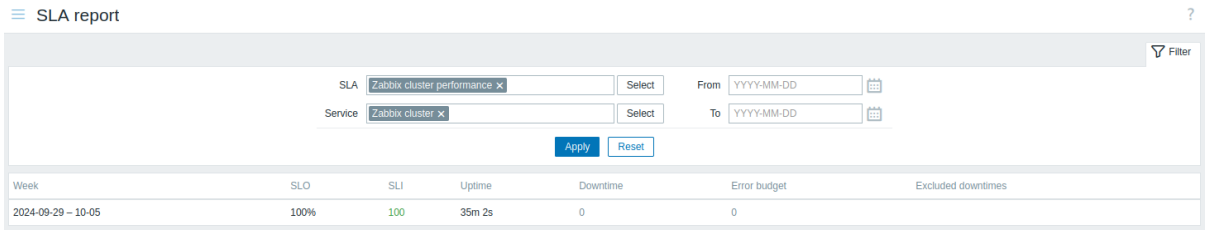
In einer realen Umgebung können Sie außerdem den gewünschten Berichtszeitraum, die Zeitzone und das Startdatum aktualisieren oder den Zeitplan von 24/7 auf benutzerdefiniert ändern. Für dieses Beispiel sind die Standardeinstellungen ausreichend.

Wechseln Sie zur Registerkarte *Excluded downtimes* und fügen Sie Ausfallzeiten für geplante Wartungszeiträume hinzu, um diese Zeiträume von der SLA-Berechnung auszuschließen. Klicken Sie im Abschnitt Excluded downtimes auf den Link Add und geben Sie den Namen der Ausfallzeit, die geplante Startzeit und die Dauer ein.

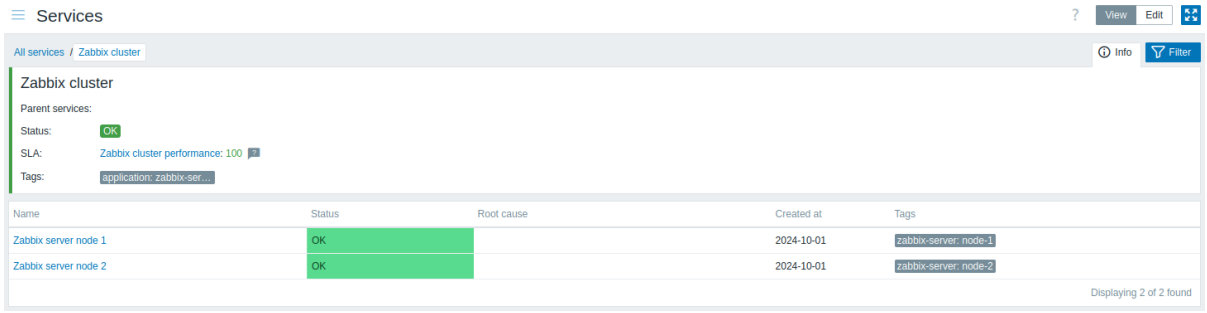


Klicken Sie auf Add, um die neue SLA zu speichern.

Wechseln Sie zum Abschnitt mit den SLA-Berichten, um den SLA-Bericht für Zabbix cluster anzuzeigen.



Die SLA-Informationen können auch im Abschnitt *Services* geprüft werden.



7 Web-Überwachung

Übersicht Mit Zabbix können Sie mehrere Verfügbarkeitsaspekte von Websites prüfen.

Attention:
Um Web-Monitoring durchzuführen, muss der Zabbix Server zunächst mit Unterstützung für cURL (libcurl) konfiguriert werden.

Um Web-Monitoring zu aktivieren, müssen Sie Webszenarien definieren. Ein Webszenario besteht aus einer oder mehreren HTTP-Anfragen oder „Schritten". Die Schritte werden vom Zabbix Server periodisch in einer vordefinierten Reihenfolge ausgeführt. Wenn ein Host durch einen Proxy überwacht wird, werden die Schritte vom Proxy ausgeführt.

Webszenarien werden auf dieselbe Weise an Hosts/Vorlagen angehängt wie Datenpunkte, Auslöser usw. Das bedeutet, dass Webszenarien auch auf Vorlagenebene erstellt und dann in einem Schritt auf mehrere Hosts angewendet werden können.

Die folgenden Informationen werden in jedem Webszenario erfasst:

- durchschnittliche Download-Geschwindigkeit pro Sekunde für alle Schritte des gesamten Szenarios
- Nummer des fehlgeschlagenen Schritts
- letzte Fehlermeldung

Die folgenden Informationen werden in jedem Schritt eines Webszenarios erfasst:

- Download-Geschwindigkeit pro Sekunde

- Antwortzeit
- Antwortcode

Weitere Details finden Sie unter Web-Monitoring-Datenpunkte.

Die aus der Ausführung von Webszenarien erfassten Daten werden in der Datenbank gespeichert. Die Daten werden automatisch für Diagramme, Auslöser und Benachrichtigungen verwendet.

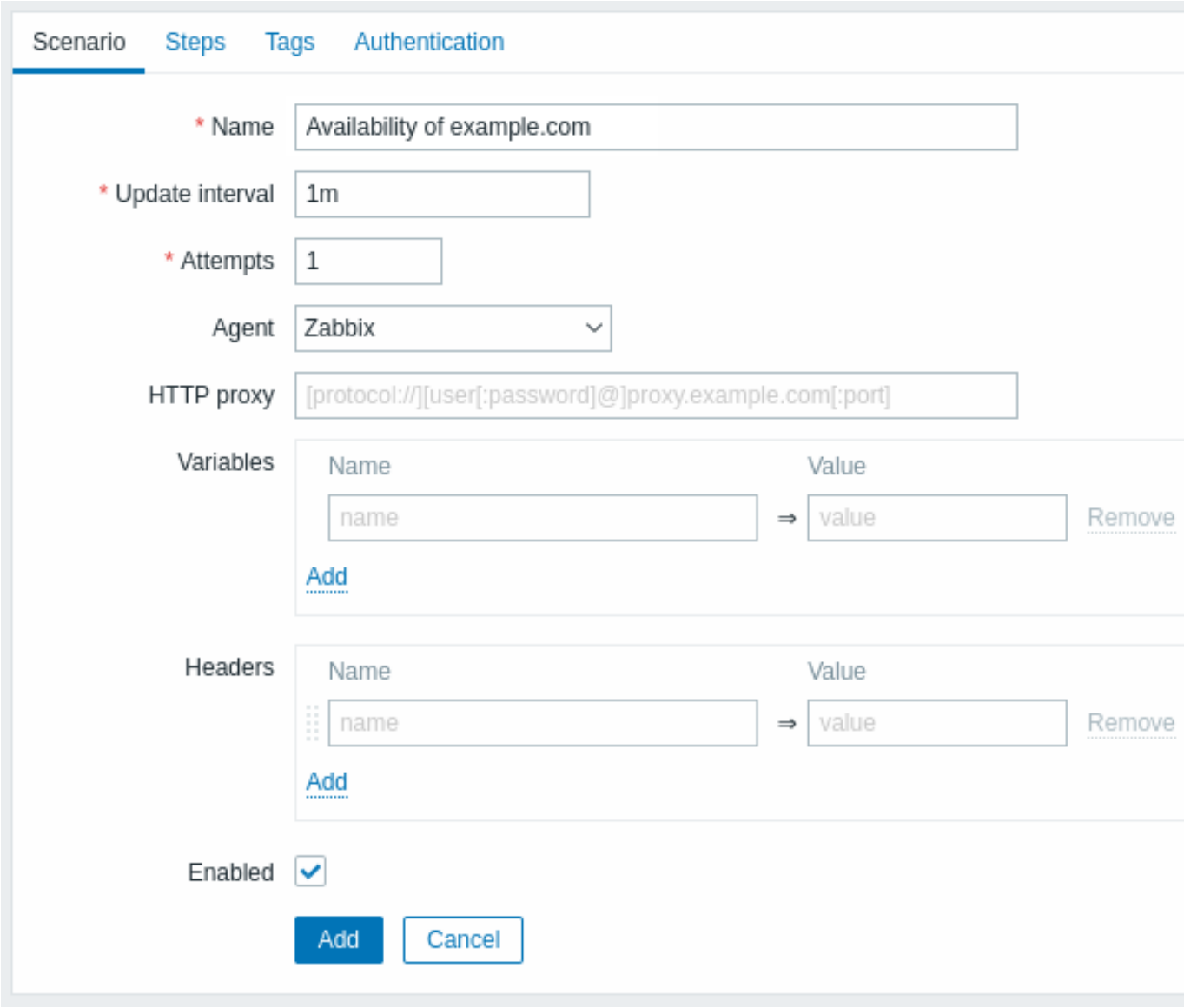
Zabbix kann auch prüfen, ob eine abgerufene HTML-Seite eine vordefinierte Zeichenfolge enthält. Es kann eine simulierte Anmeldung ausführen und einem Pfad simulierter Mausklicks auf der Seite folgen.

Das Zabbix Web-Monitoring unterstützt sowohl HTTP als auch HTTPS. Beim Ausführen eines Webszenarios folgt Zabbix 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). Alle Cookies bleiben während der Ausführung eines einzelnen Szenarios erhalten.

Konfiguration eines Webszenarios So konfigurieren Sie ein Webszenario:

- Gehen Sie zu: *Datenerfassung* → *Hosts* (oder *Vorlagen*)
- Klicken Sie in der Zeile des Hosts/der Vorlage auf *Web*
- Klicken Sie rechts auf *Webszenario erstellen* (oder auf den Namen des Szenarios, um ein vorhandenes Szenario zu bearbeiten)
- Geben Sie die Parameter des Szenarios im Formular ein

Die Registerkarte **Szenario** ermöglicht Ihnen, die allgemeinen Parameter eines Webszenarios zu konfigurieren.



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Szenarioparameter:

Parameter	Beschreibung
Name	Eindeutiger Szenarioname. Benutzermakros werden unterstützt. *Beachten Sie*, dass bei Verwendung von Benutzermakros diese Makros in den Namen von Web-Monitoring-Datenpunkten nicht aufgelöst werden.
Aktualisierungsintervall	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 Änderung seines Werts (z. B. 5m → 30s) die nächste Prüfung entsprechend 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 von Webszenario-Schritten. Bei Netzwerkproblemen (Timeout, keine Verbindung usw.) kann Zabbix die Ausführung eines Schritts mehrmals wiederholen. Der eingestellte Wert wirkt sich gleichermaßen auf jeden Schritt des Szenarios aus. Es können bis zu 10 Versuche angegeben werden, der Standardwert ist 1. *Hinweis*: Zabbix wiederholt einen Schritt nicht aufgrund eines falschen Antwortcodes oder einer Nichtübereinstimmung mit einer erforderlichen Zeichenfolge.
Agent	Wählen Sie einen Client-Agenten aus. Zabbix gibt sich als der ausgewählte Browser aus. Dies ist nützlich, wenn eine Website je nach Browser unterschiedliche Inhalte zurückgibt. Benutzermakros können in diesem Feld verwendet werden.
HTTP-Proxy	Sie können einen zu verwendenden 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 Protokollpräfixe wurde in cURL 7.21.7 hinzugefügt). Wenn kein Protokoll angegeben ist, wird der Proxy als HTTP-Proxy behandelt. Standardmäßig wird Port 1080 verwendet. Falls angegeben, überschreibt der Proxy proxybezogene Umgebungsvariablen wie http_proxy, HTTPS_PROXY. Falls nicht angegeben, überschreibt der Proxy keine proxybezogenen Umgebungsvariablen. Der eingegebene Wert wird unverändert weitergegeben; es findet keine Plausibilitätsprüfung statt. Sie können auch eine SOCKS-Proxy-Adresse eingeben. Wenn Sie das falsche Protokoll angeben, schlägt die Verbindung fehl und der Datenpunkt wird nicht unterstützt. *Beachten Sie*, dass beim HTTP-Proxy nur einfache Authentifizierung unterstützt wird. Benutzermakros können in diesem Feld verwendet werden.
Variablen	Variablen, die in Szenarioschritten 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) Zum Beispiel: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} Auf die Makros kann dann in den Schritten als {username}, {password}, {hostid} usw. verwiesen 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 erhalten, sodass der extrahierte Wert erst im darauffolgenden Schritt angewendet werden kann. Wenn der Wertteil mit `regex:` beginnt, wird der nachfolgende 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, müssen jedoch manuell URL-kodiert werden, wenn sie in rohen Post-Daten oder direkt in der URL verwendet werden.

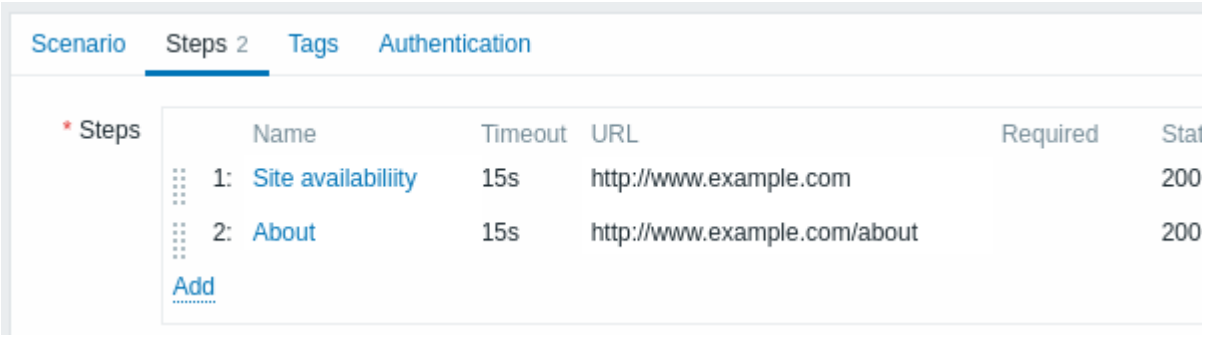
Parameter	Beschreibung
Header	HTTP-Header werden bei der Ausführung einer Anfrage verwendet. Es können Standard- und benutzerdefinierte Header verwendet werden. Header werden abhängig vom auf Szenarioebene in einer Dropdown-Liste ausgewählten Agent-Typ mit Standardeinstellungen zugewiesen und auf alle Schritte angewendet, sofern sie nicht auf Schrittebene benutzerdefiniert festgelegt werden. **Beachten Sie, dass das Definieren eines Headers auf Schrittebene automatisch alle zuvor definierten Header verwirft, mit Ausnahme eines Standard-Headers, der durch Auswahl von 'User-Agent' in einer Dropdown-Liste auf Szenarioebene zugewiesen wird.** Allerdings kann selbst der Standard-Header 'User-Agent' überschrieben werden, indem er auf Schrittebene angegeben wird. Um den Header auf Szenarioebene aufzuheben, sollte der Header auf Schrittebene mit Namen, aber ohne Wert angegeben werden. Header sollten mit derselben Syntax aufgeführt werden, wie sie im HTTP-Protokoll erscheinen würden, optional unter Verwendung einiger zusätzlicher Funktionen, die von der cURL-Option CURLOPT_HTTPHEADER unterstützt werden. Zum Beispiel: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Benutzermakros und {HOST.*}-Makros werden unterstützt.
Aktiviert	Das Szenario ist aktiv, wenn dieses Kontrollkästchen aktiviert ist, andernfalls deaktiviert.

Beachten Sie, dass beim Bearbeiten eines vorhandenen Szenarios im Formular zwei zusätzliche Schaltflächen verfügbar sind:

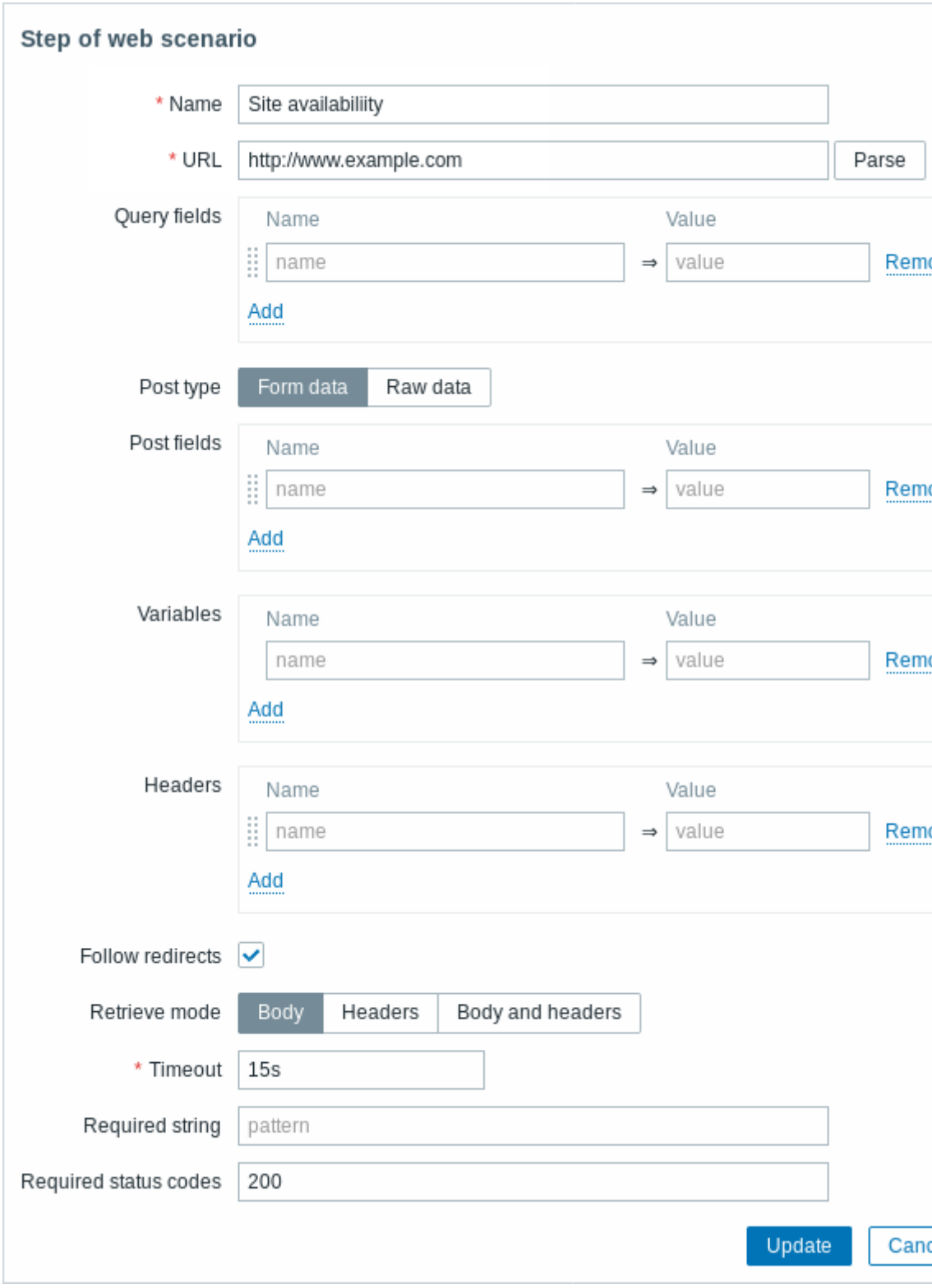
Clone	Ein weiteres Szenario auf Grundlage der Eigenschaften des vorhandenen erstellen.
Clear history and trends	Verlaufs- und Trenddaten für das Szenario löschen. Dadurch führt der Server das Szenario unmittelbar nach dem Löschen der Daten aus.

Note:
Wenn das Feld *HTTP-Proxy* leer gelassen wird, besteht eine weitere Möglichkeit zur Verwendung eines HTTP-Proxys darin, proxybezogene Umgebungsvariablen zu setzen.
Für HTTP-Prüfungen setzen Sie die Umgebungsvariable **http_proxy** für den Benutzer des Zabbix-Servers. Zum Beispiel: `http_proxy=http://proxy_ip:proxy_port`.
Für HTTPS-Prüfungen setzen Sie die Umgebungsvariable **HTTPS_PROXY**. Zum Beispiel `HTTPS_PROXY=http://proxy_ip:proxy_port`. Weitere Details erhalten Sie durch Ausführen des Shell-Befehls: *# man curl*.

Die Registerkarte **Schritte** ermöglicht Ihnen, die Schritte des Webszenarios zu konfigurieren. Um einen Webszenario-Schritt hinzuzufügen, klicken Sie im Block *Schritte* auf *Hinzufügen*.



Note:
Geheime Benutzermakros dürfen nicht in URLs verwendet werden, da sie zu "******" aufgelöst werden.



Konfigurieren von Schritten

Schrittparameter:

Parameter	Beschreibung
Name	Eindeutiger Schrittname. Benutzermakros werden unterstützt. *Beachten Sie*, dass diese Makros, wenn Benutzermakros verwendet werden, in Namen von web monitoring-Datenpunkten nicht aufgelöst werden.

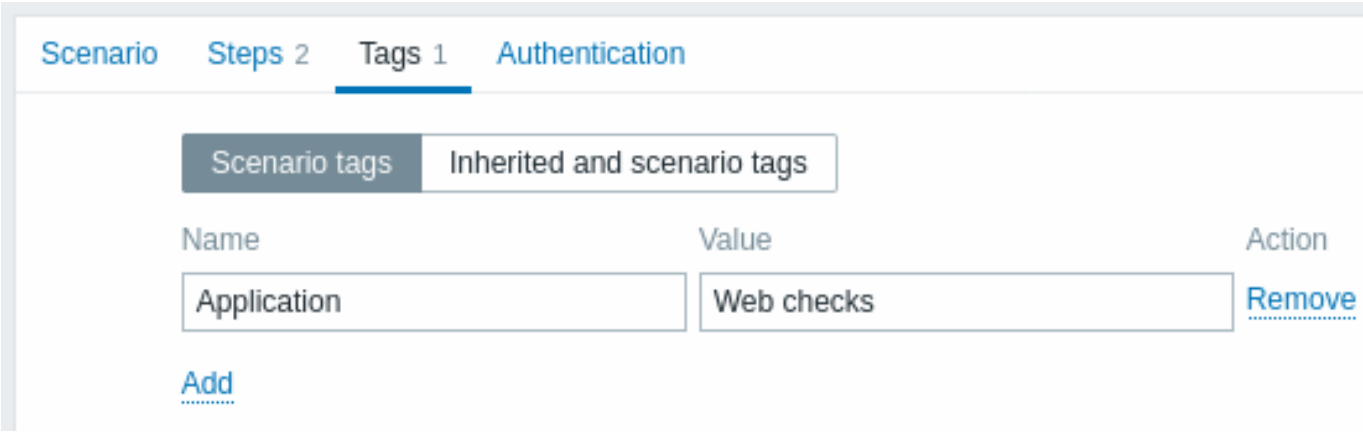
Parameter	Beschreibung
URL	URL, zu der eine Verbindung hergestellt und von der Daten abgerufen werden sollen. Zum Beispiel: https://www.example.com http://www.example.com/download Domainnamen können mit Unicode-Zeichen angegeben werden. Sie werden bei der Ausführung des Webszenario-Schritts automatisch per Punycode in ASCII umgewandelt. Mit der Schaltfläche *Parse* können optionale Abfragefelder (wie ?name=Admin&password=mypassword) von der URL getrennt werden, wobei Attribute und Werte in *Abfragefelder* verschoben werden, damit sie automatisch URL-kodiert werden. Variablen können in der URL mit der Syntax {macro} verwendet werden. Variablen können manuell URL-kodiert werden, indem die Syntax {{macro}.urlencode()} verwendet wird. Benutzermakros und {HOST.*}-Makros werden unterstützt. Begrenzt auf 2048 Zeichen.
Query fields	HTTP-GET-Variablen für die URL. Als Attribut-Wert-Paare angegeben. Werte werden automatisch URL-kodiert. Werte aus Szenariovariablen, Benutzermakros oder {HOST.*}-Makros werden aufgelöst und anschließend automatisch URL-kodiert. Die Verwendung der Syntax {{macro}.urlencode()} führt zu einer doppelten URL-Kodierung. Benutzermakros und {HOST.*}-Makros werden unterstützt.
Post	HTTP-POST-Variablen. Im Modus **Form data** 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 **Raw data** werden Attribute/Werte in einer einzelnen Zeile angezeigt und mit dem Symbol **&** verkettet. Rohwerte können manuell URL-kodiert/-dekodiert werden, indem die Syntax {{macro}.urlencode()} oder {{macro}.urldecode()} verwendet wird. 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.
Variables	Variablen auf Schrittebene, die für GET- und POST-Funktionen verwendet werden können. 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 nur auf den folgenden Schritt aus (und nicht auf den aktuellen Schritt). Sie haben das folgende Format: **{macro}**=value **{macro}**=regex:<regular expression> Weitere Informationen finden Sie in der Variablenbeschreibung auf Szenario-Ebene. Variablen werden automatisch URL-kodiert, wenn sie in Abfragefeldern oder Formulardaten für POST-Variablen verwendet werden, müssen jedoch manuell URL-kodiert werden, wenn sie in rohen POST-Daten oder direkt in der URL verwendet werden.
Headers	Benutzerdefinierte HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Als Attribut-Wert-Paare angegeben. Ein auf Schrittebene definierter Header wird für diesen bestimmten Schritt verwendet. **Beachten Sie, dass das Definieren eines Headers auf Schrittebene automatisch alle zuvor definierten Header verwirft, mit Ausnahme eines Standard-Headers, der durch Auswahl von 'User-Agent' aus einer Dropdown-Liste auf Szenarioebene zugewiesen wird.** Allerdings kann selbst der Standard-Header 'User-Agent' überschrieben werden, indem er auf Schrittebene angegeben wird. Wenn Sie beispielsweise einem Header einen Namen zuweisen, aber keinen Wert festlegen, wird der Standard-Header auf Szenarioebene aufgehoben. Benutzermakros und {HOST.*}-Makros werden unterstützt. Dies setzt die cURL-Option CURLOPT_HTTPHEADER.
Follow redirects	Aktivieren Sie das Kontrollkästchen, um HTTP-Weiterleitungen zu folgen. Dies setzt die cURL-Option CURLOPT_FOLLOWLOCATION.

Parameter	Beschreibung
Retrieve mode	Wählen Sie den Abrufmodus aus: **Body** - nur den Body aus der HTTP-Antwort abrufen **Headers** - nur die Header aus der HTTP-Antwort abrufen **Body and headers** - Body und Header aus der HTTP-Antwort abrufen
Timeout	Zabbix verwendet nicht mehr als die festgelegte Zeit für die Verarbeitung der URL (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 Timeout** Sekunden. Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 1h. Benutzermakros werden unterstützt.
Required string	Erforderliches Muster für reguläre Ausdrücke. Wenn der abgerufene Inhalt (HTML) nicht mit dem erforderlichen Muster übereinstimmt, schlägt der Schritt fehl. Wenn leer, wird keine Prüfung der erforderlichen Zeichenfolge durchgeführt. Zum Beispiel: Homepage of Zabbix Welcome.*admin *Hinweis*: Die Referenzierung von regulären Ausdrücken, die im Zabbix Frontend erstellt wurden, wird in diesem Feld nicht unterstützt. Benutzermakros und {HOST.*}-Makros werden unterstützt.
Required status codes	Liste der erwarteten HTTP-Statuscodes. Wenn Zabbix einen Code erhält, der nicht in der Liste enthalten ist, schlägt der Schritt fehl. Wenn leer, wird keine Prüfung der Statuscodes durchgeführt. Zum Beispiel: 200,201,210-299 Benutzermakros werden unterstützt.

Note:
Änderungen in Webszenario-Schritten werden erst gespeichert, wenn das gesamte Szenario gespeichert wird.

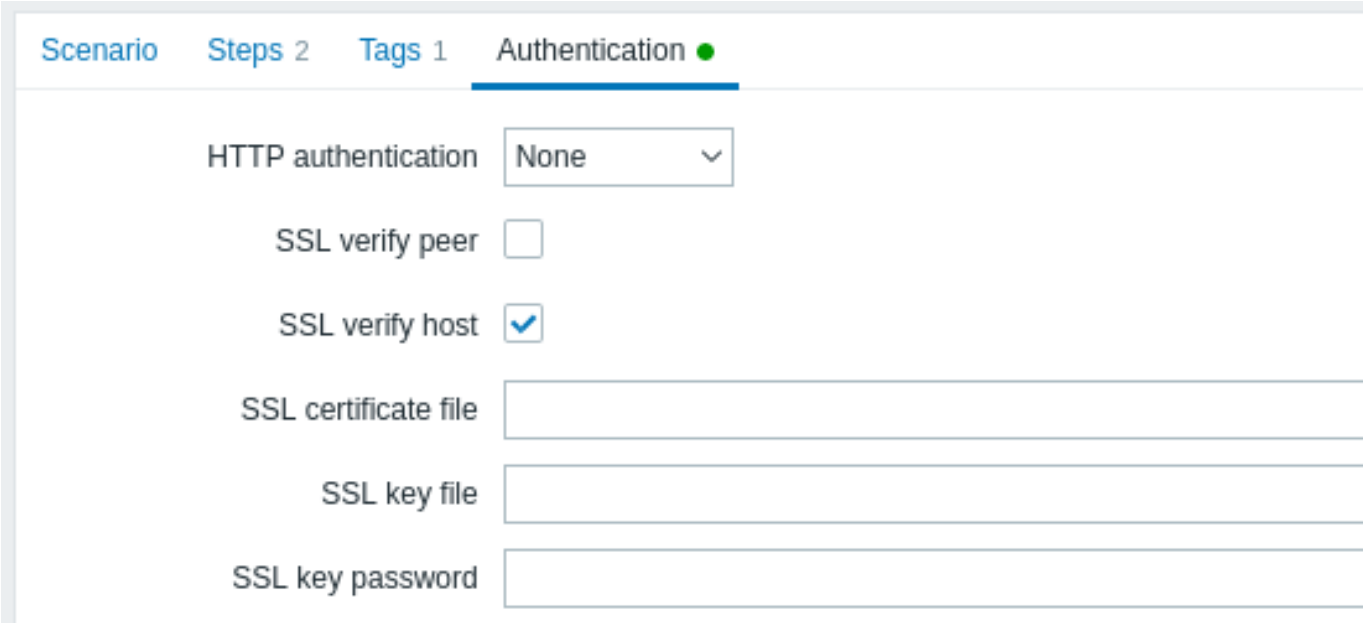
Siehe auch ein Praxisbeispiel, wie Schritte für web monitoring konfiguriert werden können.

Tags konfigurieren Auf der Registerkarte **Tags** können Tags auf Szenarioebene definiert werden.



Mithilfe von Tags können Webszenarien und Web-Monitoring-Datenpunkte gefiltert werden.

Konfigurieren der Authentifizierung Die Registerkarte **Authentifizierung** ermöglicht es Ihnen, die Authentifizierungsoptionen des Szenarios zu konfigurieren. Ein grüner Punkt neben dem Namen der Registerkarte zeigt an, dass eine Art von HTTP-Authentifizierung aktiviert ist.



Authentifizierungsparameter:

Parameter	Beschreibung
HTTP authentication	Wählen Sie die Authentifizierungsoption aus: **None** - keine Authentifizierung wird verwendet; **Basic** - Basisauthentifizierung wird verwendet; **NTLM** - NTLM-Authentifizierung (Windows NT LAN Manager) wird verwendet; **Kerberos** - Kerberos-Authentifizierung wird verwendet (siehe auch: Konfigurieren von Kerberos mit Zabbix); **Digest** - Digest-Authentifizierung wird verwendet.
User	Geben Sie den Benutzernamen ein (bis zu 255 Zeichen). Dieses Feld ist verfügbar, wenn *HTTP authentication* auf Basic, NTLM, Kerberos oder Digest gesetzt ist. Benutzermakros werden unterstützt.
Password	Geben Sie das Benutzerpasswort ein (bis zu 255 Zeichen). Dieses Feld ist verfügbar, wenn *HTTP authentication* auf Basic, NTLM, Kerberos oder Digest gesetzt ist. Benutzermakros werden unterstützt.
SSL verify peer	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des Webservers zu überprüfen. Das Serverzertifikat wird automatisch vom systemweiten Speicherort der Zertifizierungsstelle (CA) übernommen. Sie können den Speicherort der CA-Dateien mit dem Konfigurationsparameter SSLCALocation des Zabbix Server oder Proxy überschreiben. Dies setzt die cURL-Option CURLOPT_SSL_VERIFYPEER.
SSL verify host	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, dass das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserver-Zertifikats übereinstimmt. Dies setzt die cURL-Option CURLOPT_SSL_VERIFYHOST.
SSL certificate file	Name der SSL-Zertifikatsdatei, die für die Client-Authentifizierung verwendet wird. Die Zertifikatsdatei muss im PEM[1]-Format vorliegen. 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 Konfigurationsparameter SSLCertLocation des Zabbix Server oder Proxy angegeben. `HOST.*`-Makros und Benutzermakros können in diesem Feld verwendet werden. Dies setzt die cURL-Option CURLOPT_SSLCERT.
SSL key file	Name der SSL-Datei mit dem privaten Schlüssel, die für die Client-Authentifizierung verwendet wird. Die Datei mit dem privaten Schlüssel muss im PEM[1]-Format vorliegen. Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLKeyLocation des Zabbix Server oder Proxy angegeben. `HOST.*`-Makros und Benutzermakros können in diesem Feld verwendet werden. Dies setzt die cURL-Option CURLOPT_SSLKEY.
SSL key password	Passwort der SSL-Datei mit dem privaten Schlüssel. Benutzermakros können in diesem Feld verwendet werden. Dies setzt die cURL-Option CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix unterstützt Zertifikats- und private Schlüsseldateien nur im PEM-Format. Falls Ihre Zertifikats- und privaten Schlüsseldaten im Dateiformat PKCS #12 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
```

Note:
Zabbix Server übernimmt Änderungen an Zertifikaten ohne Neustart.

Note:
Wenn Sie Client-Zertifikat und privaten Schlüssel in einer einzigen Datei haben, geben Sie diese einfach im Feld "SSL certificate file" an und lassen Sie das Feld "SSL key file" leer. Zertifikat und Schlüssel müssen weiterhin im PEM-Format vorliegen. Das Kombinieren von Zertifikat und Schlüssel ist einfach:

```
cat client.crt client.key > client.pem
```

Anzeige Um für einen Host konfigurierte Webszenarien 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 Informationen zu erhalten.

Eine Übersicht der Webszenarien kann auch in *Dashboards* mit dem Widget „Web monitoring“ angezeigt werden.

Aktuelle Ergebnisse der Ausführung des Webszenarios sind im Abschnitt *Monitoring* → *Latest data* verfügbar.

Erweiterte Überwachung Manchmal ist es notwendig, den Inhalt empfangener HTML-Seiten zu protokollieren. Dies ist besonders nützlich, wenn ein Schritt in einem Webszenario fehlschlägt. Debug-Level 5 (Trace) dient genau diesem Zweck. Dieses Level kann in den Konfigurationsdateien von Server und Proxy festgelegt oder über eine Option zur Laufzeitsteuerung verwendet werden (`-R log_level_increase="http poller,N"`, wobei N die Prozessnummer ist). Die folgenden Beispiele zeigen, wie die erweiterte Überwachung gestartet werden kann, vorausgesetzt, Debug-Level 4 ist bereits gesetzt:

```
# Log-Level aller HTTP-Poller erhöhen:
zabbix_server -R log_level_increase="http poller"

# Log-Level des zweiten HTTP-Pollers erhöhen:
zabbix_server -R log_level_increase="http poller,2"
```

Wenn die erweiterte Webüberwachung nicht erforderlich ist, kann sie mit der Option `-R log_level_decrease` beendet werden.

1 Datenpunkte für Webüberwachung

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
```

2 Praxisbeispiel

Überblick

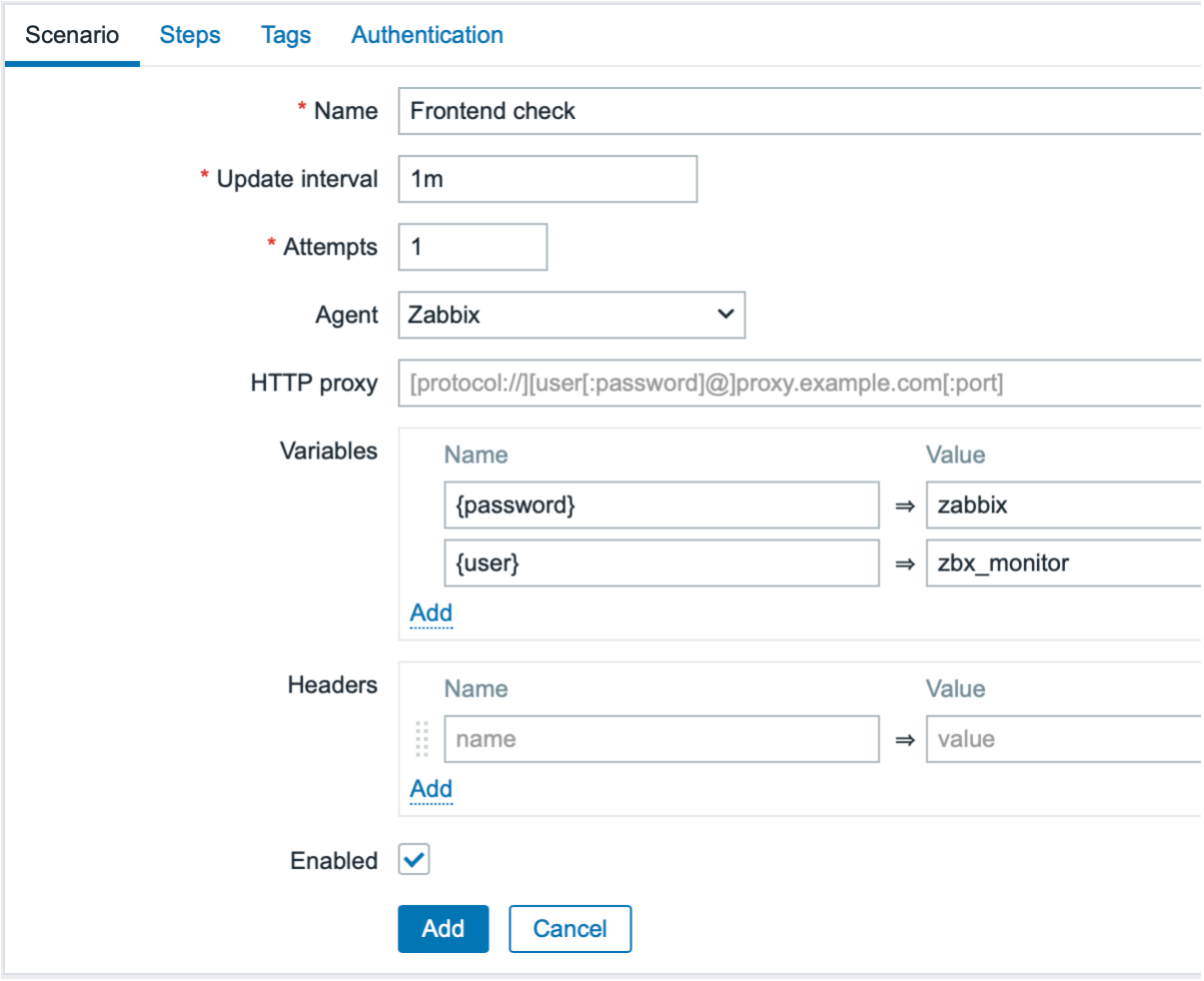
Dieser Abschnitt zeigt ein praxisnahes Schritt-für-Schritt-Beispiel dafür, wie Web-Monitoring eingesetzt werden kann.

Das Zabbix-Web-Monitoring wird verwendet, um das Zabbix Frontend zu überwachen. Ziel ist es festzustellen, ob es verfügbar ist, die richtigen Inhalte bereitstellt und wie schnell es arbeitet. Dazu sind mehrere Schritte erforderlich, darunter die Prüfung der Verfügbarkeit der ersten Seite, die Anmeldung mit Benutzername und Passwort, die Überprüfung des erfolgreichen Logins, das Abmelden und die Bestätigung der Abmeldung.

Szenario

Ein neues Webszenario hinzufügen

Gehen Sie zu *Datensammlung* → *Hosts*, wählen Sie einen Host aus und klicken Sie in der Zeile dieses Hosts auf *Web*.
Klicken Sie dann auf *Webszenario erstellen*.



Füllen Sie im Formular für das neue Szenario die folgenden Felder aus:

- **Name** - Frontend-Prüfung
- **Aktualisierungsintervall** - 1m
- **Versuche** - 1
- **Agent** - Zabbix

Fügen Sie im Abschnitt *Variablen* zwei Variablen hinzu: *{password}* und *{user}*.
Geben Sie als Werte Ihre vorhandenen Zabbix-Benutzerdaten ein.

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 Webszenario hinzufügen.

Note:
Sobald dieses Webszenario vollständig konfiguriert ist, wird dem Host automatisch ein Zabbix-Trapper-Datenpunkt hinzugefügt.
Sie können Tags des Webszenarios verwenden, um zugehörige Datenpunkte und Auslöser schnell zu identifizieren oder die gesammelten Daten zu durchsuchen.
Geeignete Tags für dieses Tutorial sind zum Beispiel `component: web-scenario` und/oder `target: frontend`.

Schritte des Webszenarios konfigurieren

Wechseln Sie zur Registerkarte *Schritte* und definieren Sie die Schritte für das Szenario. Klicken Sie auf die Schaltfläche *Hinzufügen*, um einen einzelnen Schritt hinzuzufügen.

Allgemeine Felder

Füllen Sie für jeden unten beschriebenen Schritt zusätzlich zu den schrittspezifischen Feldern die folgenden Felder aus:

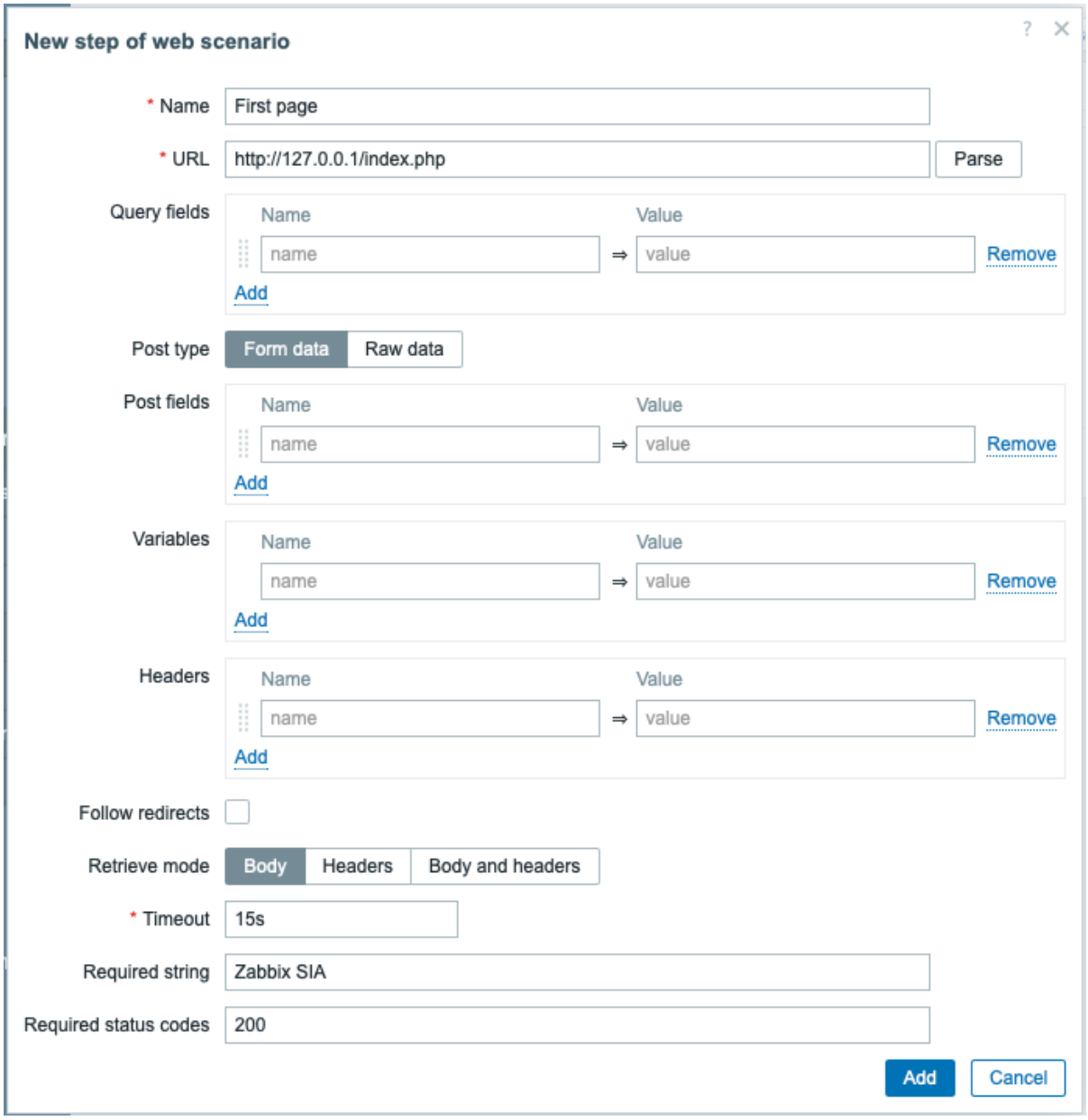
- **URL** - die URL des Zabbix Frontend
- **Timeout** - 15s
- **Erforderliche Statuscodes** - 200

Webszenario-Schritt 1

Prüfen Sie, dass die erste Seite korrekt antwortet, den HTTP-Antwortcode 200 zurückgibt und den Text „Zabbix SIA“ enthält.

- Geben Sie im Feld **Name** *First page* ein.
- Geben Sie im Feld **Required string** *Zabbix SIA* ein.
- Füllen Sie die gemeinsamen Felder aus.

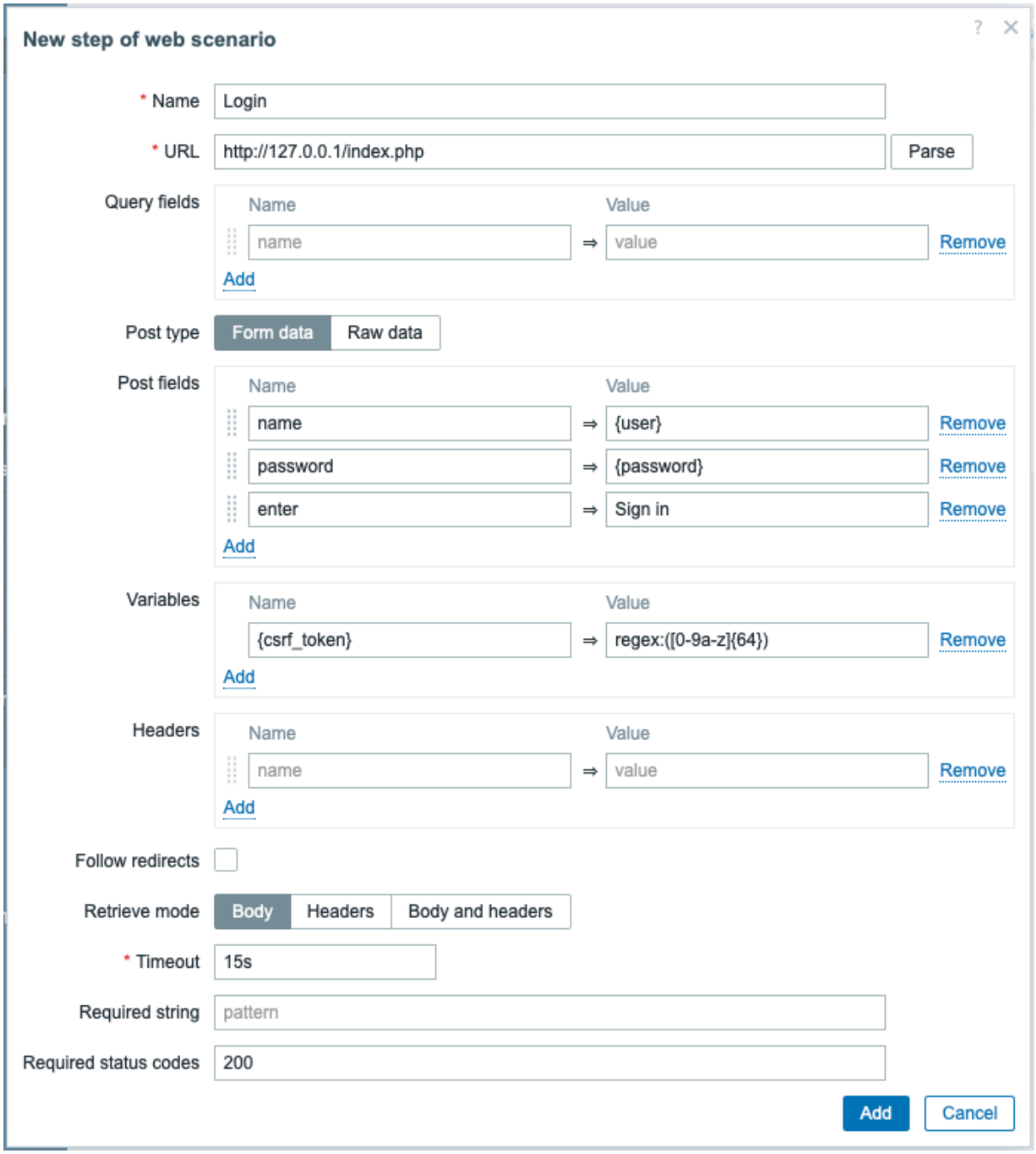
Wenn Sie die Konfiguration des Schritts abgeschlossen haben, klicken Sie auf die Schaltfläche *Add*.



Webszenario-Schritt 2

Melden Sie sich im Zabbix Frontend mit den auf Szenarioebene definierten Makros (Variablen) an – *{user}* und *{password}*.

- Geben Sie im Feld **Name** *Login* ein.
- Fügen Sie im Abschnitt **Post fields** drei Post-Felder hinzu:
  - *name* mit dem Wert *{user}*
  - *password* mit dem Wert *{password}*
  - *enter* mit dem Wert *Sign in*
- Fügen Sie im Abschnitt **Variables** eine neue Variable *{csrf_token}* mit dem Wert *regex:([0-9a-z]{64})* hinzu. Diese Variable erfasst den Wert des zugewiesenen CSRF-Tokens, um ihn in Schritt 4 erneut zu verwenden.
- Füllen Sie die gemeinsamen Felder aus.

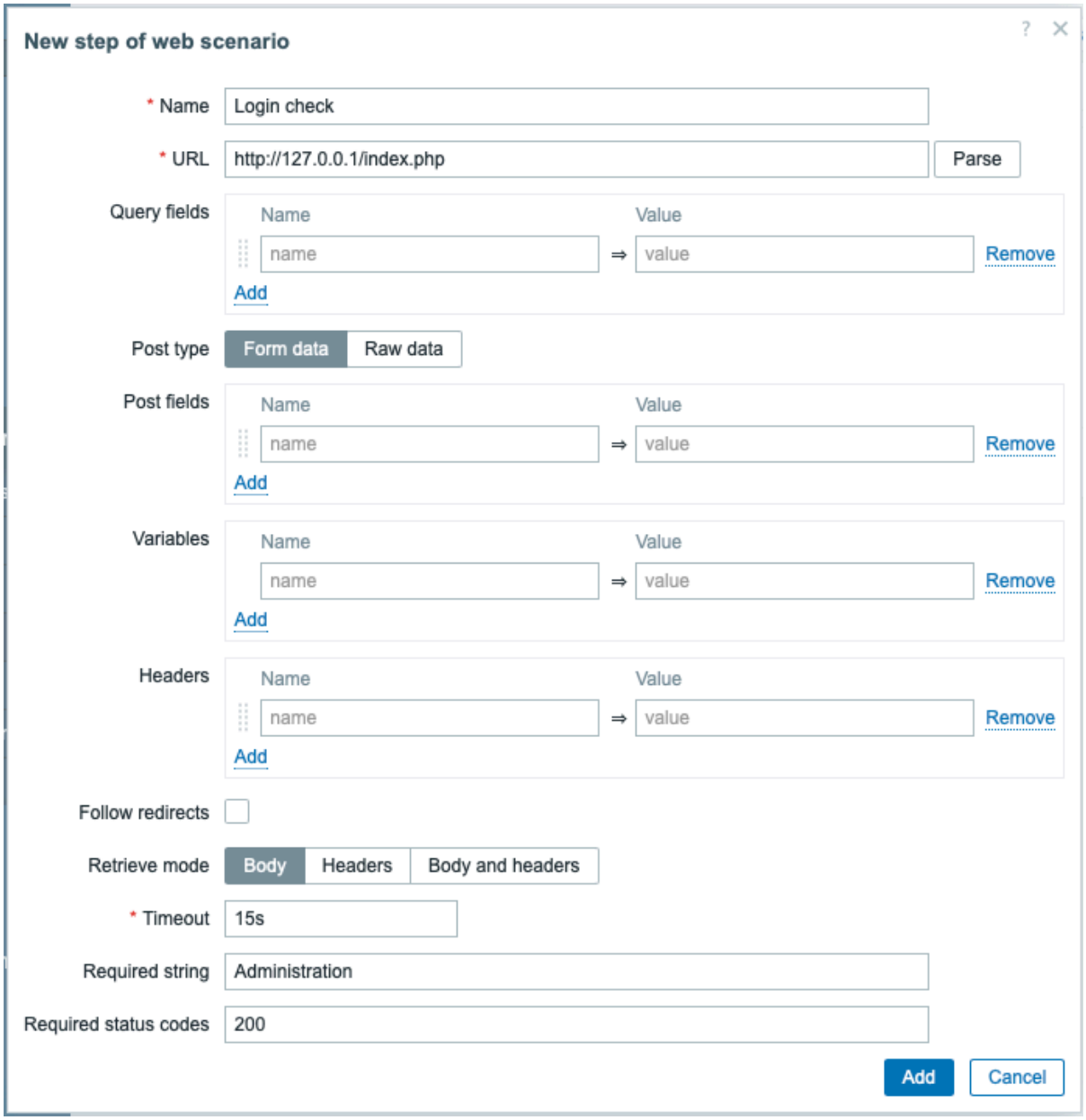


Attention:
Beachten Sie, dass das Zabbix Frontend bei der Anmeldung eine JavaScript-Weiterleitung verwendet. Daher muss die Anmeldung zuerst erfolgen, und Funktionen für angemeldete Benutzer können erst in weiteren Schritten geprüft werden. Außerdem muss der Anmeldeschritt die vollständige URL zur Datei **index.php** verwenden.

Schritt 3 des Webszenarios

Überprüfen Sie nach der Anmeldung den Erfolg, indem Sie nach einer Zeichenfolge suchen, die nur sichtbar ist, wenn man angemeldet ist – zum Beispiel *Administration*.

- Geben Sie im Feld **Name** *Login check* ein.
- Geben Sie im Feld **Required string** *Administration* ein.
- Füllen Sie die allgemeinen Felder aus.

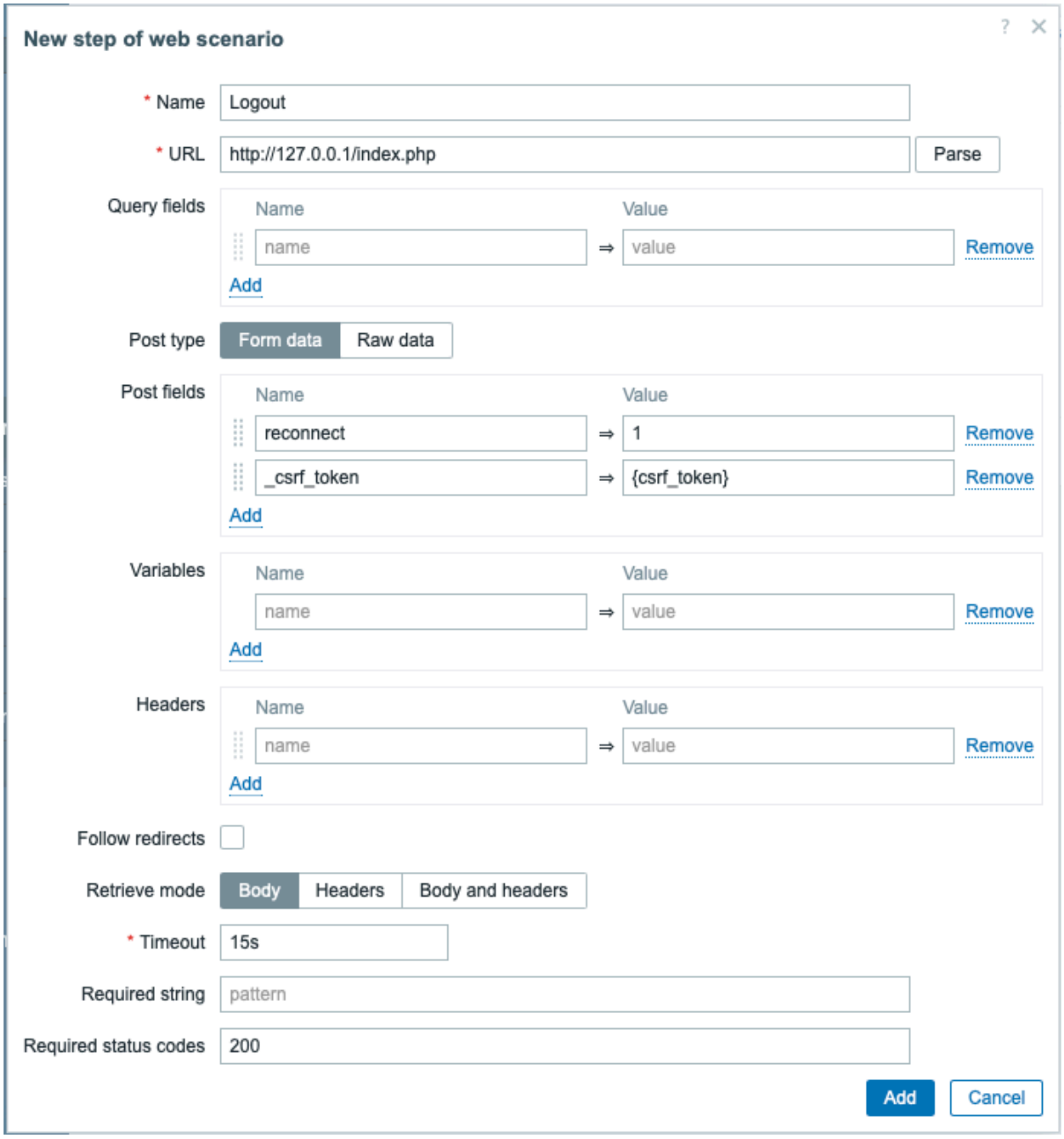


Webszenario-Schritt 4

Sobald die Erreichbarkeit und die Anmeldung des Frontends überprüft wurden, fügen Sie einen Abmeldeschritt hinzu – andernfalls wird die Zabbix-Datenbank mit vielen offenen Sitzungseinträgen überladen.

- Geben Sie im Feld **Name** *Logout* ein.
- Fügen Sie im Abschnitt **Post fields** zwei Post-Felder hinzu:
  - *reconnect* mit dem Wert *1*
  - *_csrf_token* mit dem Wert *{csrf_token}*.
- Füllen Sie die gemeinsamen Felder aus.

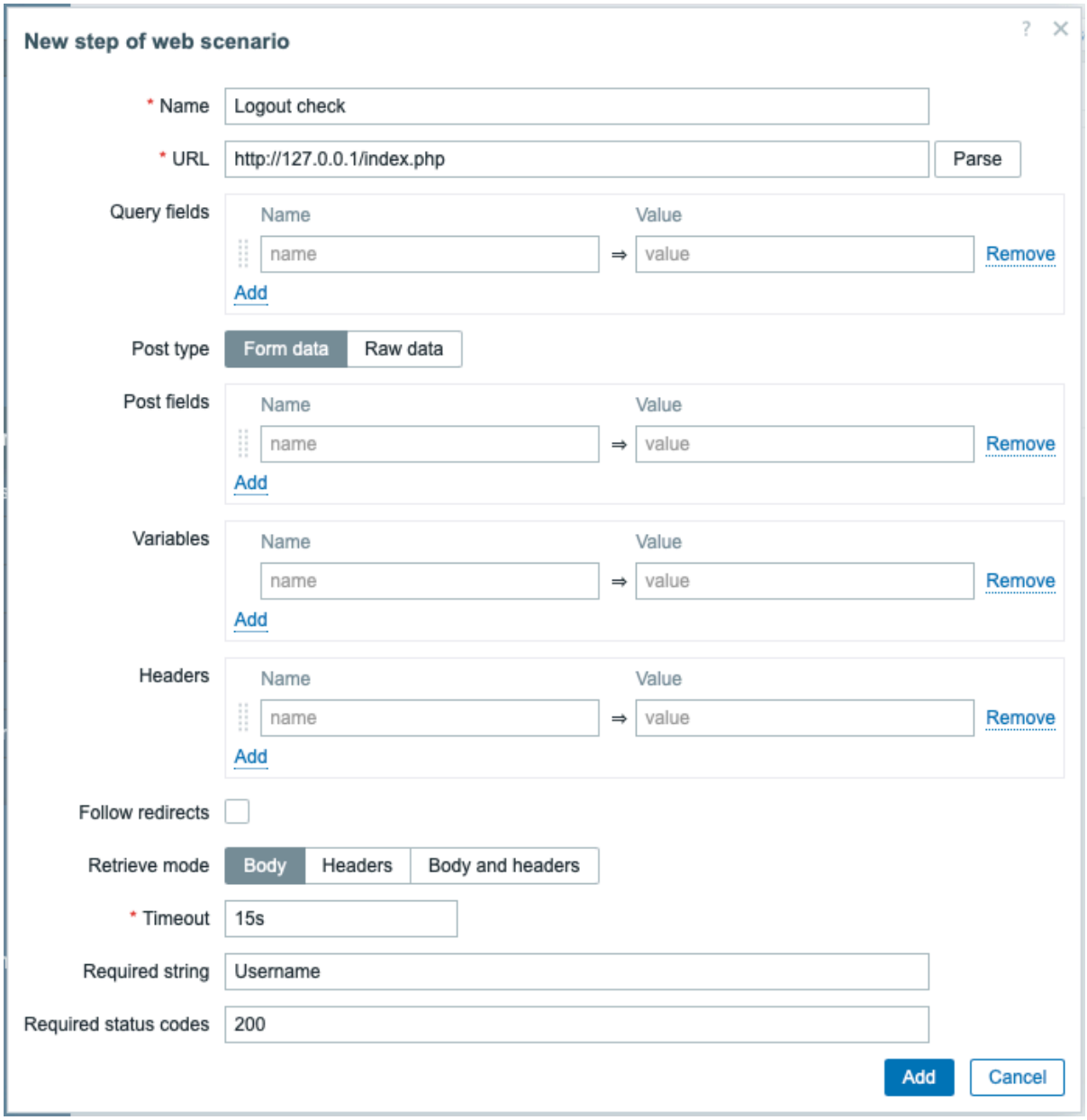
Dieser Schritt verwendet die Variable {csrf_token}, die in Schritt 2 abgerufen wurde.



Webszenario-Schritt 5

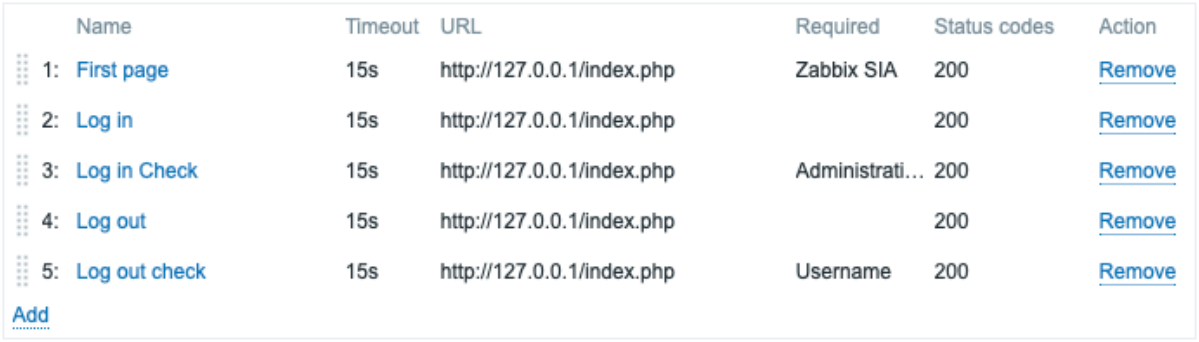
Um die Abmeldung zu bestätigen, prüfen Sie auf die Zeichenfolge **Username**.

- Geben Sie im Feld **Name** *Logout check* ein.
- Geben Sie im Feld **Required string** *Username* ein.
- Füllen Sie die gemeinsamen Felder aus.



Vollständige Konfiguration von Schritten

Eine vollständige Konfiguration von Schritten eines Webszenarios sollte wie folgt aussehen:



Ergebnisse prüfen

Speichern Sie das fertige Web-Monitoring-Szenario.

Das Szenario wird dem Host hinzugefügt. Um Informationen zum Web-Szenario 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 detailliertere Statistiken anzuzeigen:

8 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 aktivierten Optionen kompiliert.

Die folgenden Parameter der Zabbix-Server-Konfigurationsdatei können für die Überwachung virtueller Maschinen geändert werden:

- StartVMwareCollectors

Note:
Es wird empfohlen, mehr Collector-Prozesse zu aktivieren als VMware-Services überwacht werden; andernfalls kann das Abrufen von VMware-Performance-Counter-Statistiken durch das Abrufen von VMware-Konfigurationsdaten verzögert werden (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 überwachter VMware-Services betragen: Anzahl der Services < `StartVMwareCollectors` < (Anzahl der Services * 2). Wenn beispielsweise ein VMware-Service überwacht wird, setzen Sie `StartVMwareCollectors` auf 2; bei der Überwachung von drei Services setzen Sie `StartVMwareCollectors` auf 5.

 Beachten Sie, dass die erforderliche Anzahl an Collector-Prozessen auch vom Umfang der VMware-Umgebung sowie von den Konfigurationsparametern `VMwareFrequency` und `VMwarePerfFrequency` abhängt.

- VMwareCacheSize
- VMwareFrequency
- VMwarePerfFrequency
- VMwareTimeout

Attention:
Um Metriken zur Datastore-Kapazität 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 *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 erkennen. Außerdem kann Zabbix Host-Prototypen verwenden, um automatisch echte Hosts für die erkannten Entitäten zu erzeugen. Weitere Informationen finden Sie unter *Host prototypes*.

Configuration examples

For a basic example of how to set up Zabbix for monitoring VMware using the *VMware FQDN* template, see *Monitor VMware with Zabbix*.

For a more detailed example of how to create a host, a low-level discovery rule, and a host prototype for monitoring VMware, see *Setup example*.

Erweiterte Protokollierung Die von den Prozessen des *vmware collector* gesammelten 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 Prozesse des *vmware collector* 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 Prozess des *vmware collector* 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 von VMware-Collector-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 aktuellen VMware vSphere-Versionen nicht standardmäßig als nicht verfügbar markiert oder deaktiviert sind und dass keine Beschränkungen für Abfragen an die Performance-Metrik-Datenbank gesetzt sind. Weitere Informationen finden Sie unter ZBX-12094.
- Wenn der Fehler `config.vpxd.stats.maxQueryMetrics is invalid or exceeds the maximum number of characters permitted` auftritt, fügen Sie den Parameter `config.vpxd.stats.maxQueryMetrics` zu den Einstellungen des vCenter Server hinzu. Der Wert dieses Parameters sollte mit dem Wert von `maxQuerysize` in der VMware-Datei *web.xml* 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.

1 VMware-Monitoring-Datenpunktschlüssel

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 sind 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
vmware.fullname	Der vollständige Name des VMware-Dienstes.	
vmware.version	Die Version des VMware-Dienstes.	
vmware.cl.perfcounter	Die Metriken der VMware-Cluster-Performance-Counter.	Cluster
vmware.cluster.alarms.get	Die Daten der VMware-Cluster-Alarme.	
vmware.cluster.discovery	Die Erkennung von VMware-Clustern.	
vmware.cluster.property	Die VMware-Cluster-Eigenschaft.	
vmware.cluster.status	Der VMware-Cluster-Status.	
vmware.cluster.tags.get	Das Array der VMware-Cluster-Tags.	
vmware.datastore.alarms.get	Die Daten der VMware-Datastore-Alarme.	Datastore
vmware.datastore.discovery	Die Erkennung von VMware-Datastores.	
vmware.datastore.hv.list	Die Liste der Datastore-Hypervisoren.	
vmware.datastore.perfcounter	Der Wert des VMware-Datastore-Performance-Counters.	
vmware.datastore.property	Die VMware-Datastore-Eigenschaft.	
vmware.datastore.read	Die Zeitdauer für einen Lesevorgang aus dem Datastore.	
vmware.datastore.size	Der VMware-Datastore-Speicherplatz in Byte oder als Prozentsatz der Gesamtkapazität.	
vmware.datastore.tags.get	Das Array der VMware-Datastore-Tags.	
vmware.datastore.write	Die Zeitdauer für einen Schreibvorgang in den Datastore.	
vmware.dc.alarms.get	Die Daten der VMware-Datacenter-Alarme.	Datacenter
vmware.dc.discovery	Die Erkennung von VMware-Datacentern.	
vmware.dc.tags.get	Das Array der VMware-Datacenter-Tags.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
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 Daten der VMware-Hypervisor-Alarme.	Hypervisor
vmware.hv.cluster.name	Der Name des VMware-Hypervisor-Clusters.	
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 Energieverwaltung oder HT.	
vmware.hv.datacenter.name	Der Name des VMware-Hypervisor-Datacenters.	
vmware.hv.datastore.discovery	Die Erkennung von VMware-Hypervisor-Datastores.	
vmware.hv.datastore.list	Die Liste der VMware-Hypervisor-Datastores.	
vmware.hv.datastore.multipath	Die Anzahl der verfügbaren Datastore-Pfade.	
vmware.hv.datastore.read	Die durchschnittliche Zeitdauer für einen Lesevorgang aus dem Datastore.	
vmware.hv.datastore.size	Der VMware-Datastore-Speicherplatz in Byte oder als Prozentsatz der Gesamtkapazität.	
vmware.hv.datastore.write	Die durchschnittliche Zeitdauer für einen Schreibvorgang in den Datastore.	
vmware.hv.discovery	Die Erkennung von VMware-Hypervisoren.	
vmware.hv.diskinfo.get	Die VMware-Hypervisor-Festplattendaten.	
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 Prozessorthreads auf dem VMware-Hypervisor.	
vmware.hv.hw.memory	Die gesamte Speichergröße des VMware-Hypervisors.	
vmware.hv.hw.model	Das Modell des VMware-Hypervisors.	
vmware.hv.hw.sensors.get	Der Wert der Hardware-Sensoren 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 Größe des aufgeblähten Speichers des VMware-Hypervisors.	
vmware.hv.memory.used	Die Größe des verwendeten Speichers des VMware-Hypervisors.	
vmware.hv.net.if.discovery	Die Erkennung von VMware-Hypervisor-Netzwerkschnittstellen.	
vmware.hv.network.in	Die Eingangsstatistiken des VMware-Hypervisor-Netzwerks.	
vmware.hv.network.linkspeed	Die Geschwindigkeit der VMware-Hypervisor-Netzwerkschnittstelle.	
vmware.hv.network.out	Die Ausgangsstatistiken des VMware-Hypervisor-Netzwerks.	
vmware.hv.perfcounter	Der Wert des VMware-Hypervisor-Performance-Counters.	
vmware.hv.property	Die VMware-Hypervisor-Eigenschaft.	
vmware.hv.power	Der Stromverbrauch des VMware-Hypervisors.	
vmware.hv.sensor.health.state	Der zusammengefasste Sensor für den Integritätsstatus des VMware-Hypervisors.	
vmware.hv.sensors.get	Die Sensoren für den Hardware-Herstellerstatus des VMware-Hypervisors.	
vmware.hv.status	Der Status des VMware-Hypervisors.	
vmware.hv.tags.get	Das Array der VMware-Hypervisor-Tags.	
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.	Resource Pool
vmware.rp.memory	Die Speichermetriken des VMware Resource Pool.	
vmware.alarms.get	Die Daten der VMware-Virtual-Center-Alarme.	Virtual Center
vmware.vm.alarms.get	Die Daten der VMware-Alarme virtueller Maschinen.	Virtuelle Maschine
vmware.vm.attribute	Der Wert des 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 erfordert eine Konsolidierung.	
vmware.vm.cpu.latency	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.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
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 das Einlagern aus dem Swap-Bereich 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 Datacenters der VMware-virtuellen Maschine.	
vmware.vm.discovery	Die Erkennung von VMware-virtuellen Maschinen.	
vmware.vm.guest.memory.swapped	Die Menge des physischen Gast-Speichers, die in den Swap-Bereich ausgelagert wurde.	
vmware.vm.guest.osuptime	Die gesamte seit dem letzten Start des Betriebssystems verstrichene Zeit.	
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 Speichergröße der VMware-virtuellen Maschine.	
vmware.vm.memory.size.ballooned	Die Größe des aufgeblähten Speichers der VMware-virtuellen Maschine.	
vmware.vm.memory.size.compressed	Die Größe des komprimierten Speichers der VMware-virtuellen Maschine.	
vmware.vm.memory.size.consumed	Die Menge des physischen Host-Speichers, die zur Sicherung der physischen Gast-Speicherseiten verbraucht wird.	
vmware.vm.memory.size.private	Die Größe des privaten Speichers der VMware-virtuellen Maschine.	
vmware.vm.memory.size.shared	Die Größe des gemeinsam genutzten Speichers der VMware-virtuellen Maschine.	
vmware.vm.memory.size.swapped	Die Größe des ausgelagerten Speichers der VMware-virtuellen Maschine.	
vmware.vm.memory.size.usage.guest	Die Gast-Speicherauslastung der VMware-virtuellen Maschine.	
vmware.vm.memory.size.usage.host	Die Host-Speicherauslastung 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 VMware-Netzwerkschnittstellen virtueller Maschinen.	
vmware.vm.net.if.in	Die Eingangsstatistiken der Netzwerkschnittstelle der VMware-virtuellen Maschine.	
vmware.vm.net.if.out	Die Ausgangsstatistiken 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 VMware-Performance-Counters der virtuellen Maschine.	
vmware.vm.powerstate	Der Energiezustand 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 Status der VMware-virtuellen Maschine.	
vmware.vm.storage.committed	Der belegte Speicherplatz der VMware-virtuellen Maschine.	
vmware.vm.storage.readoio	Die durchschnittliche Anzahl ausstehender Leseanforderungen an die virtuelle Festplatte während des Erfassungsintervalls.	
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 belegte 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 Array der Tags 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 VMware-Festplattengeräten virtueller Maschinen.	
vmware.vm.vfs.dev.read	Die Lesestatistiken der Festplattengeräte der VMware-virtuellen Maschine.	
vmware.vm.vfs.dev.write	Die Schreibstatistiken der Festplattengeräte der VMware-virtuellen Maschine.	
vmware.vm.vfs.fs.discovery	Die Erkennung von VMware-Dateisystemen virtueller Maschinen.	
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* - die 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:

- Es darf pro URL 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 Daten zu den VMware-Cluster-Alarmen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-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 VMware-Cluster-Tags-Array.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **id** - die VMware-Cluster-ID.

Dieser Datenpunkt funktioniert mit vSphere 6.5 und neuer.

vmware.datastore.alarms.get[url,uuid]

 Die VMware-Daten zu Datastore-Alarmen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung 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 VMware-Datastore-Tags-Array.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung 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 Daten zu den VMware-Datacenter-Alarmen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-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 VMware-Datacenter-Tag-Array.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-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 Daten der Ports des VMware vSphere Distributed Switch.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware DVSwitch.
- **filter** - eine einzelne Zeichenfolge mit durch Kommas getrennten Kriterien zur Auswahl von Ports;
- **mode** - *state* (gesamtes XML ohne XML-Knoten „config“, 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 VMware-Hypervisor-Alarmdaten.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global 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 Festplattendaten des VMware-Hypervisors.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes;

- **uuid** - die global 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 URL des VMware-Dienstes;
- **uuid** - die global 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 Sensoren für den Status des Hardware-Herstellers des VMware-Hypervisors.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global 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 VMware-Hypervisor-Tags-Array.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung 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 virtuellen VMware-Centers.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Service.

vmware.vm.alarms.get[url,uuid]

 Die Alarmdaten der virtuellen VMware-Maschine.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-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 virtuellen VMware-Maschine.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-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-Tags der virtuellen Maschine.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-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 Erstellen benutzerdefinierter Leistungsindikatornamen für VMware.

[2] Der Wert dieser Datenpunkte wird aus VMware-Leistungsindikatoren abgerufen, und der 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.

2 Schlüsselfelder für die Erkennung virtueller Maschinen

Die folgende Tabelle listet Felder auf, die von Erkennungsschlüsseln im Zusammenhang mit virtuellen Maschinen zurückgegeben werden.

Datenpunktschlüssel		
Beschreibung	**Feld**	**Abgerufener Inhalt**
vmware.cluster.discovery		
Führt eine Cluster-Erkennung durch.	{#CLUSTER.ID}	Cluster-Kennung.
	{#CLUSTER.NAME}	Cluster-Name.

Datenpunktschlüssel		

	”resource_pool”	Ein Array mit Daten zum Ressourcen-Pool, einschließlich Ressourcen-Gruppen-ID, Tags-Array, Ressourcen-Pool-Pfad und Anzahl virtueller Maschinen.
		
		Array-Struktur:
		`[{`
		`"rpid":"Ressourcen-Gruppen-ID",`
		`"tags":[{}],`
		`"rpath":"Ressourcen-Gruppen-Pfad",`
		`"vm_count":0`
		`}]`
		
		Zur Array-Struktur von `"tags"` siehe das Feld ”tags”.

	”tags”	Ein Array mit Tags mit Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"Tag-Name",`
		`"tag_description":"Tag-Beschreibung",`
		`"category":"Tag-Kategorie"`
		`}]`

vmware.datastore.discovery		
Führt eine Datastore-Erkennung durch.	{#DATASTORE}	Datastore-Name.

	{#DATASTORE.EXTENT}	Ein Array mit der Partitions-ID und dem Instanznamen des Datastore-Extents.
		
		Array-Struktur:
		`[{`
		`"partitionid":1,`
		`"instance":"Name"`
		`}]`

	{#DATASTORE.TYPE}	Datastore-Typ. Wertbeispiele: `VMFS`, `NFS`, `vsan` usw.
	{#DATASTORE.UUID}	Datastore-Kennung.

	”tags”	Ein Array mit Tags mit Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"Tag-Name",`
		`"tag_description":"Tag-Beschreibung",`
		`"category":"Tag-Kategorie"`
		`}]`

vmware.dc.discovery		
Führt eine Datacenter-Erkennung durch.	{#DATACENTER}	Datacenter-Name.
	{#DATACENTERID}	Datacenter-Kennung.

	”tags”	Ein Array mit Tags mit Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"Tag-Name",`
		`"tag_description":"Tag-Beschreibung",`
		`"category":"Tag-Kategorie"`
		`}]`

vmware.dvswitch.discovery		

Datenpunktschlüssel		
Führt eine Erkennung verteilter vSphere-Switches durch.	{#DVS.NAME}	Switch-Name.
	{#DVS.UUID}	Switch-Kennung.
vmware.hv.discovery		
Führt eine Hypervisor-Erkennung durch.	{#HV.UUID}	Eindeutige Hypervisor-Kennung.
	{#HV.ID}	Hypervisor-Kennung (Name des verwalteten Objekts HostSystem).
	{#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 Auswahlpriorität für die Schnittstelle beachtet: - bevorzugt wird die IP, die sich das IP-Subnetz mit der vCenter-IP teilt; - bevorzugt wird die IP aus dem IP-Subnetz mit Standard-Gateway; - bevorzugt wird die IP der Schnittstelle mit der niedrigsten ID.
	{#CLUSTER.NAME}	Cluster-Name, kann leer sein.
	{#DATACENTER.NAME}	Datacenter-Name.
	{#PARENT.NAME}	Name des Containers, der den Hypervisor enthält.
	{#PARENT.TYPE}	Typ des Containers, in dem der Hypervisor gespeichert ist. Die Werte können `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware` sein, wobei ”VMware” für einen unbekannten Container-Typ steht.

	”resource_pool”	Ein Array mit Daten zum Ressourcen-Pool, einschließlich Ressourcen-Gruppen-ID, Tags-Array, Ressourcen-Pool-Pfad und Anzahl virtueller Maschinen.
		
		Array-Struktur:
		`[{`
		`"rpid":"Ressourcen-Gruppen-ID",`
		`"tags":[{}],`
		`"rpath":"Ressourcen-Gruppen-Pfad",`
		`"vm_count":0`
		`}]`

	”tags”	Zur Array-Struktur von `"tags"` siehe das Feld ”tags”.
		Ein Array mit Tags mit Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"Tag-Name",`
		`"tag_description":"Tag-Beschreibung",`
		`"category":"Tag-Kategorie"`
		`}]`

vmware.hv.datastore.discovery		
Führt eine Hypervisor-Datastore-Erkennung durch. Beachten Sie, dass mehrere Hypervisoren denselben Datastore verwenden können.	{#DATASTORE}	Datastore-Name.
	{#DATASTORE.TYPE}	Datastore-Typ. Wertbeispiele: `VMFS`, `NFS`, `vsan` usw.
	{#DATASTORE.UUID}	Datastore-Kennung.
	{#MULTIPATH.COUNT}	Registrierte Anzahl von Datastore-Pfaden.
	{#MULTIPATH.PARTITION.COUNT}	Anzahl verfügbarer Festplattenpartitionen.

Datenpunktschlüssel		

	”datastore_extent”	Ein Array mit dem Instanznamen und der Partitions-ID des Datastore-Extents.
		
		Array-Struktur:
		`[{`
		`"partitionid":1,`
		`"instance":"Name"`
		`}]`

	”tags”	Ein Array mit Tags mit Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"Tag-Name",`
		`"tag_description":"Tag-Beschreibung",`
		`"category":"Tag-Kategorie"`
		`}]`

vmware.hv.net.if.discovery		
Führt eine Erkennung von Hypervisor-Netzwerkschnittstellen durch.	{#IFNAME}	Schnittstellenname.
	{#IFDRIVER}	Schnittstellentreiber.
	{#IFDUPLEX}	Duplex-Einstellungen der Schnittstelle.
	{#IFSPEED}	Schnittstellengeschwindigkeit.
	{#IFMAC}	MAC-Adresse der Schnittstelle.
vmware.vm.discovery		
Führt eine Erkennung virtueller Maschinen durch.	{#VM.UUID}	Eindeutige Kennung der virtuellen Maschine.
	{#VM.ID}	Kennung der virtuellen Maschine (Name des verwalteten Objekts VirtualMachine).
	{#VM.NAME}	Name der virtuellen Maschine.
	{#HV.NAME}	Hypervisor-Name.
	{#HV.UUID}	Eindeutige Hypervisor-Kennung.
	{#HV.ID}	Hypervisor-Kennung (Name des verwalteten Objekts HostSystem).
	{#CLUSTER.NAME}	Cluster-Name, kann leer sein.
	{#DATACENTER.NAME}	Datacenter-Name.
	{#DATASTORE.NAME}	Datastore-Name.
	{#DATASTORE.UUID}	Datastore-Kennung.
	{#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 OS-Name des Gastbetriebssystems der virtuellen Maschine, kann leer sein.
	{#VM.FOLDER}	Die Kette der übergeordneten Ordner der virtuellen Maschine; kann als Wert für verschachtelte Gruppen verwendet werden. Ordnernamen werden mit ”/” kombiniert. 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}	Kennung des Ressourcen-Pools.
	{#VM.RPOOL.PATH}	Vollständiger Pfad des Ressourcen-Pools ohne den ”root”-Namen ”Resources”. Ordnernamen werden mit ”/” kombiniert.
	{#VM.SNAPSHOT.COUNT}	Anzahl der VM-Snapshots.

Datenpunktschlüssel		

	”tags”	Ein Array mit Tags mit Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"Tag-Name",`
		`"tag_description":"Tag-Beschreibung",`
		`"category":"Tag-Kategorie"`
		`}]`

	”vm_customattribute”	Ein Array benutzerdefinierter Attribute der virtuellen Maschine (falls definiert).
		
		Array-Struktur:
		`[{`
		`"name":"Name des benutzerdefinierten Felds",`
		`"value":"Wert des benutzerdefinierten Felds"`
		`}]`

	”net_if”	Ein Array von Netzwerkschnittstellen der virtuellen Maschine.
		
		Array-Struktur:
		`[{`
		`"ifname": "Schnittstellenname",`
		`"ifdesc": "Schnittstellenbeschreibung",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "Schnittstellentyp",`
		`"ifbackingdevice": "Hinterlegtes Gerät der Schnittstelle",`
		`"ifdvswitch_uuid": "UUID des Schnittstellen-Switches",`
		`"ifdvswitch_portgroup": "Portgruppe des Schnittstellen-Switches",`
		`"ifdvswitch_port": "Port des Schnittstellen-Switches",`
		`"ifip": ["IP-Adressen der Schnittstelle"]`
		`}]`
		
		Zur Beschreibung der zurückgegebenen Daten siehe den Datenpunktschlüssel ”vmware.vm.net.if.discovery”.

vmware.vm.net.if.discovery		
Führt eine Erkennung von Netzwerkschnittstellen virtueller Maschinen durch.	{#IFNAME}	Name der Netzwerkschnittstelle.
	{#IFDESC}	Schnittstellenbeschreibung.
	{#IFMAC}	MAC-Adresse der Schnittstelle.
	{#IFCONNECTED}	Verbindungsstatus der Schnittstelle (`false` - getrennt; `true` - verbunden).
	{#IFTYPE}	Schnittstellentyp.
	{#IFBACKINGDEVICE}	Name des hinterlegten Geräts.
	{#IFDVSWITCH.UUID}	Eindeutige Kennung des vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Verteilte Portgruppe.
	{#IFDVSWITCH.PORT}	Port des vSphere Distributed Switch.
	”ifip”	Ein Array von Schnittstellenadressen.
vmware.vm.vfs.dev.discovery		
Führt eine Erkennung von Festplattengeräten virtueller Maschinen durch.	{#DISKNAME}	Name des Festplattengeräts.
vmware.vm.vfs.fs.discovery		

Datenpunktschlüssel		
Führt eine Dateisystem-Erkennung virtueller Maschinen durch.	{#FSNAME}	Dateisystemname.

3 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 dienen als Beispiel):

```
{
    "alarms": [
        {
            "name": "Host-Verbindungs- und Energiezustand",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Standardalarm zur Überwachung des Host-Verbindungs- und Energiezustands",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false
        },
        {
            "name": "Host-Speicherauslastung",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Standardalarm zur Überwachung der Host-Speicherauslastung",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false
        },
        {
            // andere Alarme
        }
    ]
}
```

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
}
```


4 Beispiel für die Einrichtung der VMware-Überwachung

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.

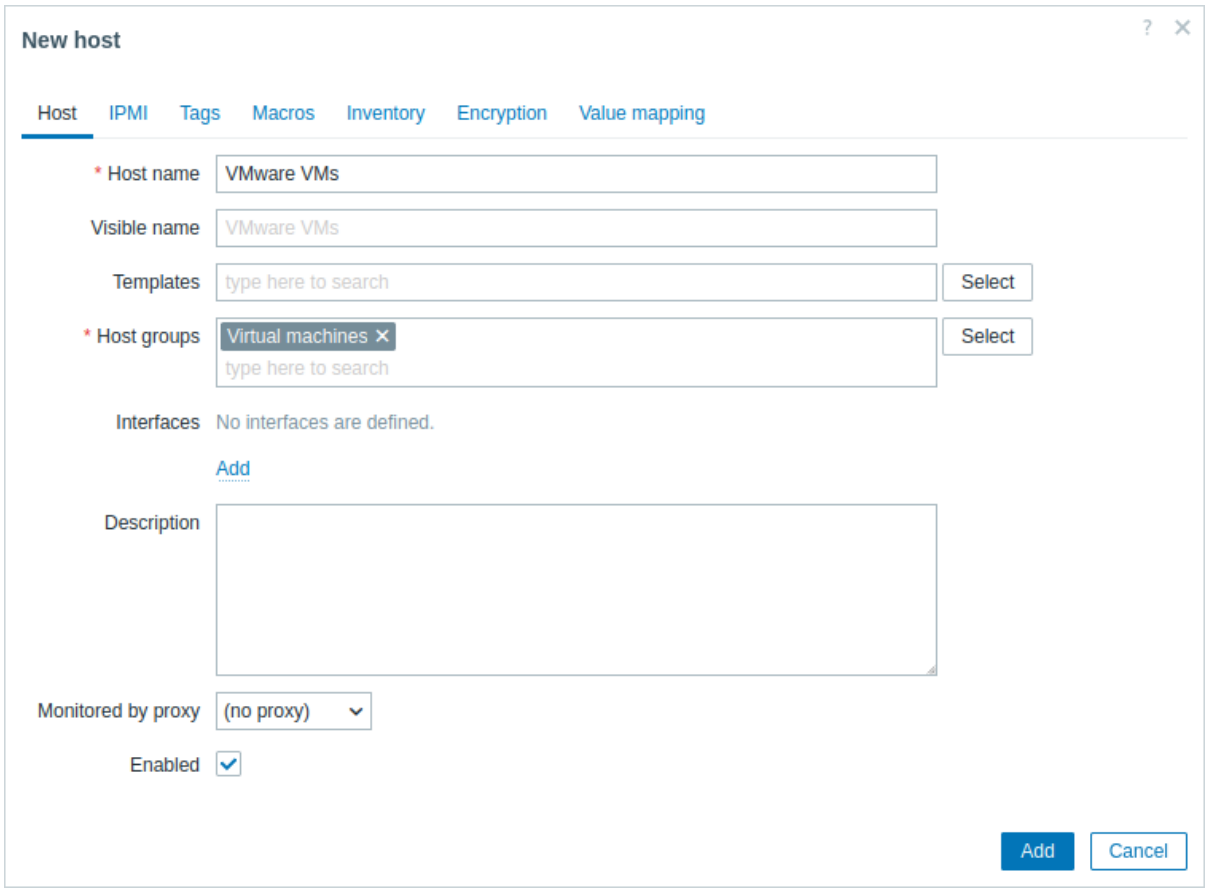
Bevor Sie fortfahren, setzen Sie den Parameter `StartVMwareCollectors` in der Konfigurationsdatei des Zabbix Server auf 2 oder höher (der Standardwert ist 0).

Einen Host erstellen

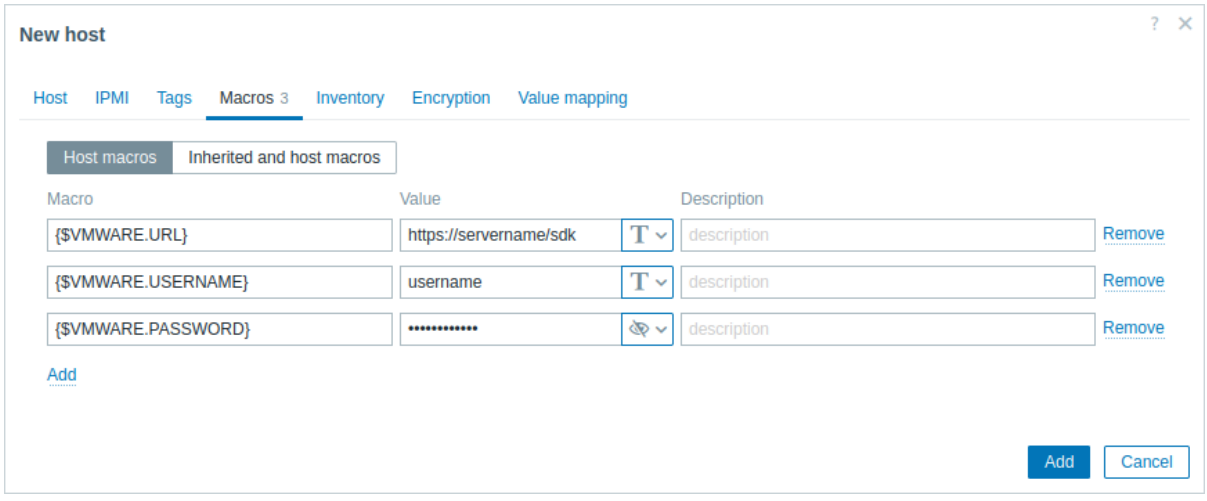
1. Gehen Sie zu *Datenerfassung* → *Hosts*.

2. Erstellen Sie einen Host:

- Geben Sie im Feld *Host name* einen Host-Namen ein (zum Beispiel „VMware VMs“).
- Geben Sie im Feld *Host groups* eine Host-Gruppe 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
  - {$VMWARE.PASSWORD} - Passwort des VMware-Dienstbenutzers {$VMWARE.USERNAME}



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

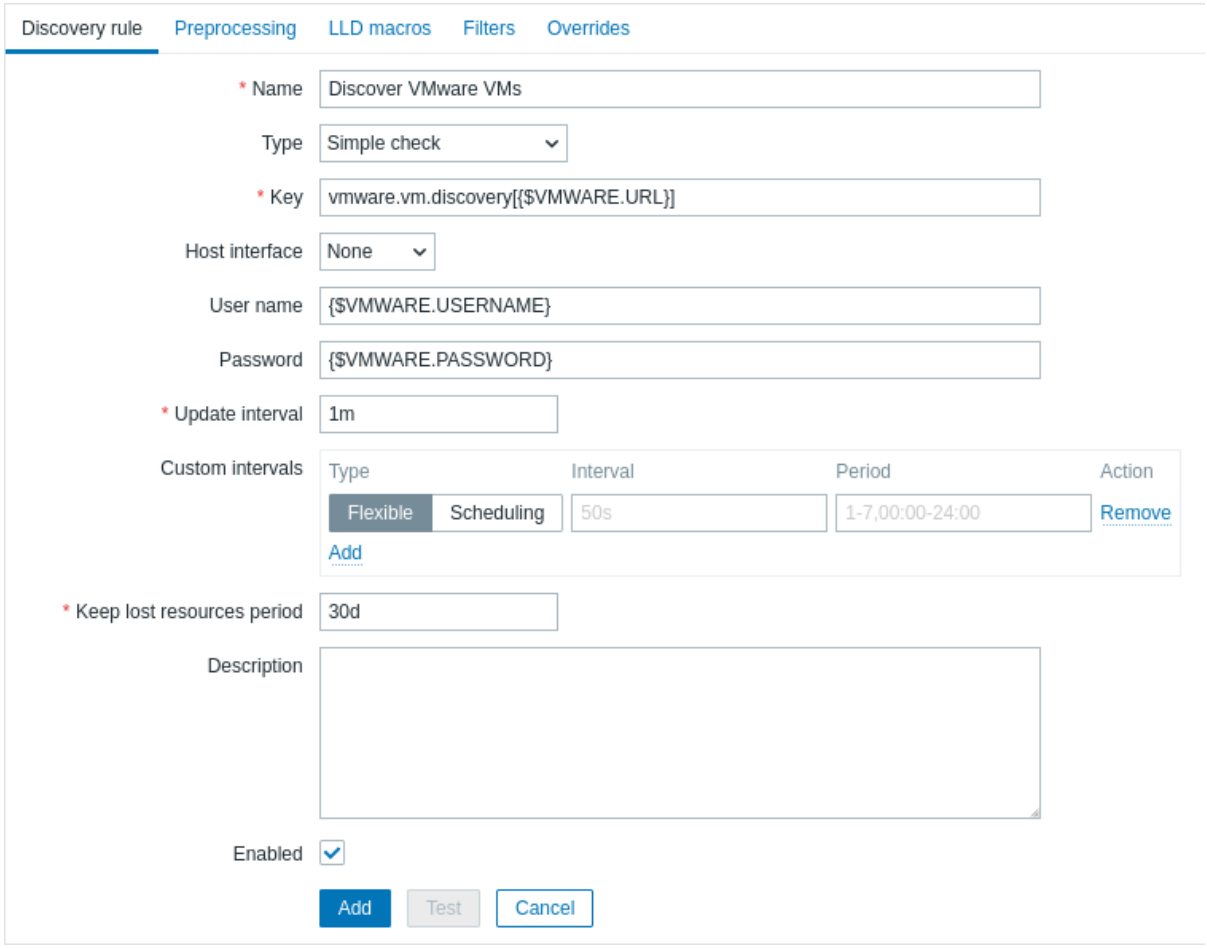
Eine Low-Level-Discovery-Regel erstellen

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 entdecken“).
- 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.



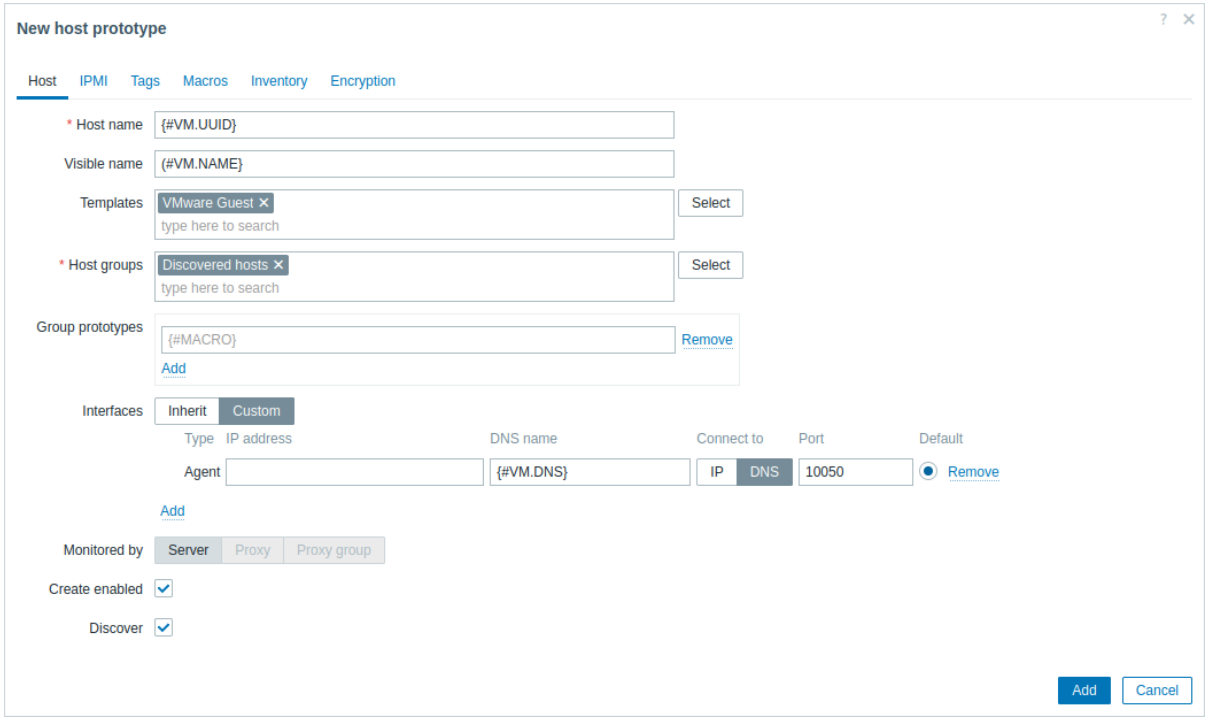
3. Klicken Sie auf die Schaltfläche *Add*, um die Low-Level-Discovery-Regel zu erstellen. Diese Discovery-Regel erkennt virtuelle Maschinen in Ihrer VMware-Umgebung.

Einen Host-Prototyp erstellen

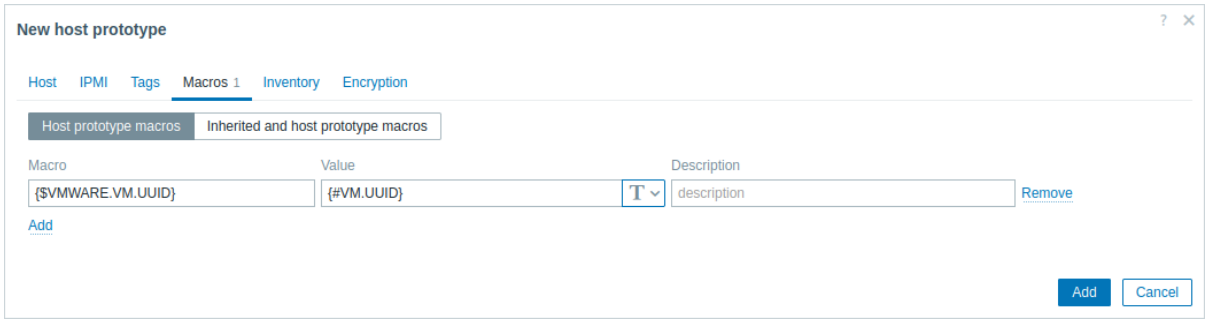
1. Klicken Sie in der Liste der Low-Level-Discovery-Regeln bei der zuvor erstellten Low-Level-Discovery-Regel auf *Host-Prototypen*.

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}]*).

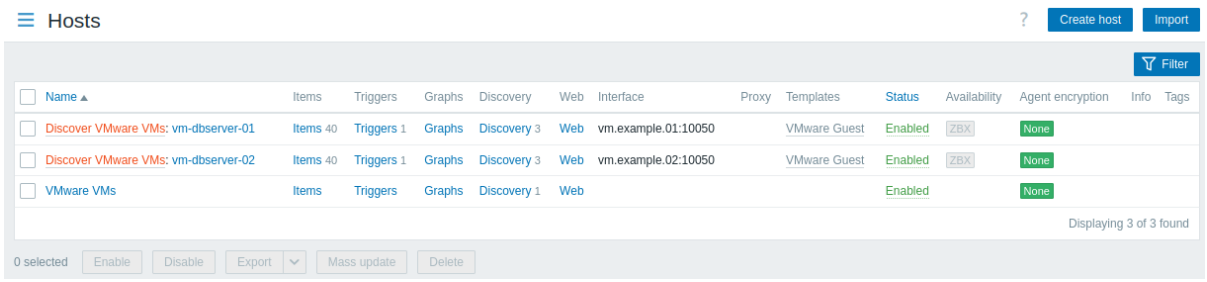


3. Klicken Sie auf die Schaltfläche *Add*, um den Host-Prototyp zu erstellen. Dieser Host-Prototyp wird verwendet, um Hosts für virtuelle Maschinen zu erstellen, die durch die zuvor erstellte Low-Level-Discovery-Regel erkannt wurden.

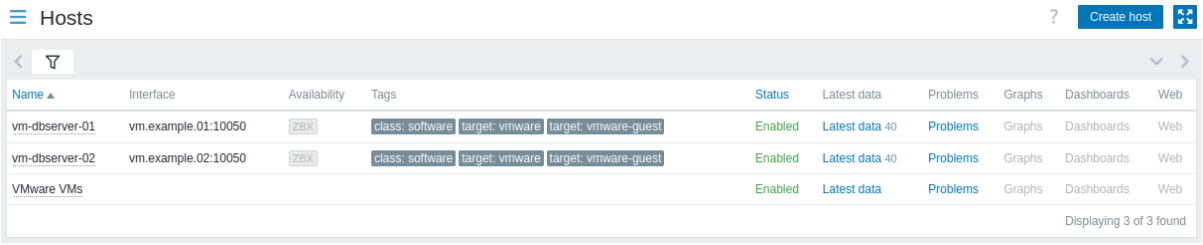
Hosts und Metriken anzeigen

Nachdem der Host-Prototyp erstellt wurde, erstellt die Low-Level-Discovery-Regel Hosts für erkannte virtuelle VMware-Maschinen, und Zabbix beginnt mit deren Überwachung. Beachten Sie, dass die Erkennung und Erstellung von Hosts bei Bedarf auch manuell ausgeführt werden kann.

Um die erstellten Hosts anzuzeigen, wechseln Sie zum Menüabschnitt *Datenerfassung → Hosts*.



Um die erfassten Metriken anzuzeigen, wechseln Sie zum Menüabschnitt *Überwachung → Hosts* und klicken Sie bei einem der Hosts auf *Letzte Daten*.



Erweiterte Konfiguration der Host-Schnittstelle

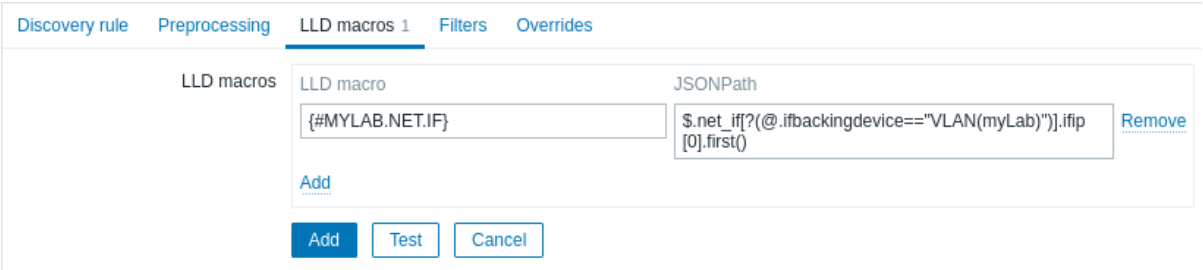
Der im Abschnitt *Eine Low-Level-Discovery-Regel erstellen* konfigurierte Datenpunktschlüssel `vmware.vm.discovery[{$VMWARE.URL}]` gibt Daten zu Netzwerkschnittstellen im Feld "net_if" zurück:

```
"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": []
      }
    ]
```

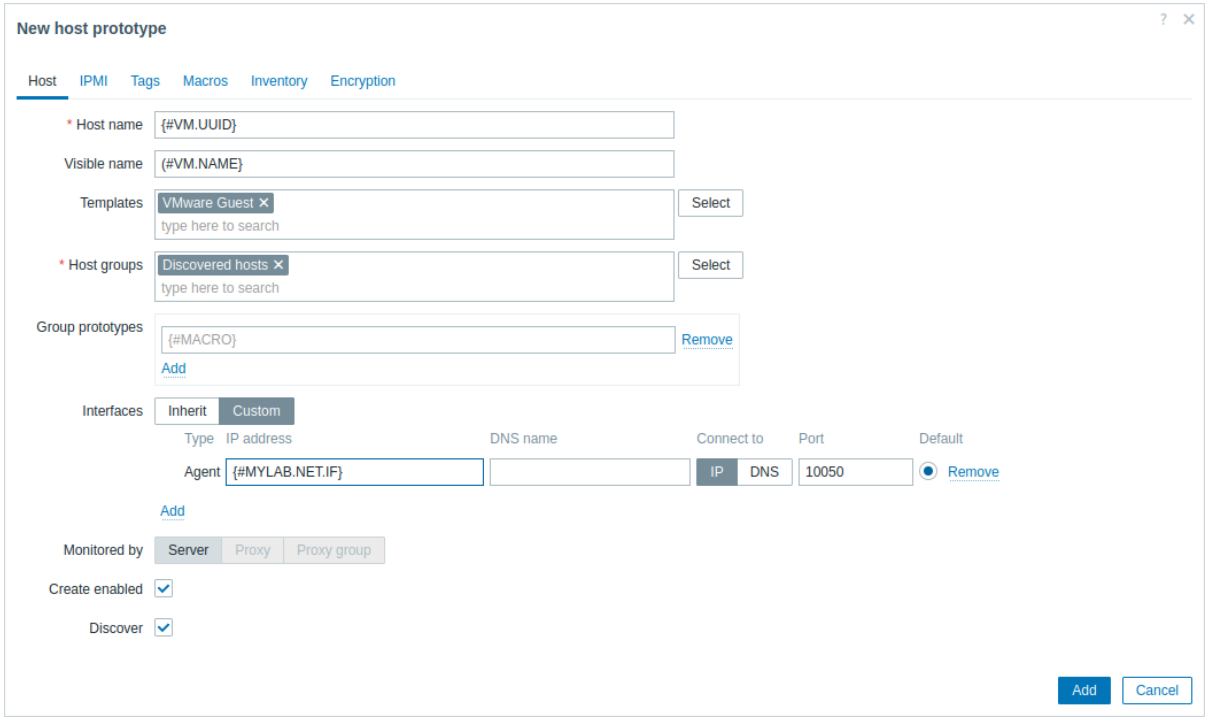
Diese Daten können verwendet werden, um eine benutzerdefinierte Host-Schnittstelle zu konfigurieren.

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.



9 Wartung

Übersicht Sie können in Zabbix Wartungszeiträume für Hosts und Host-Gruppen definieren.

Außerdem ist es möglich, die Wartung nur für einen einzelnen Auslöser (oder eine Teilmenge von Auslösern) zu definieren, indem Auslöser-Tags angegeben werden. In diesem Fall wird die Wartung nur für diese Auslöser aktiviert; alle anderen Auslöser des Hosts oder der Host-Gruppe befinden sich nicht in Wartung.

Es gibt zwei Wartungstypen – mit Datenerfassung und ohne Datenerfassung.

Während einer Wartung „mit Datenerfassung“ werden Auslöser wie gewohnt verarbeitet und bei Bedarf werden Ereignisse erstellt. Problemeskalationen werden jedoch für Hosts/Auslöser in Wartung pausiert, wenn die Option *Operationen für unterdrückte Probleme pausieren* in der Aktionskonfiguration aktiviert ist. In diesem Fall werden Eskalationsschritte, die das Senden von Benachrichtigungen oder Remote-Befehlen umfassen können, so lange ignoriert, wie der Wartungszeitraum andauert. Beachten Sie, dass Problemwiederherstellungs- und Aktualisierungsoperationen während der Wartung nicht unterdrückt werden, sondern nur Eskalationen.

Wenn beispielsweise Eskalationsschritte 0, 30 und 60 Minuten nach Beginn eines Problems geplant sind und es eine halbstündige Wartung gibt, die von 10 bis 40 Minuten nach dem Auftreten eines tatsächlichen Problems dauert, dann werden Schritt zwei und drei eine halbe Stunde später ausgeführt, also nach 60 bzw. 90 Minuten (vorausgesetzt, das Problem besteht noch). Wenn ein Problem während der Wartung auftritt, beginnt die Eskalation entsprechend erst nach der Wartung.

Um Problembenachrichtigungen während der Wartung normal (ohne Verzögerung) zu erhalten, müssen Sie die Option *Operationen für unterdrückte Probleme pausieren* in der Aktionskonfiguration deaktivieren.

Note:
Wenn sich mindestens ein Host (der im Auslöser-Ausdruck verwendet wird) nicht im Wartungsmodus befindet, sendet Zabbix eine Problembenachrichtigung.

Der Zabbix Server muss während der Wartung laufen. Wartungen werden jede Minute neu berechnet oder sofort, sobald der Konfigurations-Cache neu geladen wird, wenn es Änderungen am Wartungszeitraum gibt.

Timer-Prozesse prüfen bei 0 Sekunden jeder Minute, ob der Host-Status in bzw. aus der Wartung geändert werden muss. Zusätzlich prüft der Timer-Prozess jede Sekunde, ob Wartungen gestartet/beendet werden müssen, abhängig davon, ob es nach der Konfigurationsaktualisierung Änderungen an den [Wartungszeiträumen] gibt. Daher hängt die Geschwindigkeit beim Starten/Beenden von Wartungszeiträumen vom Konfigurations-Aktualisierungsintervall ab (standardmäßig 10 Sekunden). Beachten Sie, dass Änderungen am Wartungszeitraum die Einstellungen *Aktiv seit/Aktiv bis* nicht einschließen. Wenn außerdem ein Host/eine Host-Gruppe zu einem bereits aktiven Wartungszeitraum hinzugefügt wird, werden die Änderungen vom Timer-Prozess erst zu Beginn der nächsten Minute aktiviert.

Beachten Sie, dass beim Eintritt eines Hosts in die Wartung die Timer-Prozesse des Zabbix Servers alle offenen Probleme lesen, um zu prüfen, ob diese unterdrückt werden müssen. Dies kann sich auf die Performance auswirken, wenn es viele offene Probleme gibt. Der Zabbix Server liest beim Start ebenfalls alle offenen Probleme, auch wenn zu diesem Zeitpunkt keine Wartungen konfiguriert sind.

Beachten Sie, dass der Zabbix Server (oder Proxy) immer Daten erfasst, unabhängig vom Wartungstyp (einschließlich Wartung „ohne Daten“). Die Daten werden später vom Server ignoriert, wenn „keine Datenerfassung“ festgelegt ist.

Wenn eine Wartung „ohne Daten“ endet, lösen Auslöser, die die Funktion nodata() verwenden, vor der nächsten Prüfung innerhalb des von ihnen geprüften Zeitraums nicht aus.

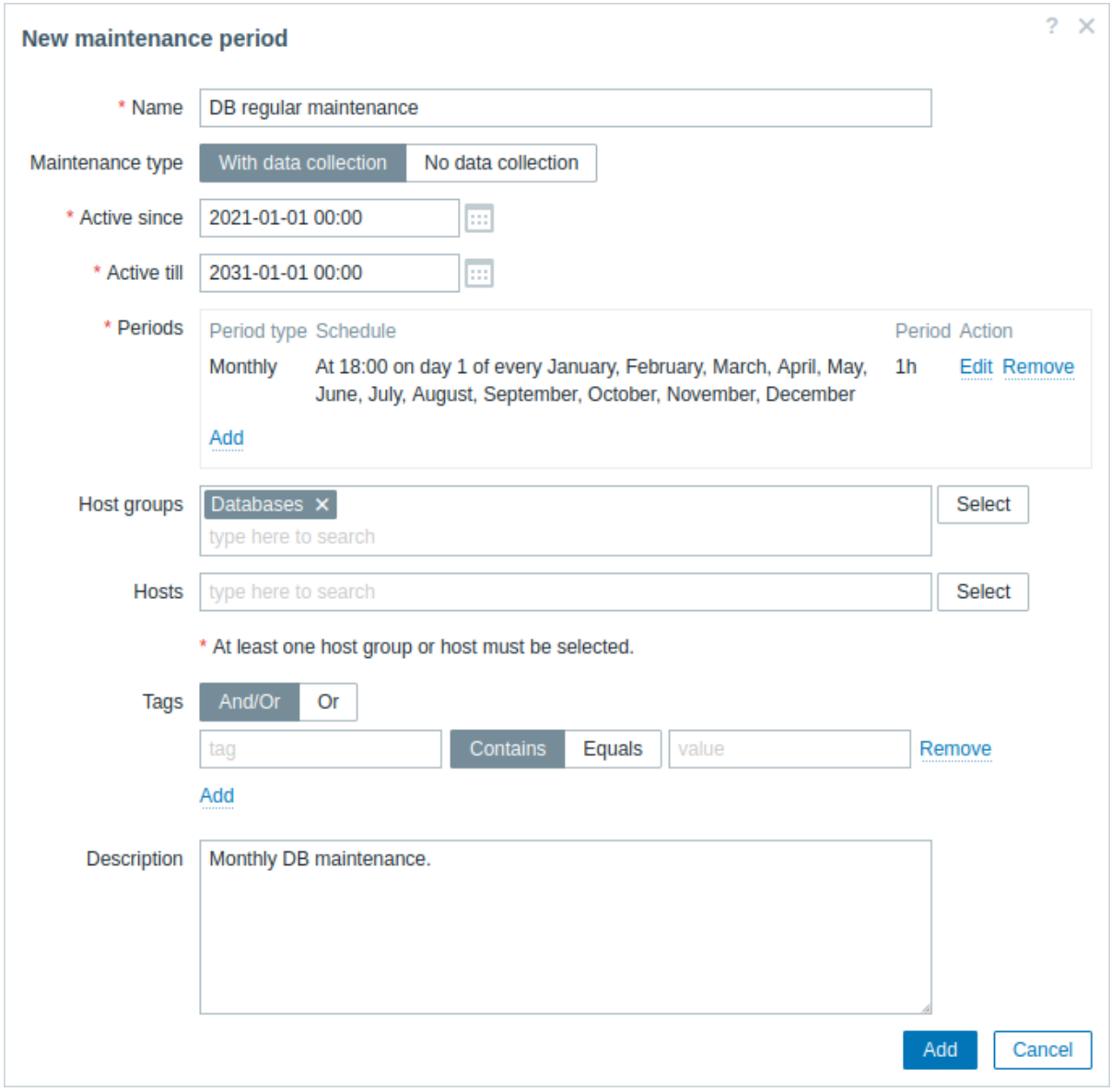
Wenn ein Log-Datenpunkt hinzugefügt wird, während sich ein Host in Wartung befindet, und die Wartung endet, werden nur neue Logdatei-Einträge seit dem Ende der Wartung erfasst.

Wenn ein mit Zeitstempel versehener Wert für einen Host gesendet wird, der sich in einer Wartung vom Typ „ohne Daten“ befindet (z. B. mit Zabbix sender), dann wird dieser Wert verworfen. Es ist jedoch möglich, einen mit Zeitstempel versehenen Wert für einen abgelaufenen Wartungszeitraum zu senden, und dieser wird akzeptiert.

Wenn Wartungszeitraum, Hosts, Gruppen oder Tags vom Benutzer geändert werden, werden die Änderungen erst nach der Synchronisierung des Konfigurations-Caches wirksam.

Konfiguration So konfigurieren Sie einen Wartungszeitraum:

1. Gehen Sie zu: *Datenerfassung* > *Wartung*.
2. Klicken Sie auf *Wartungszeitraum erstellen* (oder auf den Namen eines bestehenden Wartungszeitraums).
3. Geben Sie die Wartungsparameter im Formular ein.

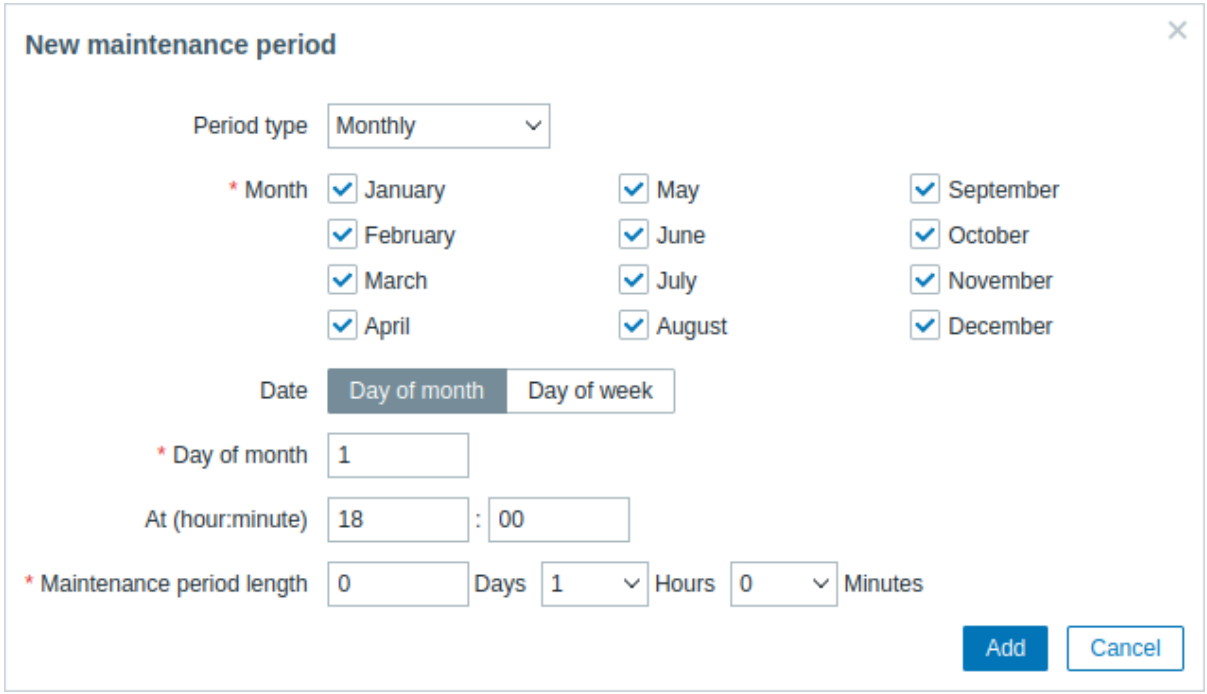


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Name des Wartungszeitraums.
Wartungstyp	Es können zwei Wartungstypen festgelegt werden: **Mit Datenerfassung** - Daten werden während der Wartung vom Server erfasst, Auslöser werden verarbeitet; **Keine Datenerfassung** - Daten werden während der Wartung nicht vom Server erfasst. Unter Auswirkung von Wartungszeiträumen finden Sie Informationen dazu, wie sich die einzelnen Typen auf den Verfügbarkeitsbericht auswirken.
Aktiv seit	Datum und Uhrzeit, ab denen die Ausführung von Wartungszeiträumen aktiv wird. *Hinweis:* Das Festlegen dieser Zeit allein aktiviert keinen Wartungszeitraum; Wartungszeiträume müssen unter *Zeiträume* konfiguriert werden (siehe unten).
Aktiv bis	Datum und Uhrzeit, ab denen die Ausführung von Wartungszeiträumen nicht mehr aktiv ist.
Zeiträume	In diesem Block können Sie die genauen Tage und Uhrzeiten festlegen, zu denen die Wartung stattfindet. Ein Klick auf Add öffnet ein Popup-Fenster mit einem flexiblen Formular *Wartungszeitraum*, in dem Sie den Wartungsplan definieren können. Eine ausführliche Beschreibung finden Sie unter Wartungszeiträume.
Host-Gruppen	Wählen Sie Host-Gruppen aus, für die die Wartung aktiviert werden soll. Die Wartung wird für alle Hosts aus den angegebenen Host-Gruppen aktiviert. Dieses Feld unterstützt Autovervollständigung; wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste aller verfügbaren Host-Gruppen angezeigt. Wenn Sie eine übergeordnete Host-Gruppe angeben, werden implizit alle untergeordneten Host-Gruppen ausgewählt. Dadurch wird die Wartung auch für Hosts aus untergeordneten Gruppen aktiviert.
Hosts	Wählen Sie Hosts aus, für die die Wartung aktiviert werden soll. Dieses Feld unterstützt Autovervollständigung; wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste aller verfügbaren Hosts angezeigt.
Tags	Geben Sie Tags an, um Probleme zu unterdrücken, deren Tags mit Tags auf Hosts in Wartung übereinstimmen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen zwei Operatoren zur Verfügung: **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilstring-Abgleich, groß-/kleinschreibungssensitiv); **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (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. Tags können nur angegeben werden, wenn der Modus *Mit Datenerfassung* ausgewählt ist.
Beschreibung	Beschreibung des Wartungszeitraums.

Wartungszeiträume

Das Fenster für den Wartungszeitraum dient zur Planung eines wiederkehrenden oder einmaligen Wartungszeitraums. Das Formular ist dynamisch; die verfügbaren Felder ändern sich abhängig vom ausgewählten *Zeitraumtyp*.



Zeitraumtyp	Beschreibung
Nur einmal	Konfigurieren Sie einen einmaligen Wartungszeitraum: *Datum* - Datum und Uhrzeit des Wartungszeitraums; *Länge des Wartungszeitraums* - wie lange die Wartung aktiv sein wird.
Täglich	Konfigurieren Sie einen täglichen Wartungszeitraum: *Alle x Tag(e)* - Wartungshäufigkeit (1 - *(Standard)* jeden Tag, 2 - alle zwei Tage usw.); *Um (Stunde:Minute)* - Uhrzeit, zu der die Wartung beginnt; *Länge des Wartungszeitraums* - wie lange die Wartung aktiv sein wird. Wenn der Parameter *Alle x Tag(e)* größer als „1“ ist, ist der Starttag der Tag, auf den die Zeit *Aktiv seit* fällt. Beispiele: - wenn *Aktiv seit* auf „2021-01-01 12:00“ gesetzt ist, *Alle x Tag(e)* auf „2“ gesetzt ist und *Um (Stunde:Minute)* auf „23:00“ gesetzt ist, 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 *Aktiv seit* auf „2021-01-01 12:00“ gesetzt ist, *Alle x Tag(e)* auf „2“ gesetzt ist und *Um (Stunde:Minute)* auf „01:00“ gesetzt ist, dann beginnt der erste Wartungszeitraum am 3. Januar um 01:00 Uhr, während der zweite Wartungszeitraum am 5. Januar um 01:00 Uhr beginnt.
Wöchentlich	Konfigurieren Sie einen wöchentlichen Wartungszeitraum: *Alle x Woche(n)* - Wartungshäufigkeit (1 - *(Standard)* jede Woche, 2 - alle zwei Wochen usw.); *Wochentag* - an welchem Tag die Wartung stattfinden soll; *Um (Stunde:Minute)* - Uhrzeit, zu der die Wartung beginnt; *Länge des Wartungszeitraums* - wie lange die Wartung aktiv sein wird. Wenn der Parameter *Alle x Woche(n)* größer als „1“ ist, ist die Startwoche die Woche, in die die Zeit *Aktiv seit* fällt. Beispiele finden Sie oben in der Beschreibung des Parameters *Täglich*.
Monatlich	Konfigurieren Sie einen monatlichen Wartungszeitraum: *Monat* - wählen Sie alle Monate aus, in denen die regelmäßige Wartung durchgeführt wird; *Datum:* ***Tag des Monats*** - wählen Sie diese Option, wenn die Wartung jeden Monat am selben Datum stattfinden soll (zum Beispiel an jedem 1. Tag des Monats), und wählen Sie dann im eingeblendeten Feld *Tag des Monats* den gewünschten Tag aus; *Datum:* ***Wochentag*** - wählen Sie diese Option, wenn die Wartung nur an bestimmten Tagen stattfinden soll (zum Beispiel an jedem ersten Montag des Monats); 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); *Um (Stunde:Minute)* - Uhrzeit, zu der die Wartung beginnt; *Länge des Wartungszeitraums* - wie lange die Wartung aktiv sein wird.

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 *Hinzufügen*, um den Wartungszeitraum zum Block *Zeiträume* hinzuzufügen.

Beachten Sie, dass Änderungen der Sommerzeit (DST) keinen Einfluss darauf haben, wie lange die Wartung dauert. Nehmen wir zum Beispiel an, dass eine zweistündige Wartung konfiguriert ist, die normalerweise um 01:00 beginnt und um 03:00 endet:

- wenn nach einer Stunde Wartung (um 02:00) eine DST-Umstellung erfolgt und die aktuelle Zeit von 02:00 auf 03:00 springt, wird die Wartung noch eine weitere Stunde fortgesetzt (bis 04:00);
- wenn nach zwei Stunden Wartung (um 03:00) eine DST-Umstellung erfolgt und die aktuelle Zeit von 03:00 auf 02:00 zurückspringt, wird die Wartung beendet, da zwei Stunden vergangen sind;
- wenn ein Wartungszeitraum während der Stunde beginnt, die durch eine DST-Umstellung übersprungen wird, startet die Wartung nicht.

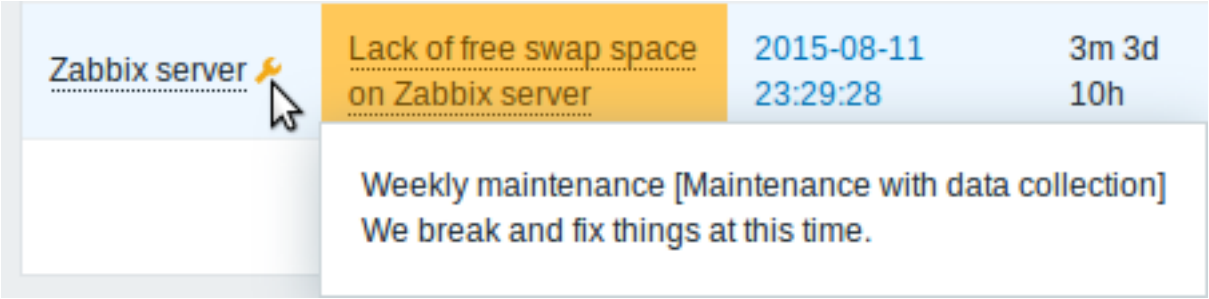
Wenn ein Wartungszeitraum auf „1 Tag“ gesetzt ist (der tatsächliche Wartungszeitraum beträgt 24 Stunden, da Zabbix Tage in Stunden berechnet), um 00:00 beginnt und am nächsten Tag um 00:00 endet:

- wird die Wartung am nächsten Tag um 01:00 beendet, wenn die aktuelle Zeit um eine Stunde vorgestellt wird;
- wird die Wartung an diesem Tag um 23:00 beendet, wenn die aktuelle Zeit um eine Stunde zurückgestellt wird.

Anzeige Hosts in Wartung anzeigen

Ein orangefarbenes Schraubenschlüssel-Symbol neben dem Hostnamen zeigt an, dass sich dieser Host in Wartung befindet in:

- *Dashboards*
- *Monitoring* > *Probleme*
- *Inventar* > *Hosts* > *Details zur Host-Inventarisierung*
- *Datensammlung* > *Hosts* (siehe Spalte „Status“)



Details zur Wartung werden angezeigt, wenn sich der Mauszeiger über dem Symbol befindet.

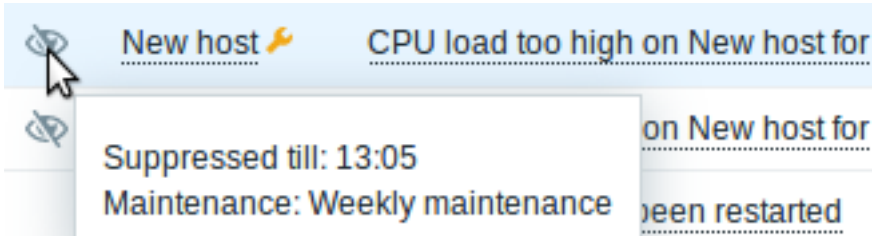
Zusätzlich erhalten Hosts in Wartung einen orangefarbenen Hintergrund in *Monitoring* > *Karten*.

Unterdrückte Probleme anzeigen

Normalerweise werden Probleme für Hosts in Wartung unterdrückt, d. h. sie werden nicht im Frontend angezeigt. Es ist jedoch auch möglich zu konfigurieren, dass unterdrückte Probleme angezeigt werden, indem Sie die Option *Unterdrückte Probleme anzeigen* an diesen Stellen auswählen:

- *Dashboards* (in der Widget-Konfiguration von *Problem-Hosts*, *Probleme*, *Probleme nach Schweregrad*, *Auslöser-Übersicht*)
- *Monitoring* > *Probleme* (im Filter)
- *Monitoring* > *Karten* (in der Kartenkonfiguration)
- Globale Benachrichtigungen (in der Benutzerprofilkonfiguration)

Wenn unterdrückte Probleme angezeigt werden, wird das folgende Symbol angezeigt: . Wenn Sie den Mauszeiger über das Symbol bewegen, werden weitere Details angezeigt.



Berechnung von Warteschlangen während der Wartung Warteschlangen, die im Zabbix Frontend angezeigt werden (Administration > Queue), werden vom Zabbix Server berechnet. Sie enthalten keine Datenpunkte in Wartung ohne Datenerfassung – die Warteschlangenlänge ist für diese Datenpunkte immer null, auch wenn ihre Werte verzögert sind. Verzögerte Datenpunkte in Wartung mit Datenerfassung werden weiterhin in der Warteschlange gezählt.

Der Zabbix Proxy kennt keine Wartungszeiträume, 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.

Daher können für dieselben Datenpunkte in Wartung ohne Datenerfassung unterschiedliche Warteschlangenlängen vom Zabbix Frontend und von internen Prüfungen auf Zabbix Proxys gemeldet werden.

10 Reguläre Ausdrücke

Übersicht Perl-kompatible reguläre Ausdrücke (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 an unterstützten Stellen manuell einen regulären Ausdruck eingeben. Beachten Sie, dass der Ausdruck nicht mit @ beginnen darf, da dieses Symbol in Zabbix für die Referenzierung globaler regulärer Ausdrücke verwendet wird.

> **Warning:**
> Bei der Verwendung regulärer Ausdrücke kann es zu einem Stack-Überlauf kommen. Weitere Informationen finden Sie auf der pcrestack-Manpage.

Beachten Sie, dass beim mehrzeiligen Abgleich die Anker ^ und $ jeweils auf den Anfang bzw. das Ende jeder Zeile passen, anstatt auf den Anfang bzw. das Ende der gesamten Zeichenkette.

Siehe auch Beispiele für korrektes Escaping in verschiedenen Kontexten.

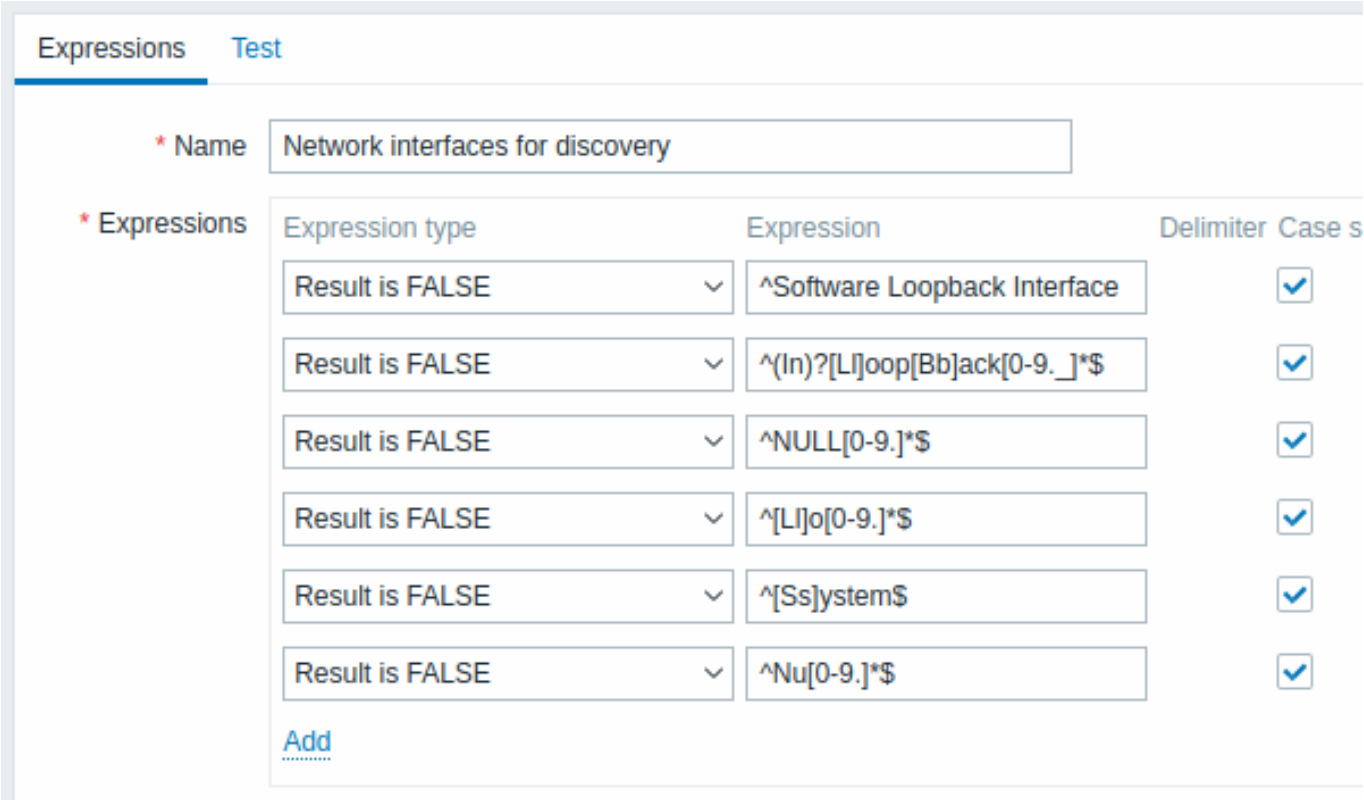
Globale reguläre Ausdrücke Im Zabbix Frontend gibt es einen erweiterten Editor zum Erstellen und Testen komplexer regulärer Ausdrücke.

Sobald ein regulärer Ausdruck auf diese Weise erstellt wurde, kann er an mehreren Stellen im Frontend verwendet werden, indem auf seinen Namen mit vorangestelltem @ verwiesen wird, zum Beispiel *@mycustomregexp*.

So erstellen Sie einen globalen regulären Ausdruck:

- Gehen Sie zu: *Administration* → *General*
- Wählen Sie *Regular expressions* aus dem Dropdown-Menü
- Klicken Sie auf *New regular expression*

Auf der Registerkarte **Expressions** können Sie den Namen des regulären Ausdrucks festlegen und Unterausdrücke hinzufügen.



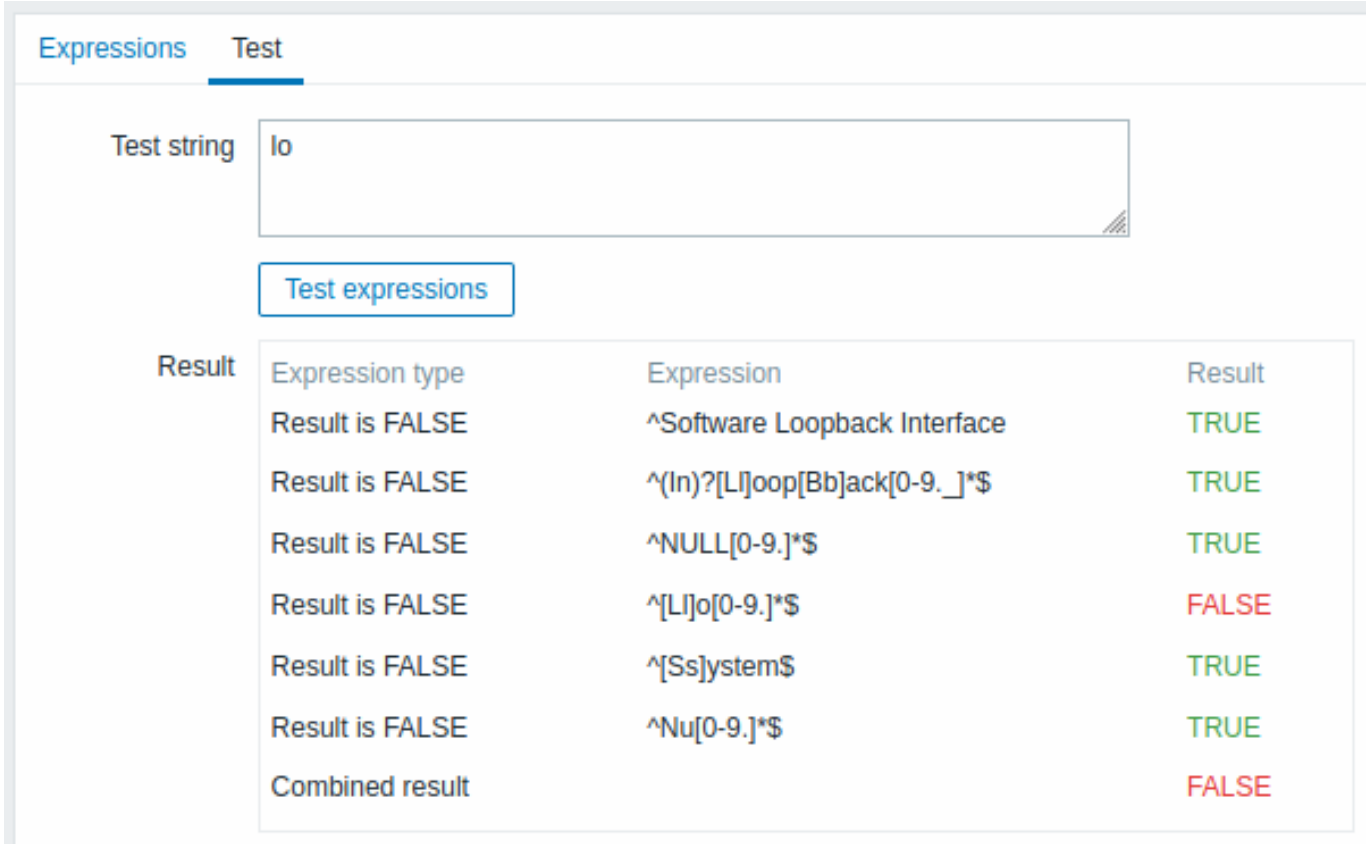
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Legen Sie den Namen des regulären Ausdrucks fest. Beliebige Unicode-Zeichen sind zulässig.
Expressions	Klicken Sie im Block *Expressions* auf *Add*, um einen neuen Unterausdruck hinzuzufügen.
Expression type	Wählen Sie den Ausdruckstyp aus: **Character string included** - stimmt mit der Teilzeichenfolge überein **Any character string included** - stimmt mit einer beliebigen Teilzeichenfolge aus einer durch Trennzeichen getrennten Liste überein. Die durch Trennzeichen getrennte Liste enthält ein Komma (,), einen Punkt (.) oder einen Schrägstrich (/). **Character string not included** - stimmt mit jeder Zeichenfolge außer der Teilzeichenfolge überein **Result is TRUE** - stimmt mit dem regulären Ausdruck überein **Result is FALSE** - stimmt nicht mit dem regulären Ausdruck überein
Expression	Geben Sie die Teilzeichenfolge bzw. den regulären Ausdruck ein.
Delimiter	Ein Komma (,), ein Punkt (.) oder ein Schrägstrich (/) zum Trennen von Textzeichenfolgen in einem regulären Ausdruck. Dieser Parameter ist nur aktiv, wenn der Ausdruckstyp *"Any character string included"* ausgewählt ist.
Case sensitive	Ein Kontrollkästchen, mit dem festgelegt wird, ob bei einem regulären Ausdruck die Groß- und Kleinschreibung beachtet wird.

Ein Schrägstrich (/) im Ausdruck wird wörtlich behandelt und nicht als Trennzeichen. Auf diese Weise können Ausdrücke, die einen Schrägstrich enthalten, fehlerfrei gespeichert werden.

Attention:
Ein benutzerdefinierter Name für einen regulären Ausdruck in Zabbix kann Kommas, Leerzeichen usw. enthalten. In Fällen, in denen dies bei der Referenzierung zu Fehlinterpretationen führen kann (zum Beispiel ein Komma im Parameter eines Datenpunktschlüssels), kann die gesamte Referenz in Anführungszeichen gesetzt werden, zum Beispiel so: *"@My custom regexp for purpose1, purpose2"*.
Namen regulärer Ausdrücke dürfen an anderen Stellen nicht in Anführungszeichen gesetzt werden (zum Beispiel in den Eigenschaften von LLD-Regeln).

Auf der Registerkarte **Test** können der reguläre Ausdruck und seine Unterausdrücke getestet werden, indem eine Testzeichenfolge angegeben wird.



Die Ergebnisse zeigen den Status jedes Unterausdrucks sowie den Gesamtstatus des benutzerdefinierten Ausdrucks.

Der Gesamtstatus des benutzerdefinierten Ausdrucks wird als *Combined result* definiert. Wenn mehrere Unterausdrücke definiert sind, verwendet Zabbix den logischen AND-Operator zur Berechnung von *Combined result*. Das bedeutet: Wenn mindestens ein Ergebnis False ist, hat auch *Combined result* den Status False.

Standardmäßige globale reguläre Ausdrücke Zabbix wird mit mehreren globalen regulären Ausdrücken in seinem Standard-Datensatz ausgeliefert.

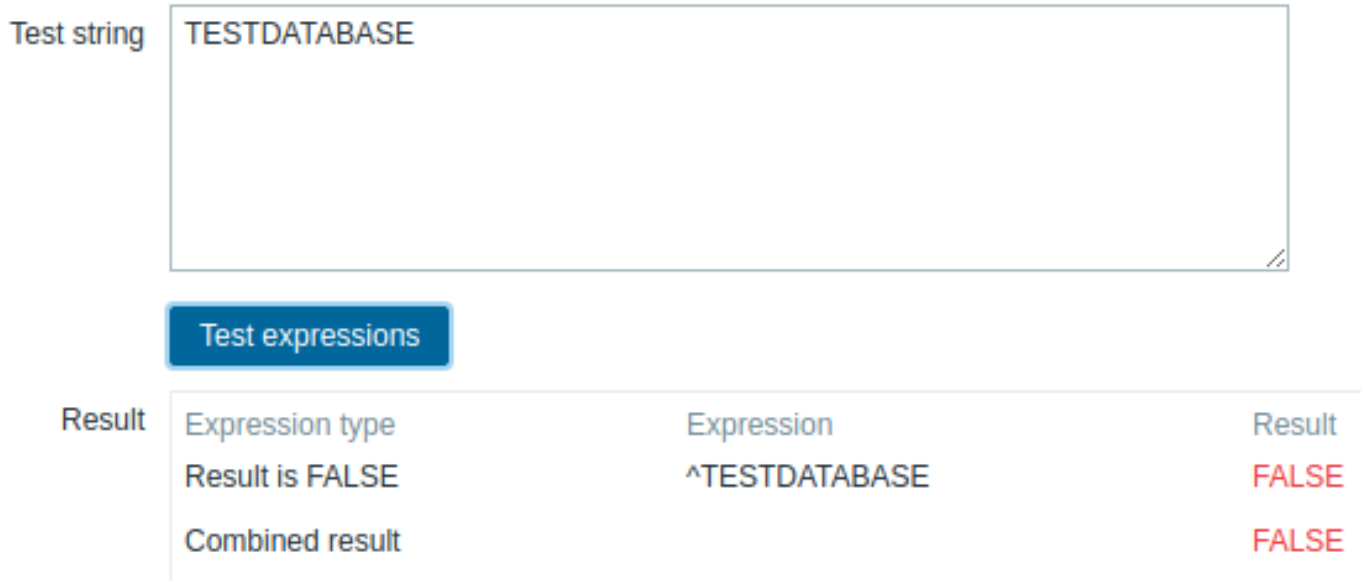
Name	Ausdruck	Entspricht
Dateisysteme für Discovery	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)$`	„btrfs“ oder „ext2“ oder „ext3“ oder „ext4“ oder „jfs“ oder „reiser“ oder „xfs“ oder „ffs“ oder „ufs“ oder „jfs“ oder „jfs2“ oder „vxfs“ oder „hfs“ oder „refs“ oder „apfs“ oder „ntfs“ oder „fat32“ oder „zfs“
Netzwerkschnittstellen für Discovery	`^Software Loopback Interface`	Zeichenfolgen, die mit „Software Loopback Interface“ beginnen.
	`^lo$`	„lo“
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Zeichenfolgen, die optional mit „In“ beginnen, dann „L“ oder „l“, dann „oop“, dann „B“ oder „b“, dann „ack“ enthalten und optional von einer beliebigen Anzahl von Ziffern, Punkten oder Unterstrichen gefolgt werden können.
	`^NULL[0-9.]*$`	Zeichenfolgen, die mit „NULL“ beginnen und optional von einer beliebigen Anzahl von Ziffern oder Punkten gefolgt werden.
	`^[Ll]o[0-9.]*$`	Zeichenfolgen, die mit „Lo“ oder „lo“ beginnen und optional von einer beliebigen Anzahl von Ziffern oder Punkten gefolgt werden.
	`^[Ss]ystem$`	„System“ oder „system“
	`^Nu[0-9.]*$`	Zeichenfolgen, die mit „Nu“ beginnen und optional von einer beliebigen Anzahl von Ziffern oder Punkten gefolgt werden.
Speichergeräte für SNMP-Discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	„Physical memory“ oder „Virtual memory“ oder „Memory buffers“ oder „Cached memory“ oder „Swap space“

Name	Ausdruck	Entspricht
Windows-Dienstnamen für Discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	MMCSS, „gupdate“, „SysmonLog“ oder Zeichenfolgen wie „clr_optimization_v2.0.50727_32“ und „clr_optimization_v4.0.30319_32“, wobei anstelle von Punkten jedes beliebige Zeichen außer einem Zeilenumbruch stehen kann.
Windows-Dienst-Startzustände für Discovery	`^(automatic|automatic delayed)$`	„automatic“ oder „automatic delayed“

Beispiele Beispiel 1

Verwendung des folgenden Ausdrucks in der Low-Level-Discovery, um Datenbanken zu erkennen, mit Ausnahme einer Datenbank mit einem bestimmten Namen:

`^TESTDATABASE$`

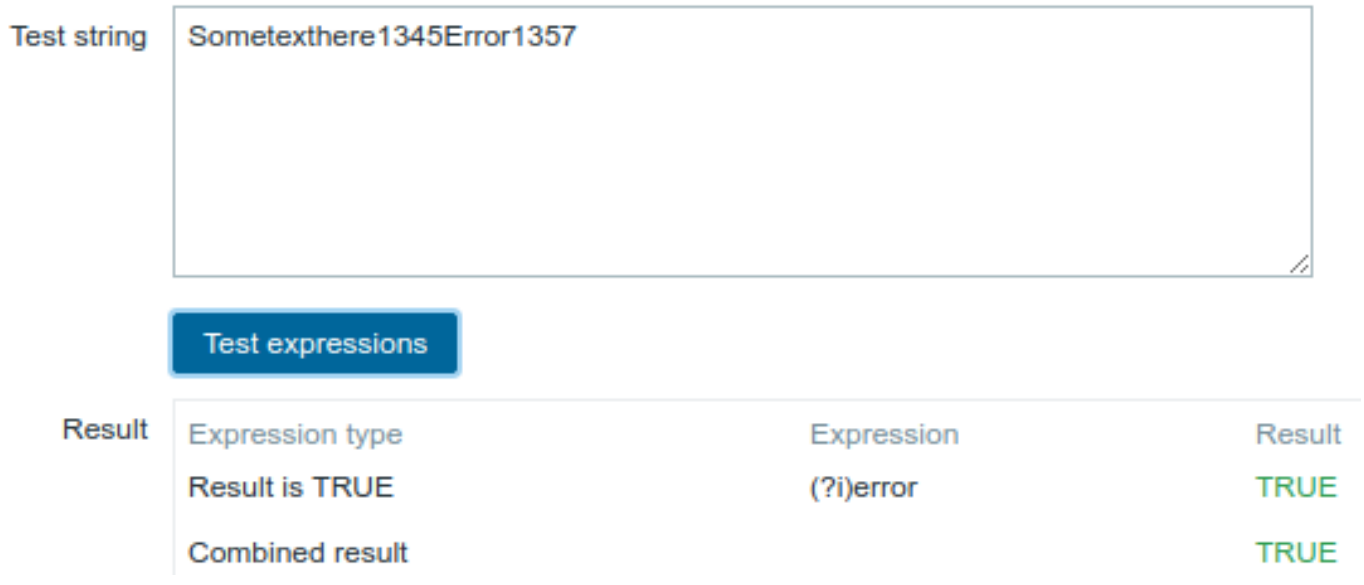


Gewählter *Ausdruckstyp*: „Ergebnis ist FALSE“. Entspricht keinem Namen, der die Zeichenfolge „*TESTDATABASE*“ enthält.

Beispiel mit einem Inline-RegEx-Modifikator

Verwendung des folgenden regulären Ausdrucks einschließlich eines Inline-Modifikators (?i), um die Zeichen „error“ abzugleichen:

`(?i)error`



Gewählter *Ausdruckstyp*: „Ergebnis ist TRUE“. Die Zeichen „error“ werden abgeglichen.

Ein weiteres Beispiel mit einem Inline-RegEx-Modifikator

Verwendung des folgenden regulären Ausdrucks mit mehreren Inline- Modifikatoren, um die Zeichen nach einer bestimmten Zeile abzugleichen:

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# we add s modifier to allow . match newline charac
```

Gewählter Ausdruckstyp: „Ergebnis ist WAHR“. Zeichen nach einer bestimmten Zeile werden abgeglichen.

Attention:
Der Modifikator **g** kann nicht in der Zeile angegeben werden. Die Liste der verfügbaren Modifikatoren finden Sie auf der pcresyntax-Manpage. Weitere Informationen zur PCRE-Syntax finden Sie in der PCRE-HTML-Dokumentation.

Unterstützung regulärer Ausdrücke nach Ort

Ort	Regulärer Ausdruck	Globaler regulärer Ausdruck	Mehrzeilenabgleich	Kommentare
Agent-Datenpunkte				
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 beides, 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`
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[]			Ja	Parameter `regexp`
vfs.file.regmatch[]				
web.page.regexp[]				

Ort	Regulärer Ausdruck	Globaler regulärer Ausdruck	Mehrzeilenabgleich	Kommentare
SNMP-Traps				
snmptrap[]	Ja	Ja	Nein	Parameter `regexp`
Vorverarbeitung von Daten-punk-twerten	Ja	Nein	Nein	Parameter `pattern`
Funktionen für Aus-löser/berechnete Daten-punkte				
count()	Ja	Ja	Ja	Parameter `pattern`, wenn der Parameter `operator` *regexp* oder *iregexp* ist
countunique()	Ja	Ja		
find()	Ja	Ja		
logeventid()	Ja	Ja	Nein	Parameter `pattern`
logsource()				
Low-Level-Discovery				
Filter	Ja	Ja	Nein	Feld *Regular expression*
Überschreibungen	Ja	Nein		In den Optionen *matches*, *does not match* für *Operation*-Bedingungen
Aktionsbedingungen	Ja	Nein	Nein	In den Optionen *matches*, *does not match* für Autoregistrierungsbedingungen von *Host name* und *Host metadata*
Skripte	Ja	Ja	Nein	Feld *Input validation rule*
Web-Überwachung	Ja	Nein	Ja	*Variables* mit dem Präfix **regex:** Feld *Required string*
Benutzer-Makrokontext	Ja	Nein	Nein	Im Makrokontext mit dem Präfix **regex:**
Makrofunktionen				
regsub()	Ja	Nein	Nein	Parameter `pattern`
iregsub()				
Link-Indikatoren in Karten	Ja	Nein	Nein	Feld *Pattern* (für Text-Datenpunkte)
Symbolzuordnung	Ja	Ja	Nein	Feld *Expression*
Wertzuordnung	Ja	Nein	Nein	Feld *Value*, wenn der Zuordnungstyp `regexp` ist

11 Problemquittierung

Übersicht Problemereignisse in Zabbix können von Benutzern bestätigt werden.

Wenn ein Benutzer über ein Problemereignis benachrichtigt wird, kann er zum Zabbix Frontend gehen, das Popup-Fenster zur Problemaktualisierung dieses Problems auf eine der unten aufgeführten Arten öffnen und das Problem bestätigen. Bei der Bestätigung kann er einen Kommentar eingeben, zum Beispiel dass er bereits daran arbeitet oder was auch immer er dazu mitteilen möchte.

Auf diese Weise sieht ein anderer Systembenutzer, wenn er dasselbe Problem bemerkt, sofort, ob es bereits bestätigt wurde und welche Kommentare bisher dazu vorliegen.

So kann der Workflow zur Behebung von Problemen mit mehr als einem Systembenutzer koordiniert ablaufen.

Der Bestätigungsstatus wird auch beim Definieren von Aktionsoperationen verwendet. Sie können zum Beispiel festlegen, dass eine Benachrichtigung nur dann an einen Manager der nächsthöheren Ebene gesendet wird, wenn ein Ereignis für eine gewisse Zeit nicht bestätigt wurde.

Um Ereignisse zu bestätigen und zu kommentieren, muss ein Benutzer mindestens Leserechte für die entsprechenden Auslöser haben. Um den Schweregrad eines Problems zu ändern oder ein Problem zu schließen, muss ein Benutzer Lese-/Schreibrechte für die entsprechenden Auslöser haben.

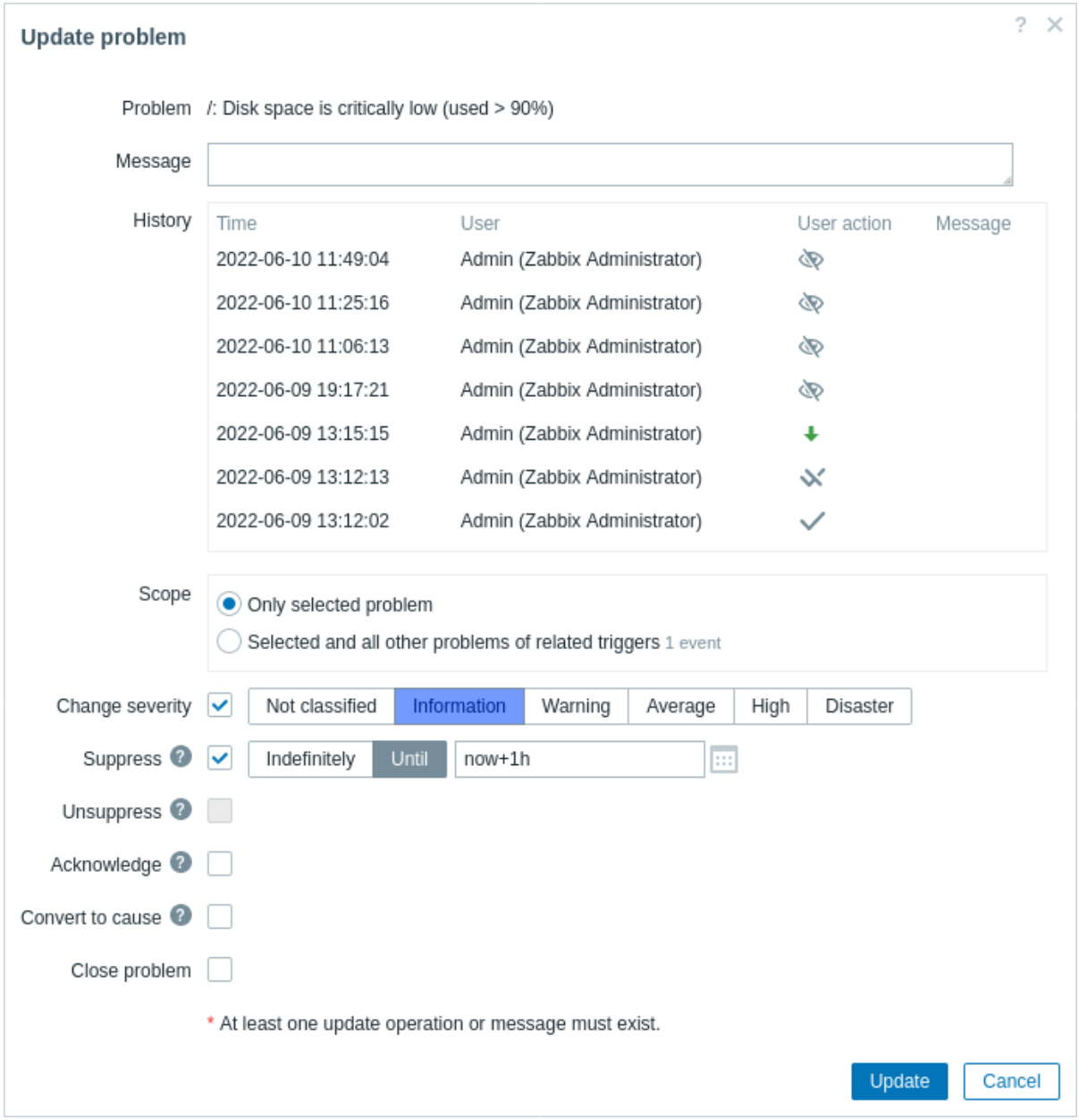
Es gibt **mehrere** Möglichkeiten, auf das Popup-Fenster zur Problemaktualisierung zuzugreifen, in dem ein Problem bestätigt werden kann.

- 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 klicken in:
  - *Dashboards* (*Trigger overview*-Widget)

Das Popup-Menü enthält die Option *Update*, über die Sie zum Fenster für die Problemaktualisierung gelangen.

Probleme aktualisieren Das Popup zum Aktualisieren von Problemen ermöglicht Folgendes:

- das Problem zu kommentieren
- bisherige Kommentare und Aktionen anzuzeigen
- den Schweregrad des Problems zu ändern
- das Problem zu unterdrücken/die Unterdrückung aufzuheben
- das Problem zu quittieren/die Quittierung aufzuheben
- ein Symptomproblem in ein Ursachenproblem umzuwandeln
- das Problem manuell zu schließen



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

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	Frühere Aktivitäten und Kommentare zum Problem werden zusammen mit Zeit- und Benutzerdetails aufgelistet. Die Bedeutung der Symbole zur Kennzeichnung von Benutzeraktionen finden Sie auf der Seite event detail. Beachten Sie, dass der Verlauf nur angezeigt wird, wenn für die Aktualisierung nur ein Problem ausgewählt ist.
Scope	Definieren Sie den Umfang solcher Aktionen wie das Ändern des Schweregrads, das Quittieren oder das manuelle Schließen von Problemen: **Only selected problem** - wirkt sich nur auf dieses Ereignis aus **Selected and all other problems of related triggers** - beim Quittieren/Schließen eines Problems wirkt sich dies auf dieses Ereignis und alle anderen Probleme aus, die bisher nicht quittiert/geschlossen wurden. Wenn der Umfang bereits quittierte oder geschlossene Probleme enthält, werden diese Probleme nicht erneut quittiert/geschlossen. Andererseits ist die Anzahl der Nachrichten- und Schweregradänderungsvorgänge nicht begrenzt.

Parameter	Beschreibung
Change severity	Aktivieren Sie das Kontrollkästchen und klicken Sie auf die Schweregrad-Schaltfläche, um den Schweregrad des Problems zu aktualisieren. Das Kontrollkästchen zum Ändern des Schweregrads ist verfügbar, wenn für mindestens eines der ausgewählten Probleme Lese-/Schreibberechtigungen bestehen. Beim Klicken auf *Update* werden nur die Probleme aktualisiert, für die Lese-/Schreibzugriff besteht. Wenn für keinen der ausgewählten Auslöser Lese-/Schreibberechtigungen bestehen, ist das Kontrollkästchen deaktiviert.
Suppress	Aktivieren Sie das Kontrollkästchen, um das Problem zu unterdrücken: **Indefinitely** - auf unbestimmte Zeit unterdrücken **Until** - bis zu einem angegebenen Zeitpunkt unterdrücken. Es werden sowohl absolute als auch relative Zeitformate 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 zu einem absoluten Datum/Zeitpunkt Beachten Sie, dass ein einfacher Zeitraum (z. B. `1d`, `1w`) nicht unterstützt wird. Die Verfügbarkeit dieser Option hängt von den Benutzerrollen-Einstellungen für "Suppress problems" ab. Siehe auch: Problem suppression
Unsuppress	Aktivieren Sie das Kontrollkästchen, um die Unterdrückung des Problems aufzuheben. Dieses Kontrollkästchen ist nur aktiv, wenn mindestens eines der ausgewählten Probleme unterdrückt ist. Die Verfügbarkeit dieser Option hängt von den Benutzerrollen-Einstellungen für "Suppress problems" ab.
Acknowledge	Aktivieren Sie das Kontrollkästchen, um das Problem zu quittieren. Dieses Kontrollkästchen ist verfügbar, wenn sich unter den ausgewählten mindestens ein nicht quittiertes Problem befindet. Es ist nicht möglich, einem bereits quittierten Problem eine weitere Quittierung hinzuzufügen (ein weiterer Kommentar ist jedoch möglich).
Unacknowledge	Aktivieren Sie das Kontrollkästchen, um die Quittierung des Problems aufzuheben. Dieses Kontrollkästchen ist verfügbar, wenn sich unter den ausgewählten mindestens ein quittiertes Problem befindet.
Convert to cause	Aktivieren Sie das Kontrollkästchen, um das/die Symptomproblem(e) in Ursachenproblem(e) umzuwandeln.
Close problem	Aktivieren Sie das Kontrollkästchen, um das/die ausgewählte(n) Problem(e) manuell zu schließen. Das Kontrollkästchen zum Schließen eines Problems ist verfügbar, wenn die Option *Allow manual close* in der Auslöser-Konfiguration für mindestens eines der ausgewählten Probleme aktiviert ist. Beim Klicken auf *Update* werden nur die Probleme geschlossen, deren Schließen erlaubt ist. Wenn kein Problem manuell geschlossen werden kann, ist das Kontrollkästchen deaktiviert. Bereits geschlossene Probleme werden nicht erneut geschlossen.

Anzeige Basierend auf Bestätigungsinformationen kann konfiguriert werden, wie die Anzahl der Probleme im Dashboard oder auf Karten angezeigt wird. Dazu müssen Sie im *Problem display*-Parameter eine Auswahl treffen, der sowohl in der Kartenkonfiguration als auch im Dashboard-Widget *Problems by severity* verfügbar ist. Es ist möglich, die Gesamtzahl aller Probleme, die Anzahl unbestätigter Probleme getrennt von der Gesamtzahl oder nur die Anzahl unbestätigter Probleme anzuzeigen.

Basierend auf Informationen zu Problemaktualisierungen (Bestätigung usw.) können Aktualisierungsvorgänge konfiguriert werden – das Senden einer Nachricht oder das Ausführen von Remote-Befehlen.

1 Problemunterdrückung

Überblick

Die Problemunterdrückung bietet eine Möglichkeit, ein Problem vorübergehend auszublenden, das später behandelt werden kann. Dies ist nützlich, um die Problemliste zu bereinigen und den dringendsten Problemen die höchste Priorität zuzuweisen. Beispielsweise kann manchmal am Wochenende ein Problem auftreten, das nicht dringend genug ist, um sofort behandelt zu werden, sodass es "verschoben" werden kann bis Montagmorgen.

Die Problemunterdrückung ermöglicht es, ein *einzelnes* Problem zu verbergen, im Gegensatz zur Problemunterdrückung durch Hostwartung, bei der alle Probleme des Wartungshosts verborgen werden.

Vorgänge für Auslöseraktionen werden für unterdrückte Probleme auf die gleiche Weise angehalten wie bei der Hostwartung.

Konfiguration

Ein Problem kann über das Fenster **Problemaktualisierung** unterdrückt werden, wobei die Unterdrückung eine der Optionen für die Problemaktualisierung ist, zusammen mit Kommentieren, Ändern des Schweregrads, Bestätigen usw.

Ein Problem kann auch über dasselbe Fenster zur Problemaktualisierung unterdrückt werden.

Anzeige

Nach der Unterdrückung wird das Problem durch ein blinkendes Unterdrückungssymbol in der Spalte *Info* markiert, bevor es ausgeblendet wird.

Das Unterdrückungssymbol blinkt, solange sich die Unterdrückungsaufgabe in der Warteliste befindet. Sobald der Task-Manager das Problem unterdrückt hat, hört das Symbol auf zu blinken. Wenn das Unterdrückungssymbol längere Zeit blinkt, kann dies auf ein Serverproblem hinweisen, zum Beispiel wenn der Server ausgefallen ist und der Task-Manager die Aufgabe nicht abschließen kann. Das Gleiche gilt für die Aufhebung der Unterdrückung. In der kurzen Zeitspanne, nachdem die Aufgabe übermittelt wurde und der Server sie noch nicht abgeschlossen hat, blinkt das Symbol für die Aufhebung der Unterdrückung.

Ein unterdrücktes Problem kann je nach den Einstellungen des Problemfilters/Widgets sowohl ausgeblendet als auch angezeigt werden.

Wenn ein unterdrücktes Problem in der Problemliste angezeigt wird, ist es durch das Symbol für die Unterdrückung gekennzeichnet und die Details zur Unterdrückung werden beim Darüberfahren mit der Maus angezeigt:

Unterdrückungsdetails werden auch in einem Pop-up-Fenster angezeigt, wenn Sie den Mauszeiger auf das Unterdrückungssymbol in der Spalte *Aktionen*.

12 Konfigurations-Export/-Import

Überblick Die Zabbix-Export/Import-Funktion ermöglicht den Austausch verschiedener Konfigurationen zwischen zwei Zabbix-Systemen.

Typische Anwendungsfälle für diese Funktionalität sind:

- Austausch von Vorlagen oder Netzwerkkarten - Zabbix-Benutzer können ihre Konfigurationsparameter austauschen
- eine Vorlage in Zabbix Community templates hochladen. Dann können andere die Vorlage herunterladen und die Datei in Zabbix importieren.
- Integration mit Drittanbieter-Tools - universelle YAML-, XML- und JSON-Formate ermöglichen die Integration und den Datenimport/-export mit Drittanbieter Tools und Anwendungen

Was kann exportiert/importiert werden?

Objekte, die exportiert/importiert werden können, sind:

- Host Gruppen (*nur über die Zabbix-API*)
- Template Gruppen (*nur über Zabbix API*)
- Templates
- Hosts
- Network maps
- Mediatypen
- Bilder

Exportformat

Die Daten können über das Zabbix-Webfrontend oder die Zabbix API. Unterstützte Exportformate sind YAML, XML und JSON.

Details zum Export

- Alle unterstützten Elemente werden in eine Datei exportiert.
- Von verknüpften Vorlagen geerbte Host- und Vorlagen-Entitäten (Datenpunkte, Auslöser, Graphen, Discovery-Regeln) werden nicht exportiert. Alle Änderungen, die an diesen Entitäten auf Host-Ebene vorgenommen wurden (z. B. geändertes Datenpunkt-Intervall, geänderter regulärer Ausdruck oder hinzugefügte Prototypen zur Low-Level-Discovery-Regel), gehen beim Export verloren; beim Import werden alle Entitäten aus verknüpften Vorlagen wie in der ursprünglich verknüpften Vorlage neu erstellt.
- Durch Low-Level-Discovery erstellte Entitäten sowie alle von ihnen abhängigen Entitäten werden nicht exportiert. Zum Beispiel wird ein für einen durch eine LLD-Regel erzeugten Datenpunkt erstellter Auslöser 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 Importieren von Hosts/Vorlagen mit der Option „Delete missing“ werden Host-/Vorlagen-Makros, die in der Importdatei nicht vorhanden sind, nach dem Import vom Host/von der Vorlage gelöscht.
- Leere Tags für Datenpunkte, Auslöser, Diagramme, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes sind bedeutungslos, d. h. es ist dasselbe, als würden sie fehlen.
- Wenn Entitäten des importierten Hosts/der importierten Vorlage eigene Timeouts konfiguriert haben, werden diese angewendet; andernfalls werden Proxy-/globale Timeouts angewendet.
- Der Import unterstützt YAML, XML und JSON; die Importdatei muss die korrekte Dateierweiterung haben: .yaml und .yml für YAML, .xml für XML und .json für JSON. Siehe Kompatibilitätsinformationen zu 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 Konvertierungsfehler beim Import.

YAML-Basisformat Das YAML-Exportformat enthält die folgenden Knoten:

- Stammknoten für den Zabbix-YAML-Export
- Exportversion

```
zabbix_export:
  version: '8.0'
```

Andere 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:

- Root-Objekt für den Zabbix-JSON-Export
- Exportversion

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Andere Objekte sind von den exportierten Objekten abhängig.

1 Vorlagengruppen

Übersicht

Im Frontend können Vorlagengruppen nur zusammen mit dem Vorlagenexport exportiert werden. Wenn eine Vorlage exportiert wird, werden alle Gruppen, zu denen sie gehört, automatisch mit exportiert.

Die API ermöglicht den Export von Vorlagengruppen unabhängig von Vorlagen.

Exportformat

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Netzwerkgeräte'
```

Exportierte Elemente

Element	Type	Beschreibung
uuid	string	Eindeutige Kennung für diese Vorlagengruppe.
name	string	Gruppenname.

2 Host-Gruppen

Übersicht

Im Frontend können Host-Gruppen nur zusammen mit dem Host-Export exportiert werden. Wenn ein Host exportiert wird, werden alle Gruppen, zu denen er gehört, automatisch mit exportiert.

Die API ermöglicht den Export von Host-Gruppen unabhängig von Hosts.

Exportformat

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix-Server'
```

Exportierte Elemente

Element	Type	Beschreibung
uuid	string	Eindeutige Kennung für diese Host-Gruppe.
name	string	Name der Gruppe.

3 Vorlagen

Übersicht

Vorlagen werden zusammen mit vielen zugehörigen Objekten und Objektbeziehungen exportiert.

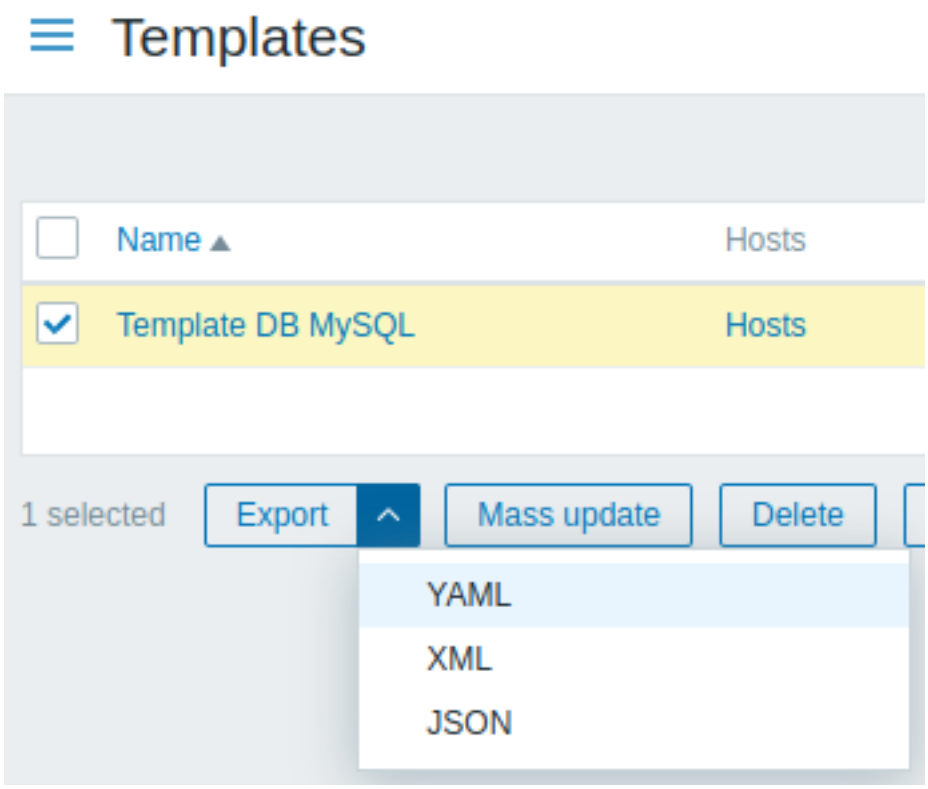
Der Export einer Vorlage enthält:

- Verknüpfte Vorlagengruppen
- Verknüpfte Host-Gruppen (falls in der Konfiguration von Host-Prototypen verwendet)
- Vorlagendaten
- Verknüpfungen zu anderen Vorlagen
- Verknüpfungen zu Vorlagengruppen
- Direkt verknüpfte Datenpunkte
- Direkt verknüpfte Auslöser
- Direkt verknüpfte Diagramme
- Direkt verknüpfte Dashboards
- Direkt verknüpfte Discovery-Regeln mit allen Prototypen
- Direkt verknüpfte Webszenarien
- Wertezuordnungen

Exportieren

Um Vorlagen zu exportieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datensammlung* → *Vorlagen*.
2. Markieren Sie die Kontrollkästchen der zu exportierenden Vorlagen.
3. Klicken Sie unterhalb der Liste auf *Exportieren*.



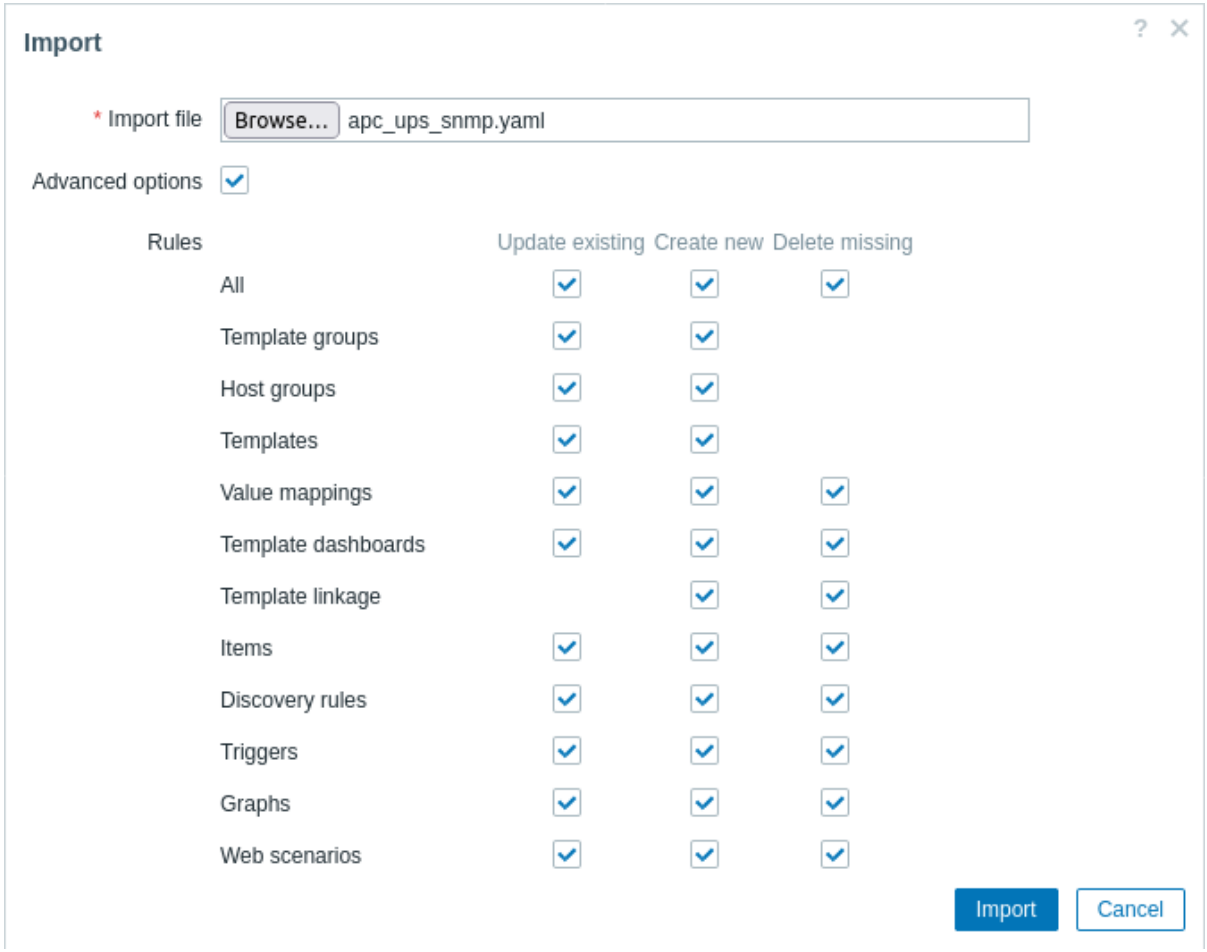
Je nach ausgewähltem Format werden Vorlagen in eine lokale Datei mit einem Standardnamen exportiert:

- `zabbix_export_templates.yaml` - beim YAML-Export (Standardoption für den Export);
- `zabbix_export_templates.xml` - beim XML-Export;
- `zabbix_export_templates.json` - beim JSON-Export.

Importieren

Um Vorlagen zu importieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* → *Vorlagen*.
2. Klicken Sie oben rechts auf *Importieren*.
3. Wählen Sie die Importdatei aus.
4. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.



Wenn Sie das Kontrollkästchen *Erweiterte Optionen* aktivieren, wird eine detaillierte Liste aller importierbaren Elemente angezeigt – markieren oder deaktivieren Sie jede Importregel nach Bedarf.

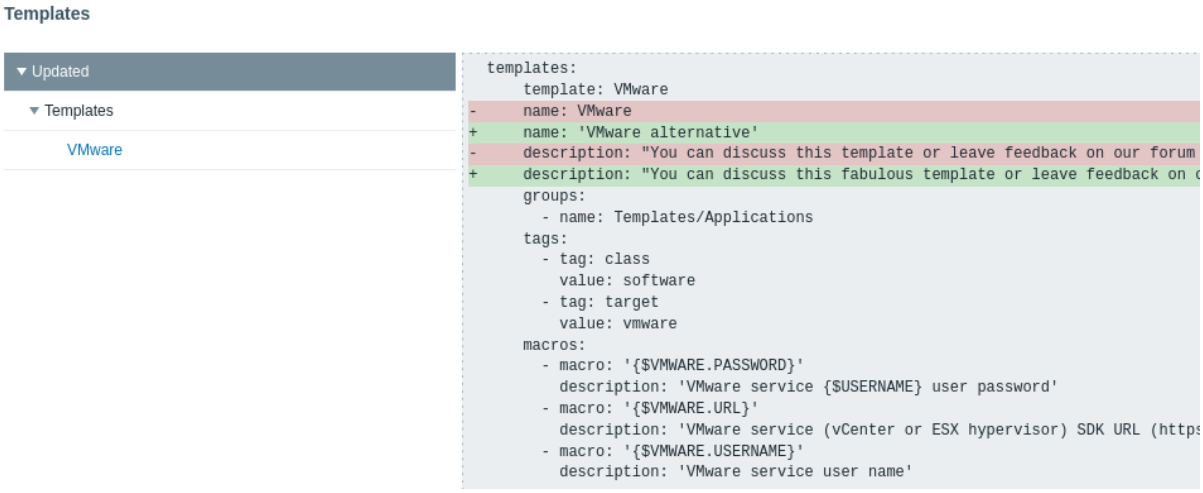
Wenn Sie auf das Kontrollkästchen in der Zeile *Alle* klicken, werden alle darunterliegenden Elemente markiert/deaktiviert.

Importregeln:

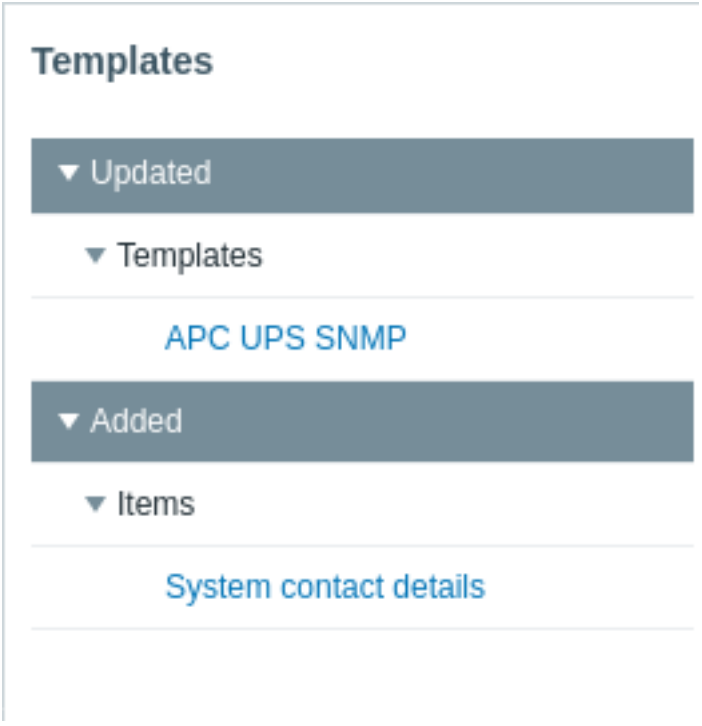
Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene Elemente werden mit Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Neu erstellen	Neue Elemente werden mit Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.
Fehlende löschen	Vorhandene Elemente, die in der Importdatei nicht vorhanden sind, werden entfernt. Andernfalls werden sie nicht entfernt. Wenn *Fehlende löschen* für *Vorlagenverknüpfung* markiert ist, wird die aktuelle Vorlagenverknüpfung, die in der Importdatei nicht vorhanden ist, aufgehoben. Entitäten (Datenpunkte, Auslöser, Graphen usw.), die von den aufgehobenen Verknüpfungen geerbt wurden, werden nicht entfernt (es sei denn, die Option *Fehlende löschen* ist auch für jede Entität ausgewählt).

Attention:
Wenn Vorlagen mit denselben Namen bereits vorhanden sind, sollten beim Importieren die Optionen *Fehlende löschen* aktiviert werden, um einen sauberen Import zu erreichen. Auf diese Weise werden die alten Datenpunkte entfernt, die in der aktualisierten Vorlage nicht mehr vorhanden sind (beachten Sie, dass dies den Verlust der Historie dieser alten Datenpunkte bedeutet).

Auf dem nächsten Bildschirm können Sie den Inhalt einer zu importierenden Vorlage anzeigen. Wenn es sich um eine neue Vorlage handelt, werden alle Elemente grün aufgelistet. Wenn eine vorhandene Vorlage aktualisiert wird, werden neue Vorlagenelemente grün hervorgehoben; entfernte Vorlagenelemente werden rot hervorgehoben; Elemente, die sich nicht geändert haben, werden auf grauem Hintergrund aufgelistet.



Das Menü auf der linken Seite kann verwendet werden, um durch die Liste der Änderungen zu navigieren. Der Abschnitt *Aktualisiert* hebt alle Änderungen hervor, die an vorhandenen Vorlagenelementen vorgenommen wurden. Der Abschnitt *Hinzugefügt* listet neue Vorlagenelemente auf. Die Elemente in jedem Abschnitt sind nach Elementtyp gruppiert; klicken Sie auf den grauen Pfeil, um die Elementgruppe ein- oder auszuklappen.



Überprüfen Sie die Änderungen an der Vorlage und klicken Sie dann auf *Importieren*, um den Vorlagenimport durchzuführen. Im Frontend wird eine Erfolgs- oder Fehlermeldung zum Import angezeigt.

Exportformat Wenn eine Vorlage exportiert wird, erzeugt 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.

In den folgenden Abschnitten wird jedes Element im Exportformat beschrieben. In den Beispielen wird die Vorlage Linux by Zabbix agent verwendet.

Eine Auslassung (`...`) kennzeichnet Elemente, die der Kürze halber ausgelassen wurden. Der Hinweis (`see table below`) 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)
```

Metadaten der Vorlage

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 Auslöser-Elemente der Vorlage, die mit den Auslösern von Vorlagen-Datenpunkten identisch sind.
graphs		Wurzelelement für Diagramm-Elemente der Vorlage, die mit den Host-Diagrammen identisch sind.

```
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: 'Templates/Operating systems'
      items: (siehe Tabelle unten)
      discovery_rules: (siehe Tabelle unten)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Zeitüberschreitung, nach der der Agent als nicht verfügbar betrachtet wird. Funkti
          config: (siehe Tabelle unten)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Kritischer Schwellenwert der CPU-Auslastung, ausgedrückt in %.'
          config: (siehe Tabelle unten)
        (...)
      dashboards: (siehe Tabelle unten)
      valuemaps: (siehe Tabelle unten)
    (...)
```

Vorlagen

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für diese Vorlage.
template	string	(erforderlich) Eindeutiger Vorlagenname.
name	string	Sichtbarer Vorlagenname.
description	text	Beschreibung der Vorlage.
wizard_ready	text	Gibt an, ob die Vorlage im Host Wizard zur Auswahl verfügbar ist. Mögliche Werte:[1] NO (0, Standard), YES (1). Siehe auch: Template object (`wizard_ready`).
readme	text	Vorlagenspezifische Konfigurationsanweisungen zur Anzeige im Host Wizard. Unterstützt Markdown-Formatierung.
vendor		Stammelement für den Vorlagenanbieter (vorhanden, wenn die exportierte Vorlage Anbieterdaten enthält).
name	string	(erforderlich) Name des Vorlagenanbieters.
version	string	(erforderlich) Version der Vorlage. Bei out-of-the-box templates 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 Benutzermakros der Vorlage.
macro	string	(erforderlich) Name des Benutzermakros.
type	string	Typ des Benutzermakros. Mögliche Werte:[1] TEXT (0, Standard), SECRET_TEXT (1), VAULT (2). Siehe auch: User macro object (`type`).
value	string	Wert des Benutzermakros.
description	string	Beschreibung des Benutzermakros.
config		Stammelement für die Konfiguration von Vorlagenmakros, die dafür verantwortlich ist, wie das Makro im Host Wizard angezeigt wird.
dashboards		Stammelement für Vorlagen-Dashboards.
valuemaps		Stammelement für Wertzuordnungen der Vorlage.

Vorlagen-Datenpunkte

```
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 `nc
          valuemap:
            name: 'Zabbix-Agent-Ping-Status'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'Anzahl installierter 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	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für diesen Datenpunkt.
name	string	(erforderlich) Name des Datenpunkts.
type	string	Typ des Datenpunkts. 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).
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 von Zeitsuffixen, Benutzermakros oder LLD-Makros), wie lange die Verlaufsdaten gespeichert werden sollen. Standard: 31d.
trends	string	Zeitraum (unter Verwendung von Zeitsuffixen, Benutzermakros oder LLD-Makros), wie lange die 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	Liste von durch Kommas getrennten 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; das 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	Type	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 sollte auch `password` (siehe unten) angegeben werden, oder beide Elemente sollten 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 sollte auch `username` (siehe oben) angegeben werden, oder beide Elemente sollten 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 inventory finden Sie die unterstützten Felder).
valuemap		Wurzelelement für Datenpunkt-Wertzuordnungen.
name	string	(erforderlich) Name der Wertzuordnung, die für den Datenpunkt verwendet werden soll.
logtimefmt	string	Format der Zeit in Protokolleinträgen. Unterstützt für Datenpunkte vom Werttyp LOG.
preprocessing		Wurzelelement für die Vorverarbeitung von Datenpunktwerten.
step		Wurzelelement für Vorverarbeitungsschritte von Vorlagen-Datenpunktwerten.
jmx_endpoint	string	JMX-Endpunkt. Unterstützt für JMX-Datenpunkte.
master_item		(erforderlich für DEPENDENT-Datenpunkte) Wurzelelement für den Master-Datenpunkt eines abhängigen Datenpunkts.
key	string	(erforderlich) Schlüssel des Master-Datenpunkts des abhängigen Datenpunkts.
timeout	string	Zeitüberschreitung für die Abfrageanforderung von Datenpunktdaten. Unterstützt für die Liste der Datenpunkttypen unter Timeouts.
url	string	(erforderlich für HTTP_AGENT-Datenpunkte) URL-Zeichenfolge.
query_fields		Wurzelelement 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		Wurzelelement 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	HTTP(S)-Request-Body-Daten. Unterstützt für HTTP_AGENT-Datenpunkte.
status_codes	string	Bereiche erforderlicher HTTP-Statuscodes, durch Kommas getrennt. Unterstützt für HTTP_AGENT-Datenpunkte.
follow_redirects	string	Antwort-Weiterleitungen bei der Datenabfrage folgen. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0), YES (1, Standard).
post_type	string	Typ des Post-Data-Bodys. 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		Wurzelelement für HTTP(S)-Request-Header. Unterstützt für HTTP_AGENT-Datenpunkte.
name	string	(erforderlich für HTTP_AGENT-Datenpunkte) Header-Name.
value	string	(erforderlich für HTTP_AGENT-Datenpunkte) Header-Wert.
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 Request-Methode. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] GET (0, Standard), POST (1), PUT (2), HEAD (3).

Element	Type	Beschreibung
output_format	string	Wie die Antwort verarbeitet werden soll. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] RAW (0, Standard), JSON (1).
allow_traps	string	Erlaubt das Befüllen des Werts ähnlich wie bei einem Trapper-Datenpunkt. 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 Datei mit dem SSL-Schlüssel. Unterstützt für HTTP_AGENT-Datenpunkte.
verify_peer	string	Gibt an, ob geprüft 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 geprüft 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).
tags		Wurzelelement für Datenpunkt-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
triggers		Wurzelelement für Auslöser von Vorlagen-Datenpunkten.

Note:
Siehe auch: Item object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Vorlagen-Datenpunkt-Wertvorverarbeitungsschritte

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Element	Type	Description
type	string	(erforderlich) Der Typ des Datenpunkt-Wertvorverarbeitungsschritts. 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).

Element	Type	Description
parameters		(erforderlich) Wurzelelement für Parameter des Datenpunkt-Wertvorverarbeitungsschritts.
parameter	string	Einzelner Parameter des Datenpunkt-Wertvorverarbeitungsschritts.
error_handler	string	Aktionstyp, der im Fall eines Fehlers im Vorverarbeitungsschritt verwendet wird. Mögliche Werte:[1] ORIGINAL_ERROR (0, Standard), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parameter für die Fehlerbehandlung.

Note:
Siehe auch: Item preprocessing object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Auslöser von Vorlagen-Datenpunkten

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Number of installed packages has been changed'
              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)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: High memory utilization'
              event_name: 'Linux: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)'
              priority: AVERAGE
              description: 'The system is running out of free memory.'
              dependencies:
                - name: 'Linux: Lack of available memory'
                  expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
              tags:
                - tag: scope
                  value: capacity
                - tag: scope
                  value: performance
        (...)
```

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für diesen Auslöser.
expression	string	(erforderlich) Auslöser-Ausdruck.
recovery_mode	string	Grundlage für die Erzeugung 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.

Element	Type	Beschreibung
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	Auslöserstatus. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
priority	string	Auslöser-Schweregrad. 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 Ereigniserzeugung (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 (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Regeln für Low-Level-Discovery in Vorlagen

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Erkennung von Netzwerkschnittstellen'
          key: net.if.discovery
          delay: 1h
          filter: (siehe Tabelle unten)
          description: 'Die Erkennung von Netzwerkschnittstellen.'
          item_prototypes: (siehe Tabelle unten)
          trigger_prototypes: (siehe Tabelle unten)
          graph_prototypes: (siehe Tabelle unten)
        (...)
```

Attention:
Die meisten Elemente von Regeln für Low-Level-Discovery in Vorlagen sind dieselben wie bei Vorlagen-Datenpunkten. Die folgende Tabelle beschreibt die Elemente, die sich von Vorlagen-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		Wurzelelement für Filter von Regeln für Low-Level-Discovery in Vorlagen.
lifetime	string	Zeitraum (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro), nach dem nicht mehr erkannte Ressourcen gelöscht werden. Standard: 7d.

Element	Type	Beschreibung
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		Wurzelelement für Elemente von Datenpunktprototypen in Vorlagen, die dieselben sind wie bei Vorlagen-Datenpunkten.
trigger_prototypes		Wurzelelement für Elemente von Auslöserprototypen in Vorlagen, die dieselben sind wie bei Auslösern von Vorlagen-Datenpunkten.
graph_prototypes		Wurzelelement für Elemente von Graphprototypen in Vorlagen, die dieselben sind wie bei Host-Graphen.
host_prototypes		Wurzelelement für Elemente von Host-Prototypen in Vorlagen, die dieselben sind wie bei Hosts.
parent_discovery_rule		Wurzelelement für die übergeordnete Low-Level-Discovery-Regel (oder den Regelprototyp) des Low-Level-Discovery-Regelprototyps. Diese Eigenschaft kennzeichnet ihn als LLD-Regelprototyp, direktes untergeordnetes Element der referenzierten Regel/des referenzierten Regelprototyps.
key	string	(erforderlich) Der Schlüssel der übergeordneten Low-Level-Discovery-Regel (oder des Regelprototyps).
master_item	string	(erforderlich für DEPENDENT-Regeln) Wurzelelement für den Master-Datenpunkt der abhängigen Regel.
lld_macro_paths		Wurzelelement für Makropfade der Low-Level-Discovery-Regel.
lld_macro	string	(erforderlich) Makroname der Low-Level-Discovery-Regel.
path	string	(erforderlich) Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird.
preprocessing		Wurzelelement für die Vorverarbeitung von Werten der Low-Level-Discovery-Regel.
step		Wurzelelement für Elemente von Vorverarbeitungsschritten für Werte der Low-Level-Discovery-Regel, die dieselben sind wie bei Vorverarbeitungsschritten für Werte von Vorlagen-Datenpunkten, jedoch mit weniger möglichen Werten. Siehe auch: LLD-Regel-Vorverarbeitungsobjekt.
overrides		Wurzelelement für Überschreibungsregeln der Low-Level-Discovery-Regel.
name	string	(erforderlich) Eindeutiger Name der Überschreibung.
step	string	(erforderlich) Eindeutige Reihenfolgenummer der Überschreibung.
stop	string	Verarbeitung der nächsten Überschreibungen stoppen, wenn Übereinstimmung vorliegt.
filter		Wurzelelement für Filterelemente von Überschreibungsregeln für Regeln für Low-Level-Discovery in Vorlagen, die dieselben sind wie bei Filtern von Regeln für Low-Level-Discovery in Vorlagen.
operations		Wurzelelement für Überschreibungsoperationen von Regeln für Low-Level-Discovery in Vorlagen.

Note:
Siehe auch: LLD-Regelobjekt (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Filter für Low-Level-Discovery-Regeln von Vorlagen

```
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		Wurzelelement für Filterbedingungen.
macro	string	(erforderlich) Der Name des Low-Level-Discovery-Makros, für das die Prüfung durchgeführt werden soll.
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 aus dem benutzerdefinierten Ausdruck auf eine 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.

Note:
Siehe auch: LLD rule filter object (beziehen Sie sich auf die entsprechende Eigenschaft mit übereinstimmendem Namen).

Operationen zum Überschreiben von Regeln der Low-Level-Discovery von Vorlagen

Element	Type	Beschreibung
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 Zeichenkette für den Operator der Überschreibungsbedingung.
status	string	Status des Objekts bei der Überschreibungsoperation.
discover	string	Gibt an, ob das Objekt als Ergebnis der Discovery hinzugefügt wird.
delay	string	Aktualisierungsintervall, das für den Datenpunkt-Prototyp bei der Überschreibungsoperation festgelegt wird.
history	string	Verlaufsaufbewahrungszeitraum, der für den Datenpunkt-Prototyp bei der Überschreibungsoperation festgelegt wird.
trends	string	Trend-Aufbewahrungszeitraum, der für den Datenpunkt-Prototyp bei der Überschreibungsoperation festgelegt wird.
severity	string	Schweregrad des Auslöser-Prototyps, der bei der Überschreibungsoperation festgelegt wird.
tags		Stammelement für die Tags, die für das Objekt bei der Überschreibungsoperation festgelegt werden.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
templates		Stammelement für die Vorlagen, die mit dem Host-Prototyp bei der Überschreibungsoperation verknüpft werden.

Element	Type	Beschreibung
name	string	(erforderlich) Name der Vorlage.
inventory_mode	string	Inventarmodus des Host-Prototyps, der bei der Überschreibungsoperation festgelegt wird.

Note:
Siehe auch: LLD rule override operation object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Vorlagen-Webszenarien

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für dieses Webszenario.
name	string	(erforderlich) Name des Webszenarios.
delay	string	Häufigkeit der Ausführung des Webszenarios (unter Verwendung von Sekunden, Zeitsuffixen oder Benutzermakros). Standard: 1m.
attempts	integer	Die Anzahl der Versuche zur Ausführung der Schritte des Webszenarios. Mögliche Werte:[1] 1-10 (Standard: 1).
agent	string	Client-Agent. Zabbix gibt sich als der ausgewählte Browser aus. 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 den Szenarioschritten verwendet werden können.
name	string	(erforderlich) Variablenname.
value	text	(erforderlich) Variablenwert.
headers		Stammelement für HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Header sollten mit derselben Syntax aufgeführt werden, wie sie im HTTP-Protokoll erscheinen würden.
name	string	(erforderlich) Header-Name.
value	text	(erforderlich) Header-Wert.
status	string	Status des Webszenarios. Mögliche Werte:[1] AKTIVIERT (0, Standard), DEAKTIVIERT (1).
authentication	string	Authentifizierungsmethode. Mögliche Werte:[1] NONE (0, Standard), BASIC (1), NTLM (2), Kerberos (3) oder Digest (4).
http_user	string	Benutzername für die Authentifizierung mit BASIC (HTTP), NTLM, Kerberos oder Digest.
http_password	string	Passwort für die Authentifizierung mit BASIC (HTTP), NTLM, Kerberos oder Digest.
verify_peer	string	Überprüfen des SSL-Zertifikats des Webservers. Mögliche Werte:[1] NO (0, Standard), YES (1).
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-Datei mit dem privaten Schlüssel, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).
ssl_key_password	string	Passwort der SSL-Datei mit dem privaten Schlüssel.
steps		(erforderlich) Stammelement für Schritte des Vorlagen-Webszenarios.
tags		Stammelement für Webszenario-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.

Note:
Siehe auch: Webszenario-Objekt (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Schritte des Webszenarios der Vorlage

Element	Type	Beschreibung
name	string	(erforderlich) Name des Webszenario-Schritts.
url	string	(erforderlich) URL für die Überwachung.
query_fields		Stammelement für Abfrageparameter (ein Array von HTTP-Feldern, die bei der Ausführung einer Anfrage zur URL hinzugefügt werden).
name	string	(erforderlich) Name des Abfrageparameters.
value	string	Wert des Abfrageparameters.
posts		Stammelement für HTTP-POST-Variablen (eine Zeichenkette (Rohdaten für POST) oder ein Array von HTTP-Feldern (Formularfelddaten)).
name	string	(erforderlich) Name des POST-Felds.
value	string	(erforderlich) Wert des POST-Felds.
variables		Stammelement von Variablen (Makros) auf Schritt-Ebene, die nach diesem Schritt angewendet werden sollen. Wenn der Variablenwert das Präfix 'regex:' hat, wird sein Wert aus den von diesem Schritt zurückgegebenen Daten entsprechend dem regulären Ausdrucksmuster nach dem Präfix 'regex:' extrahiert
name	string	(erforderlich) Name der Variablen.
value	text	(erforderlich) Wert der Variablen.
headers		Stammelement für HTTP-Header, die bei der Ausführung einer Anfrage gesendet werden.
name	string	(erforderlich) Name des Headers.
value	text	(erforderlich) 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	Timeout (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro) für die Schrittausführung. 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: Web scenario step object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Konfiguration von Vorlagenmakros

```
zabbix_export:
  (...)
  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: Schwellenwerte
            label: 'Schwellenwert der CPU-Auslastung, ausgedrückt'
            description: 'Kritischer Schwellenwert der CPU-Auslastung, ausgedrückt in %. Im Bereich von 0
            regex: '^-?([0-9]+|([0-9]+)\.([0-9]+))$'
        (...)
        - macro: '{$IFCONTROL}'
```

```
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Schnittstellensteuerung'
            description: 'Einen Auslöser auslösen, wenn sich der Betriebsstatus der Schnittstelle auf "Lin
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Element	Type	Beschreibung
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	Beschriftung des einklappbaren Abschnitts, in dem das Makro gruppiert ist.
label	string	(erforderlich für TEXT-, LIST- und CHECKBOX-Makros) Makrobeschriftung.
description	text	Makro-Hilfetext. 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-Einträge oder CHECKBOX-Werte.
value	string	(erforderlich für LIST-Makros) Wert des LIST-Eintrags.
text	string	(erforderlich für LIST-Makros) Text des LIST-Eintrags.
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: Macro configuration object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Vorlagen-Dashboards

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (siehe Tabelle unten)
        (...)
```

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für dieses Dashboard.
name	string	(erforderlich) Name des Vorlagen-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 Vorlagen-Dashboards.
name	string	Seitenname.

Element	Type	Beschreibung
display period	integer	Anzeigedauer der Seite.
sortorder	integer	Sortierreihenfolge der Seite.
widgets		Stammelement für Widgets des Vorlagen-Dashboards.

Note:
Siehe auch: Template dashboard object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Widgets des Vorlagen-Dashboards

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                    - type: STRING
                      name: reference
                      value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Beschreibung
type	string	(erforderlich) Widget-Typ.
name	string	Widget-Name.
x	integer	Horizontale Position von der linken Seite des Vorlagen-Dashboards aus. Mögliche Werte:[1] 0-71.
y	integer	Vertikale Position von der Oberseite des Vorlagen-Dashboards aus. Mögliche Werte:[1] 0-63.

Element	Type	Beschreibung
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 Widget-Felder des Vorlagen-Dashboards.
type	string	(erforderlich) Widget-Feldtyp. 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) Widget-Feldname.
value	mixed	(erforderlich) Widget-Feldwert, abhängig vom Feldtyp.

Note:
Siehe auch: Template dashboard widget object (siehe die entsprechende Property mit übereinstimmendem 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		Stammelement für Zuordnungen.
type	string	Abgleichstyp der Zuordnung. 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, auf den der ursprüngliche Wert abgebildet wird.

Note:
Siehe auch: Value map object (beziehen Sie sich auf die entsprechende Eigenschaft mit einem übereinstimmenden Namen).

Fußnoten [1] API-Ganzzahlwerte 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.

4 Hosts

Übersicht

Hosts werden exportiert zusammen mit vielen zugehörigen Objekten und Objektbeziehungen.

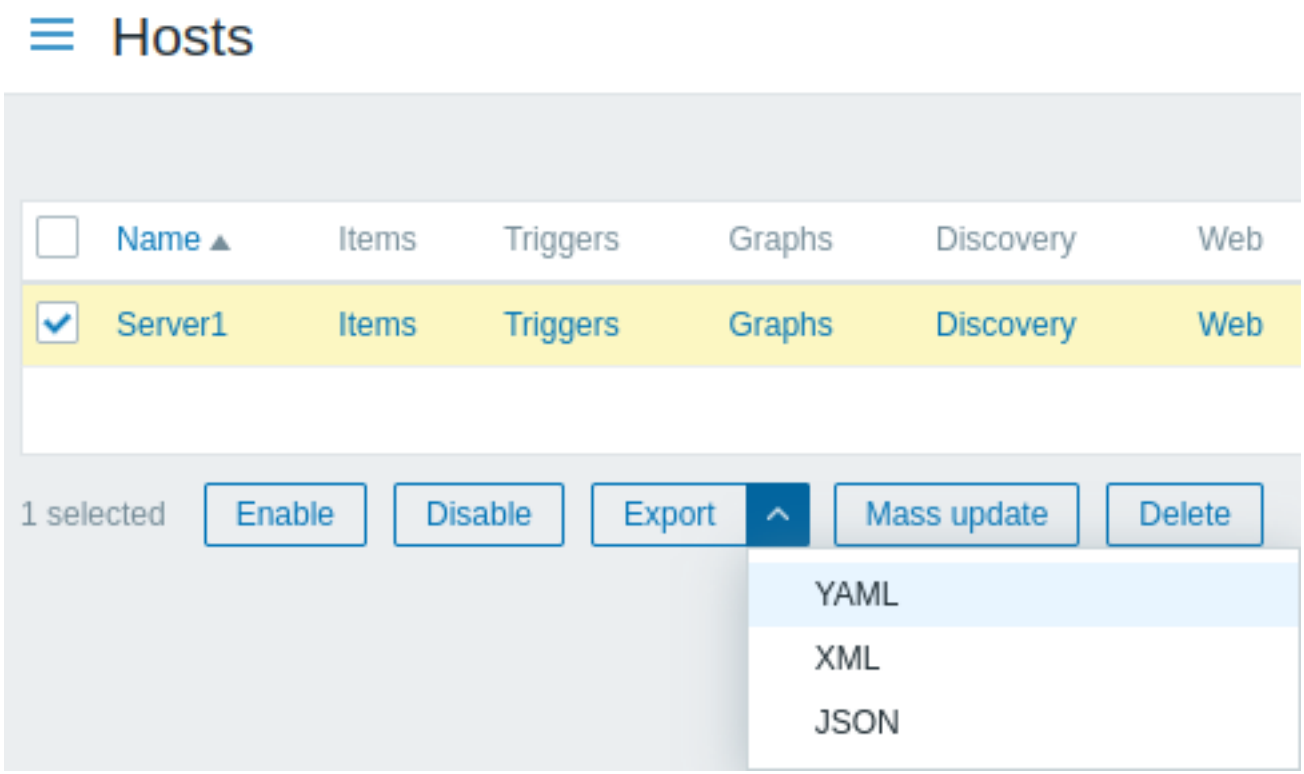
Der Host-Export enthält:

- Verknüpfte Host-Gruppen
- Host-Daten
- Vorlagen-Verknüpfung
- Host-Gruppen-Verknüpfung
- Host-Schnittstellen
- Direkt verknüpfte Datenpunkte
- Direkt verknüpfte Auslöser
- Direkt verknüpfte Discovery-Regeln mit allen Prototypen
- Direkt verknüpfte Webszenarien
- Host-Makros
- Host-Inventardaten
- Wertezuordnungen
- Verknüpfte Graphen

Exportieren

Um Hosts zu exportieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* → *Hosts*.
2. Aktivieren Sie die Kontrollkästchen der zu exportierenden Hosts.
3. Klicken Sie unterhalb der Liste auf *Exportieren*.



Je nach ausgewähltem Format werden Hosts in eine lokale Datei mit einem Standardnamen exportiert:

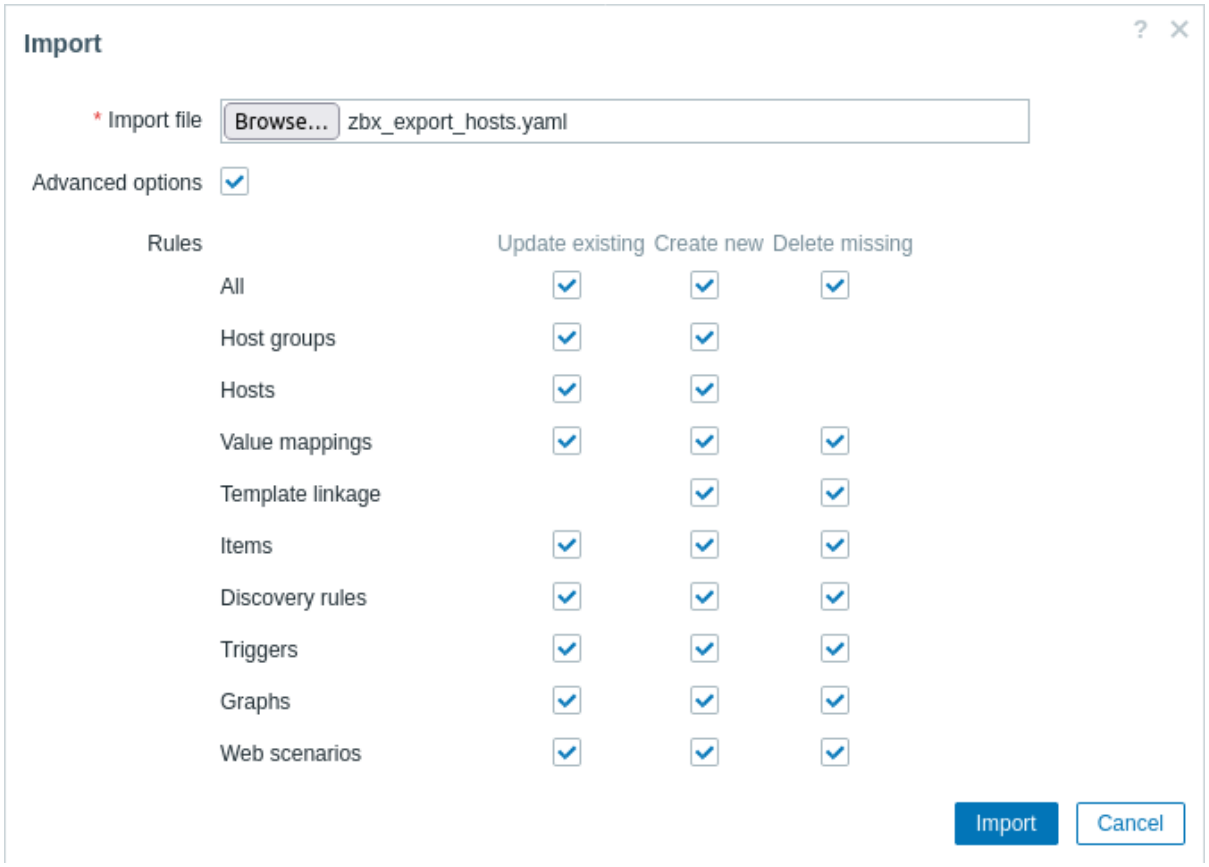
- `zabbix_export_hosts.yaml` - beim YAML-Export (Standardoption für den Export);
- `zabbix_export_hosts.xml` - beim XML-Export;
- `zabbix_export_hosts.json` - beim JSON-Export.

Importieren

Um Hosts zu importieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* → *Hosts*.

2. Klicken Sie oben rechts auf *Importieren*.
3. Wählen Sie die Importdatei aus.
4. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.



Wenn Sie das Kontrollkästchen *Erweiterte Optionen* aktivieren, wird eine detaillierte Liste aller importierbaren Elemente angezeigt – markieren oder deaktivieren Sie jede Importregel nach Bedarf.

Wenn Sie auf das Kontrollkästchen in der Zeile *Alle* klicken, werden alle darunterliegenden Elemente markiert/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 in der Importdatei nicht enthalten sind, werden entfernt. Andernfalls werden sie nicht entfernt. Wenn *Fehlende löschen* für *Vorlagenverknüpfung* markiert ist, wird die aktuelle Vorlagenverknüpfung, die in der Importdatei nicht vorhanden ist, aufgehoben. Entitäten (Datenpunkte, Auslöser, Diagramme usw.), die von den getrennten Vorlagen geerbt wurden, werden nicht entfernt (es sei denn, die Option *Fehlende löschen* ist auch für jede Entität ausgewählt).

Im Frontend wird eine Erfolgs- oder Fehlermeldung des Imports angezeigt.

Exportformat

Wenn ein Host exportiert wird, erzeugt Zabbix ein strukturiertes YAML-, JSON- oder XML-Format. Der Export umfasst Host-Elemente wie Host-Metadaten, Datenpunkte, Makros, Auslöser und mehr.

Jedes Element erfüllt einen bestimmten Zweck und kann verschachtelte Elemente enthalten.

In den folgenden Abschnitten wird jedes Element im Exportformat beschrieben. In den Beispielen wird ein Host mit der Vorlage MySQL by Zabbix agent 2 verwendet. Zu Demonstrationszwecken für die Elemente wird die Vorlage nach der Erstellung des Hosts getrennt. Durch das Trennen bleiben alle Elemente in der Konfiguration erhalten (anders als beim Trennen und Löschen, wodurch sie entfernt werden).

Eine Auslassung (`...`) kennzeichnet Elemente, die der Kürze halber weggelassen wurden. 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	Beschreibung
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.

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (siehe Tabelle unten)
      items: (siehe Tabelle unten)
      discovery_rules: (siehe Tabelle unten)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'Passwort des MySQL-Benutzers.'
        (...)
      valuemaps: (siehe Tabelle unten)
  (...)
```

Hosts

Element	Type	Beschreibung
host	string	(erforderlich) Eindeutiger Host-Name.
name	string	Sichtbarer Host-Name.
description	text	Host-Beschreibung.
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 Proxy (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.

Element	Type	Beschreibung
status	string	Host-Status. 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).
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) Name der Vorlage.
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 Host-Inventar.
<inventory_property>	string	Inventareigenschaft. Alle Eigenschaften haben ihr jeweiliges Element (`type`, `name`, `os` usw.; siehe zum Beispiel Export format).
inventory_mode	string	Inventarmodus. Mögliche Werte:[1] DISABLED (-1), MANUAL (0, Standard), AUTOMATIC (1).
valuemaps		Stammelement für Host-Wertzuordnungen.

Note:
Siehe auch: Host object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host-Schnittstellen

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          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 für die Verbindung zum Host die IP-Adresse als Schnittstelle verwendet werden soll (andernfalls wird DNS verwendet). Mögliche Werte:[1] NO (0), YES (1, Standard).
ip	string	(für IP-Verbindungen erforderlich) IP-Adresse (IPv4 oder IPv6).
dns	string	(für DNS-Verbindungen erforderlich) DNS-Name.
port	string	Portnummer.

Element	Type	Beschreibung
details		Stammelement für Schnittstellendetails.
version	string	Diese SNMP-Version verwenden. Mögliche Werte:[1] SNMPV1 (1), SNMP_V2C (2, Standard), SNMP_V3 (3).
community	string	(für SNMPv1- und SNMPv2-Datenpunkte erforderlich) SNMP-Community.
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 zur Verwendung in Datenpunkten (Format: `if<N>`).

Note:
Siehe auch: Host interface object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host-Datenpunkte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog-Cache-Festplattennutzung'
          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-Auslastung'
          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 zur Gesamtzahl der Seiten im Buffer Pool.'
        tags:
          - tag: component
            value: memory
        triggers: (siehe Tabelle unten)
      (...)
      - name: Laufzeit
        type: DEPENDENT
        key: mysql.uptime
        units: uptime
        description: 'Anzahl der Sekunden, die der Server bereits läuft.'
        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	Type	Beschreibung
name	string	(erforderlich) Name des Datenpunkts.
type	string	Typ des Datenpunkts. 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).
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 von Zeitsuffixen, Benutzermakros oder LLD-Makros), wie lange die Verlaufsdaten gespeichert werden sollen. Standard: 31d.
trends	string	Zeitraum (unter Verwendung von Zeitsuffixen, Benutzermakros oder LLD-Makros), wie lange die 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	Liste von durch Kommas getrennten 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; das Skript für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte).
ipmi_sensor	string	IPMI-Sensor. Unterstützt für IPMI-Datenpunkte.

Element	Type	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 sollte auch `password` (siehe unten) angegeben werden, oder beide Elemente sollten 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 sollte auch `username` (siehe oben) angegeben werden, oder beide Elemente sollten 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. (siehe Host-Inventar für unterstützte Felder).
valuemap		Wurzelelement für Datenpunkt-Wertzuordnungen.
name	string	(erforderlich) Name der Wertzuordnung, die für den Datenpunkt verwendet werden soll.
logtimefmt	string	Format der Zeit in Logeinträgen. Unterstützt für Datenpunkte vom Werttyp LOG.
preprocessing		Wurzelelement für die Datenpunkt-Wertvorverarbeitung.
step		Wurzelelement 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) Wurzelelement für den Master-Datenpunkt eines abhängigen Datenpunkts.
key	string	(erforderlich) Schlüssel des Master-Datenpunkts eines abhängigen Datenpunkts.
timeout	string	Timeout für die Datenabfrage des Datenpunkts. Unterstützt für die Liste der Datenpunkttypen unter Timeouts.
url	string	(erforderlich für HTTP_AGENT-Datenpunkte) URL-Zeichenfolge.
query_fields		Wurzelelement 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		Wurzelelement 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	HTTP(S)-Request-Body-Daten. Unterstützt für HTTP_AGENT-Datenpunkte.
status_codes	string	Bereiche erforderlicher HTTP-Statuscodes, durch Kommas getrennt. Unterstützt für HTTP_AGENT-Datenpunkte.
follow_redirects	string	Antwort-Weiterleitungen bei der Datenabfrage folgen. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0), YES (1, Standard).
post_type	string	Typ des Post-Data-Bodys. 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		Wurzelelement für HTTP(S)-Request-Header. Unterstützt für HTTP_AGENT-Datenpunkte.
name	string	(erforderlich für HTTP_AGENT-Datenpunkte) Header-Name.
value	string	(erforderlich für HTTP_AGENT-Datenpunkte) Header-Wert.

Element	Type	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 Request-Methode. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] GET (0, Standard), POST (1), PUT (2), HEAD (3).
output_format	string	Wie die Antwort verarbeitet werden soll. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] RAW (0, Standard), JSON (1).
allow_traps	string	Erlaubt das Befüllen des Werts ähnlich wie bei einem Trapper-Datenpunkt. 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 Datei mit dem SSL-Schlüssel. Unterstützt für HTTP_AGENT-Datenpunkte.
verify_peer	string	Gibt an, ob validiert 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 validiert 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).
tags		Wurzelelement für Datenpunkt-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
triggers		Wurzelelement für Host-Datenpunkt-Auslöser.

Note:
Siehe auch: Item object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Vorverarbeitungsschritte für Datenpunktwerte des Host

```
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 bei einem Fehler im Vorverarbeitungsschritt verwendet wird. Mögliche Werte:[1] ORIGINAL_ERROR (0, Standard), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parameter für die Fehlerbehandlung.

Note:
Siehe auch: Item preprocessing object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host-Datenpunkt-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 has not received any data for items for the last 30 minutes.'
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für diesen Auslöser.
expression	string	(erforderlich) Auslöserausdruck.

Element	Type	Beschreibung
recovery_mode	string	Grundlage für die Erzeugung 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	Auslöserstatus. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
priority	string	Auslöserschweregrad. 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 Ereigniserzeugung (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 (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host-Regeln für Low-Level-Discovery

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (siehe Tabelle unten)
          description: 'Durchsuchen von Datenbanken im DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Größe der Datenbank {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Datenbankgröße.'
              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-Regeln für Low-Level-Discovery 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-Regeln für Low-Level-Discovery.
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 Elemente von Host-Datenpunktprototypen, die dieselben sind wie bei Host-Datenpunkten.
trigger_prototypes		Stammelement für Elemente von Host-Auslöserprototypen, 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 Host-Prototypen, die dieselben sind wie bei Hosts.
parent_discovery_rule		Stammelement für die übergeordnete Low-Level-Discovery-Regel (oder den Regelprototyp) des Low-Level-Discovery-Regelprototyps. Diese Eigenschaft kennzeichnet ihn als LLD-Regelprototyp, direktes untergeordnetes Element der referenzierten Regel bzw. des referenzierten Regelprototyps.
key	string	(erforderlich) Der Schlüssel der übergeordneten Low-Level-Discovery-Regel (oder des Regelprototyps).
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) Makroname der Low-Level-Discovery-Regel.
path	string	(erforderlich) Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird.
preprocessing		Stammelement für die Wertvorverarbeitung von Low-Level-Discovery-Regeln.
step		Stammelement für Elemente von Schritten der Wertvorverarbeitung von Low-Level-Discovery-Regeln, die dieselben sind wie bei Schritten der Wertvorverarbeitung von Host-Datenpunkten, jedoch mit weniger möglichen Werten. Siehe auch: LLD-Regel-Vorverarbeitungsobjekt.
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	Verarbeitung der nächsten Überschreibungen stoppen, wenn Übereinstimmung vorliegt.
filter		Stammelement für Filterelemente von Überschreibungsregeln für Low-Level-Discovery-Regeln, die dieselben sind wie bei Filtern von Host-Regeln für Low-Level-Discovery.

Element	Type	Beschreibung
operations		Stammelement für Überschreibungsoperationen von Host-Regeln für Low-Level-Discovery.

Note:
Siehe auch: LLD-Regelobjekt (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Filter für Host-Regeln der Low-Level-Discovery

```
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	Methode zur Auswertung von 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, für das die Prüfung durchgeführt werden soll.
value	string	Zu vergleichender Wert.
operator	string	Bedingungsoperator. Mögliche Werte:[1] MATCHES_REGEX (8, Standard), NOT_MATCHES_REGEX (9).
formulaid	string	(erforderlich) Beliebige eindeutige ID, die verwendet wird, um aus dem benutzerdefinierten Ausdruck auf eine 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.

Note:
Siehe auch: LLD rule filter object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Überschreibungsoperationen für Host-Regeln der Low-Level-Discovery

Element	Type	Beschreibung
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).
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 Zeichenkette für den Operator der Überschreibungsbedingung.
status	string	Status des Objekts bei der Überschreibungsoperation.
discover	string	Gibt an, ob das Objekt als Ergebnis der Discovery hinzugefügt wird.

Element	Type	Beschreibung
delay	string	Für den Datenpunkt-Prototyp bei der Überschreibungsoperation festgelegtes Aktualisierungsintervall.
history	string	Für den Datenpunkt-Prototyp bei der Überschreibungsoperation festgelegter Verlaufsspeicherzeitraum.
trends	string	Für den Datenpunkt-Prototyp bei der Überschreibungsoperation festgelegter Trendspeicherzeitraum.
severity	string	Bei der Überschreibungsoperation festgelegter Schweregrad des Auslöser-Prototyps.
tags		Stammelement für die Tags, die für das Objekt bei der Überschreibungsoperation festgelegt werden.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
templates		Stammelement für die Vorlagen, die bei der Überschreibungsoperation mit dem Host-Prototyp verknüpft werden.
name	string	(erforderlich) Name der Vorlage.
inventory_mode	string	Bei der Überschreibungsoperation festgelegter Inventarmodus des Host-Prototyps.

Note:
Siehe auch: LLD rule override operation object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host-Webszenarien

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für dieses Webszenario.
name	string	(erforderlich) Name des Webszenarios.
delay	string	Häufigkeit der Ausführung des Webszenarios (unter Verwendung von Sekunden, Zeitsuffixen oder Benutzermakros). Standard: 1m.
attempts	integer	Die Anzahl der Versuche zur Ausführung der Webszenario-Schritte. Mögliche Werte: 1-10 (Standard: 1).
agent	string	Client-Agent. Zabbix gibt sich als der ausgewählte Browser aus. 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 Szenario-Schritten verwendet werden können.
name	string	(erforderlich) Variablenname.
value	text	(erforderlich) Variablenwert.
headers		Stammelement für HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Header sollten mit derselben Syntax aufgeführt werden, wie sie im HTTP-Protokoll erscheinen würden.
name	string	(erforderlich) Header-Name.
value	text	(erforderlich) Header-Wert.
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 für die BASIC-(HTTP)-, NTLM-, Kerberos- oder Digest-Authentifizierung.
http_password	string	Passwort für die BASIC-(HTTP)-, NTLM-, Kerberos- oder Digest-Authentifizierung.
verify_peer	string	Das SSL-Zertifikat des Webservers überprüfen. Mögliche Werte:[1] NO (0, Standard), YES (1).
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-Private-Key-Datei, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).

Element	Type	Beschreibung
ssl_key_password	string	Passwort der SSL-Private-Key-Datei.
steps		(erforderlich) Stammelement für Host-Webszenario-Schritte.
tags		Stammelement für Webszenario-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.

Note:
Siehe auch: Web scenario object (siehe die entsprechende Property mit übereinstimmendem Namen).

Schritte des Host-Webszenarios

Element	Typ	Beschreibung
name	string	(erforderlich) Name des Webszenario-Schritts.
url	string	(erforderlich) URL für die Überwachung.
query_fields		Stammelement für Abfrageparameter (ein Array von HTTP-Feldern, die beim Ausführen einer Anfrage zur URL hinzugefügt werden).
name	string	(erforderlich) Name des Abfrageparameters.
value	string	Wert des Abfrageparameters.
posts		Stammelement für HTTP-POST-Variablen (eine Zeichenkette (Rohdaten des POST) oder ein Array von HTTP-Feldern (Formularfelddaten)).
name	string	(erforderlich) Name des POST-Felds.
value	string	(erforderlich) Wert des POST-Felds.
variables		Stammelement der Variablen (Makros) auf Schritt-Ebene, die nach diesem Schritt angewendet werden sollen. Wenn der Variablenwert das Präfix 'regex:' hat, wird sein Wert aus den von diesem Schritt zurückgegebenen Daten entsprechend dem regulären Ausdrucksmuster nach dem Präfix 'regex:' extrahiert
name	string	(erforderlich) Name der Variablen.
value	text	(erforderlich) Wert der Variablen.
headers		Stammelement für HTTP-Header, die beim Ausführen einer Anfrage gesendet werden.
name	string	(erforderlich) Name des Headers.
value	text	(erforderlich) 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	Timeout (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro) für die Schrittausführung. 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 für Webszenario-Schritte (siehe die entsprechende Property mit übereinstimmendem Namen).

Host-Diagramme

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandbreite'
      graph_items: (siehe Tabelle unten)
    (...)
```

Element	Type	Beschreibung
uuid	string	Eindeutige Kennung des Diagramms.
name	string	(erforderlich) Diagrammname.

Element	Type	Beschreibung
width	integer	Diagrammbreite in Pixeln. Wird für die Vorschau und für Kreis-/Explosionsdiagramme verwendet. Mögliche Werte:[1] 20-65535 (Standard: 900).
height	integer	Diagrammhöhe in Pixeln. Wird für die Vorschau und für Kreis-/Explosionsdiagramme verwendet. Mögliche Werte:[1] 20-65535 (Standard: 900).
yaxismin	double	Minimalwert der Y-Achse. Unterstützt für den FIXED-Mindestwert der Y-Achse. Standard: 0.
yaxismax	double	Maximalwert der Y-Achse. Unterstützt für den FIXED-Mindestwert 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	Mindestwert 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 individuelle Datenpunktdetails.
host	string	(erforderlich) Datenpunkt-Host.
key	string	(erforderlich) Datenpunktschlüssel.
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 individuelle Datenpunktdetails.
host	string	(erforderlich) Datenpunkt-Host.
key	string	(erforderlich) Datenpunktschlüssel.
graph_items		(erforderlich) Wurzelelement für Host-Diagrammelemente.

Note:
Siehe auch: Graph object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host-Grafik-Datenpunkte

```
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	Reihenfolge der Darstellung. Der kleinere Wert wird zuerst gezeichnet. Kann verwendet werden, um Linien oder Bereiche hinter (oder vor) einem anderen zu zeichnen.
drawtype		string	Zeichenstil des Grafik-Datenpunkts. Unterstützt für NORMAL-Grafiken. Mögliche Werte:[1] SINGLE_LINE (0, Standard), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color		string	Elementfarbe (6 Symbole, hexadezimal).
yaxisside		string	Seite der Grafik, auf der die Y-Skala des Grafik-Datenpunkts gezeichnet wird. Unterstützt für NORMAL- und STACKED-Grafiken.
calc_fnc		string	Zu zeichnende Daten, 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 Grafiken), LAST (9, unterstützt für Kreis-/Explosionsgrafiken).
type		string	Typ des Grafik-Datenpunkts. Mögliche Werte:[1] SIMPLE (0, Standard), GRAPH_SUM (2; der Wert des Datenpunkts stellt das gesamte Kreisdiagramm dar; unterstützt für Kreis-/Explosionsgrafiken).
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 übereinstimmendem Namen).

Host-Wertzuordnungen

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element		Type	Beschreibung
uuid		string	(erforderlich) Eindeutige Kennung für diese Wertzuordnung.
name		string	(erforderlich) Name der Wertzuordnung.
mapping			Stammelement für Zuordnungen.
	type	string	Abgleichstyp der Zuordnung. 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 einem übereinstimmenden Namen).

Fußnoten [1] API-Ganzzahlwerte 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.

5 Netzwerkkarten

Übersicht

Der Export von Netzwerkkarten enthält:

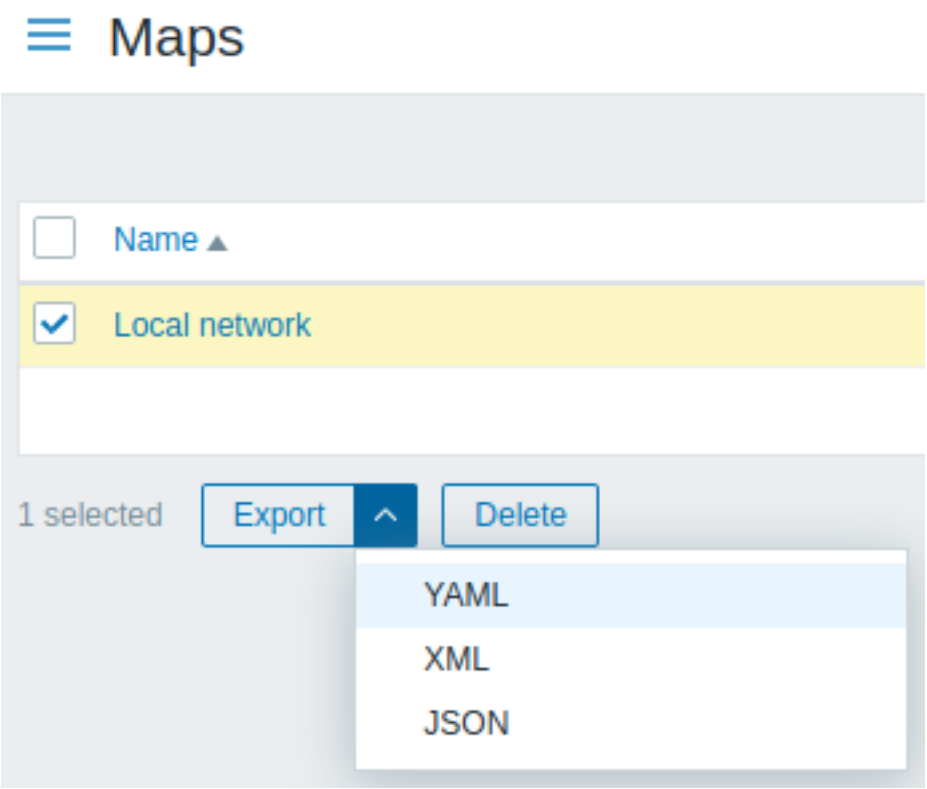
- Alle zugehörigen Bilder
- Die Kartenstruktur (alle Karteneinstellungen, alle enthaltenen Elemente mit ihren Einstellungen, Kartenverknüpfungen und Statusindikatoren für Kartenverknüpfungen)

Warning:
Alle Host-Gruppen, Hosts, Auslöser, anderen Karten oder sonstigen Elemente, die mit der exportierten Karte in Verbindung stehen könnten, werden nicht exportiert. Wenn daher mindestens eines der Elemente, auf die sich die Karte bezieht, fehlt, schlägt der Import fehl.

Exportieren

Um Netzwerkkarten zu exportieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Monitoring* → *Maps*.
2. Aktivieren Sie die Kontrollkästchen der zu exportierenden Netzwerkkarten.
3. Klicken Sie unterhalb der Liste auf *Export*.



Je nach ausgewähltem Format werden Karten in eine lokale Datei mit einem Standardnamen exportiert:

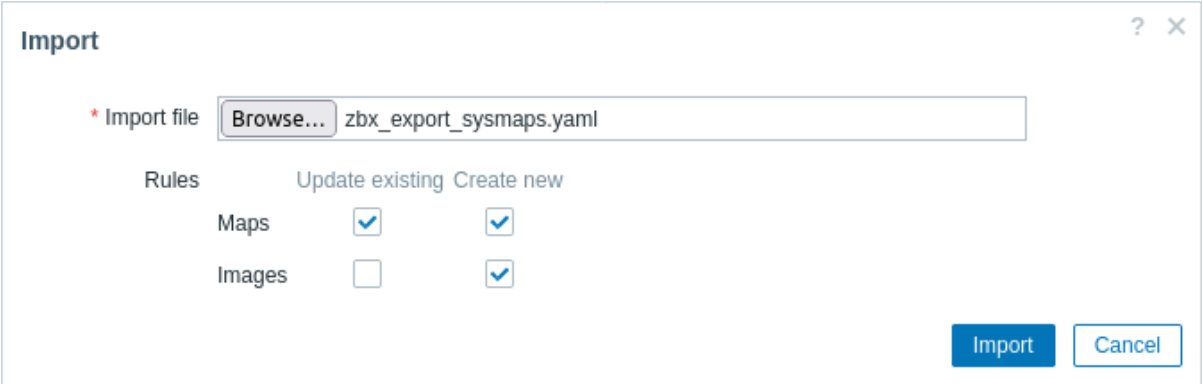
- `zabbix_export_maps.yaml` - beim YAML-Export (Standardoption für den Export);
- `zabbix_export_maps.xml` - beim XML-Export;
- `zabbix_export_maps.json` - beim JSON-Export.

Importieren

Um Netzwerkkarten zu importieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Monitoring* → *Maps*.
2. Klicken Sie oben rechts auf *Import*.
3. Wählen Sie die Importdatei aus.

4. Markieren Sie die erforderlichen Optionen in den Importregeln.
5. Klicken Sie unten rechts im Konfigurationsformular auf *Import*.



Importregeln:

Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene Karten werden mithilfe der Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Neue erstellen	Neue Karten werden mithilfe der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.

Wenn Sie beide Kartenoptionen deaktivieren und die entsprechenden Optionen für Bilder aktivieren, werden nur Bilder importiert. Der Bildimport ist nur für Benutzer mit der Rolle *Super admin* verfügbar.

Im Frontend wird eine Erfolgs- oder Fehlermeldung zum Import angezeigt.

Warning:
Wenn ein vorhandenes Bild ersetzt wird, wirkt sich dies auf alle Karten 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	Beschreibung
images		Stammelement für Bilder.
name	string	Eindeutiger Bildname.
imagetype	integer	Bildtyp. Mögliche Werte: 1 - Bild; 2 - Hintergrund.
encodedImage	string	Base64-kodiertes Bild.

Element	Type	Beschreibung
maps		Stammelement für Karten.

Karten

Element	Typ	Beschreibung
name	string	Eindeutiger Kartenname.
width	integer	Kartenbreite in Pixeln.
height	integer	Kartenhöhe in Pixeln.
label_type	integer	Typ der Beschriftung von Kartenelementen. Mögliche Werte: 0 - Beschriftung; 1 - Host-IP-Adresse; 2 - Elementname; 3 - Nur Status; 4 - Nichts.
label_location	integer	Standardposition der Beschriftung von Kartenelementen. Mögliche Werte: 0 - Unten; 1 - Links; 2 - Rechts; 3 - Oben.
highlight	integer	Aktiviert die Hervorhebung von Symbolen für aktive Auslöser und Host-Status. Mögliche Werte: 0 - Nein; 1 - Ja.
expandproblem	integer	Zeigt den Problem-Auslöser für Elemente mit einem einzelnen Problem an. Mögliche Werte: 0 - Nein; 1 - Ja.
markelements	integer	Hebt Kartenelemente hervor, deren Status sich kürzlich geändert hat. Mögliche Werte: 0 - Nein; 1 - Ja.
show_unack	integer	Problemanzeige. Mögliche Werte: 0 - Anzahl aller Probleme; 1 - Anzahl nicht bestätigter Probleme; 2 - Anzahl bestätigter und nicht bestätigter Probleme getrennt.
severity_min	integer	Minimale Auslöser-Wichtigkeit, die standardmäßig auf der Karte angezeigt wird. Mögliche Werte: 0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe.
show_suppressed	integer	Zeigt Probleme an, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden. Mögliche Werte: 0 - Nein; 1 - Ja.
grid_size	integer	Zellengröße eines Kartenrasters in Pixeln. Unterstützt, wenn `grid_show` auf 0 gesetzt ist. Mögliche Werte: 20, 40, 50, 75 oder 100.
grid_show	integer	Zeigt ein Raster in der Kartenkonfiguration an. Mögliche Werte: 0 - Ja; 1 - Nein.

Element	Typ	Beschreibung
grid_align	integer	Richtet Symbole in der Kartenkonfiguration automatisch aus. Mögliche Werte: 0 - Ja; 1 - Nein.
label_format	integer	Verwendet die erweiterte Beschriftungskonfiguration. Mögliche Werte: 0 - Nein; 1 - Ja.
label_type_host	integer	Zeigt die Beschriftung als Host-Beschriftung an. 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	Zeigt die Beschriftung als Hostgruppen-Beschriftung an. Unterstützt, wenn `label_format` auf 1 gesetzt ist. Mögliche Werte: 0 - Beschriftung; 2 - Elementname; 3 - Nur Status; 4 - Nichts; 5 - Benutzerdefinierte Beschriftung.
label_type_trigger	integer	Zeigt die Beschriftung als Auslöser-Beschriftung an. 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	Zeigt die Beschriftung als Kartenbeschriftung an. 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	Zeigt die Beschriftung als Bildbeschriftung an. 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. Unterstützt, wenn `label_type_host` auf 5 gesetzt ist.
label_string_hostgroup	string	Benutzerdefinierte Beschriftung für Hostgruppen-Elemente. Unterstützt, wenn `label_type_hostgroup` auf 5 gesetzt ist.
label_string_trigger	string	Benutzerdefinierte Beschriftung für Auslöser-Elemente. Unterstützt, wenn `label_type_trigger` auf 5 gesetzt ist.
label_string_map	string	Benutzerdefinierte Beschriftung für Kartenelemente. Unterstützt, wenn `label_type_map` auf 5 gesetzt ist.
label_string_image	string	Benutzerdefinierte Beschriftung für Bildelemente. Unterstützt, wenn `label_type_image` auf 5 gesetzt ist.
expand_macros	integer	Erweitert Makros in Beschriftungen in der Kartenkonfiguration. Mögliche Werte: 0 - Nein; 1 - Ja.

Element	Typ	Beschreibung
background		Stammelement für das Hintergrundbild (falls vorhanden). Unterstützt, wenn `imagetype` auf 2 gesetzt ist.
name	string	Name des Hintergrundbildes.
iconmap		Stammelement für die Symbolzuordnung (falls vorhanden).
name	string	Name der Symbolzuordnung.
urls		Stammelement 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		Stammelement für Karten-selements.
shapes		Stammelement für Kartenformen.
type	integer	Formtyp. Mögliche Werte: 0 - Rechteck; 1 - Ellipse.
x	integer	X-Koordinaten der Form in Pixeln.
y	integer	Y-Koordinaten der Form in Pixeln.
width	integer	Breite der Form.
height	integer	Höhe der Form.
text	string	Text innerhalb der Form.
font	integer	Schriftstil 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, dargestellt als Hexadezimalcode.
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 - Mittig; 1 - Oben; 2 - Unten.
border_type	integer	Typ des Rahmens für die Form. Mögliche Werte: 0 - Keiner; 1 - Fette Linie; 2 - Gepunktete Linie; 3 - Gestrichelte Linie.
border_width	integer	Breite des Rahmens in Pixeln.

Element		Typ	Beschreibung
	border_color	string	Rahmenfarbe, dargestellt als Hexadezimalcode.
	background_color	string	Hintergrundfarbe (Füllfarbe), dargestellt als Hexadezimalcode.
	zindex	integer	Wert für die Anordnung aller Formen und Linien (z-index).
lines			Stammelement für Kartenlinien.
	x1	integer	X-Koordinaten des ersten Linienpunkts in Pixeln.
	y1	integer	Y-Koordinaten des ersten Linienpunkts in Pixeln.
	x2	integer	X-Koordinaten des zweiten Linienpunkts in Pixeln.
	y2	integer	Y-Koordinaten des zweiten Linienpunkts in Pixeln.
	line_type	integer	Linientyp. Mögliche Werte: 0 - Keiner; 1 - Fette Linie; 2 - Gepunktete Linie; 3 - Gestrichelte Linie.
	line_width	integer	Linienbreite in Pixeln.
	line_color	string	Linienfarbe, dargestellt als Hexadezimalcode.
	zindex	integer	Wert für die Anordnung aller Formen und Linien (z-index).
links			Stammelement für Verknüpfungen zwischen Kartenelementen.
	drawtype	integer	Verknüpfungsstil. Mögliche Werte: 0 - Linie; 2 - Fette Linie; 3 - Gepunktete Linie; 4 - Gestrichelte Linie.
	color	string	Verknüpfungsfarbe (6 Zeichen, hex).
	label	string	Verknüpfungsbeschriftung.
	selementid1	id	ID eines zu verbindenden Elements.
	selementid2	id	ID des anderen zu verbindenden Elements.
linktriggers			Stammelement für Verknüpfungsstatusindikatoren.

Note:
Siehe auch: Map object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Karten-Selemente

Element	Typ	Beschreibung
elementtype	integer	Typ des Kartenelements. Mögliche Werte: 0 - Host; 1 - Karte; 2 - Auslöser; 3 - Host-Gruppe; 4 - Bild.
elements		Wurzelelement für Zabbix-Entitäten (Host, Host-Gruppe, Karte usw.), die auf der Karte dargestellt werden. Alle Entitäten haben ihr jeweiliges Element (`host` usw.; siehe zum Beispiel Exportformat).
label	string	Beschriftung des Symbols.
label_location	integer	Position der Beschriftung. Mögliche Werte: -1 - Standard der Karte verwenden; 0 - Unten; 1 - Links; 2 - Rechts; 3 - Oben.
x	integer	Position auf der X-Achse.
y	integer	Position auf der Y-Achse.

Element	Typ	Beschreibung
elementsubtype	integer	Untertyp des Elements. Unterstützt, wenn `elementtype` auf 3 gesetzt ist. Mögliche Werte: 0 - Einzelne Host-Gruppe; 1 - Alle Host-Gruppen.
areatype	integer	Bereichsgröße. Unterstützt, wenn `elementtype` auf 1 gesetzt ist. Mögliche Werte: 0 - Wie die gesamte Karte; 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 für die Platzierung im Bereich. Unterstützt, wenn `elementsubtype` auf 1 gesetzt ist. Mögliche Werte: 0 - Gleichmäßig im Bereich platzieren.
use_iconmap	integer	Symbolzuordnung für dieses Element verwenden. Nur relevant, wenn die Symbolzuordnung auf Kartenebene 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 Status `OK` befindet.
name	string	Eindeutiger Bildname.
icon_on		Wurzelelement für das Bild, das verwendet wird, wenn sich das Element im Status `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 Karten oder von jedem einzelnen Kartenelement 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-Gruppen-Elemente). Wenn Tags angegeben sind, werden auf der Karte nur Probleme mit diesen Tags angezeigt.
tag	string	Tag-Name.
value	string	Tag-Wert.
operator	integer	Operator.

Note:
Siehe auch: Map element object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Statusindikatoren für Kartenverknüpfungen

Element	Type	Beschreibung
drawtype	integer	Verknüpfungsstil, wenn sich der Auslöser im Zustand „Problem“ befindet. Mögliche Werte: 0 - Linie; 2 - Fette Linie; 3 - Gepunktete Linie; 4 - Gestrichelte Linie.

Element	Type	Beschreibung
color	string	Verknüpfungsfarbe (6 Zeichen, hexadezimal), wenn sich der Auslöser im Zustand „Problem“ befindet.
trigger		Stammelement für den Auslöser, der zur Anzeige des Verknüpfungsstatus verwendet wird.
description	string	Name des Auslösers.
expression	string	Auslöserausdruck.
recovery_expression	string	Wiederherstellungsausdruck des Auslösers.

Note:
Siehe auch: Map link trigger object (beziehen Sie sich auf die entsprechende Eigenschaft mit demselben Namen).

6 Medientypen

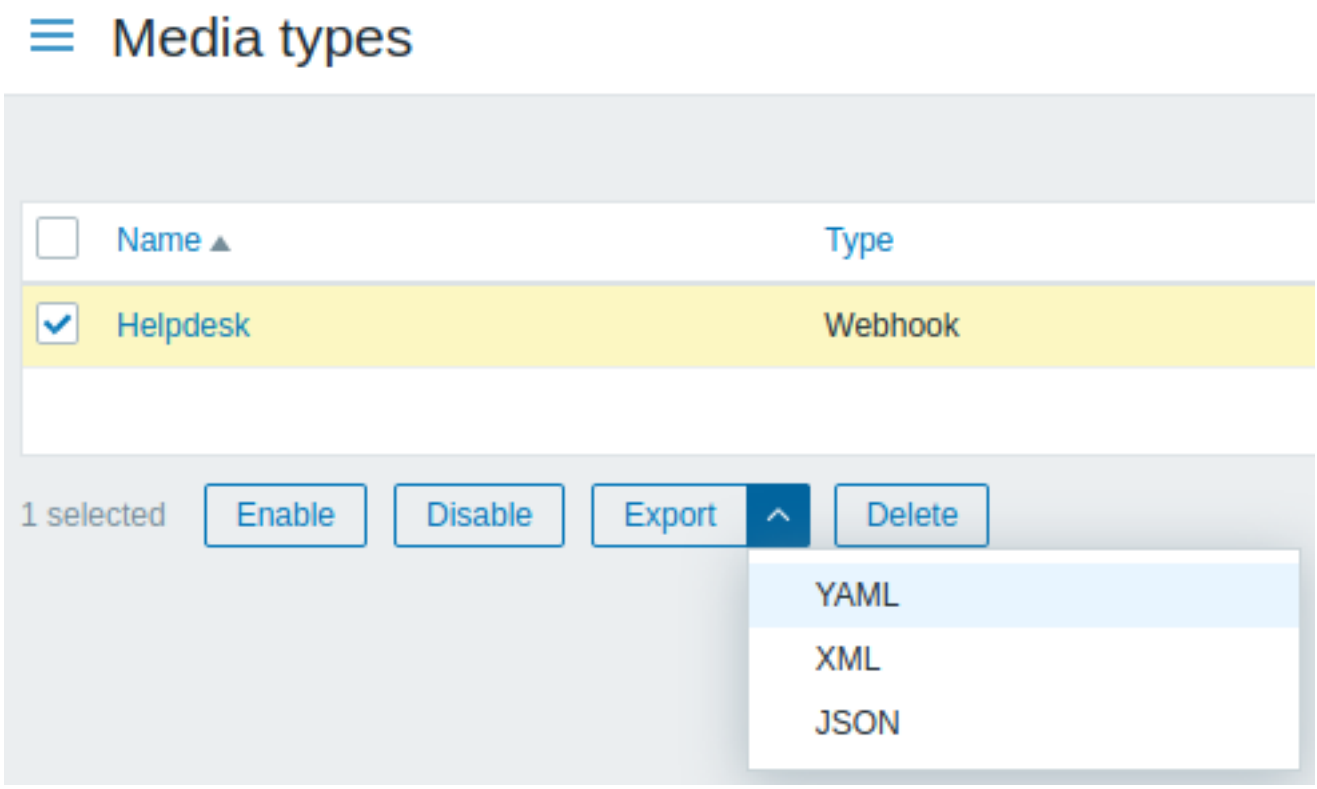
Übersicht

Medientypen werden zusammen mit allen zugehörigen Objekten und Objektbeziehungen exportiert.

Exportieren

Gehen Sie wie folgt vor, um Medientypen zu exportieren:

1. Gehen Sie zu *Benachrichtigungen* → *Medientypen*.
2. Aktivieren Sie die Kontrollkästchen der zu exportierenden Medientypen.
3. Klicken Sie unterhalb der Liste auf *Exportieren*.



Je nach ausgewähltem Format werden Medientypen mit einem Standardnamen in eine lokale Datei exportiert:

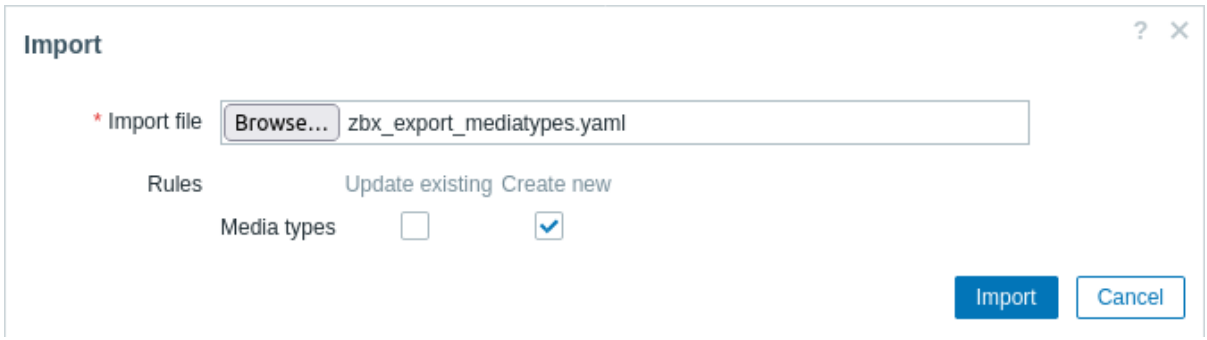
- `zabbix_export_mediatypes.yaml` - beim YAML-Export (Standardoption für den Export);
- `zabbix_export_mediatypes.xml` - beim XML-Export;
- `zabbix_export_mediatypes.json` - beim JSON-Export.

Importieren

Um Medientypen zu importieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Benachrichtigungen* → *Medientypen*.

2. Klicken Sie oben rechts auf *Importieren*.
3. Wählen Sie die Importdatei aus.
4. Markieren Sie die erforderlichen Optionen in den Importregeln.
5. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.



Importregeln:

Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene Elemente werden mithilfe der Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Neue erstellen	Neue Elemente werden mithilfe der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.

Im Frontend wird eine Erfolgs- oder Fehlermeldung zum Import 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 "prio
    }

    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 ] Sende Anfrage: ' + params.endpoint + '\n' + data);

request.addHeader('Content-Type: application/json');
response = request.post(params.endpoint, data);

Zabbix.log(4, '[ Pushover Webhook ] Antwort mit Statuscode ' + request.getStatus() + ' erhalte

if (response !== null) {
    try {
        response = JSON.parse(response);
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Die von Pushover erhaltene Antwort konnte nicht ge
        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 'Unbekannter Fehler. Prüfen Sie das Debug-Log für weitere Informationen.';
    }
}
```

```
        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover-Benachrichtigung fehlgeschlagen: ' + error);
        throw 'Pushover-Benachrichtigung fehlgeschlagen: ' + error;
    }
description: |
  Bitte beachten Sie die Einrichtungsanleitung hier: https://git.zabbix.com/projects/ZBX/repos/zabbi

  Setzen Sie den Parameter token auf Ihren Pushover-Anwendungsschlüssel.
  Wenn Sie dem Zabbix-Benutzer Pushover-Medien zuweisen, fügen Sie den Benutzerschlüssel in das Feld
message_templates:
  - event_source: TRIGGERS
    operation_mode: PROBLEM
    subject: 'Problem: {EVENT.NAME}'
    message: |
      Problem begann um {EVENT.TIME} am {EVENT.DATE}
      Problemname: {EVENT.NAME}
      Host: {HOST.NAME}
      Schweregrad: {EVENT.SEVERITY}
      Betriebsdaten: {EVENT.OPDATA}
      Ursprüngliche Problem-ID: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: RECOVERY
    subject: 'Gelöst in {EVENT.DURATION}: {EVENT.NAME}'
    message: |
      Das Problem wurde um {EVENT.RECOVERY.TIME} am {EVENT.RECOVERY.DATE} behoben
      Problemname: {EVENT.NAME}
      Problemdauer: {EVENT.DURATION}
      Host: {HOST.NAME}
      Schweregrad: {EVENT.SEVERITY}
      Ursprüngliche Problem-ID: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: UPDATE
    subject: 'Aktualisiertes Problem in {EVENT.AGE}: {EVENT.NAME}'
    message: |
      {USER.FULLNAME} hat das Problem am {EVENT.UPDATE.DATE} um {EVENT.UPDATE.TIME} {EVENT.UPDATE.AC
      {EVENT.UPDATE.MESSAGE}

      Der aktuelle Problemstatus ist {EVENT.STATUS}, das Alter ist {EVENT.AGE}, bestätigt: {EVENT.AC
  - event_source: DISCOVERY
    operation_mode: PROBLEM
    subject: 'Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
    message: |
      Discovery-Regel: {DISCOVERY.RULE.NAME}

      Geräte-IP: {DISCOVERY.DEVICE.IPADDRESS}
      Geräte-DNS: {DISCOVERY.DEVICE.DNS}
      Gerätestatus: {DISCOVERY.DEVICE.STATUS}
      Geräte-Uptime: {DISCOVERY.DEVICE.UPTIME}

      Name des Geräteservices: {DISCOVERY.SERVICE.NAME}
      Port des Geräteservices: {DISCOVERY.SERVICE.PORT}
      Status des Geräteservices: {DISCOVERY.SERVICE.STATUS}
      Uptime des Geräteservices: {DISCOVERY.SERVICE.UPTIME}
  - event_source: AUTOREGISTRATION
    operation_mode: PROBLEM
    subject: 'Autoregistrierung: {HOST.HOST}'
    message: |
      Host-Name: {HOST.HOST}
```

```
            Host-IP: {HOST.IP}
            Agent-Port: {HOST.PORT}
```

Exportierte Elemente

Exportierte Elemente werden in der folgenden Tabelle erläutert.

Element	Type	Beschreibung
name	string	(erforderlich) Name des Medientyps.
type	string	(erforderlich) Für den Medientyp verwendeter Transport. Mögliche Werte:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Gibt an, 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 (unter Verwendung von Sekunden oder Zeitsuffixen) 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).
operation_mode	string	Betriebsmodus. Mögliche Werte:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Betreff der Nachricht.
message	string	Nachrichtentext.

Note:
Siehe auch: Media type object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

E-Mail

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *E-Mail* exportiert.

Element	Type	Beschreibung
provider	string	E-Mail-Anbieter.
smtp_server	string	SMTP-Server.
smtp_port	integer	Port des SMTP-Servers, zu dem die Verbindung hergestellt wird. Standard: 25.
smtp_helo	string	SMTP helo.
smtp_email	string	E-Mail-Adresse, von der Benachrichtigungen gesendet werden.
smtp_security	string	Zu verwendende Sicherheitsstufe für die 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-Peer-Prüfung 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	Authentifizierungspasswort.
redirection_url	string	URL des Zabbix Frontend, zu der nach der 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 Aktualisierungstoken.
token_url	string	OAuth-URL zum Austausch des Autorisierungstokens gegen Zugriffs- und Aktualisierungstoken.

Element	Type	Beschreibung
message_format	string	Nachrichtenformat. Mögliche Werte:[1] TEXT (0), HTML (1, Standard).

Note:
Siehe auch: Media type object (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

SMS

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *SMS* exportiert.

Element	Type	Beschreibung
gsm_modem	string	(erforderlich) Name des seriellen Geräts des GSM-Modems.

Note:
Siehe auch: Media type object (bezieht sich auf die entsprechende Eigenschaft mit demselben Namen).

Skript

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *Skript* exportiert.

Element	Type	Beschreibung
Skriptname	string	(erforderlich) Name des Skripts.
Parameter		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: Medientyp-Objekt (siehe die entsprechende Eigenschaft mit demselben Namen).

webhook

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *webhook* exportiert.

Element	Type	Beschreibung
script	string	Skript.
timeout	string	Timeout-Intervall für HTTP-Anfragen 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 Ereignismenüeintrags. Unterstützt das Makro {EVENT.TAGS.*}.
event_menu_name	string	Name des Ereignismenüeintrags. 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 (siehe die entsprechende Eigenschaft mit übereinstimmendem Namen).

Fußnoten

[1] API-Ganzzahlwerte 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.

13 Discovery

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte im Abschnitt „Discovery“ zuzugreifen.

1 Netzwerk-Erkennung

Überblick

Zabbix bietet eine automatische Netzwerkerkennungsfunktion, 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 Administrationsaufwand verwenden

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 (es wird nur der unverschlüsselte Modus unterstützt)
- Informationen, die von einem SNMP-Agent empfangen werden

Sie bietet NICHT:

- Erkennung der Netzwerktopologie

Die Netzwerkerkennung besteht im Wesentlichen aus zwei Phasen: Erkennung und Aktionen.

Discovery

Zabbix scannt regelmäßig die in den Netzwerkerkennungsregeln definierten IP-Bereiche. Die Häufigkeit der Prüfung ist für jede Regel individuell konfigurierbar.

Jede Regel verfügt über eine Reihe von Service-Prüfungen, die für den IP-Bereich ausgeführt werden sollen.

Erkennungsregeln werden vom Discovery-Manager verarbeitet. Der Discovery-Manager erstellt für jede Regel einen Auftrag mit einer Liste von Aufgaben (Netzwerkprüfungen). Netzwerkprüfungen werden parallel von den verfügbaren Discovery-Workern ausgeführt (die Anzahl ist im Frontend für jede Regel konfigurierbar). Nur Prüfungen mit derselben IP und demselben Port werden nacheinander eingeplant, da einige Geräte keine parallelen Verbindungen auf demselben Port akzeptieren.

Die Warteschlangengröße für Netzwerkprüfungen ist auf ungefähr 2000000 oder 4 GB Speicher begrenzt. Wenn die Warteschlange voll wird, wird die Erkennungsregel übersprungen und eine Warnmeldung im Log ausgegeben. Sie können den internen 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 einzelne Prüfungen keinen Service finden (oder fehlschlagen), werden andere Prüfungen dennoch verarbeitet.

> **Note:**
> Wenn eine Erkennungsregel während der Ausführung geändert wird, dann wird die aktuelle Discovery-Ausführung abgebrochen.

Jede Prüfung eines Service und eines Hosts (IP), die vom Netzwerkerkennungsmodul durchgeführt wird, erzeugt ein Discovery-Ereignis.

Ereignis	Ergebnis der Service-Prüfung
Service entdeckt	Der Service ist ’up’, nachdem er ’down’ war, oder wenn er zum ersten Mal entdeckt wurde.
Service Up	Der Service ist ’up’, nachdem er bereits ’up’ war.
Service verloren	Der Service ist ’down’, nachdem er ’up’ war.
Service Down	Der Service ist ’down’, nachdem er bereits ’down’ war.
Host entdeckt	Mindestens ein Service eines Hosts ist ’up’, nachdem alle Services dieses Hosts ’down’ waren, oder es wird ein Service entdeckt, der zu einem nicht registrierten Host gehört.
Host Up	Mindestens ein Service eines Hosts ist ’up’, nachdem bereits mindestens ein Service ’up’ war.
Host verloren	Alle Services eines Hosts sind ’down’, nachdem mindestens einer ’up’ war.
Host Down	Alle Services 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.

Erstellung von Hosts

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 zu Aktionen auf einem Host führen. Solche Operationen sind:

- Host aktivieren
- Host deaktivieren
- Host zu einer Hostgruppe hinzufügen
- Vorlage mit einem Host verknüpfen

Erstellte Hosts werden zur Gruppe *Discovered hosts* hinzugefügt (standardmäßig, konfigurierbar unter *Administration* > *General* > *Other*). Wenn Sie möchten, dass Hosts zu einer anderen Gruppe hinzugefügt werden, fügen Sie eine Operation *Aus Hostgruppen entfernen* hinzu (mit Angabe von „Discovered hosts“) und fügen Sie außerdem eine Operation *Zu Hostgruppen hinzufügen* hinzu (mit Angabe einer anderen Hostgruppe), da ein Host zu einer Hostgruppe gehören muss.

Die IP-Adresse des erkannten Geräts wird zusammen mit der Erkennungsquelle (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 Erkennungsquelle existiert, wird dieser Host als Ziel für die Ausführung von Operationen verwendet. Wenn sich die Erkennungsquelle unterscheidet, wird die erkannte Entität als anderer Host behandelt und es kann ein neuer Host erstellt werden.

Wenn die IP-Adresse des erkannten Hosts geändert wird oder die Schnittstelle gelöscht wird, wird bei der nächsten Erkennung 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.

Erstellung von Schnittstellen beim Hinzufügen von Hosts

Wenn Hosts als Ergebnis der Netzwerkentdeckung hinzugefügt werden, werden für sie Schnittstellen gemäß den folgenden Regeln erstellt:

- Die erkannten Dienste – wenn zum Beispiel eine SNMP-Prüfung erfolgreich war, wird eine SNMP-Schnittstelle erstellt.
- Wenn ein Host sowohl auf Anfragen des Zabbix Agent als auch auf SNMP-Anfragen geantwortet hat, werden beide Schnittstellentypen erstellt.
- Wenn die Eindeutigkeitskriterien Daten des Zabbix Agent oder von SNMP zurückgegebene Daten sind, wird die erste für einen Host gefundene Schnittstelle als Standardschnittstelle erstellt. Weitere IP-Adressen werden als zusätzliche Schnittstellen hinzugefügt. Die Bedingungen der Aktion (wie z. B. Host-IP) haben keinen Einfluss auf das Hinzufügen von Schnittstellen. *Beachten Sie*, dass dies funktioniert, wenn alle Schnittstellen durch dieselbe Discovery-Regel erkannt werden. Wenn eine andere Discovery-Regel eine andere Schnittstelle desselben Hosts erkennt, wird ein zusätzlicher Host hinzugefügt.
- Wenn ein Host nur auf Agent-Prüfungen geantwortet hat, wird er nur mit einer Agent-Schnittstelle erstellt. Wenn er später auch auf SNMP antwortet, werden zusätzliche SNMP-Schnittstellen hinzugefügt.
- Wenn anfänglich 3 separate Hosts erstellt wurden, nachdem sie anhand des Eindeutigkeitskriteriums „IP“ erkannt wurden, und die Discovery-Regel dann so geändert wird, dass die Hosts A, B und C ein identisches Ergebnis der Eindeutigkeitskriterien haben, werden B und C als zusätzliche Schnittstellen für A, den zuerst erstellten Host, angelegt. Die einzelnen Hosts B und C bleiben bestehen. Unter *Monitoring > Discovery* werden die hinzugefügten Schnittstellen in der Spalte „Discovered device“ in schwarzer Schrift und eingerückt angezeigt, aber in der Spalte „Monitored host“ wird nur A, der zuerst erstellte Host, angezeigt. „Uptime/Downtime“ wird für IPs, die als zusätzliche Schnittstellen betrachtet werden, nicht gemessen.

Proxy-Einstellung ändern

Die von verschiedenen Proxys erkannten Hosts werden nicht immer als unterschiedliche Hosts behandelt. Erkennungs- und Eindeutigkeitsprüfungen hängen von der Struktur der Proxy-Gruppe ab: Wenn ein Proxy eine Erkennungsregel ausführt und einen Host erstellt, wird dieser Host zur übergeordneten Proxy-Gruppe des Proxys hinzugefügt und nicht dem Proxy selbst zugewiesen. Wenn Zabbix während der Erkennung die Eindeutigkeit der IP-Adresse auswertet, prüft es die 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.

Zum Beispiel die Schritte zum Ersetzen eines 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 Hosts, die von einzelnen Proxys in dieser Gruppe überwacht werden, aktualisiert werden, um Duplikate zu vermeiden)
5. Erkennungsregel aktivieren

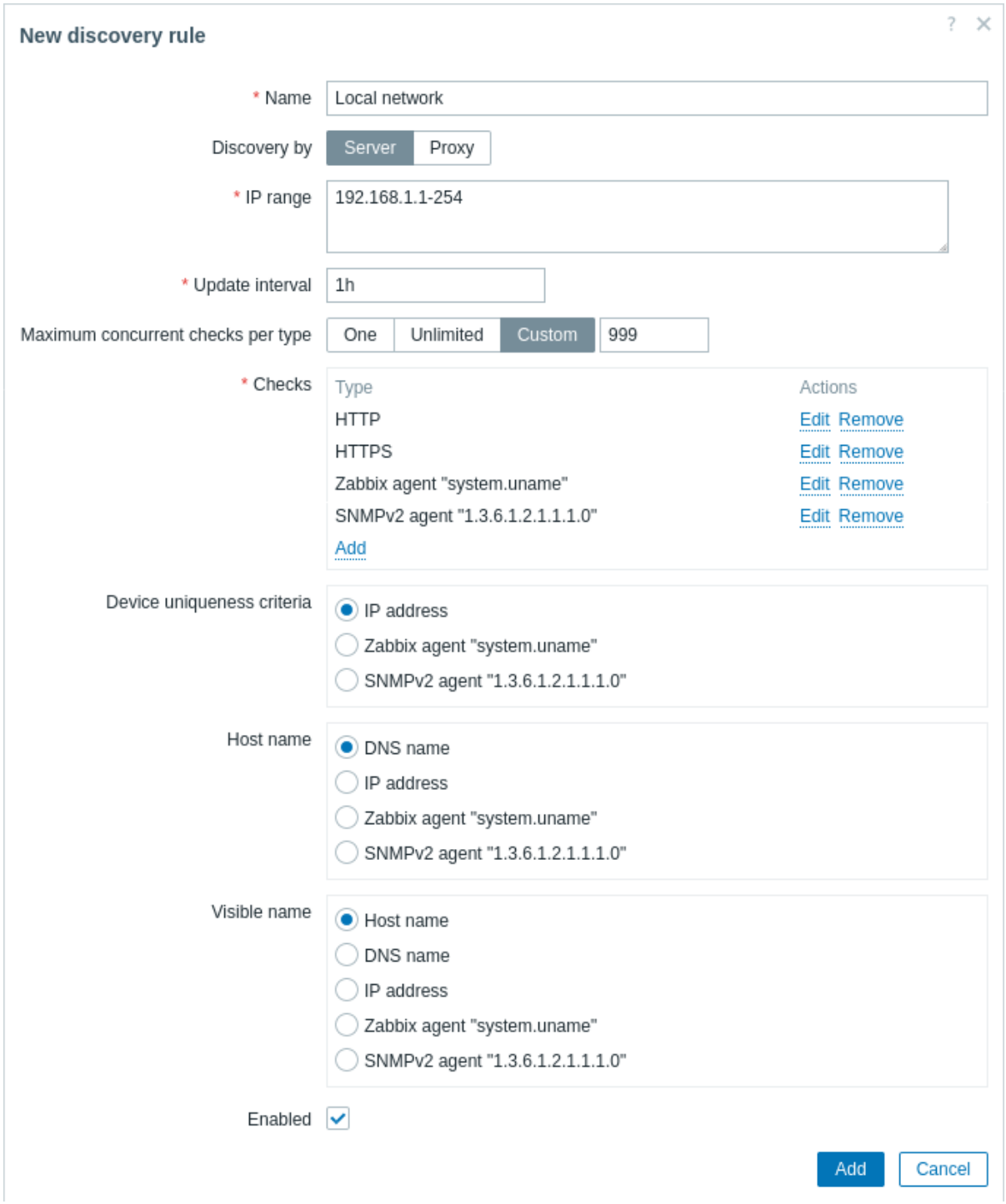
1 Konfigurieren einer Netzwerkerkennungsregel

Übersicht

So konfigurieren Sie eine von Zabbix verwendete Netzwerkerkennungsregel, um Hosts und Dienste zu erkennen:

- Gehen Sie zu *Datenerfassung* → *Erkennung*
- Klicken Sie auf *Erkennungsregel erstellen* (oder auf den Regelnamen, um eine vorhandene Regel zu bearbeiten)
- Bearbeiten Sie die Attribute der Erkennungsregel

Regelattribute



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Eindeutiger Name der Regel. Zum Beispiel „Lokales Netzwerk“.
Discovery by	Die Discovery wird durchgeführt von: **Server** - durch den Zabbix Server **Proxy** - durch den Zabbix Proxy (ausgewählt im Feld für den Proxy-Namen)

Parameter	Beschreibung
IP range	Der Bereich der IP-Adressen für die Discovery. Er kann die folgenden 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.
Update interval	Dieser Parameter definiert, wie oft Zabbix die Regel ausführt. Das Intervall wird ab dem Ende der Ausführung der vorherigen Discovery-Instanz gemessen, sodass es keine Überlappung gibt. 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. 1w → 1h) die nächste Prüfung entsprechend dem vorherigen Wert ausgeführt wird (bei den Beispielwerten also erst weit in der Zukunft).
Maximum concurrent checks per type	Legen Sie die maximale Anzahl von Discovery-Threads (Workern) pro Service-Prüfung für die parallele Verarbeitung von Discovery-Prüfungen fest: **One** - ein Thread **Unlimited** - unbegrenzte Anzahl von Threads (jedoch nicht mehr als im Parameter StartDiscoverers) **Custom** - legen Sie eine benutzerdefinierte Anzahl von Threads fest (0-999) Beachten Sie, dass alle Discovery-Regeln mit asynchronen SNMPv3-Service-Prüfungen aufgrund der Besonderheiten der libsnmp-Implementierung von einem Worker verarbeitet werden, d. h. eine Erhöhung der Anzahl der Worker erhöht die Discovery-Geschwindigkeit nicht.
Checks	Zabbix verwendet diese Liste von Prüfungen für die Discovery. 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 Discovery verwendet die Funktionalität **net.tcp.service[]**, um jeden Host zu testen, außer bei SNMP, wo eine SNMP-OID abgefragt wird. Der Zabbix Agent wird durch Abfrage eines Datenpunkts im unverschlüsselten Modus getestet. Weitere Informationen finden Sie unter Agent-Datenpunkte. Der Parameter „Ports“ kann einer der folgenden sein: Einzelner Port: 22 Portbereich: 22-45 Liste: 22-45,55,60-70 Seit Zabbix 7.0. werden alle Service-Prüfungen asynchron durchgeführt, außer LDAP-Prüfungen. Seit Zabbix 7.0 erfolgt die HTTP/HTTPs-Prüfung über libcurl. Wenn Zabbix Server/Proxy ohne libcurl kompiliert ist, funktionieren HTTP-Prüfungen wie in früheren Versionen (d. h. als TCP-Prüfungen), HTTPS-Prüfungen funktionieren jedoch nicht.
Device uniqueness criteria	Eindeutigkeitskriterien können sein: **IP address** - verarbeiten Sie nicht mehrere Geräte mit einzelner IP. Wenn bereits ein Gerät mit derselben IP existiert, wird es als bereits entdeckt betrachtet und es wird kein neuer Host hinzugefügt. **<discovery check>** - entweder Zabbix Agent oder SNMP-Agent-Prüfung. Beachten Sie, dass die während der Discovery verwendeten Eindeutigkeitskriterien nicht mit der Host-Identifizierung im System bei der Ausführung von Aktionen identisch sind. Die Eindeutigkeitskriterien während der Discovery definieren, ob zwei oder mehr entdeckte Geräte gleich (oder verschieden) sind, während in Zabbix nur die IP-Adresse das Kriterium für die Host-Identifizierung ist (siehe Host-Erstellung).
Host name	Legen Sie den technischen Host-Namen eines erstellten Hosts fest mit: **DNS name** - DNS-Name (Standard) **IP address** - IP-Adresse **<discovery check>** - empfangener Zeichenfolgenwert der Discovery-Prüfung (z. B. Zabbix Agent, SNMP-Agent-Prüfung) Siehe auch: Host-Benennung.

Parameter	Beschreibung
Visible name	Legen Sie den sichtbaren Host-Namen eines erstellten Hosts fest mit: **Host name** - technischer Host-Name (Standard) **DNS name** - DNS-Name **IP address** - IP-Adresse **<discovery check>** - empfangener Zeichenfolgenwert der Discovery-Prüfung (z. B. Zabbix Agent, SNMP-Agent-Prüfung) Siehe auch: Host-Benennung.
Enabled	Wenn das Kontrollkästchen markiert ist, ist die Regel aktiv und wird vom Zabbix Server ausgeführt. Wenn es nicht markiert ist, ist die Regel nicht aktiv. Sie wird nicht ausgeführt.

Überschreitung des Limits für Dateideskriptoren

Bei einer großen Anzahl gleichzeitiger Prüfungen kann das Limit für Dateideskriptoren des discovery manager ausgeschöpft werden.

Die Anzahl der für die Erkennung erforderlichen Dateideskriptoren entspricht der Anzahl der Discovery-Worker * 1000. Standardmäßig gibt es 5 Discovery-Worker, während das Soft-Limit des Systems bei ungefähr 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 für *Maximum concurrent checks per type* einen höheren Wert festgelegt hat als den von Zabbix berechneten Wert, verwendet Zabbix den benutzerdefinierten Wert für einen Worker.

Ein praxisnahes Szenario

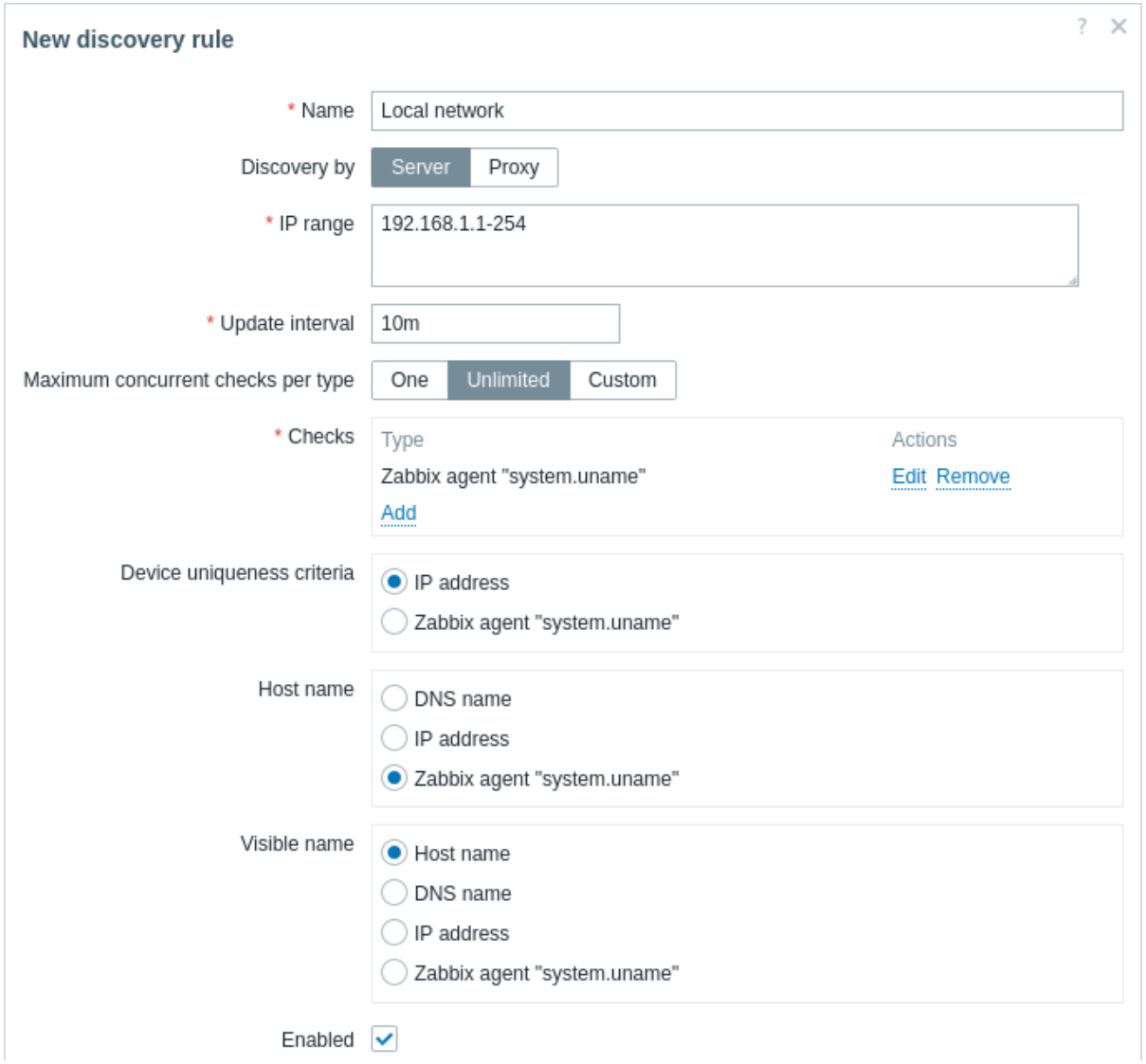
In diesem Beispiel möchten wir die Netzwerk-Erkennung für das lokale Netzwerk mit einem IP-Bereich von 192.168.1.1-192.168.1.254 einrichten.

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.



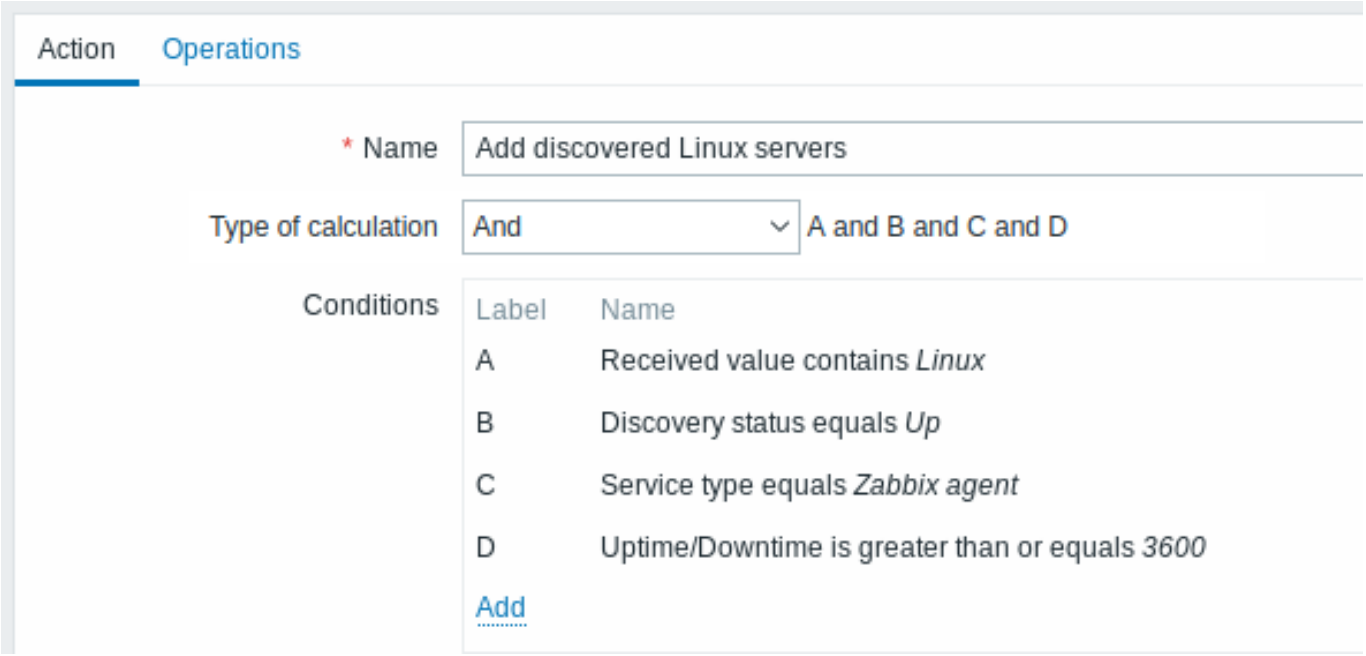
Zabbix wird versuchen, Hosts im IP-Bereich 192.168.1.1-192.168.1.254 zu erkennen, indem eine Verbindung zu Zabbix Agents hergestellt und der Wert des Schlüssels **system.uname** abgerufen wird. Der vom Agent empfangene Wert kann verwendet werden, um die Hosts zu benennen und auch, um unterschiedliche 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.

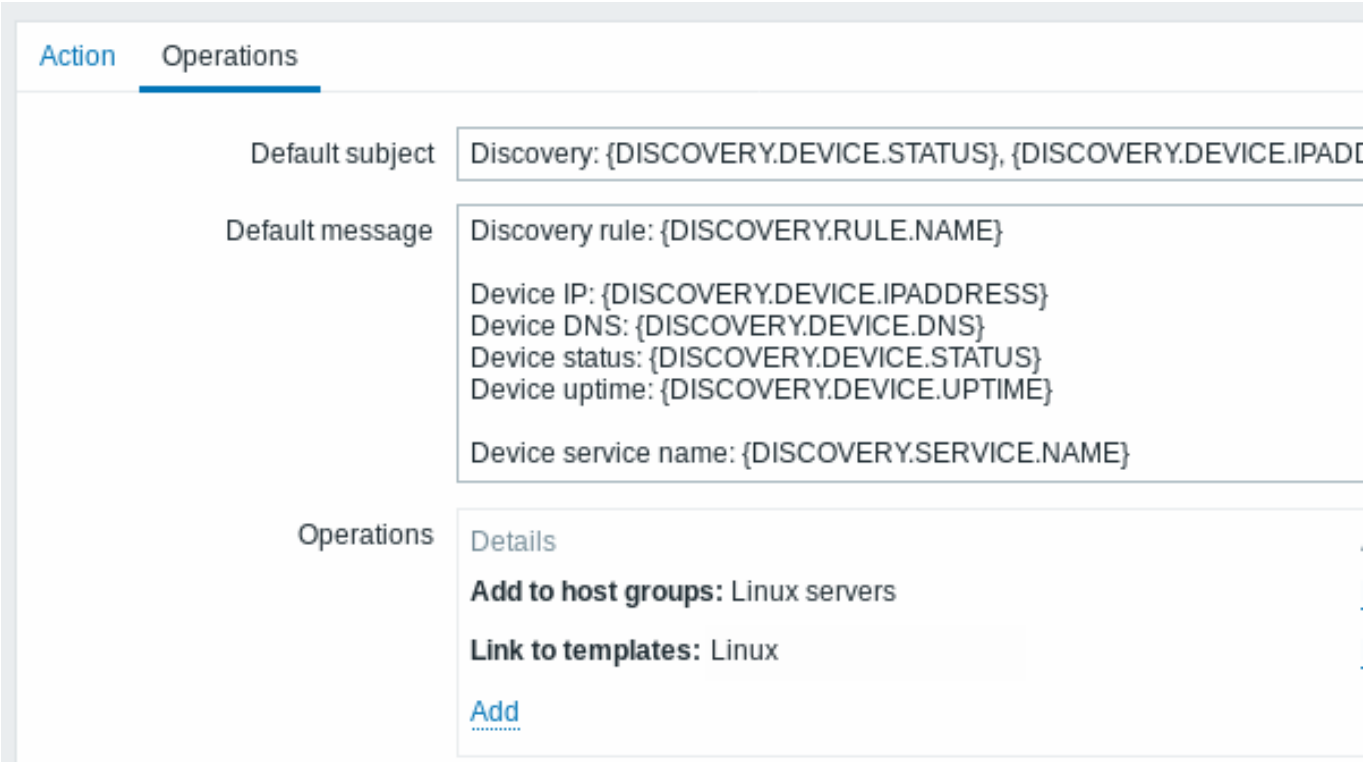
Schritt 2

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

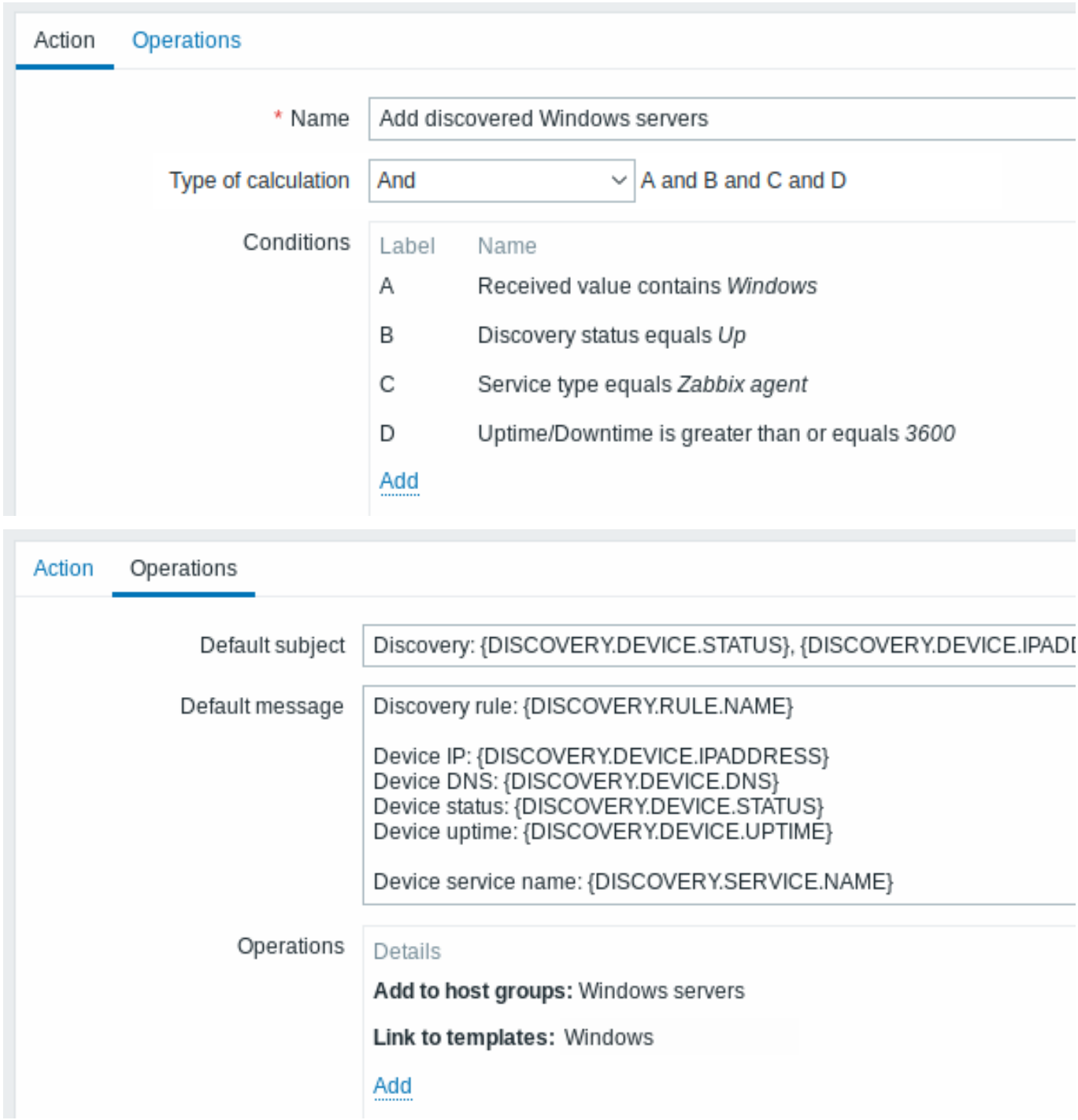


Die Aktion führt die folgenden Operationen aus:

- 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“.

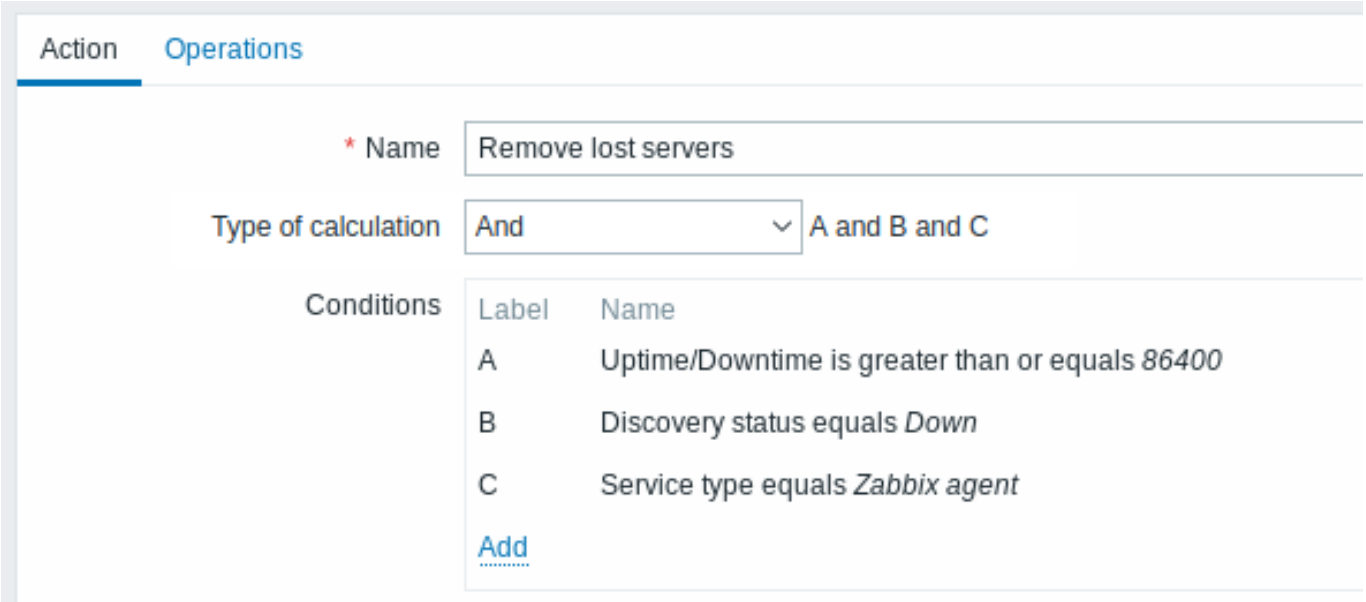
Schritt 3

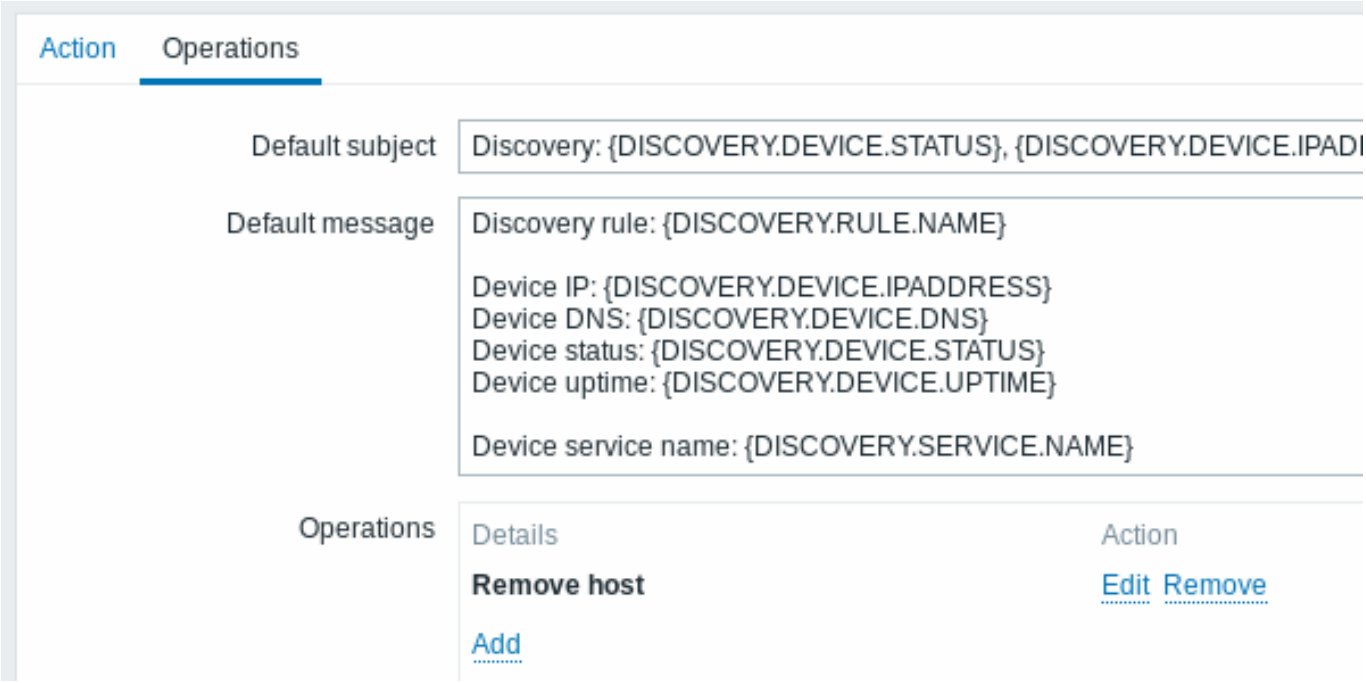
Definieren einer Discovery-Aktion, um die erkannten Windows-Server zur jeweiligen Gruppe/Vorlage hinzuzufügen.



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.

2 Aktive Agent-Autoregistrierung

Übersicht

Es ist möglich, die automatische Registrierung aktiver Zabbix-Agenten zu erlauben, woraufhin der Server mit ihrer 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 Prüfungen anfordert.

Diese Funktion kann für die automatische Überwachung neuer Cloud-Knoten sehr nützlich sein. Sobald Sie einen neuen Knoten in der 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 Prüfungen anfordert und in seiner Konfigurationsdatei die Konfigurationsparameter `ListenIP` oder `ListenPort` definiert sind, werden diese ebenfalls an den Server gesendet. 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 Agent zu konfigurieren. Wenn kein IP-Adresswert empfangen wird, wird der für die eingehende Verbindung verwendete Wert

genutzt. Wenn kein Portwert empfangen wird, wird 10050 verwendet.

Es ist möglich festzulegen, dass der Host mit einem DNS-Namen als Standard-Agent-Schnittstelle automatisch registriert werden soll.

Die automatische Registrierung wird erneut ausgeführt:

- wenn sich die Informationen zu den Host-Metadaten ändern:
  - weil HostMetadata geändert und der Agent neu gestartet wurde
  - weil sich der von HostMetadataItem zurückgegebene Wert geändert hat
- 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 der automatischen Registrierung aktiver Agenten für Zabbix Server und Zabbix Proxy beträgt 120 Sekunden. Falls also ein entdeckter Host gelöscht wird, wird die automatische Registrierung nach 120 Sekunden erneut ausgeführt.

Konfiguration

Server angeben

Stellen Sie sicher, dass Sie den Zabbix Server in der Agent-Konfigurationsdatei - `zabbix_agentd.conf` - angegeben haben:

```
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 zur Benennung des Host. Den System-Hostnamen unter Linux können Sie mit dem Befehl `hostname` ermitteln.

Wenn *Hostname* in der Zabbix-Agent-Konfiguration als kommagetrennte Liste von Hosts definiert ist, werden Hosts für alle aufgeführten Hostnamen erstellt.

Starten Sie den Agent nach allen Änderungen an der Konfigurationsdatei neu.

Aktion für die aktive Agent-Autoregistrierung

Wenn der Server eine Autoregistrierungsanfrage von einem Agent empfängt, ruft er eine Aktion auf. Für die Agent-Autoregistrierung muss eine Aktion mit der Ereignisquelle „Autoregistrierung“ konfiguriert werden.

> **Note:**
> Die Einrichtung der Netzwerkerkennung ist nicht erforderlich, damit sich aktive Agents automatisch registrieren.

Gehen Sie im Zabbix Frontend zu *Benachrichtigungen* → *Aktionen*, wählen Sie *Autoregistrierungsaktionen* aus und klicken Sie auf *Aktion erstellen*:

- Geben Sie im Reiter „Aktion“ Ihrer Aktion einen Namen.
- Geben Sie optional Bedingungen an. Sie können in den Bedingungen eine Teilzeichenfolgenübereinstimmung oder einen regulären Ausdruck für Hostname/Host-Metadaten verwenden. Wenn Sie die Bedingung „Host-Metadaten“ verwenden möchten, lesen Sie den nächsten Abschnitt.
- Fügen Sie im Reiter „Operationen“ die entsprechenden Operationen hinzu, z. B. „Host hinzufügen“, „Zu Hostgruppe hinzufügen“ (zum Beispiel *Discovered hosts*), „Vorlagen verknüpfen“ usw.

> **Note:**
> Wenn die Hosts, die automatisch registriert werden, voraussichtlich nur für aktives Monitoring 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* zum Verknüpfen erstellen.

Erstellte Hosts werden zur Gruppe *Discovered hosts* hinzugefügt (standardmäßig, konfigurierbar unter *Administration* > *Allgemein* > *Andere*). Wenn Sie möchten, dass Hosts zu einer anderen Gruppe hinzugefügt werden, fügen Sie eine Operation *Aus Hostgruppe entfernen* hinzu (mit Angabe von „Discovered hosts“) und zusätzlich eine Operation *Zu Hostgruppe hinzufügen* (mit Angabe einer anderen Hostgruppe), da ein Host zu einer Hostgruppe gehören muss.

Sichere Autoregistrierung

Eine sichere Methode der Autoregistrierung ist durch die Konfiguration einer PSK-basierten Authentifizierung mit verschlüsselten Verbindungen möglich.

Die Verschlüsselungsstufe wird global unter *Administration* > *General* > *Autoregistration* konfiguriert. Es ist möglich, keine Verschlüsselung, TLS-Verschlüsselung mit PSK-Authentifizierung oder beides auszuwählen (sodass sich einige Hosts ohne Verschlüsselung registrieren können, während andere über Verschlüsselung registriert werden).

Die Authentifizierung per PSK wird vom Zabbix Server überprüft, bevor ein Host hinzugefügt wird. Bei erfolgreicher Prüfung wird der Host hinzugefügt und *Connections from/to host* werden ausschließlich auf 'PSK' gesetzt, wobei Identität/Pre-shared Key mit den globalen Einstellungen für die Autoregistrierung übereinstimmen.

Attention:
Um die Sicherheit der Autoregistrierung in Installationen mit Proxys zu gewährleisten, sollte die Verschlüsselung zwischen Zabbix Server und Proxy aktiviert werden.

DNS als Standardschnittstelle verwenden

Mit den Konfigurationsparametern HostInterface und HostInterfaceItem kann während der automatischen Registrierung ein benutzerdefinierter Wert für die Host-Schnittstelle angegeben werden.

Genauer gesagt sind sie nützlich, wenn der Host bei der automatischen Registrierung mit einem DNS-Namen als Standard-Agent-Schnittstelle statt mit seiner IP-Adresse 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 – zum Beispiel von einer IP-Adresse zu einem DNS-Namen oder umgekehrt –, wird die Standardschnittstelle des automatisch registrierten Hosts entsprechend aktualisiert. Diese Aktualisierung wird auf den bestehenden Host angewendet, ohne 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 Reverse-DNS-Abfrage der IP-Adresse des Agenten ermittelt. Wenn Reverse DNS nicht korrekt konfiguriert ist oder einen ungültigen Namen zurückgibt, kann dies aufgrund eines ungültigen Schnittstellenwerts zu einer fehlerhaften oder fehlgeschlagenen automatischen Registrierung führen.

Verwendung von Host-Metadaten

Wenn der Agent eine Autoregistrierungsanfrage an den Server sendet, übermittelt er seinen Hostnamen. In einigen Fällen (zum Beispiel bei Amazon-Cloud-Knoten) reicht ein Hostname für den Zabbix Server nicht aus, um entdeckte Hosts zu unterscheiden. Host-Metadaten können optional verwendet werden, um weitere Informationen von einem Agent an den Server zu senden.

Host-Metadaten werden in der Konfigurationsdatei des Agent konfiguriert – `zabbix_agentd.conf`. Es gibt 2 Möglichkeiten, Host-Metadaten in der Konfigurationsdatei anzugeben:

```
HostMetadata
HostMetadataItem
```

Siehe die Beschreibung der Optionen im obigen Link.

Der Parameter HostMetadataItem kann bis zu 65535 UTF-8-Codepunkte zurückgeben. Ein längerer Wert wird abgeschnitten.

Beachten Sie, dass unter MySQL die tatsächlich maximale Länge in Zeichen geringer ist, wenn der zurückgegebene Wert Multibyte-Zeichen enthält. Beispielsweise ist ein Wert, der nur 3-Byte-Zeichen enthält, insgesamt auf 21844 Zeichen begrenzt, während ein Wert, der nur 4-Byte-Zeichen enthält, insgesamt auf 16383 Symbole begrenzt ist.

Attention:
Ein Autoregistrierungsversuch erfolgt jedes Mal, wenn ein aktiver Agent eine Anfrage an den Server sendet, um aktive Prüfungen zu aktualisieren. Die Verzögerung zwischen den Anfragen wird im Parameter RefreshActiveChecks des Agent angegeben. 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.

3 Low-level-Discovery

Überblick Low-Level-Discovery (LLD) bietet eine Möglichkeit, Datenpunkte, Auslöser und Diagramme für verschiedene Entitäten auf einem Host automatisch zu erstellen. Beispielsweise kann Zabbix automatisch mit der Überwachung von Dateisystemen oder Netzwerkschnittstellen auf Ihrem Rechner beginnen, ohne dass Datenpunkte für jedes Dateisystem oder jede Netzwerkschnittstelle manuell erstellt werden müssen. LLD kann auch Hosts erstellen, zum Beispiel um auf einem Hypervisor erkannte virtuelle Maschinen zu befüllen, sowie verschachtelte Discovery-Regeln ermöglichen, wodurch eine mehrstufige Discovery möglich wird. Zusätzlich kann Zabbix so konfiguriert werden, dass nicht mehr benötigte Entitäten automatisch anhand der tatsächlichen Ergebnisse regelmäßig durchgeführter Discovery entfernt werden.

Ein Benutzer kann eigene Discovery-Typen definieren, sofern sie einem bestimmten JSON-Protokoll folgen.

Die allgemeine Architektur des Discovery-Prozesses ist wie folgt.

Zunächst erstellt ein Benutzer eine Discovery-Regel unter *Datensammlung* > *Vorlagen* in der Spalte *Discovery*. Eine Discovery-Regel besteht aus (1) einem Datenpunkt, der die erforderlichen Entitäten erkennt (zum Beispiel Dateisysteme oder Netzwerkschnittstellen), und (2) Prototypen von Datenpunkten, Auslösern und Diagrammen, die basierend auf dem Wert dieses Datenpunkts erstellt werden sollen.

Ein Datenpunkt, der die erforderlichen Entitäten erkennt, ist wie ein regulärer Datenpunkt, wie er auch an anderer Stelle vorkommt: Der Server fragt einen Zabbix Agent (oder was auch immer als Typ des Datenpunkts eingestellt ist) nach einem Wert dieses Datenpunkts, der Agent antwortet mit einem Textwert. Der Unterschied besteht darin, dass der vom Agent zurückgegebene Wert eine Liste erkannter Entitäten im JSON-Format enthalten muss. Während die Details dieses Formats nur für Implementierer benutzerdefinierter Discovery-Prüfungen wichtig sind, ist es notwendig zu wissen, dass der zurückgegebene Wert eine Liste von Makro→Wert-Paaren enthält. Zum Beispiel könnte der Datenpunkt "net.if.discovery" zwei Paare zurückgeben: "{#IFNAME}" → "lo" und "{#IFNAME}" → "eth0".

Diese Makros werden in Namen, Schlüsseln und anderen Prototypfeldern verwendet, wo sie dann durch die empfangenen Werte ersetzt werden, um für jede erkannte Entität reale Datenpunkte, Auslöser, Diagramme oder sogar Hosts zu erstellen. Siehe die vollständige Liste der Optionen zur Verwendung von LLD-Makros.

Wenn der Server einen Wert für einen Discovery-Datenpunkt empfängt, betrachtet er die Makro→Wert-Paare und erzeugt für jedes Paar auf Basis der Prototypen reale Datenpunkte, Auslöser und Diagramme. Im obigen Beispiel mit "net.if.discovery" würde der Server einen Satz von Datenpunkten, Auslösern und Diagrammen für die Loopback-Schnittstelle "lo" und einen weiteren Satz für die Schnittstelle "eth0" erzeugen.

Beachten Sie, dass sich seit **Zabbix 4.2** das Format des von Low-Level-Discovery-Regeln zurückgegebenen JSON geändert hat. Es wird nicht mehr erwartet, dass das JSON das Objekt "data" enthält. Low-Level-Discovery akzeptiert nun ein normales JSON, das ein Array enthält, um neue Funktionen wie die Vorverarbeitung von Datenpunktwerten und benutzerdefinierte Pfade zu Low-Level-Discovery-Makrowerten in einem JSON-Dokument zu unterstützen.

Integrierte Discovery-Schlüssel wurden aktualisiert, sodass sie ein Array von LLD-Zeilen an der Wurzel des JSON-Dokuments zurückgeben. Zabbix extrahiert automatisch ein Makro und einen Wert, wenn ein Array-Feld die Syntax {#MACRO} als Schlüssel verwendet. Alle neuen nativen Discovery-Prüfungen verwenden die neue Syntax ohne die Elemente "data". Bei der Verarbeitung eines Low-Level-Discovery-Werts wird zuerst die Wurzel ermittelt (Array bei `$.` oder `$.data`).

Während das Element "data" aus allen nativen, mit Discovery verbundenen Datenpunkten entfernt wurde, akzeptiert Zabbix aus Gründen der Abwärtskompatibilität weiterhin die JSON-Notation mit einem "data"-Element, auch wenn von ihrer Verwendung abgeraten wird. Wenn das JSON ein Objekt mit nur einem Array-Element "data" enthält, wird der Inhalt des Elements automatisch mit JSONPath `$.data` extrahiert. Low-Level-Discovery akzeptiert jetzt optionale benutzerdefinierte LLD-Makros mit einem benutzerdefinierten Pfad, der in JSONPath-Syntax angegeben ist.

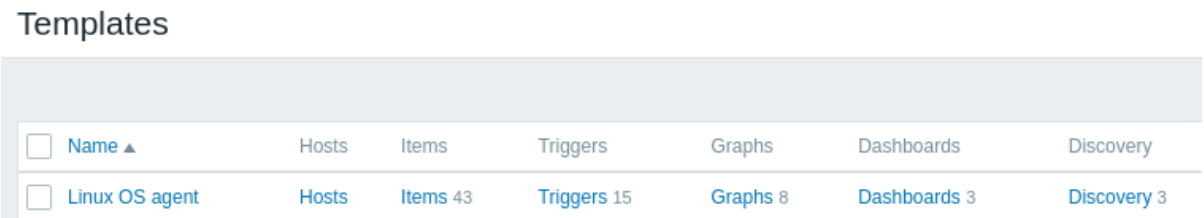
Warning:
Als Ergebnis der oben genannten Änderungen können neuere Agents nicht mehr mit einem älteren Zabbix Server zusammenarbeiten.

Siehe auch: Erkannte Entitäten

Konfiguration der Low-Level-Discovery Wir veranschaulichen die Low-Level-Discovery anhand eines Beispiels zur Dateisystemerkennung.

Gehen Sie wie folgt vor, um die Discovery zu konfigurieren:

- Gehen Sie zu: *Datenerfassung* > *Vorlagen* oder *Hosts*.
- Klicken Sie in der Zeile einer geeigneten Vorlage/eines geeigneten Hosts auf *Discovery*.



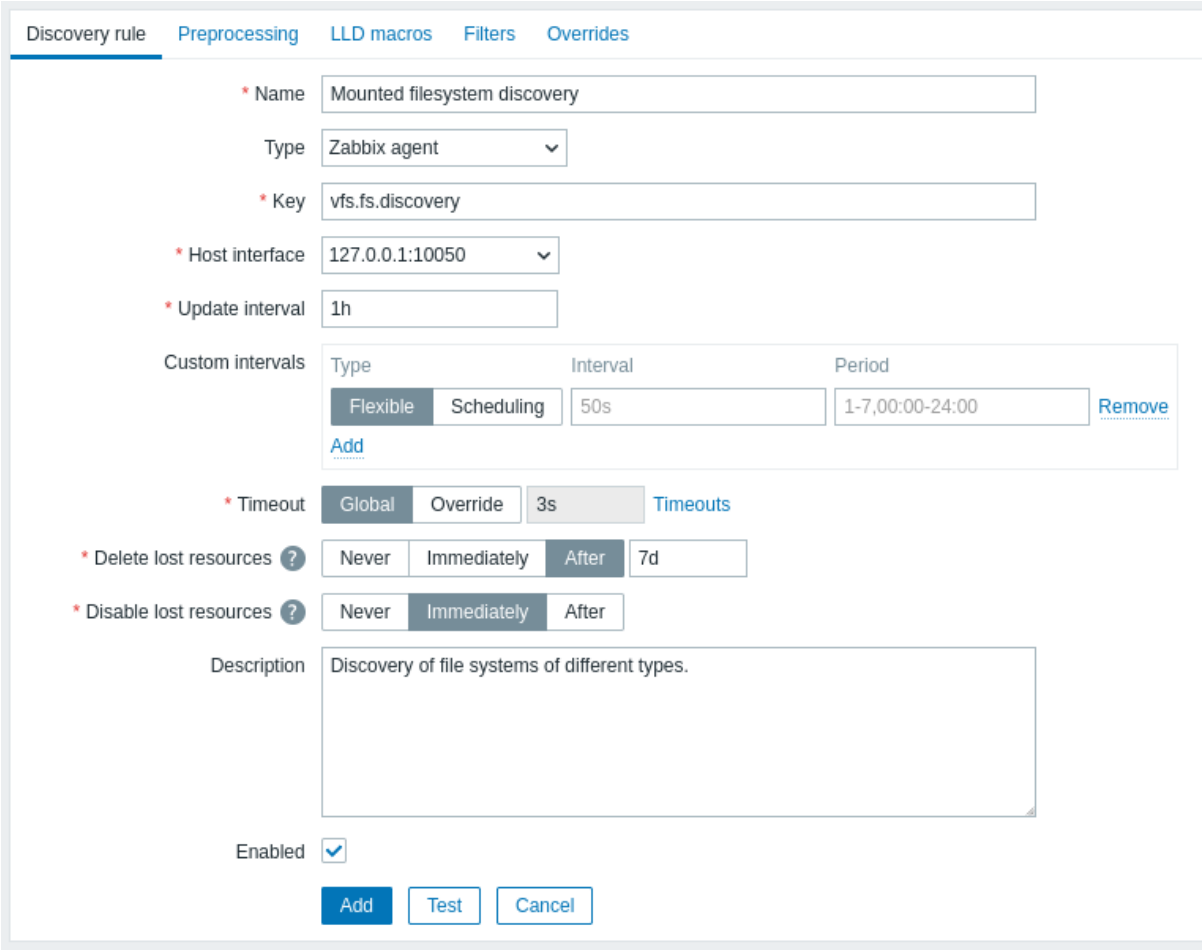
- Klicken Sie oben rechts auf dem Bildschirm auf *Discovery-Regel erstellen*.
- Füllen Sie das Formular der Discovery-Regel mit den erforderlichen Angaben aus.

Discovery-Regel

Das Formular der Discovery-Regel enthält fünf Registerkarten, die von links nach rechts den Datenfluss während der Discovery darstellen:

- *Discovery-Regel* - gibt vor allem den integrierten Datenpunkt oder ein benutzerdefiniertes Skript an, um Discovery-Daten abzurufen.
- *Vorverarbeitung* - wendet eine Vorverarbeitung auf die ermittelten Daten an.
- *LLD-Makros* - ermöglicht das Extrahieren einiger Makrowerte zur Verwendung in ermittelten Datenpunkten, Auslösern usw.
- *Filter* - ermöglicht das Filtern der ermittelten Werte.
- *Überschreibungen* - ermöglicht das Ändern von Datenpunkten, Auslösern, Diagrammen 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 erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Name der Discovery-Regel.
Type	Der Typ der Prüfung, mit der die Discovery durchgeführt wird. In diesem Beispiel verwenden wir den Datenpunkttyp *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 zum Beispiel den integrierten Datenpunktschlüssel „vfs.fs.discovery“ verwenden, um eine JSON-Zeichenfolge mit der Liste der auf dem Computer vorhandenen Dateisysteme, ihren 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).
Update interval	Dieses Feld gibt an, wie oft Zabbix die Discovery durchführt. Zu Beginn, wenn Sie die Dateisystem-Discovery gerade einrichten, möchten Sie möglicherweise ein kleines Intervall festlegen. Sobald Sie jedoch wissen, dass sie funktioniert, können Sie es auf 30 Minuten oder mehr setzen, da sich Dateisysteme normalerweise nicht sehr häufig ändern. Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzermakros werden unterstützt. *Hinweis*: Das Aktualisierungsintervall kann nur dann 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. Neue Discovery-Regeln werden innerhalb von 60 Sekunden nach ihrer Erstellung geprüft, es sei denn, sie haben Scheduling oder Flexible update interval und das *Update interval* ist auf 0 gesetzt. *Beachten Sie*, dass bei einer vorhandenen Discovery-Regel die Discovery sofort durchgeführt werden kann, indem die Schaltfläche *Execute now* gedrückt wird.

Parameter	Beschreibung
Custom intervals	Sie können benutzerdefinierte Regeln für die Prüfung des Datenpunkts erstellen: **Flexible** - erstellt eine Ausnahme zum *Update interval* (Intervall mit anderer 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: **Global** - es wird das Proxy-/globale Timeout verwendet (angezeigt im ausgegrauten Feld *Timeout*); **Override** - es wird ein benutzerdefiniertes Timeout verwendet (im Feld *Timeout* festgelegt; zulässiger Bereich: 1 - 600s). Zeitsuffixe, z. B. 30s, 1m, sowie Benutzermakros werden unterstützt. Durch Klicken auf den Link *Timeouts* können Sie Proxy-Timeouts oder globale Timeouts konfigurieren (wenn kein Proxy verwendet wird). Beachten Sie, dass der Link *Timeouts* nur für Benutzer des Typs *Super admin* sichtbar ist, die Berechtigungen für die Frontend-Bereiche *Administration* > *General* oder *Administration* > *Proxies* haben.
Delete lost resources	Geben Sie an, wie schnell die ermittelte Entität gelöscht wird, sobald ihr Discovery-Status zu „Not discovered anymore“ wird: *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*. Zeitsuffixe werden unterstützt, z. B. 2h, 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 Verlaufsdaten 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, sobald ihr Discovery-Status zu „Not discovered anymore“ wird: *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 Low-Level-Discovery erneut ermittelt werden. Manuell deaktivierte Ressourcen werden bei erneuter Ermittlung nicht wieder aktiviert. Dieses Feld wird nicht angezeigt, wenn *Delete lost resources* auf „Immediately“ gesetzt ist. Zeitsuffixe werden unterstützt, z. B. 2h, 1d. Benutzermakros werden unterstützt.
Description	Geben Sie eine Beschreibung ein.
Enabled	Wenn diese Option aktiviert ist, wird die Regel verarbeitet.

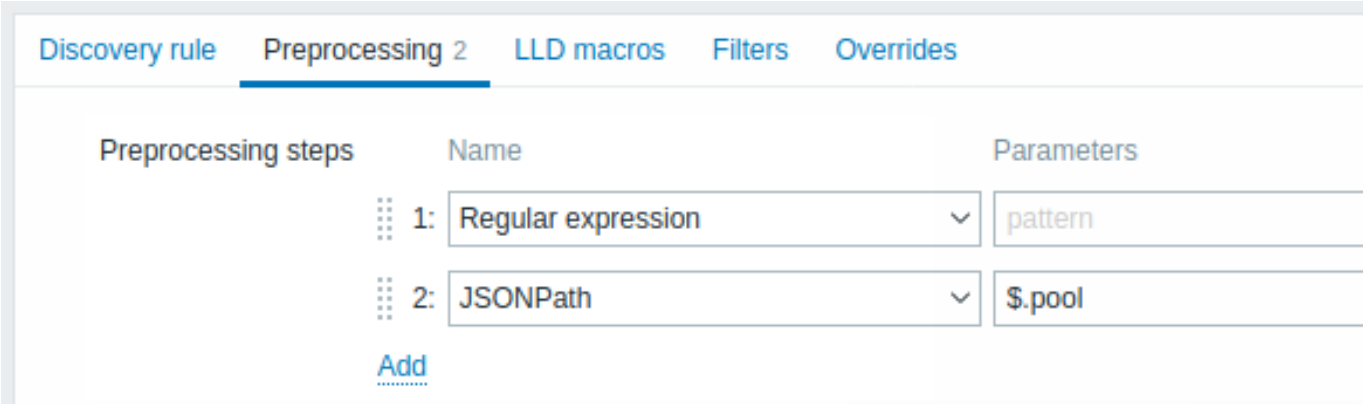
Note:
Der Verlauf der Discovery-Regel wird nicht beibehalten.

Vorverarbeitung

Die Registerkarte **Vorverarbeitung** ermöglicht es, Transformationsregeln zu definieren, die auf das Ergebnis der Discovery angewendet werden. In diesem Schritt sind eine oder mehrere Transformationen möglich. Transformationen werden in der Reihenfolge ausgeführt, in der sie definiert sind. Die gesamte Vorverarbeitung wird vom Zabbix Server durchgeführt.

Siehe auch:

- Details zur Vorverarbeitung
- Testen der Vorverarbeitung



Typ		
	Transformation	Beschreibung
Text		
	Regulärer Ausdruck	Vergleichen Sie den empfangenen Wert mit dem regulären Ausdruck <pattern> und ersetzen Sie den Wert durch das extrahierte <output>. Der reguläre Ausdruck unterstützt die Extraktion von maximal 10 erfassten Gruppen mit der Sequenz \N. Parameter: **pattern** - regulärer Ausdruck **output** - Ausgabevorlage für die Formatierung. Eine Escape-Sequenz \N (wobei N=1…9) wird durch die N-te gefundene Gruppe ersetzt. Eine Escape-Sequenz \0 wird durch den gefundenen Text ersetzt. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
	Ersetzen	Suchen Sie die Suchzeichenfolge und ersetzen Sie sie durch eine andere (oder nichts). Alle Vorkommen der Suchzeichenfolge werden ersetzt. Parameter: **search string** - die zu suchende und zu ersetzende Zeichenfolge, Groß-/Kleinschreibung wird beachtet (erforderlich) **replacement** - die Zeichenfolge, durch die die Suchzeichenfolge ersetzt wird. Die Ersetzungszeichenfolge kann auch leer sein, wodurch die Suchzeichenfolge beim Auffinden effektiv gelöscht wird. Es ist möglich, Escape-Sequenzen zu verwenden, um nach Zeilenumbrüchen, Wagenrücklauf, Tabulatoren und Leerzeichen zu suchen oder diese zu ersetzen: ”\n \r \t \s”; der Backslash kann als ”\\” maskiert werden und Escape-Sequenzen können als ”\\n” maskiert werden. Das Escaping von Zeilenumbrüchen, Wagenrücklauf und Tabulatoren erfolgt bei der Low-Level-Discovery automatisch.
Strukturierte Daten		
	JSONPath	Extrahieren Sie einen Wert oder ein Fragment aus JSON-Daten mithilfe der JSONPath-Funktionalität. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.

	XML XPath	Extrahieren Sie einen Wert oder ein Fragment aus XML-Daten mithilfe der XPath-Funktionalität.
		Damit diese Option funktioniert, muss der Zabbix Server mit libxml-Unterstützung kompiliert 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 *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.

Typ	
CSV zu JSON	Konvertieren Sie CSV-Dateidaten in das JSON-Format. Weitere Informationen finden Sie unter: CSV-zu-JSON-Vorverarbeitung.
XML zu JSON	Konvertieren Sie Daten im XML-Format in JSON. Weitere Informationen finden Sie unter: Serialisierungsregeln. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
SNMP	
SNMP-Walk-Wert	Extrahieren Sie einen Wert anhand des angegebenen OID-/MIB-Namens und wenden Sie Formatierungsoptionen an: **Unverändert** - Hex-STRING als nicht maskierte Hex-Zeichenfolge zurückgeben (*beachten* Sie, dass Anzeigehinweise weiterhin angewendet werden); **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, ausgedrückt als Folge von Hex-Zeichen (z. B. ”1A 2B 3C 4D”), in einen vorzeichenlosen 64-Bit-Integer konvertieren. Bei Bitzeichenfolgen, die länger als 8 Bytes sind, werden nachfolgende Bytes ignoriert. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
SNMP-Walk zu JSON	Konvertieren Sie SNMP-Werte in JSON. Geben Sie einen Feldnamen im JSON und den entsprechenden SNMP-OID-Pfad an. Feldwerte werden mit Werten aus dem angegebenen SNMP-OID-Pfad befüllt. Sie können diesen Vorverarbeitungsschritt für die SNMP-OID-Discovery verwenden. Ähnliche Wertformatierungsoptionen wie im Schritt *SNMP-Walk-Wert* sind verfügbar. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
SNMP-Get-Wert	Wenden Sie Formatierungsoptionen auf den SNMP-Get-Wert an: **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, ausgedrückt als Folge von Hex-Zeichen (z. B. ”1A 2B 3C 4D”), in einen vorzeichenlosen 64-Bit-Integer konvertieren. Bei Bitzeichenfolgen, die länger als 8 Bytes sind, werden nachfolgende Bytes ignoriert. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
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
Validierung	
Entspricht nicht dem regulären Ausdruck	Geben Sie einen regulären Ausdruck an, dem ein Wert nicht entsprechen darf. Z. B. `Error:(.*?)\.` Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
Auf Fehler in JSON prüfen	Prüfen Sie auf eine Fehlermeldung auf Anwendungsebene, die sich an JSONPath befindet. Beenden Sie die Verarbeitung bei Erfolg und wenn die Meldung nicht leer ist; andernfalls setzen Sie die Verarbeitung mit dem Wert fort, der vor diesem Vorverarbeitungsschritt vorhanden war. Beachten Sie, dass diese Fehler externer Dienste dem Benutzer unverändert gemeldet werden, ohne Informationen zum Vorverarbeitungsschritt hinzuzufügen. Z. B. `$.errors`. Wenn ein JSON wie `{"errors":"e1"}` empfangen wird, wird der nächste Vorverarbeitungsschritt nicht ausgeführt. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.

Typ	
Auf Fehler in XML prüfen	Prüfen Sie auf eine Fehlermeldung auf Anwendungsebene, die sich an XPath befindet. Beenden Sie die Verarbeitung bei Erfolg und wenn die Meldung nicht leer ist; andernfalls setzen Sie die Verarbeitung mit dem Wert fort, 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 Fehlschlag beim Parsen von ungültigem XML wird kein Fehler gemeldet. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
Entspricht dem regulären Ausdruck	Geben Sie einen regulären Ausdruck an, dem ein Wert entsprechen muss. Wenn Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktivieren, können benutzerdefinierte Optionen zur Fehlerbehandlung angegeben werden: entweder den Wert verwerfen, einen angegebenen Wert setzen oder eine angegebene Fehlermeldung setzen.
Drosselung	
Unveränderte Werte mit Heartbeat verwerfen	Verwerfen Sie einen Wert, wenn er sich innerhalb des definierten Zeitraums (in Sekunden) nicht geändert hat. Positive Ganzzahlwerte werden zur Angabe der Sekunden unterstützt (Minimum - 1 Sekunde). Zeitsuffixe können in diesem Feld verwendet werden (z. B. 30s, 1m, 2h, 1d). Benutzermakros und Low-Level-Discovery-Makros können in diesem Feld verwendet werden. Für einen Discovery-Datenpunkt kann nur eine Drosselungsoption angegeben werden. Z. B. `1m`. Wenn identischer Text innerhalb von 60 Sekunden zweimal an diese Regel übergeben wird, wird er verworfen. *Hinweis*: Änderungen an Datenpunkt-Prototypen setzen die Drosselung nicht zurück. Die Drosselung wird nur zurückgesetzt, wenn Vorverarbeitungsschritte geändert werden.
Prometheus	
Prometheus zu JSON	Konvertieren Sie erforderliche Prometheus-Metriken in JSON. Weitere Details finden Sie unter Prometheus-Prüfungen.

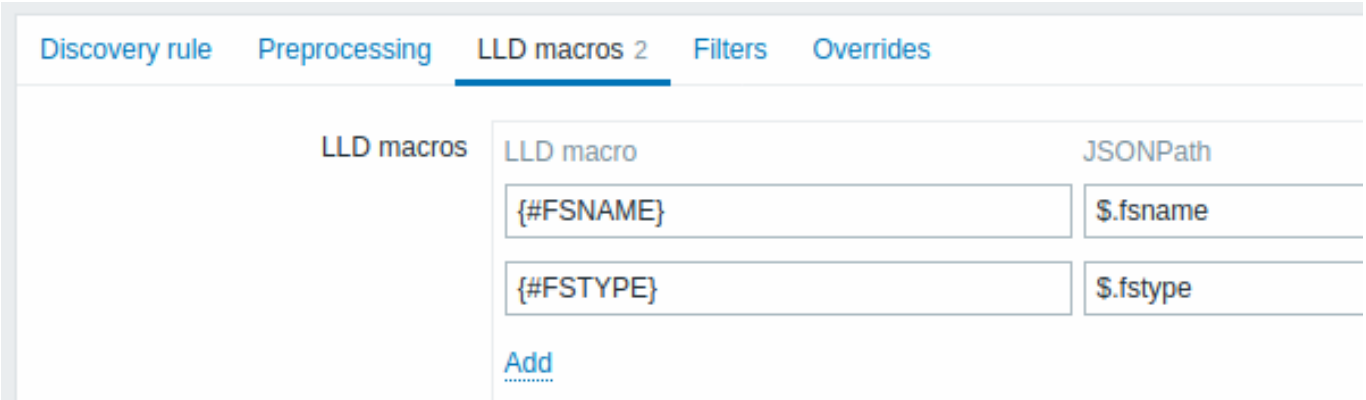
Beachten Sie, dass der Inhalt dieser Registerkarte schreibgeschützt ist, wenn die Discovery-Regel über eine Vorlage auf den Host angewendet wurde.

Benutzerdefinierte Makros

Die Registerkarte **LLD-Makros** ermöglicht es, benutzerdefinierte Low-Level-Discovery-Makros anzugeben.

Benutzerdefinierte Makros sind nützlich, wenn das zurückgegebene JSON die erforderlichen Makros noch nicht definiert enthält. Zum Beispiel:

- Der native Schlüssel `vfs.fs.discovery` für die Dateisystemerkennung gibt ein JSON mit einigen vordefinierten LLD-Makros wie {#FSNAME}, {#FSTYPE} zurück. Diese Makros können direkt in Datenpunkt- und Auslöser-Prototypen (siehe die folgenden Abschnitte auf dieser Seite) verwendet werden; das Definieren benutzerdefinierter Makros ist nicht erforderlich;
- Der Agent-Datenpunkt `vfs.fs.get` gibt ebenfalls ein JSON mit Dateisystemdaten zurück, jedoch ohne vordefinierte LLD-Makros. In diesem Fall können Sie die Makros selbst definieren und sie mithilfe von JSONPath den Werten im JSON zuordnen:



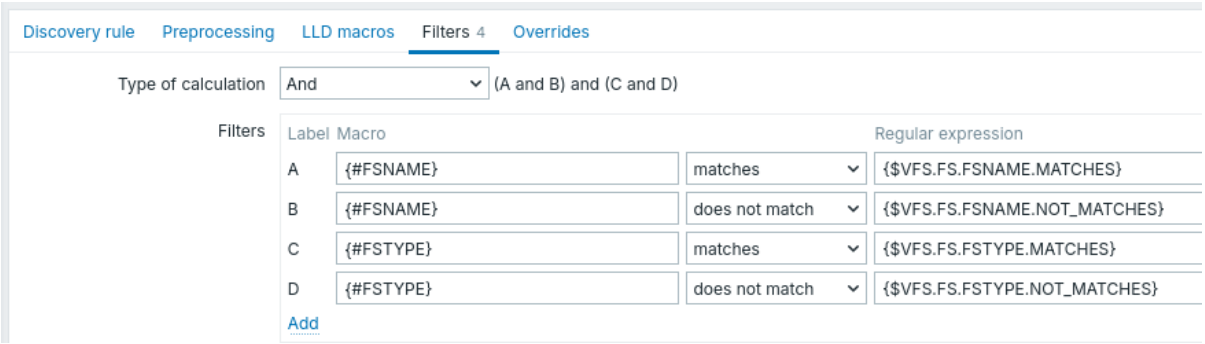
Die extrahierten Werte können in erkannten Datenpunkten, Auslösern usw. verwendet werden. Beachten Sie, dass die Werte aus dem Erkennungsergebnis und allen bisherigen Vorverarbeitungsschritten extrahiert werden.

Parameter	Beschreibung
LLD-Makro	Name des Low-Level-Discovery-Makros unter Verwendung der folgenden Syntax: {#MACRO}.

Parameter	Beschreibung
JSONPath	Pfad, der verwendet wird, um den Wert des LLD-Makros aus einer LLD-Zeile mithilfe der JSONPath-Syntax zu extrahieren. Die aus dem zurückgegebenen JSON extrahierten Werte werden verwendet, um die LLD-Makros in den Prototypfeldern von Datenpunkten, Auslösern usw. zu ersetzen. JSONPath kann entweder in Punktnotation oder in Klammernotation angegeben werden. Die Klammernotation sollte bei Sonderzeichen und Unicode verwendet werden, z. B. `$['unicode + special chars #1']['unicode + special chars #2']`. Zum Beispiel extrahiert `$.foo` „bar“ und „baz“ aus diesem JSON: `[{"foo":"bar"}, {"foo":"baz"}]` Beachten Sie, dass `$.foo` „bar“ und „baz“ auch aus diesem JSON extrahiert: `{"data":[{"foo":"bar"}, {"foo":"baz"}]}`, da ein einzelnes „data“-Objekt automatisch verarbeitet wird (zur Abwärtskompatibilität mit der Low-Level-Discovery-Implementierung in Zabbix-Versionen vor 4.2).

Filter

Ein Filter kann verwendet werden, um echte Datenpunkte, Auslöser und Diagramme nur für Entitäten zu erzeugen, die den Kriterien entsprechen. Die Registerkarte **Filter** enthält Filterdefinitionen für Discovery-Regeln, mit denen Discovery-Werte gefiltert werden können:



Parameter	Beschreibung
Berechnungstyp	Die folgenden Optionen zur Berechnung von Filtern sind verfügbar: **Und** - alle Filter müssen erfüllt sein; **Oder** - es genügt, wenn ein Filter erfüllt ist; **Und/Oder** - verwendet *Und* bei unterschiedlichen Makronamen und *Oder* bei demselben Makronamen; **Benutzerdefinierter Ausdruck** - bietet die Möglichkeit, eine benutzerdefinierte Berechnung der Filter zu definieren. Die Formel muss alle Filter in der Liste enthalten. Begrenzt auf 255 Zeichen.

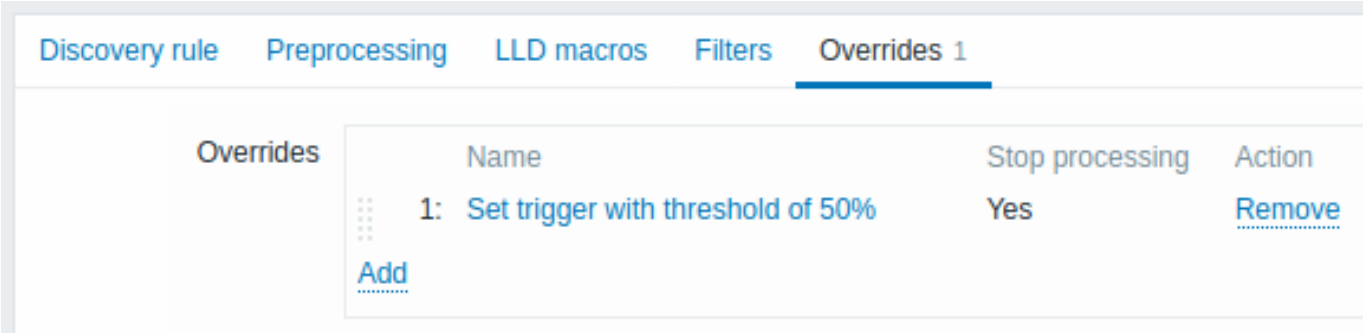
Parameter	Beschreibung
Filter	Die folgenden Operatoren für Filterbedingungen sind verfügbar: *entspricht*, *entspricht nicht*, *existiert*, *existiert nicht*. Die Operatoren *entspricht* und *entspricht nicht* erwarten einen Perl-kompatiblen regulären Ausdruck (PCRE). Wenn Sie beispielsweise nur an den Dateisystemen C:, D: und E: interessiert sind, können Sie {#FSNAME} in das Feld „Makro“ und den regulären Ausdruck ”^C|^D|^E” in das Textfeld „Regulärer Ausdruck“ eintragen. Das Filtern ist auch nach Dateisystemtypen mit dem Makro {#FSTYPE} (z. B. ”^ext|^reiserfs”) und nach Laufwerkstypen mit dem Makro {#FSDRIVETYPE} (z. B. ”fixed”) möglich (nur vom Windows-Agent unterstützt). Sie können im Feld „Regulärer Ausdruck“ einen regulären Ausdruck eingeben oder auf einen globalen regulären Ausdruck verweisen. Um einen regulären Ausdruck zu testen, können Sie „grep -E“ verwenden, zum Beispiel: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` Die Operatoren *existiert* und *existiert nicht* ermöglichen es, Entitäten anhand des Vorhandenseins oder Fehlens des angegebenen LLD-Makros in der Antwort zu filtern. 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 eine Bedingung „existiert nicht“ angegeben. Es wird eine Warnung angezeigt, wenn das Fehlen eines Makros das Ergebnis des Ausdrucks beeinflusst. Zum Beispiel, wenn {#B} fehlt in: {#A} entspricht 1 und {#B} entspricht 2 - ergibt eine Warnung {#A} entspricht 1 oder {#B} entspricht 2 - keine Warnung

Warning:
Ein Fehler oder Tippfehler im regulären Ausdruck, der in der LLD-Regel verwendet wird (zum Beispiel ein falscher regulärer Ausdruck „Dateisysteme für Discovery“), kann zur Löschung von Tausenden von Konfigurationselementen, historischen Werten und Ereignissen für viele Hosts führen.

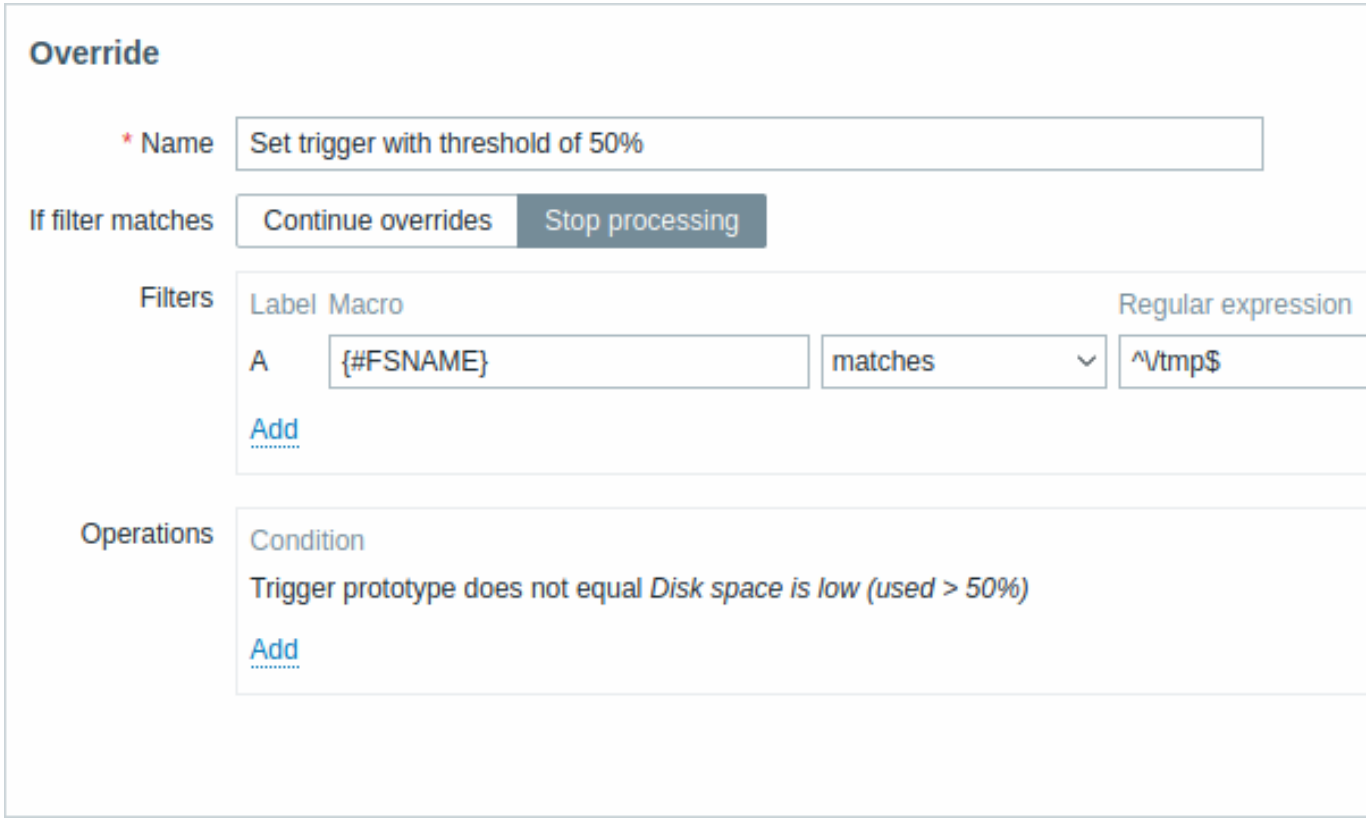
Attention:
Die Zabbix-Datenbank in MySQL muss mit Groß-/Kleinschreibungssensitivität erstellt werden, damit Dateisystemnamen, die sich nur durch Groß-/Kleinschreibung unterscheiden, korrekt erkannt werden.

Überschreiben

Die Registerkarte **Überschreibungen** ermöglicht das Festlegen von Regeln zum Ändern der Liste von Datenpunkt-, Auslöser-, Graph-, Host- und Discovery-Prototypen oder ihrer Attribute für entdeckte Objekte, die die angegebenen Kriterien erfüllen.



Überschreibungen (falls vorhanden) werden in einer per Drag-and-drop neu anordenbaren Liste angezeigt und in der Reihenfolge ausgeführt, in der sie definiert sind. Um die Details einer neuen Überschreibung zu konfigurieren, klicken Sie auf Add im Block *Überschreibungen*. Um eine vorhandene Überschreibung zu bearbeiten, klicken Sie auf den Namen der Überschreibung. Es öffnet sich ein Popup-Fenster, in dem Sie die Details der Überschreibungsregel bearbeiten können.

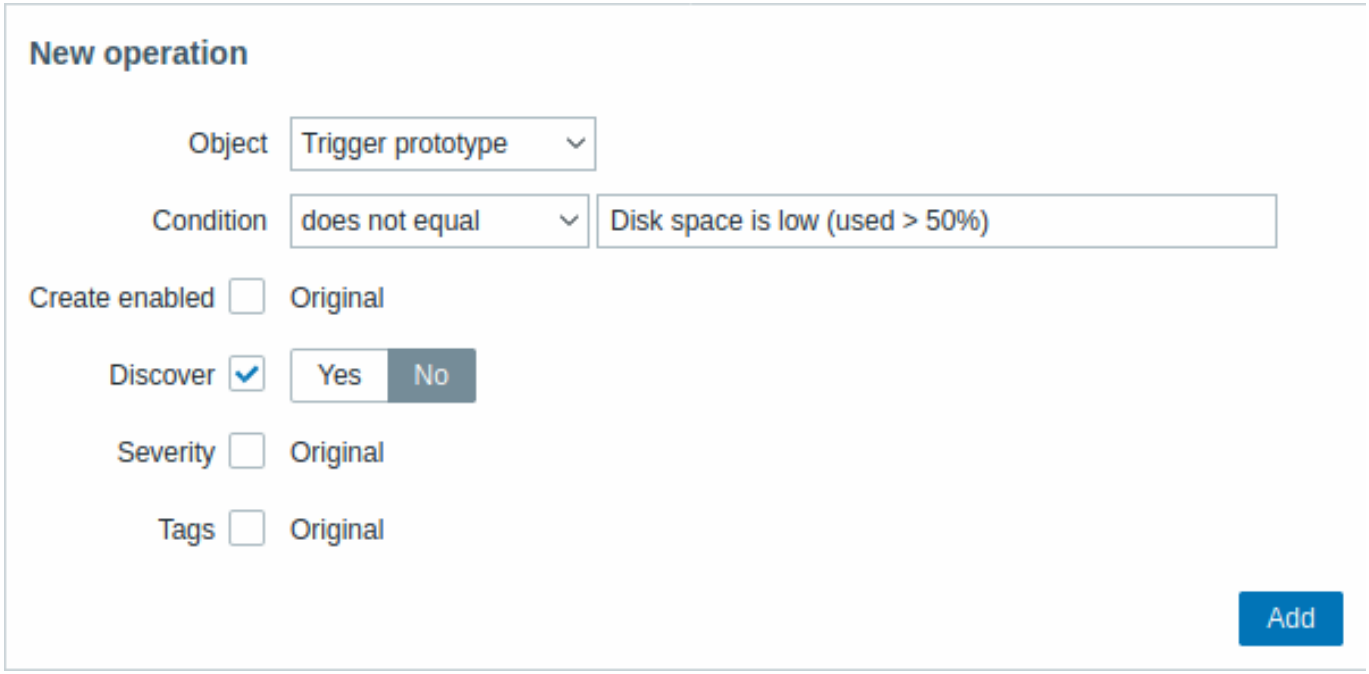


Alle Pflichtparameter sind mit roten Sternchen markiert.

Parameter	Beschreibung
Name	Ein eindeutiger Name der Überschreibung (pro LLD-Regel).
If filter matches	Definiert, ob die nächsten Überschreibungen verarbeitet werden sollen, wenn die Filterbedingungen erfüllt sind: **Continue overrides** - nachfolgende Überschreibungen werden verarbeitet. **Stop processing** - Operationen aus vorhergehenden (falls vorhanden) und dieser Überschreibung werden ausgeführt; nachfolgende Überschreibungen werden für übereinstimmende LLD-Zeilen ignoriert.
Filters	Bestimmt, auf welche entdeckten Entitäten die Überschreibung angewendet werden soll. Überschreibungsfilter werden nach den Filtern der Discovery-Regel verarbeitet und haben dieselbe Funktionalität.
Operations	Überschreibungsoperationen werden mit folgenden Details angezeigt: **Condition** - ein Objekttyp und eine Bedingung, die für den Objektnamen erfüllt sein muss; zum Beispiel: Auslöser-Prototyp ist nicht gleich *Disk space is low (used > 50%)*. **Actions** - Links zum Bearbeiten und Entfernen einer Operation werden angezeigt.

Konfigurieren einer Operation

Um die Details einer neuen Operation zu konfigurieren, klicken Sie auf Add im Block „Operationen“. Um eine vorhandene Operation zu bearbeiten, klicken Sie auf Edit neben der Operation. Es öffnet sich ein Popup-Fenster, in dem Sie die Details der Operation bearbeiten können.



Parameter	Beschreibung
Object	Fünf Objekttypen sind verfügbar: Datenpunkt-Prototyp Auslöser-Prototyp Graph-Prototyp Host-Prototyp Discovery-Prototyp
Condition	Ermöglicht das Filtern von Entitäten, auf die die Operation angewendet werden soll.
Operator	Unterstützte Operatoren: **equals** - auf diesen Prototyp anwenden **does not equal** - auf alle Prototypen außer diesem anwenden **contains** - anwenden, wenn der Prototypname diese Zeichenfolge enthält **does not contain** - anwenden, wenn der Prototypname diese Zeichenfolge nicht enthält **matches** - anwenden, wenn der Prototypname dem regulären Ausdruck entspricht **does not match** - anwenden, wenn der Prototypname dem regulären Ausdruck nicht entspricht
Pattern	Ein regulärer Ausdruck oder eine zu suchende Zeichenfolge.
Object: *Item prototype*	
Create enabled	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Datenpunkt-Prototyps überschrieben werden können: *Yes* - der Datenpunkt wird im aktivierten Zustand hinzugefügt. *No* - der Datenpunkt wird zu einer entdeckten Entität hinzugefügt, jedoch im deaktivierten Zustand.
Discover	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Datenpunkt-Prototyps überschrieben werden können: *Yes* - der Datenpunkt wird hinzugefügt. *No* - der Datenpunkt wird nicht hinzugefügt.
Update interval	Wenn das Kontrollkästchen aktiviert ist, werden zwei Optionen angezeigt, mit denen ein anderes Intervall für den Datenpunkt festgelegt werden kann: *Delay* - Aktualisierungsintervall des Datenpunkts. Benutzermakros und Zeitsuffixe (z. B. 30s, 1m, 2h, 1d) werden unterstützt. Sollte auf 0 gesetzt werden, wenn *Custom interval* verwendet wird. *Custom interval* - klicken Sie auf Add, um flexible/geplante Intervalle anzugeben. Detaillierte Informationen finden Sie unter Benutzerdefinierte Intervalle.

Parameter		Beschreibung
	History	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen für den Datenpunkt ein anderer Verlaufsspeicherzeitraum festgelegt werden kann: *Do not store* - wenn ausgewählt, wird der Verlauf nicht gespeichert. *Store up to* - wenn ausgewählt, erscheint rechts ein Eingabefeld zur Angabe des Speicherzeitraums. Benutzermakros und LLD-Makros werden unterstützt.
	Trends	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen für den Datenpunkt ein anderer Trendspeicherzeitraum festgelegt werden kann: *Do not store* - wenn ausgewählt, werden die Trends nicht gespeichert. *Store up to* - wenn ausgewählt, erscheint rechts ein Eingabefeld zur Angabe des Speicherzeitraums. Benutzermakros und LLD-Makros werden unterstützt.
	Tags	Wenn das Kontrollkästchen aktiviert ist, erscheint ein neuer Block, in dem Tag-Wert-Paare angegeben werden können. Diese Tags werden an die im Datenpunkt-Prototyp angegebenen Tags angehängt, auch wenn die Tag-Namen übereinstimmen.
Object: *Trigger prototype*		
	Create enabled	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Auslöser-Prototyps überschrieben werden können: *Yes* - der Auslöser wird im aktivierten Zustand hinzugefügt. *No* - der Auslöser wird zu einer entdeckten Entität hinzugefügt, jedoch im deaktivierten Zustand.
	Discover	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Auslöser-Prototyps überschrieben werden können: *Yes* - der Auslöser wird hinzugefügt. *No* - der Auslöser wird nicht hinzugefügt.
	Severity	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen für den Auslöser-Schweregrad angezeigt, mit denen der Auslöser-Schweregrad geändert werden kann.
	Tags	Wenn das Kontrollkästchen aktiviert ist, erscheint ein neuer Block, in dem Tag-Wert-Paare angegeben werden können. Diese Tags werden an die im Auslöser-Prototyp angegebenen Tags angehängt, auch wenn die Tag-Namen übereinstimmen.
Object: *Graph prototype*		
	Discover	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Graph-Prototyps überschrieben werden können: *Yes* - der Graph wird hinzugefügt. *No* - der Graph wird nicht hinzugefügt.
Object: *Host prototype*		
	Create enabled	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Host-Prototyps überschrieben werden können: *Yes* - der Host wird im aktivierten Zustand erstellt. *No* - der Host wird im deaktivierten Zustand erstellt.
	Discover	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen die ursprünglichen Einstellungen des Host-Prototyps überschrieben werden können: *Yes* - der Host wird entdeckt. *No* - der Host wird nicht entdeckt.

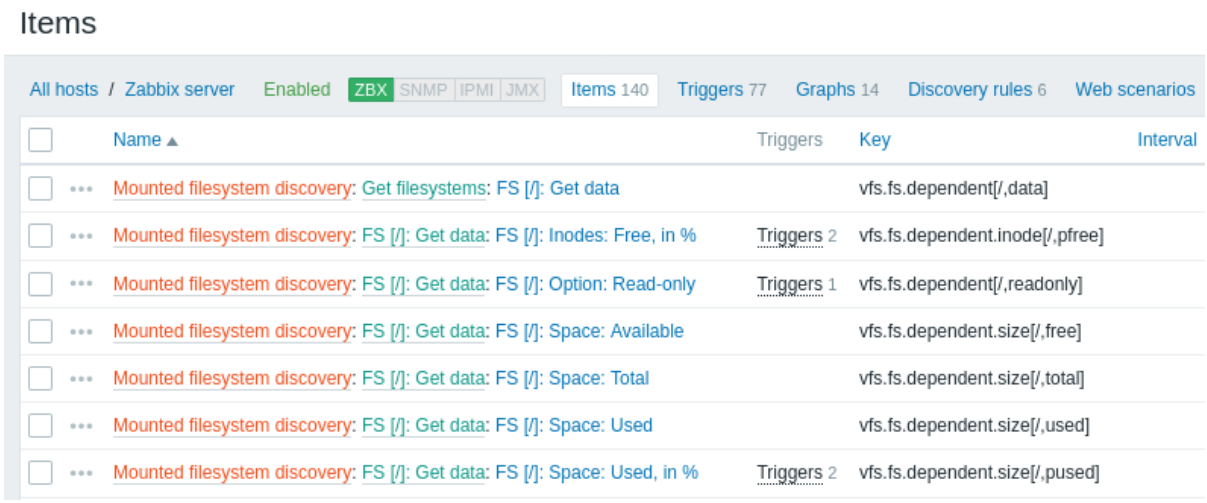
Parameter	Beschreibung
Link templates	Wenn das Kontrollkästchen aktiviert ist, erscheint ein Eingabefeld zur Angabe von Vorlagen. Beginnen Sie mit der Eingabe des Vorlagennamens oder klicken Sie neben dem Feld auf *Select* und wählen Sie Vorlagen aus der Liste im Popup-Fenster aus. Vorlagen aus dieser Überschreibung werden zu allen Vorlagen hinzugefügt, die bereits mit dem Host-Prototyp verknüpft sind.
Tags	Wenn das Kontrollkästchen aktiviert ist, erscheint ein neuer Block, in dem Tag-Wert-Paare angegeben werden können. Diese Tags werden an die im Host-Prototyp angegebenen Tags angehängt, auch wenn die Tag-Namen übereinstimmen.
Host inventory	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen ein anderer Modus für das Host-Inventar des Host-Prototyps ausgewählt werden kann: *Disabled* - Host-Inventar nicht befüllen *Manual* - Details manuell angeben *Automated* - Host-Inventardaten basierend auf erfassten Metriken automatisch ausfüllen.

Formularschaltflächen

Mit den Schaltflächen am unteren Rand des Formulars können mehrere Operationen ausgeführt werden.

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 auf Grundlage der Eigenschaften der aktuellen Discovery-Regel erstellen.
Execute now	Die Discovery sofort auf Grundlage der Discovery-Regel ausführen. Die Discovery-Regel muss bereits vorhanden sein. Siehe weitere Details. *Hinweis:* Wenn die Discovery sofort ausgeführt wird, wird der Konfigurations-Cache nicht aktualisiert; daher spiegelt das Ergebnis keine sehr aktuellen Änderungen an der Konfiguration der Discovery-Regel 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.

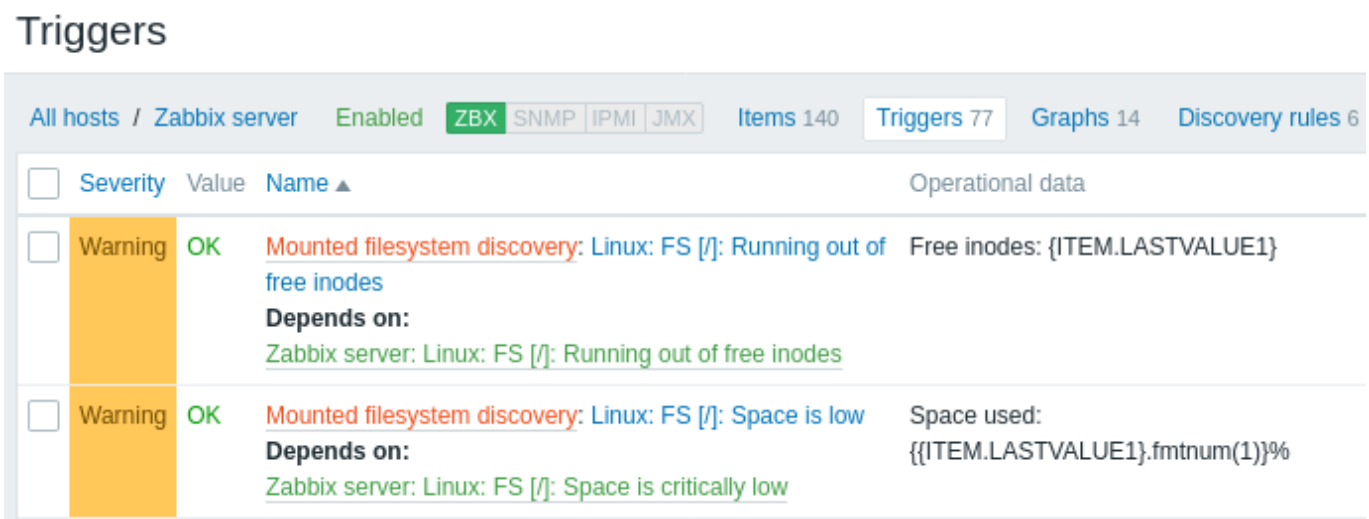
Verlorene Ressourcen können basierend auf dem Wert des Parameters *Disable lost resources* automatisch deaktiviert werden. Dies betrifft verlorene Hosts, Datenpunkte und Auslöser.

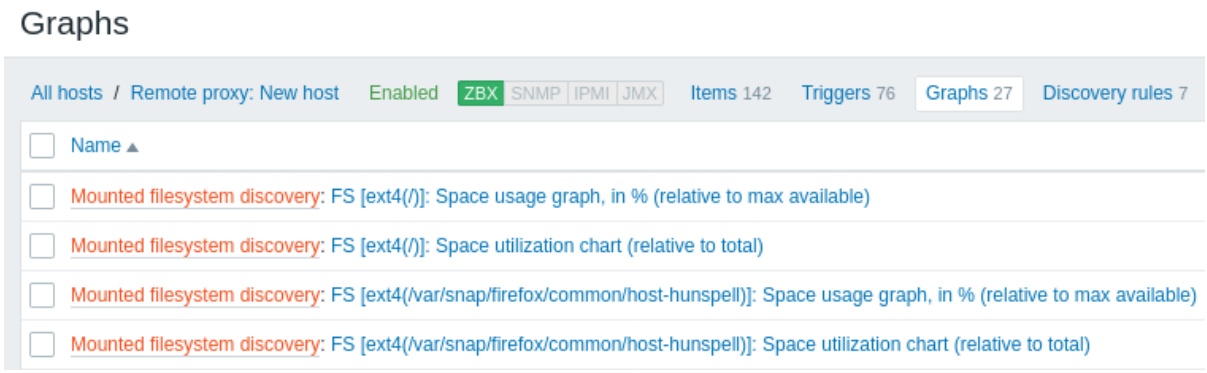
Verlorene Ressourcen können basierend auf dem Wert des Parameters *Delete lost resources* automatisch gelöscht werden. Dies betrifft verlorene Hosts, Hostgruppen, Datenpunkte, Auslöser und Diagramme.

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:

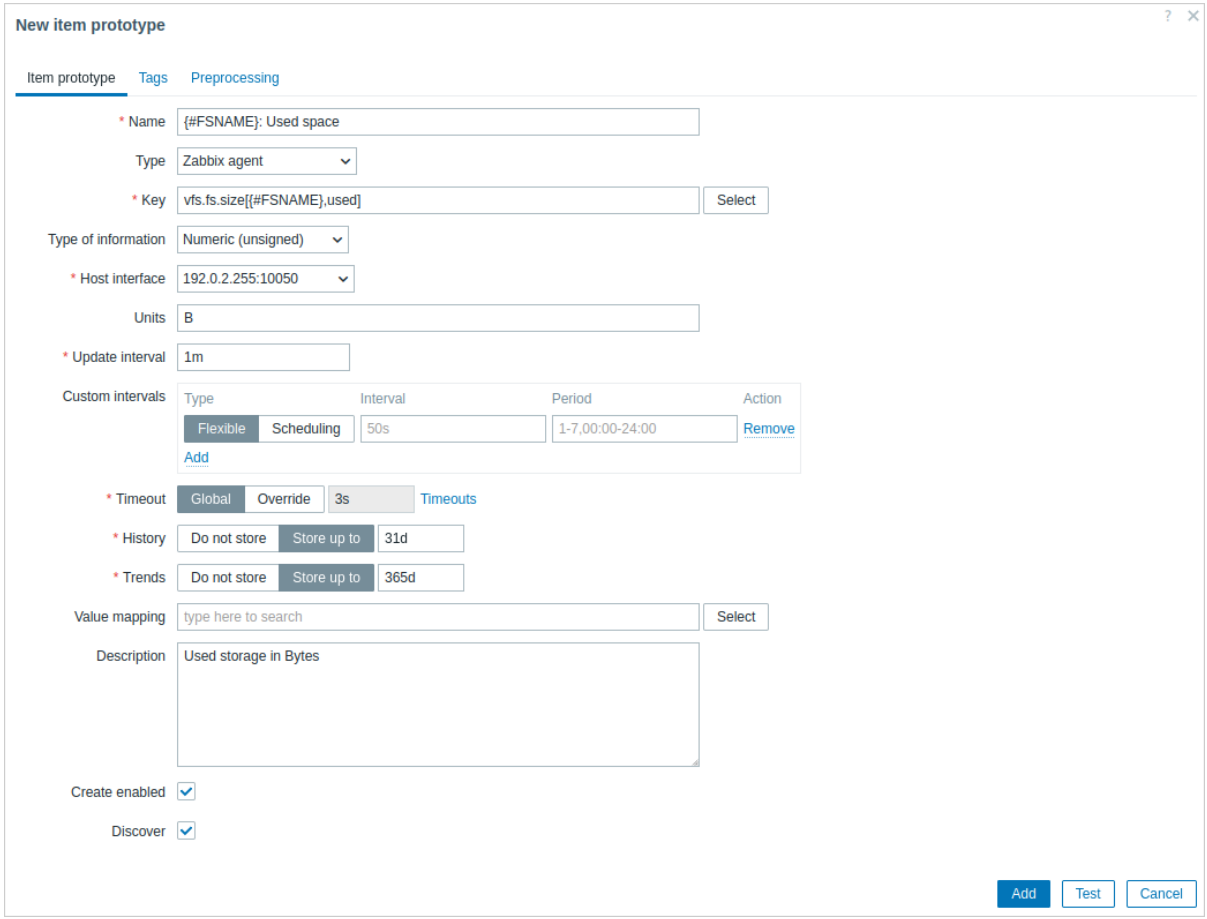
- Discovery von Netzwerkschnittstellen
- Discovery von CPUs und CPU-Kernen
- Discovery von SNMP-OIDs
- Discovery von JMX-Objekten;
- Discovery mit ODBC-SQL-Abfragen
- Discovery von Windows-Diensten
- Discovery von Host-Schnittstellen in Zabbix

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.

1 Datenpunkt-Prototypen

Sobald eine Regel erstellt wurde, gehen Sie zu den Datenpunkten für diese Regel und klicken Sie auf „Datenpunkt-Prototyp erstellen“, um einen Datenpunkt-Prototyp zu erstellen.

Beachten Sie, wie das Makro {#FSNAME} dort verwendet wird, wo ein Dateisystemname erforderlich ist. Die Verwendung eines Low-Level-Discovery-Makros ist im Datenpunktschlüssel zwingend erforderlich, um sicherzustellen, dass die Discovery korrekt verarbeitet wird. Wenn die Discovery-Regel verarbeitet wird, wird dieses Makro durch das erkannte Dateisystem ersetzt.



Note:
Wenn keine Parameter für den Datenpunktschlüssel verwendet werden, platzieren Sie das LLD-Makro innerhalb der Datenpunktschlüssel-Klammern `[...]` (zum Beispiel `v_[{#MACRO}]`).

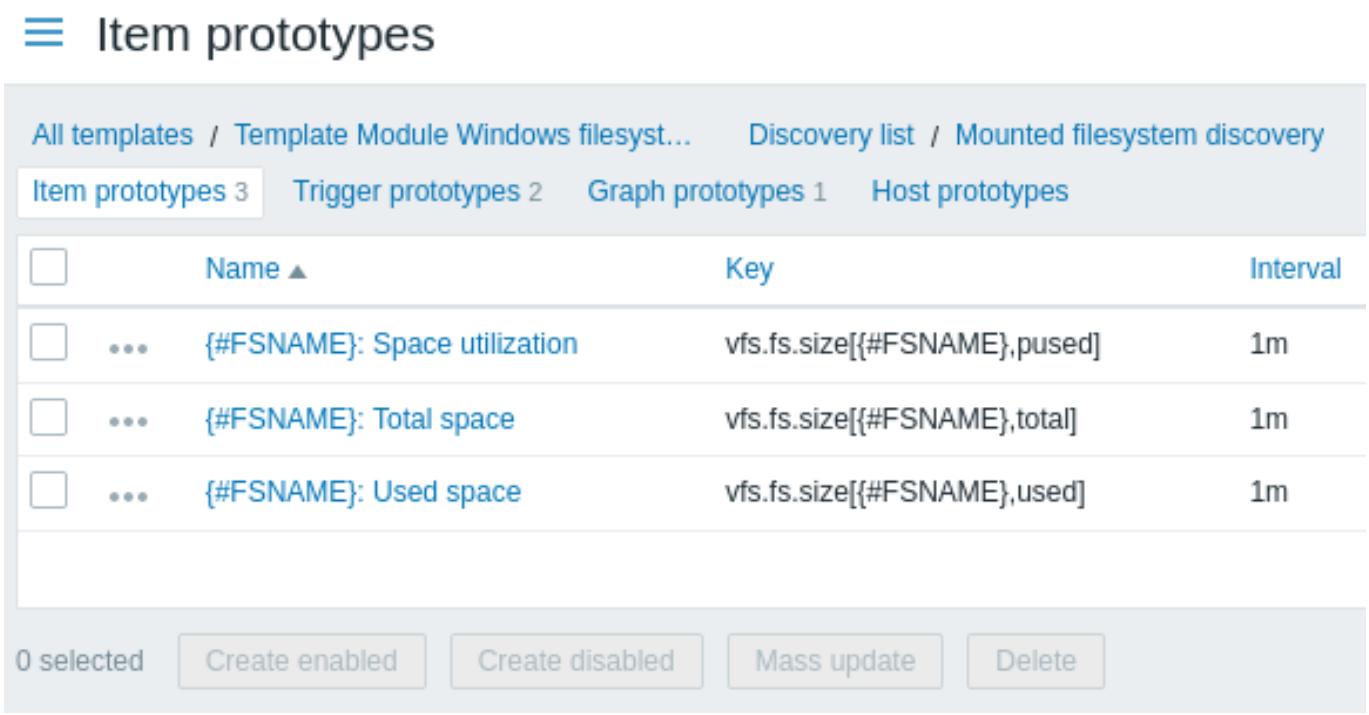
Low-Level-Discovery-Makros und Benutzer-Makros werden in der Konfiguration von Datenpunkt-Prototypen und in den Parametern der Datenpunkt-Wertvorverarbeitung unterstützt.Beachten Sie, dass bei Verwendung in Aktualisierungsintervallen ein einzelnes Makro das gesamte Feld ausfüllen muss. Mehrere Makros in einem Feld oder mit Text gemischte Makros werden nicht unterstützt.

Note:
Kontextspezifisches Escaping von Low-Level-Discovery-Makros wird durchgeführt, damit sie sicher in Parametern der Vorverarbeitung für reguläre Ausdrücke und XPath verwendet werden können.

Attribute, die spezifisch für Datenpunkt-Prototypen sind:

Parameter	Beschreibung
Create enabled	Wenn diese Option aktiviert ist, wird der Datenpunkt im aktivierten Zustand hinzugefügt. Wenn diese Option deaktiviert ist, wird der Datenpunkt zu einer erkannten Entität hinzugefügt, jedoch im deaktivierten Zustand.
Discover	Wenn diese Option aktiviert ist (Standard), wird der Datenpunkt zu einer erkannten Entität hinzugefügt. Wenn diese Option deaktiviert ist, wird der Datenpunkt nicht zu einer erkannten Entität hinzugefügt, es sei denn, diese Einstellung wird in der Discovery-Regel überschrieben.

Wir können mehrere Datenpunkt-Prototypen für jede Dateisystemmetrik erstellen, die uns interessiert:



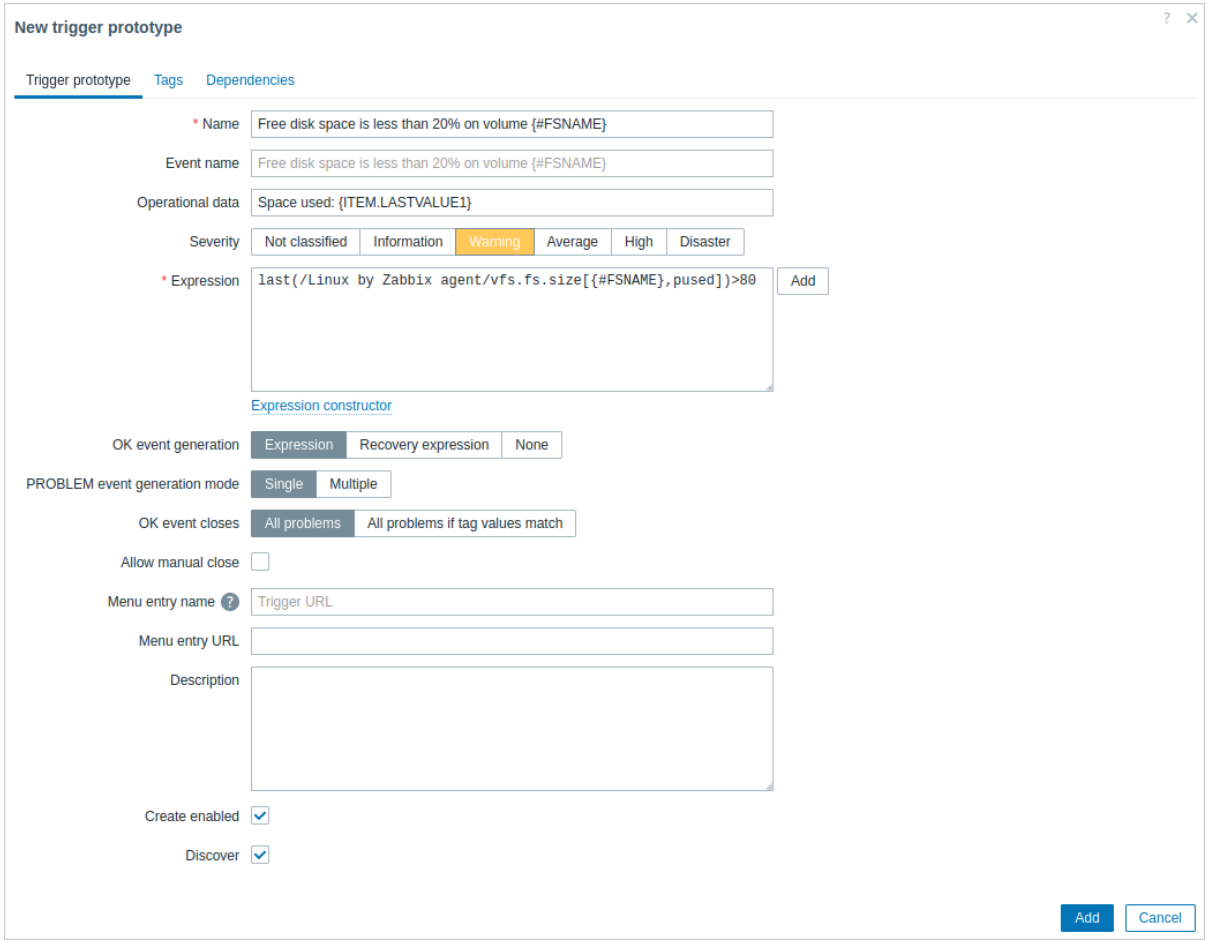
Klicken Sie auf das Symbol mit den drei Punkten, um das Menü für den jeweiligen Datenpunkt-Prototyp mit folgenden Optionen zu öffnen:

- *Create trigger prototype* - einen Auslöser-Prototyp basierend auf diesem Datenpunkt-Prototyp erstellen
- *Trigger prototypes* - klicken Sie hier, um eine Liste mit Links zu bereits konfigurierten Auslöser-Prototypen dieses Datenpunkt-Prototyps anzuzeigen
- *Create dependent item* - einen abhängigen Datenpunkt für diesen Datenpunkt-Prototyp erstellen

Die Option *Mass update* ist verfügbar, wenn Sie Eigenschaften mehrerer Datenpunkt-Prototypen gleichzeitig aktualisieren möchten.

2 Auslöser-Prototypen

Wir erstellen Auslöser-Prototypen auf ähnliche Weise wie Datenpunkt-Prototypen:

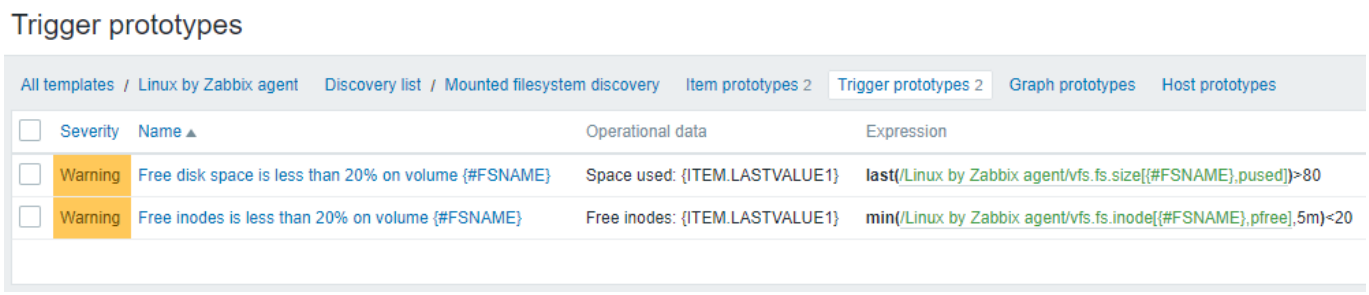


Attribute, die spezifisch für Auslöser-Prototypen sind:

Parameter	Beschreibung
Create enabled	Wenn diese Option aktiviert ist, wird der Auslöser im aktivierten Zustand hinzugefügt. Wenn diese Option deaktiviert ist, wird der Auslöser zu einer entdeckten Entität hinzugefügt, jedoch im deaktivierten Zustand.
Discover	Wenn diese Option aktiviert ist (Standard), wird der Auslöser zu einer entdeckten Entität hinzugefügt. Wenn diese Option deaktiviert ist, wird der Auslöser nicht zu einer entdeckten Entität hinzugefügt, es sei denn, diese Einstellung wird in der Discovery-Regel überschrieben.

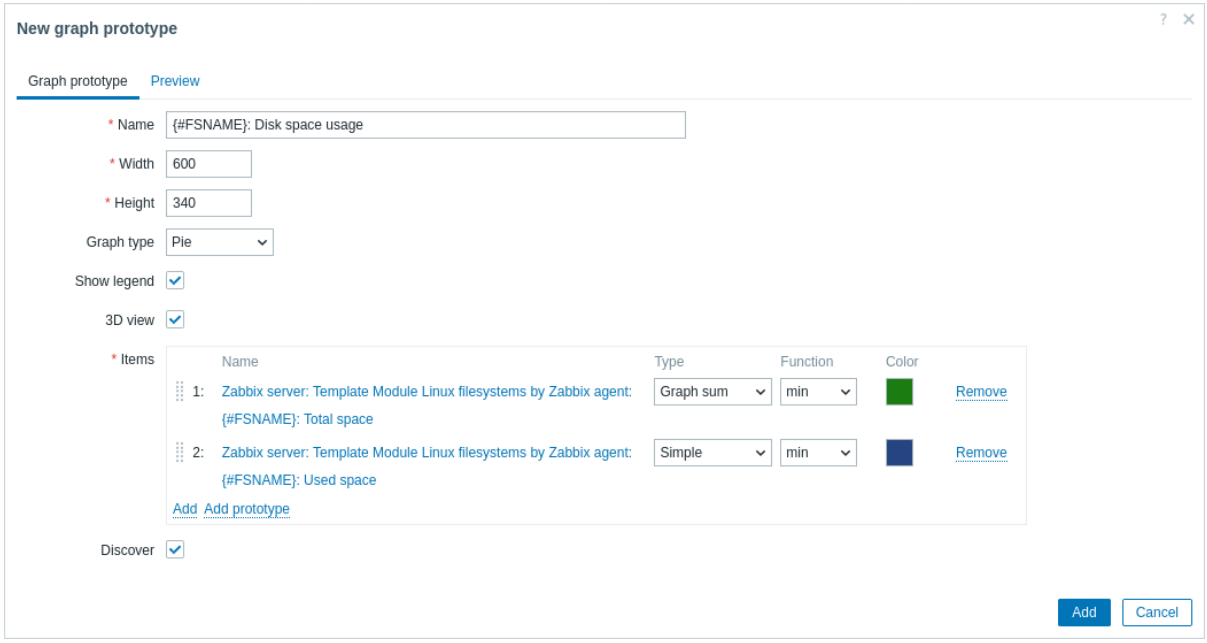
Wenn aus den Prototypen echte Auslöser erstellt werden, kann es erforderlich sein, flexibel festzulegen, welche Konstante ('20' in unserem Beispiel) für den Vergleich im Ausdruck verwendet wird. Siehe, wie Benutzermakros mit Kontext nützlich sein können, um eine solche Flexibilität zu erreichen.

Sie können Abhängigkeiten zwischen Auslöser-Prototypen definieren. Gehen Sie dazu auf die Registerkarte *Dependencies*. 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.



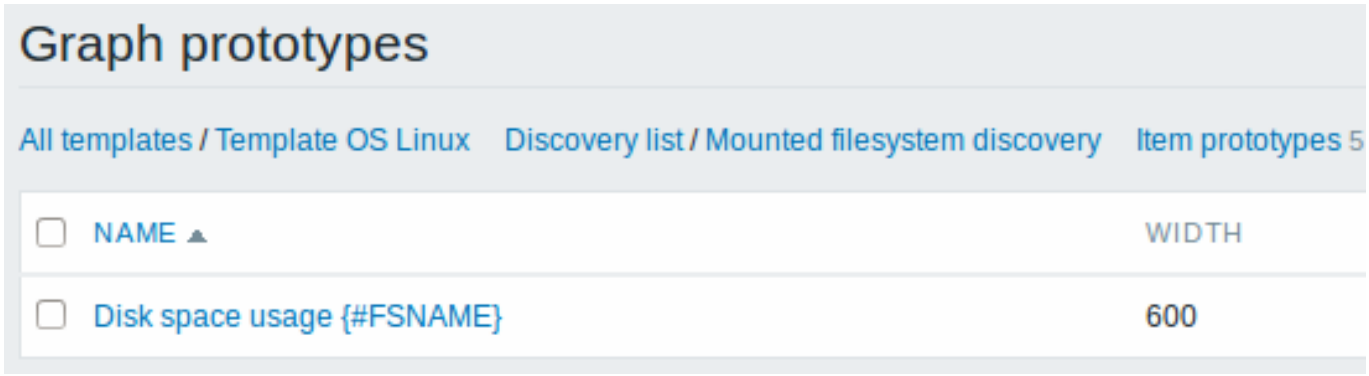
3 Graph-Prototypen

Wir können auch Graph-Prototypen erstellen:

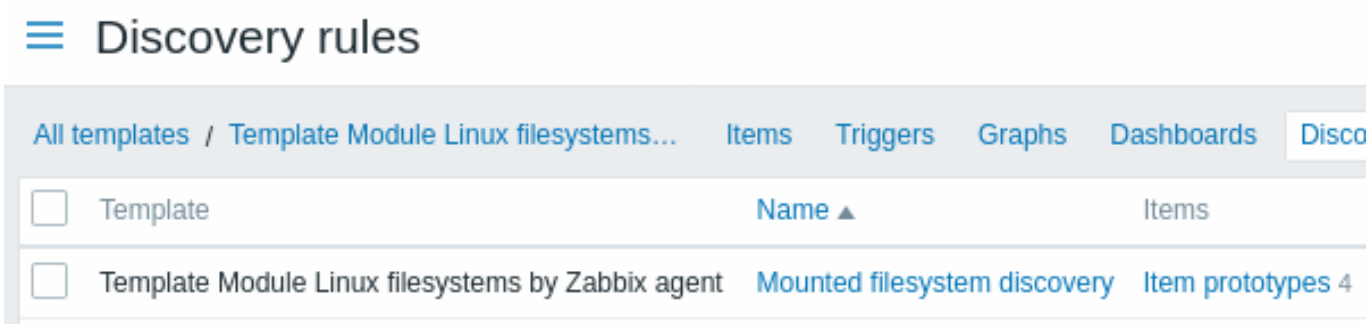


Attribute, die spezifisch für Graph-Prototypen sind:

Parameter	Beschreibung
Discover	Falls aktiviert (Standard), wird der Graph zu einer entdeckten Entität hinzugefügt. Falls deaktiviert, wird der Graph nicht zu einer entdeckten Entität hinzugefügt, es sei denn, diese Einstellung wird in der Discovery-Regel überschrieben.



Schließlich haben wir eine Discovery-Regel erstellt, die wie unten dargestellt aussieht. Sie hat fünf Datenpunkt-Prototypen, zwei Auslöser-Prototypen und einen Graph- Prototyp.



4 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-Prototypen werden unter Low-Level-Discovery-Regeln konfiguriert.

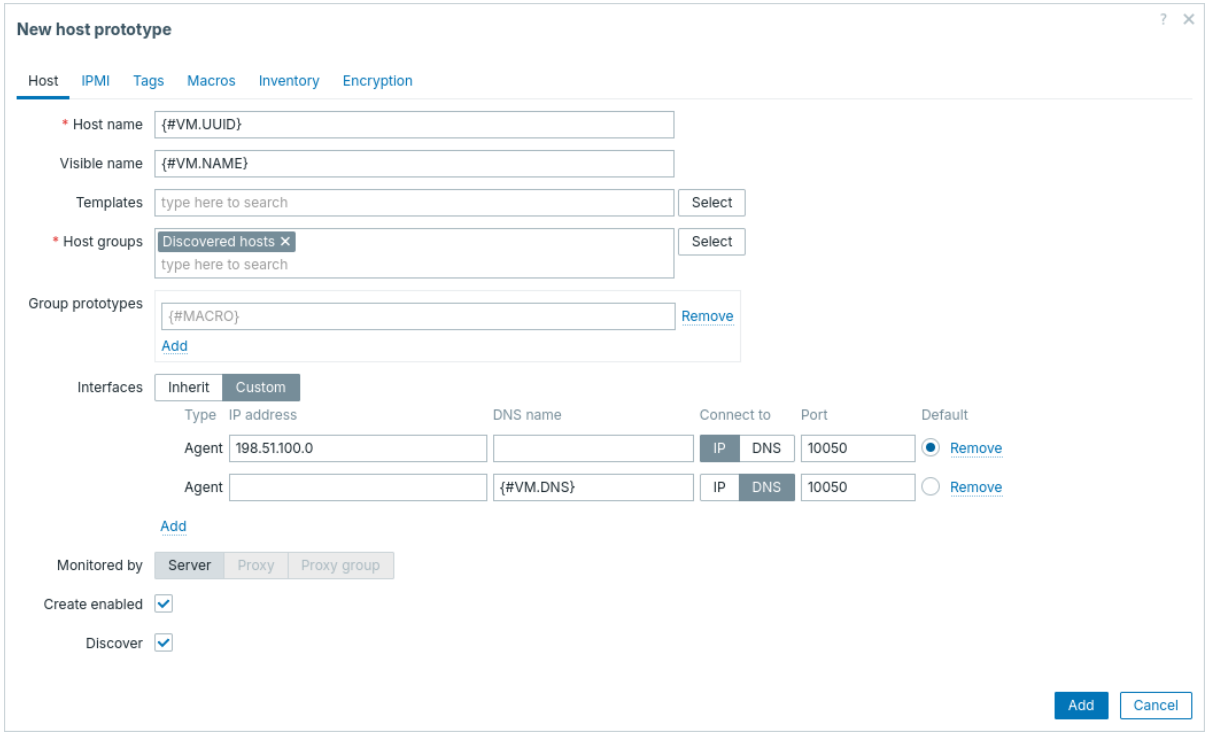
So erstellen Sie einen Host-Prototypen:

1. Gehen Sie zu *Datenerfassung → Hosts*.

2. Klicken Sie beim gewünschten Host auf *Discovery*, um zur Liste der für diesen Host konfigurierten Low-Level-Discovery-Regeln zu gelangen.

3. Klicken Sie bei der gewünschten Discovery-Regel auf *Host prototypes*.

4. Klicken Sie oben rechts auf die Schaltfläche *Create host prototype* und füllen Sie das Konfigurationsformular aus.



Alle erforderlichen Eingabefelder 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-Prototypen mithilfe von Low-Level-Discovery-Makros. Basierend auf den angegebenen Gruppen-Prototypen werden Hostgruppen erkannt, erstellt und mit den erstellten Hosts verknüpft; erkannte Gruppen, die bereits durch andere Low-Level-Discovery-Regeln erstellt wurden, werden ebenfalls mit den erstellten Hosts verknüpft. Bereits erkannte Hostgruppen, die mit manuell erstellten Hostgruppen übereinstimmen, werden jedoch nicht mit den erstellten Hosts verknüpft.
Interfaces	Legen Sie fest, ob erkannte Hosts die IP von dem 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 erkannter Hosts fest; wenn diese Option nicht aktiviert ist, werden Hosts als deaktiviert erstellt.
Discover	Legen Sie fest, ob Hosts aus dem Host-Prototyp erstellt werden; wenn diese Option nicht aktiviert ist, werden keine Hosts aus dem Host-Prototyp 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 Werte von Benutzermakros in Host-Prototypen unterstützt.
 Wertzuordnungen werden für Host-Prototypen nicht unterstützt.

Ein Beispiel für die Konfiguration eines Host-Prototyps 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).

5 Discovery-Prototypen

Übersicht

Discovery-Prototypen sind **verschachtelte Low-Level-Discovery**-Regeln innerhalb einer „übergeordneten“ Discovery-Regel und ermöglichen die mehrstufige Discovery von Objekten mit ihren eigenen Datenpunkten, Auslösern usw. Beispielsweise möchten Sie möglicherweise zunächst alle Datenbankinstanzen auf einem Datenbank-Server entdecken, dann für jede Instanz die Tablespaces und anschließend für jeden Tablespace die Tabellen.

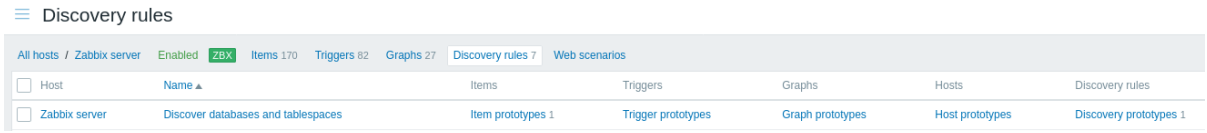
Discovery-Prototypen verfügen über eigene Datenpunkt-, Auslöser-, Graph-, Host- und Discovery-Prototypen. Ein verschachtelter Discovery-Prototyp verwendet denselben JSON-Wert wie die übergeordnete Regel, wenn Sie den Typ *Verschachtelt* angeben.

Die Anzahl der Verschachtelungsebenen für Discovery-Prototypen ist unbegrenzt.

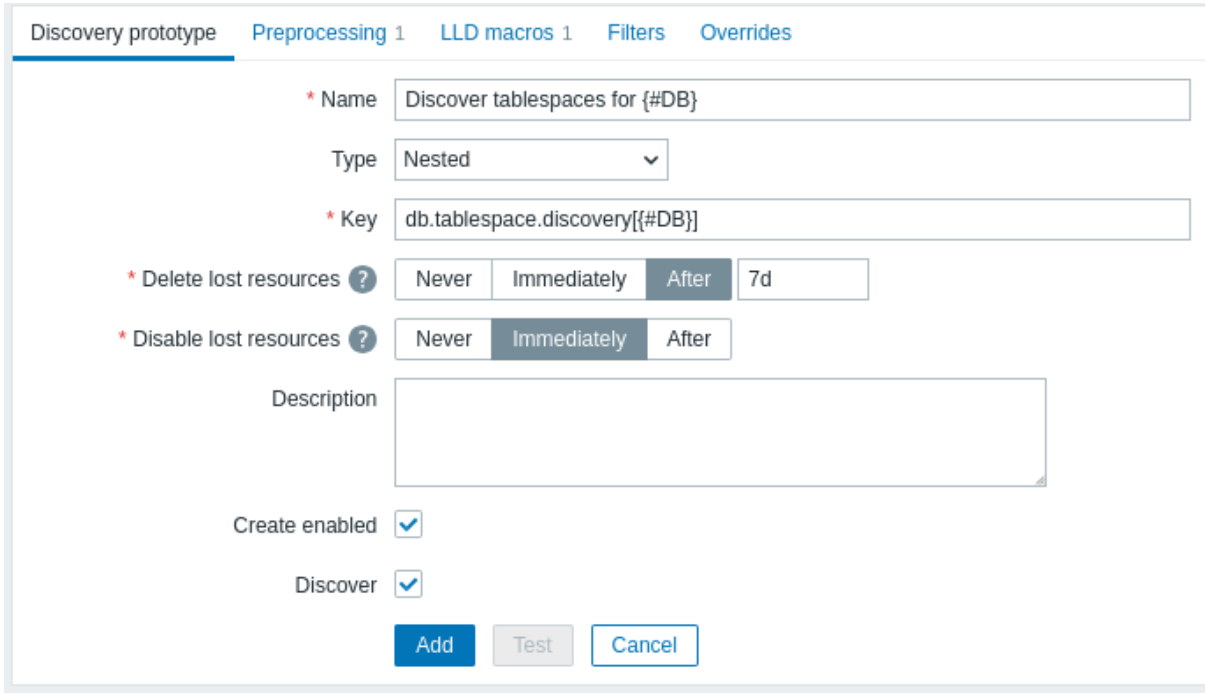
Konfiguration

So erstellen Sie einen Suchlaufprototypen:

- Klicken Sie in der Zeile einer vorhandenen Suchlaufregel auf *Suchlaufprototypen*



- Klicken Sie auf *Suchlaufprototyp erstellen*



Die Konfigurationsfelder dieses Formulars entsprechen denen der regulären Low-Level-Discovery.

Wenn Sie im geöffneten Formular für den Suchlaufprototypen als *Typ* „Verschachtelt“ auswählen, werden Suchlaufregeln (aus dem Suchlaufprototypen) auf Basis eines JSON-Objekts aus demselben JSON-Wert wie bei der übergeordneten Suchlaufregel erzeugt. Wenn das ursprüngliche JSON beispielsweise [<Objekt A>, <Objekt B>] ist und es einen verschachtelten Suchlaufregelprototypen gibt, dann würden entsprechend zwei Suchlaufregeln auf Basis der Daten von Objekt A bzw. Objekt B erzeugt.

In diesem Fall wird der Suchlaufprototyp gleichzeitig mit der übergeordneten Regel aktiviert. Die verschachtelte Regel kann daher die 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 Suchlaufregeln 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 von mehrstufigem 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

Erkennen von Datenbankinstanzen auf einem Datenbank-Server und anschließendes Erkennen der Tablespaces für jede Instanz.

1. Sie haben mindestens einen Host, der mit der Erkennung des Datenbank-Servers verknüpft ist.
2. Erstellen Sie für diesen Host eine LLD-Regel mit dem Namen *Datenbanken und Tablespaces erkennen*.
3. Wechseln Sie bei 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 zu jeder Datenbank gehörenden Datenpunkte 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: Verschachtelt, Schlüssel: `db.tablespace.discovery[{#DB}]`).
7. Wechseln Sie beim diesem Discovery-Prototyp auf den Reiter *Vorverarbeitung* und fügen Sie den Schritt `JSONPath=$.tablespaces` hinzu.
8. Wechseln Sie bei diesem Discovery-Prototyp auf den Reiter *LLD-Makros* und fügen Sie das Makro `{#TSNAME}=$.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 zu jedem Tablespace jeder Datenbank gehörenden Datenpunkte 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 Datenpunktprototyp mit dem Namen *Größe des Tablespace {#TSNAME} für {#DB}* (Typ: Agent, Schlüssel: `db.ts.size[{#DB}, {#TSNAME}]`).
8. Erstellen Sie wieder 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

6 Hinweise zur Low-Level-Discovery

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, der die Verwendung geänderter Discovery-Datenpunktschlüssel in verschiedenen Discovery-Regeln ermöglicht, 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.

7 Discovery-Regeln

Bitte verwenden Sie die Seitenleiste, um Beispiele für die Konfiguration von Discovery-Regeln für verschiedene Fälle anzuzeigen.

1 Erkennung eingehängter Dateisysteme

Übersicht

Es ist möglich, eingehängte Dateisysteme und ihre Eigenschaften zu erkennen:

- Name des Einhängepunkts
- Dateisystemtyp
- Dateisystemgröße
- Inode-Statistiken
- Einhängeoptionen

Dazu können Sie eine Kombination aus Folgendem verwenden:

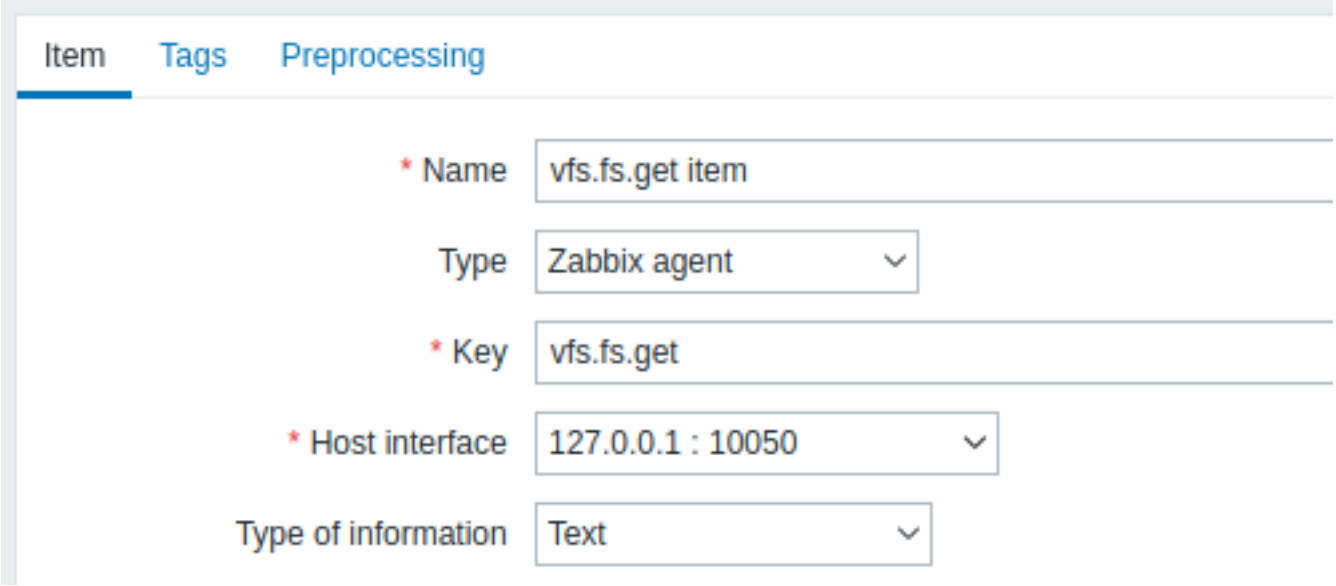
- dem Agent-Datenpunkt `vfs.fs.get` als Master-Datenpunkt
- einer abhängigen Low-Level-Discovery-Regel und Datenpunkt-Prototypen

Konfiguration

Master-Datenpunkt

Erstellen Sie einen Zabbix-Agent-Datenpunkt mit dem folgenden Schlüssel:

`vfs.fs.get`



Setzen Sie den Informationstyp auf „Text“ für möglicherweise große JSON-Daten.

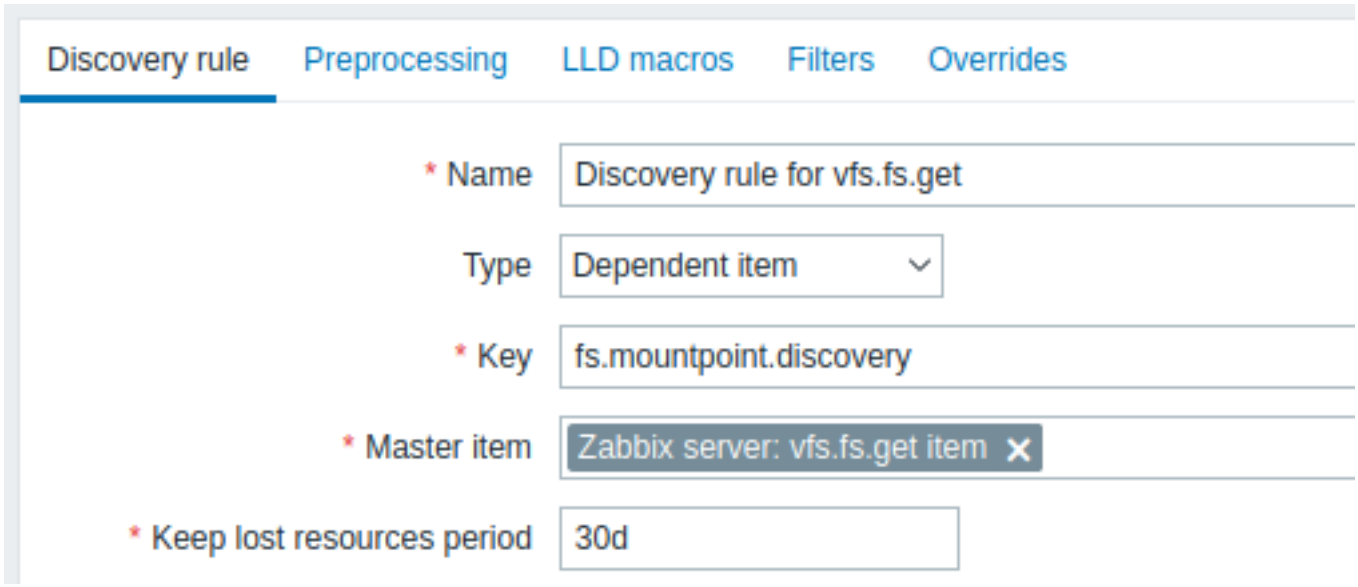
Die von diesem Datenpunkt zurückgegebenen Daten enthalten für ein eingehängtes Dateisystem etwa Folgendes:

```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
```

```
  }
]
```

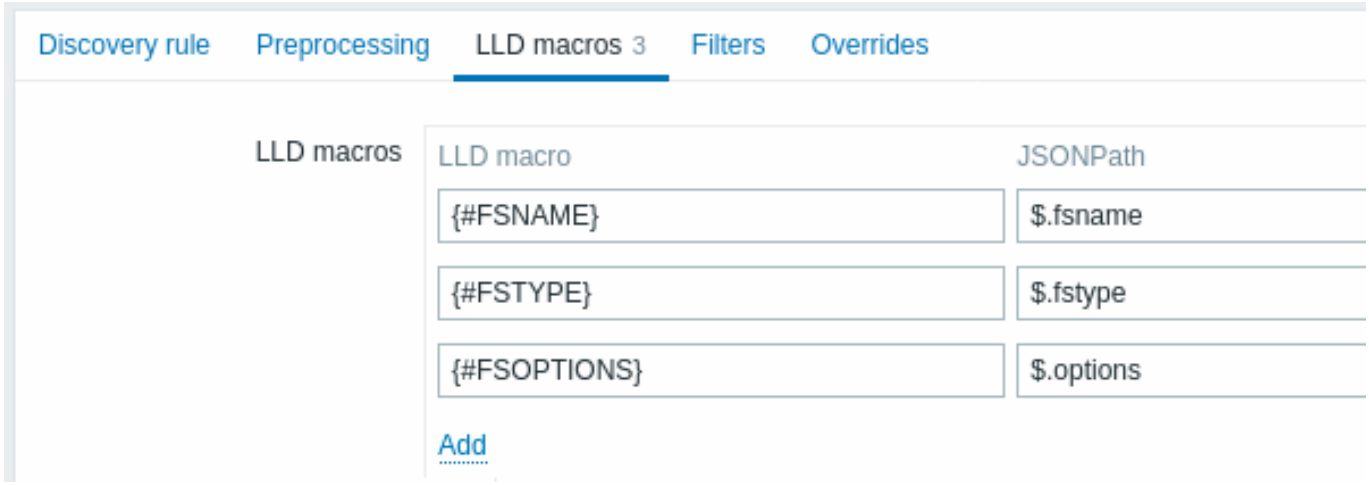
Abhängige LLD-Regel

Erstellen Sie eine Low-Level-Discovery-Regel vom Typ „Dependent item“:

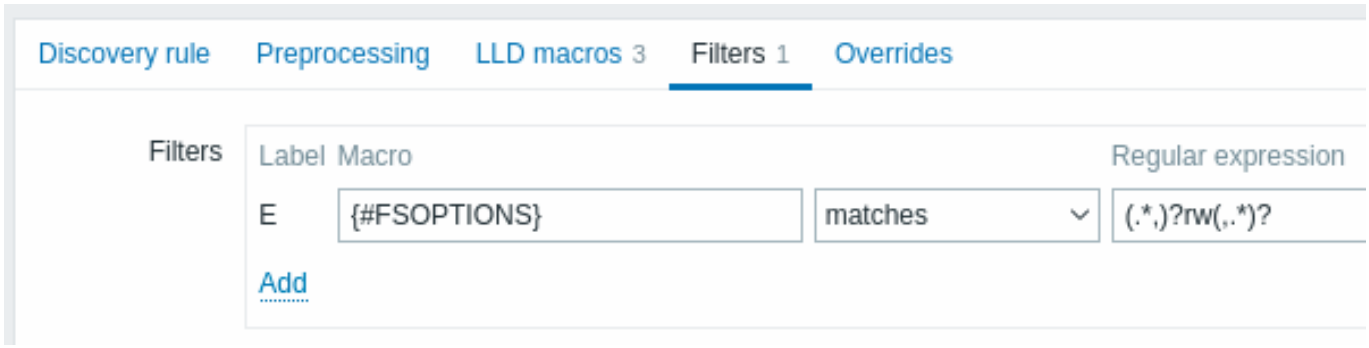


Wählen Sie als Master-Datenpunkt den Datenpunkt `vfs.fs.get` aus, den wir erstellt haben.

Definieren Sie auf der Registerkarte „LLD-Makros“ benutzerdefinierte Makros mit dem entsprechenden JSONPath:

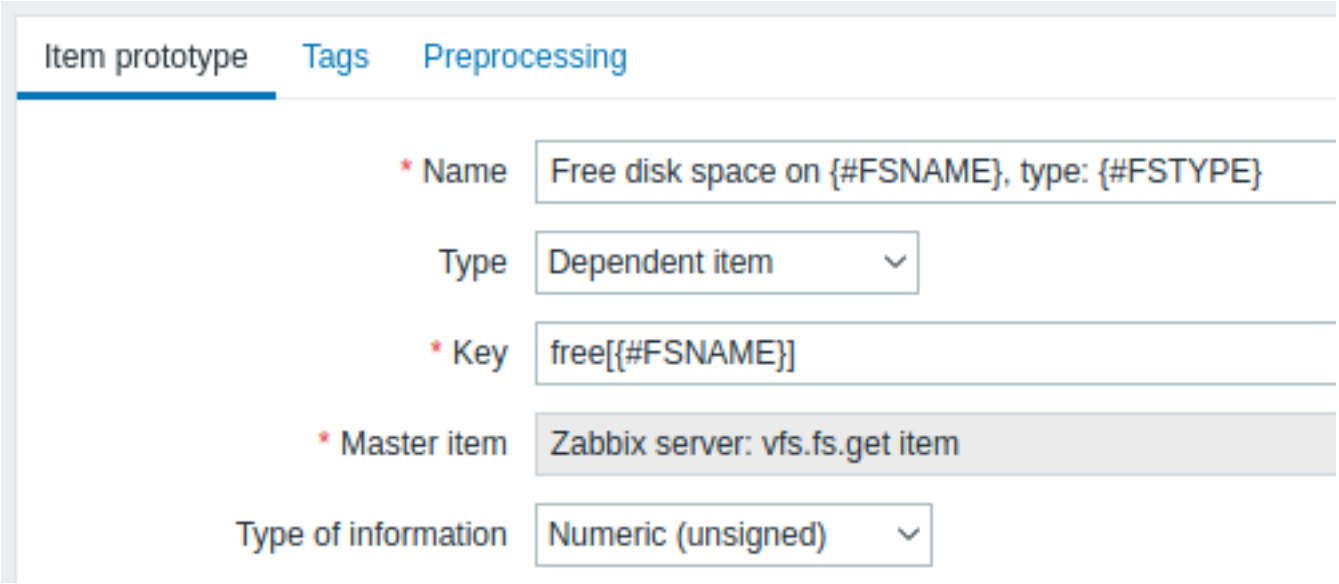


Auf der Registerkarte „Filter“ können Sie einen regulären Ausdruck hinzufügen, der nur **read-write**-Dateisysteme filtert:



Prototyp für abhängigen Datenpunkt

Erstellen Sie in dieser LLD-Regel einen Datenpunkt-Prototypen vom Typ „Dependent item“. Wählen Sie als Master-Datenpunkt für diesen Prototypen den Datenpunkt `vfs.fs.get` aus, den wir erstellt haben.



Beachten Sie die Verwendung benutzerdefinierter Makros im Namen und Schlüssel des Datenpunkt-Prototyps:

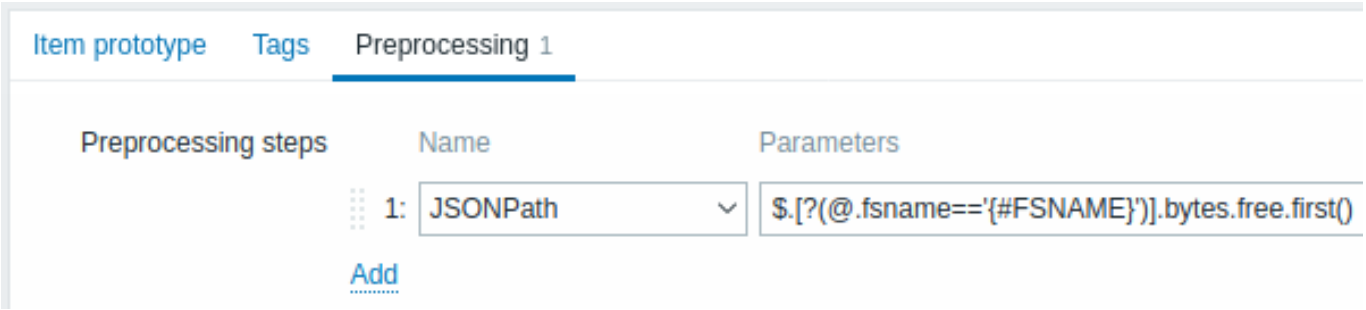
- *Name*: Freier Festplattenspeicher auf {#FSNAME}, Typ: {#FSTYPE}
- *Key*: Free[{#FSNAME}]

Verwenden Sie als Informationstyp:

- *Numeric (unsigned)* für Metriken wie „free“, „total“, „used“
- *Numeric (float)* für Metriken wie „pfree“, „pused“ (Prozent)

Wählen Sie im Reiter „Preprocessing“ des Datenpunkt-Prototyps JSONPath aus und verwenden Sie den folgenden JSONPath-Ausdruck als Parameter:

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



Sobald die Discovery startet, wird ein Datenpunkt für jeden einzelnen Einhängepunkt erstellt. Dieser Datenpunkt gibt die Anzahl freier Bytes für den jeweiligen Einhängepunkt zurück.

2 Erkennung von Netzwerkschnittstellen

Ähnlich wie Dateisysteme erkannt werden, ist es auch möglich, Netzwerkschnittstellen zu erkennen.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunktschlüssel ist

```
net.if.discovery
```

Unterstützte Makros

Sie können das Makro {#IFNAME} im Filter der Discovery-Regel und in Prototypen von Datenpunkten, Auslösern und Diagrammen verwenden.

Beispiele für Datenpunkt-Prototypen, die Sie möglicherweise auf Basis von ”net.if.discovery” erstellen möchten:

- ”net.if.in[{#IFNAME},bytes]”,
- ”net.if.out[{#IFNAME},bytes]”.

Beachten Sie, dass unter Windows auch {#IFGUID} zurückgegeben wird.

3 Erkennung von CPUs und CPU-Kernen

Ähnlich wie Dateisysteme erkannt werden, ist es auch möglich, CPUs und CPU-Kerne zu erkennen.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunktschlüssel ist

`system.cpu.discovery`

Unterstützte Makros

Dieser Discovery-Schlüssel gibt zwei Makros zurück – {#CPU.NUMBER} und {#CPU.STATUS}, die jeweils die Reihennummer und den Status der CPU angeben. Er zählt Prozessoren von 0 bis N - 1 auf, wobei N die Gesamtzahl der Prozessoren ist.

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" gibt es nur unter Windows */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

Beachten Sie, dass keine klare Unterscheidung zwischen tatsächlichen, physischen Prozessoren, Kernen und Hyperthreads getroffen werden kann. {#CPU.STATUS} gibt auf Linux-, UNIX- und BSD-Systemen den Status des Prozessors zurück, der entweder "online" oder "offline" sein kann. Auf Windows-Systemen kann dasselbe Makro auch einen dritten Wert darstellen – "unknown" –, was darauf hinweist, dass ein Prozessor erkannt wurde, für ihn jedoch noch keine Informationen erfasst wurden.

Die CPU-Erkennung stützt sich auf den Collector-Prozess des Agent, um mit den vom Collector bereitgestellten Daten konsistent zu bleiben und Ressourcen bei der Datenermittlung zu sparen. Dies führt dazu, dass dieser Datenpunktschlüssel nicht mit dem Kommandozeilen-Flag (-t) der Agent-Binärdatei funktioniert; stattdessen wird der Status NOT_SUPPORTED zusammen mit einer Meldung zurückgegeben, die angibt, dass der Collector-Prozess nicht gestartet wurde.

Datenpunkt-Prototypen, die auf Basis der CPU-Erkennung erstellt werden können, sind zum Beispiel:

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

Eine detaillierte Beschreibung der Datenpunktschlüssel finden Sie unter Zabbix-Agent-Datenpunktschlüssel.

Hinweise zu erkannten Performance-Counter-Datenpunkten auf NUMA-Systemen

Einige Windows-Performance-Counter sind für einige der logischen Prozessoren auf NUMA-Systemen möglicherweise nicht verfügbar.

Zum Beispiel funktionieren Datenpunkte, die mit einem Datenpunktprototyp mit dem untenstehenden Schlüssel erkannt werden, möglicherweise nur für den ersten NUMA-Knoten. Datenpunkte für die anderen NUMA-Knoten können sich im nicht unterstützten Zustand befinden.

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

Außerdem liefert eine Discovery-Regel mit dem Schlüssel "system.cpu.discovery" keine Low-Level-Discovery-Makrowerte, die in Performance-Counter wie die folgenden eingesetzt werden könnten:

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

Während system.cpu.util-Datenpunkte je nach Anzahl der Prozessorgruppen und Anzahl der Prozessoren unterschiedliche Performance-Counter zur Überwachung der CPU-Auslastung in Prozent verwenden. Außerdem gibt es einen seltenen Bug im Zusammenhang mit Prozessorgruppen unter Windows.

4 Erkennung von SNMP-OIDs

Übersicht

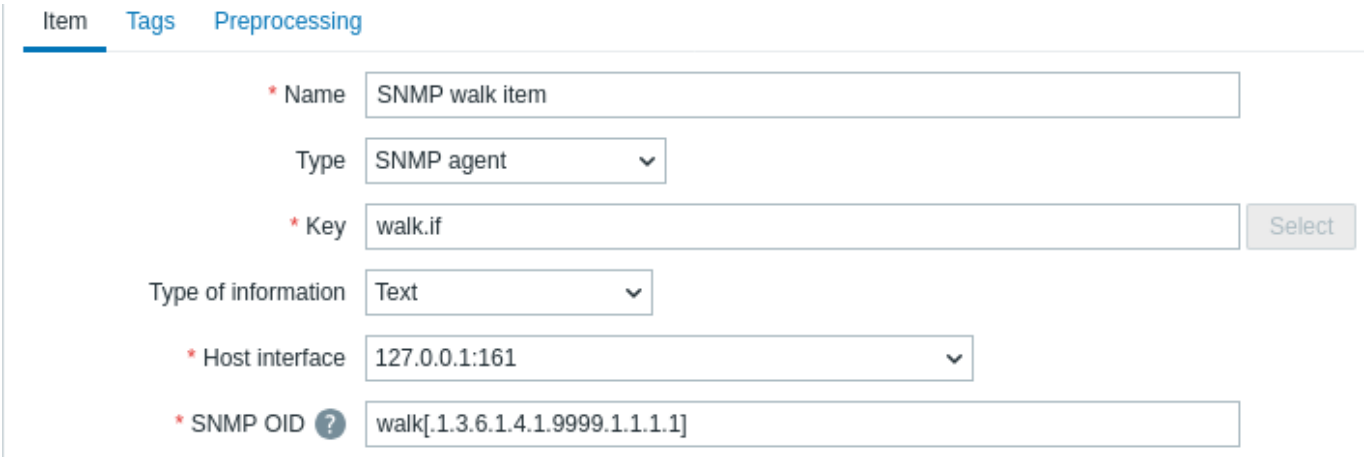
In diesem Abschnitt führen wir eine Low-Level-Discovery auf einem SNMP-Gerät durch.

Diese Erkennungsmethode für SNMP-OIDs wird seit Zabbix Server/Proxy 6.4 unterstützt.

Beispielkonfiguration

1. Erstellen Sie einen SNMP-Agent-Datenpunkt mit einem Schlüssel wie diesem:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



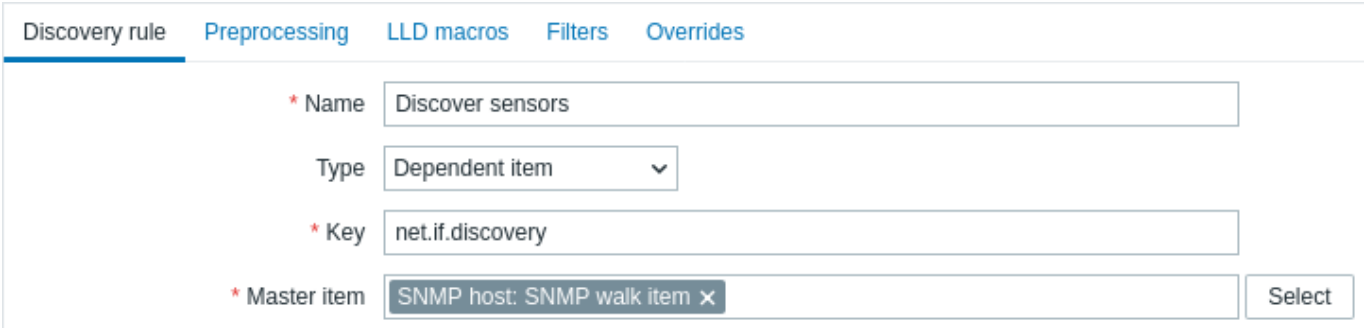
Dieser Datenpunkt führt einen einzelnen SNMP-Tabellen-Walk aus und gibt alle Tabelleneinträge in einer Anfrage zurück, in einem Format, das der Ausgabe des Dienstprogramms `snmpwalk` mit den Formatierungsoptionen `-Oe -Ot -On` entspricht.

Er gibt den folgenden mehrzeiligen Textwert zurück:

```
.1.3.6.1.4.1.9999.1.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.1.3.2 = 200
```

2. Erstellen Sie eine Discovery-Regel:

- Geben Sie im Feld *Name* einen aussagekräftigen Namen für die Discovery-Regel ein (z. B. „Sensoren erkennen“).
- Wählen Sie im Feld *Type* „Dependent item“ aus.
- Geben Sie im Feld *Key* einen aussagekräftigen Schlüssel ein (z. B. „net.if.discovery“).
- Wählen Sie im Feld *Master item* „SNMP walk item“ aus.



3. Fügen Sie auf der Registerkarte *Preprocessing* einen Vorverarbeitungsschritt mit „SNMP walk to JSON“ in der Dropdown-Liste *Name* mit 3 Parametern hinzu:

- *Field name*: „{#SENSORNAME}“; *OID prefix*: „.1.3.6.1.4.1.9999.1.1.1.1.1“: *Format*: „Unchanged“.
- *Field name*: „{#SENSORTYPE}“; *OID prefix*: „.1.3.6.1.4.1.9999.1.1.1.1.2“: *Format*: „Unchanged“.
- *Field name*: „{#SENSORVALUE}“; *OID prefix*: „.1.3.6.1.4.1.9999.1.1.1.1.3“: *Format*: „Unchanged“.

Nach der Vorverarbeitung gibt die Discovery-Regel ein JSON-Array von Makrosätzen zurück.

Zum Beispiel:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

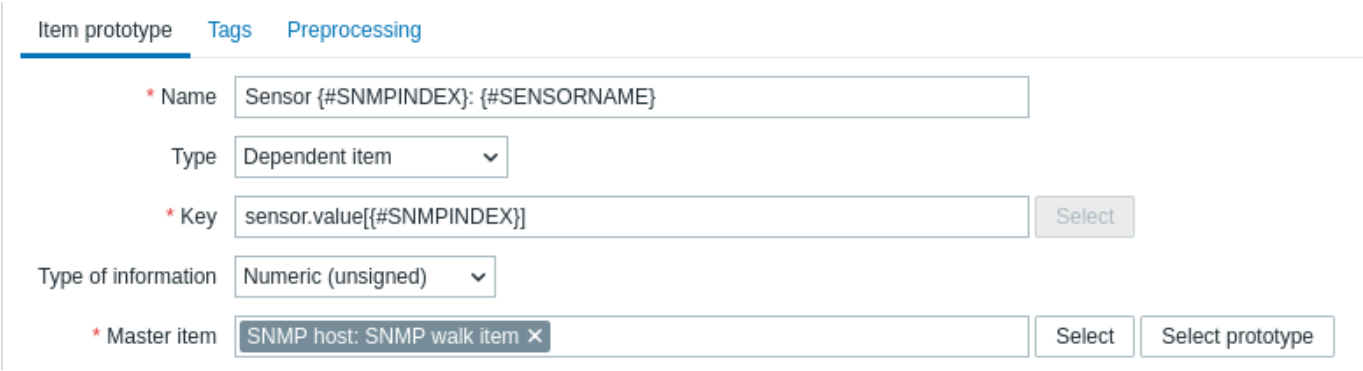
Jedes Objekt repräsentiert einen erkannten Sensor und stellt Makros wie `{#SNMPINDEX}`, `{#SENSORNAME}`, `{#SENSORTYPE}` und `{#SENSORVALUE}` bereit.

Sie werden nach dem SNMP-Index gruppiert, der das numerische Suffix am Ende jeder OID ist (z. B. .1, .2) — dieser Index identifiziert jede Zeile in der SNMP-Tabelle eindeutig und wird automatisch als `{#SNMPINDEX}` extrahiert.

4. Erstellen Sie unter der Discovery-Regel einen oder mehrere Datenpunkt-Prototypen (mit der Discovery-Regel als Master-Datenpunkt).

Zum Beispiel ein abhängiger Datenpunkt für den Sensorwert:

- Geben Sie im Feld *Name* „Sensor {#SNMPINDEX}: {#SENSORNAME}“ ein.
- Wählen Sie im Feld *Type* „Dependent item“ aus.
- Geben Sie im Feld *Key* „sensor.value[{#SNMPINDEX}]“ ein.
- Wählen Sie im Feld *Master item* „SNMP walk item“ aus.



Fügen Sie auf der Registerkarte *Preprocessing* einen Vorverarbeitungsschritt mit dem Namen „SNMP walk value“ hinzu, mit der OID „.1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}“ im Feld *Parameter*. *Format*: „Unchanged“.

Die folgenden Datenpunkte werden erkannt:

Name	Key	OID, aus der der Wert extrahiert wird	Datenpunktwert
Sensor 1: Temperature Sensor	sensor.value[1]	.1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	.1.3.6.1.4.1.9999.1.1.1.1.3.2	200

Wenn die Discovery-Regel ausgeführt wird, werden Datenpunkte wie `sensor.value[1]`, `sensor.value[2]` erstellt.

Jeder abhängige Datenpunkt extrahiert seinen Wert per Vorverarbeitung aus dem SNMP-Walk-Ergebnis des Master-Datenpunkts, ohne selbst separate SNMP-Anfragen auszuführen.

5. Referenzieren Sie abhängige Datenpunkt-Prototypen in Auslöser-Prototypen unter Verwendung derselben Makros aus der Discovery-Regel. Beispiel:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

Dadurch wird für jeden erkannten Sensor ein Auslöser erzeugt (zum Beispiel sensor.value[1], sensor.value[2]) und ausgelöst, wenn der letzte Wert (Temperatur oder Luftfeuchtigkeit) 75 überschreitet.

6. Schließen Sie abhängige Datenpunkte für jede erkannte Entität ein. Beispiel für einen Diagramm-Datenpunktschlüssel:

```
sensor.value[{#SNMPINDEX}]
```

Für jedes {#SNMPINDEX} wird ein Diagramm erstellt, das Temperatur und Luftfeuchtigkeit im Zeitverlauf darstellt.

Diese Konfiguration führt pro Abfragezyklus nur eine einzige SNMP-Walk-Anfrage aus, unabhängig von der Anzahl der erkannten Datenpunkte. Alle abhängigen Datenpunkte extrahieren ihre Werte per Vorverarbeitung aus dem SNMP-Walk-Ergebnis des Masters, wodurch SNMP-Datenverkehr und Last erheblich reduziert werden.

Dynamische Indizes mit walk[]

Dynamische Indizes (zum Beispiel Schnittstellenindizes) können sich ändern, wenn Hardware neu konfiguriert wird. Um dieses Verhalten zu berücksichtigen, wird eine Master-SNMP-Walk-Erkennungsregel mit einem Schlüssel wie diesem erstellt:

```
walk[1.3.6.1.2.1.2.2.1.10]
```

Nach der Vorverarbeitung „SNMP walk to JSON“ könnte das Ergebnis wie folgt aussehen:

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

Ein Prototyp für einen abhängigen Datenpunkt verwendet das Makro {#SNMPINDEX}, um den Schlüssel zu erstellen:

```
net.if.in[{#SNMPINDEX}]
```

Die Vorverarbeitung für diesen Prototyp umfasst den Namen „SNMP walk value“ mit der OID „1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}“ im Feld *Parameter*. *Format*: „Unchanged“.

Zur Laufzeit werden tatsächliche Datenpunkte wie net.if.in[2] und net.if.in[3] erstellt. Wenn sich ein bestimmter Schnittstellenindex ändert (zum Beispiel wenn der Index 2 in der SNMP-Tabelle durch 5 ersetzt wird), dann gilt beim nächsten Lauf der Erkennungsregel:

- Der alte abhängige Datenpunkt net.if.in[2] wird als „lost“ markiert oder entfernt, und für diesen Datenpunkt werden keine neuen Daten mehr erfasst.
- Ein neuer abhängiger Datenpunkt net.if.in[5] wird erstellt und beginnt mit einer leeren Historie.
- Verlaufsdaten von net.if.in[2] werden nicht automatisch nach net.if.in[5] verschoben.

Beispiel für einen Auslöser-Prototyp:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

Ein Beispiel für einen Graph-Prototyp enthält folgende Datenpunkte:

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

Diese Konfiguration gewährleistet eine zuverlässige Überwachung von Tabellen mit dynamischen Indizes und minimiert gleichzeitig den SNMP-Datenverkehr — pro Abfragezyklus ist nur ein einziger SNMP-Walk erforderlich, wobei die Prototypen abhängiger Datenpunkte die benötigten Werte extrahieren.

Erkannte Entitäten

Wenn der Server läuft, erstellt er reale abhängige Datenpunkte, Auslöser und Diagramme basierend auf den Werten, die die SNMP-Erkennungsregel zurückgibt.

5 Erkennung von SNMP-OIDs (veraltet)

Übersicht

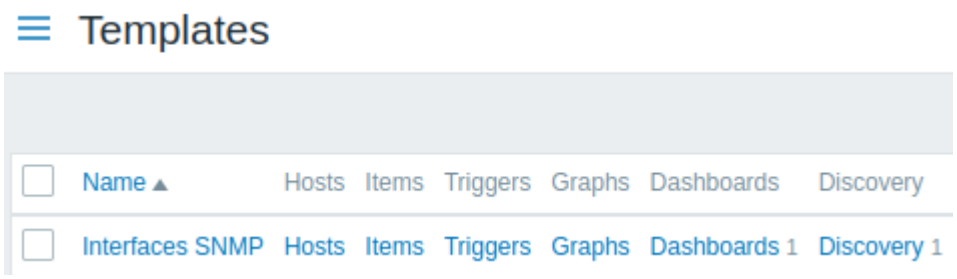
In diesem Abschnitt führen wir eine SNMP-Discovery auf einem Switch durch.

Datenpunktschlüssel

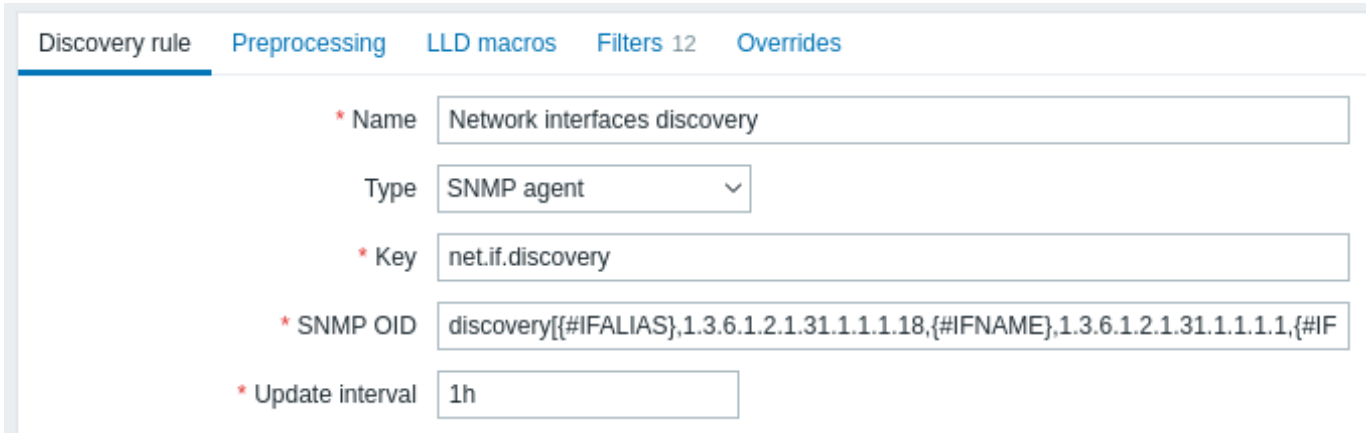
Anders als bei der Erkennung von Dateisystemen und Netzwerkschnittstellen muss der Datenpunkt nicht zwingend einen Schlüssel `"snmp.discovery"` haben - der Datenpunkttyp SNMP-Agent ist ausreichend.

Gehen Sie wie folgt vor, um die Discovery-Regel zu konfigurieren:

- Gehen Sie zu: *Datenerfassung* → *Vorlagen*
- Klicken Sie in der Zeile einer passenden Vorlage auf *Discovery*



- Klicken Sie oben rechts auf dem Bildschirm auf *Discovery-Regel erstellen*
- Füllen Sie das Formular der Discovery-Regel mit den erforderlichen Angaben wie im Screenshot unten aus



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Die zu erkennenden OIDs werden im Feld SNMP OID im folgenden Format definiert: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

wobei *{#MACRO1}*, *{#MACRO2}* ... gültige LLD-Makronamen sind und *oid1*, *oid2* ... OIDs sind, die aussagekräftige Werte für diese Makros liefern können. Ein eingebautes Makro *{#SNMPINDEX}*, das den Index der erkannten OID enthält, wird auf erkannte Entitäten angewendet. Die erkannten Entitäten werden nach dem Makrowert *{#SNMPINDEX}* gruppiert.

Hier ist ein Beispiel, bei dem einige snmpwalks auf dem Switch ausgeführt werden:

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

Und die SNMP OID ist gesetzt auf: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

Diese Regel erkennt nun Entitäten mit auf **WAN**, **LAN1** und **LAN2** gesetzten Makros {#IFDESCR}, auf **8:0:27:90:7a:75**, **8:0:27:90:7a:76** und **8:0:27:2b:af:9e** gesetzten Makros {#IFPHYSADDRESS} sowie auf die erkannten OID-Indizes **1**, **2** und **3** gesetzten Makros {#SNMPINDEX}:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
```

```
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
    }
]
```

Wenn eine Entität die angegebene OID nicht hat, wird das entsprechende Makro für diese Entität weggelassen. Wenn wir zum Beispiel die folgenden Daten haben:

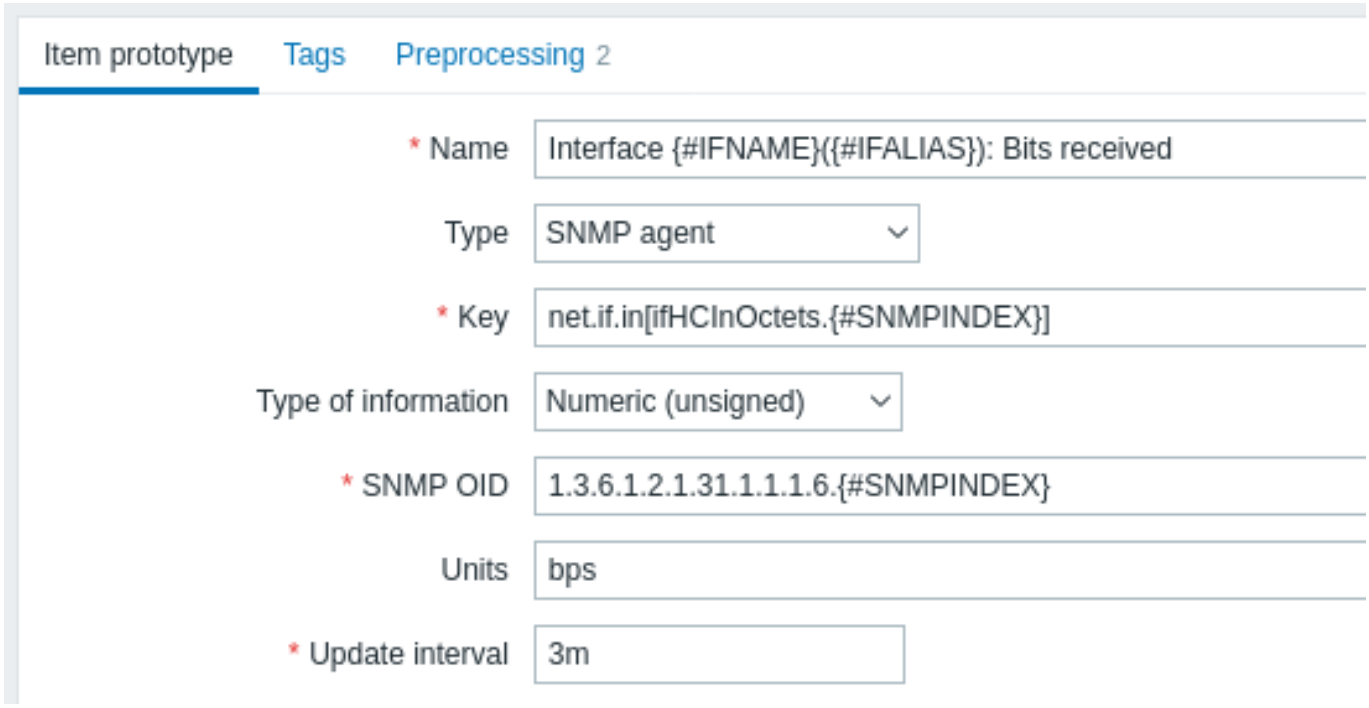
```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

Dann liefert die SNMP-Erkennung `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` in diesem Fall die folgende Struktur zurück:

```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```

Datenpunkt-Prototypen

Der folgende Screenshot veranschaulicht, wie wir diese Makros in Datenpunkt- Prototypen verwenden können:



Sie können so viele Datenpunkt-Prototypen erstellen, wie benötigt werden:

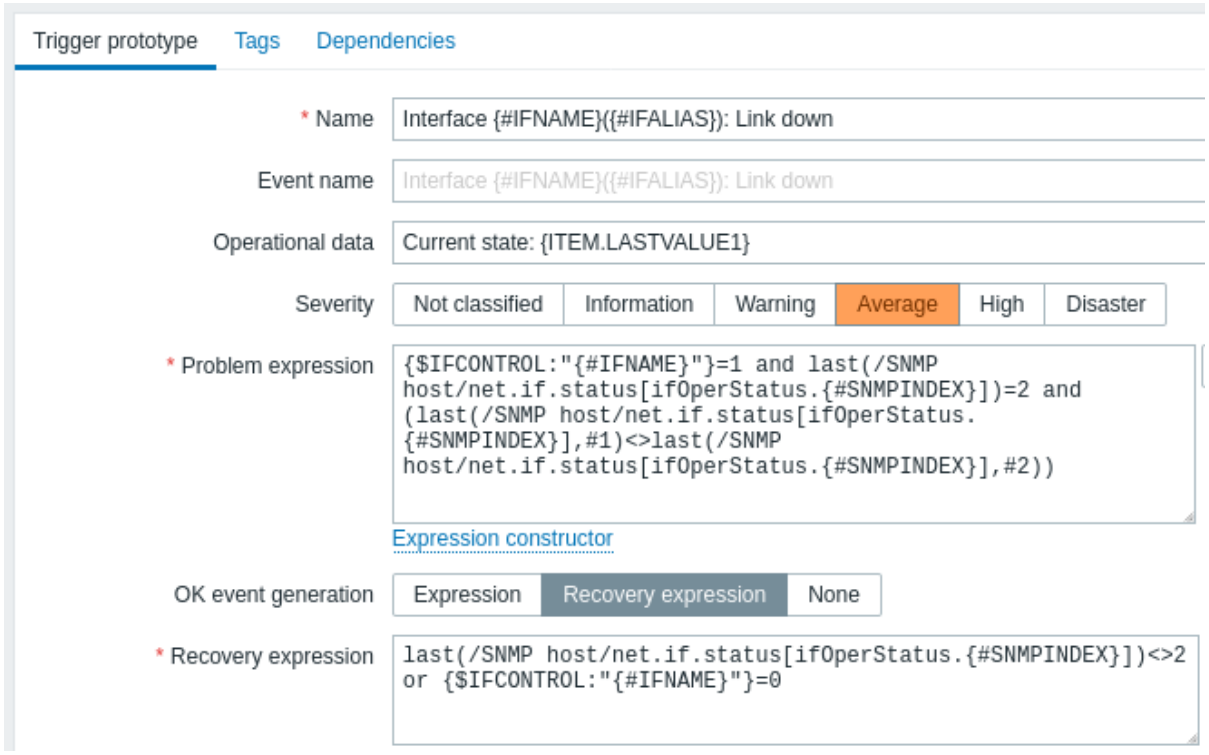
Item prototypes

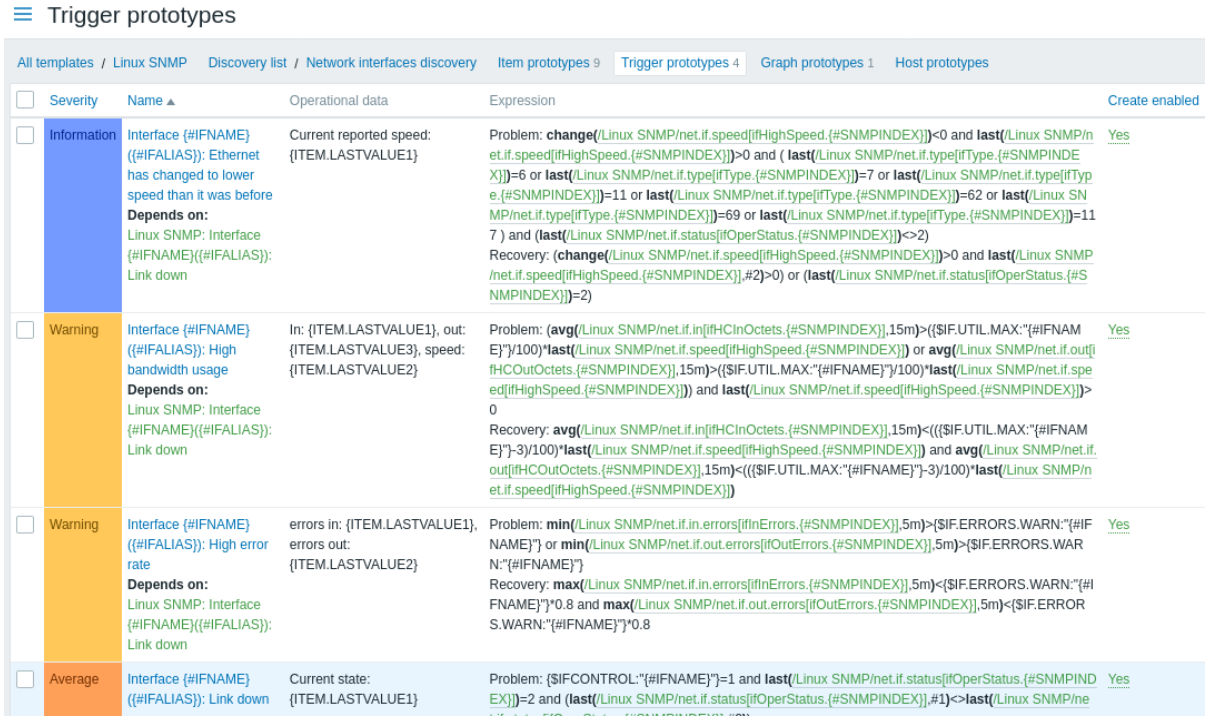
All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 Graph prototypes 1 Host prototypes

Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Interface type	net.if.type[ifType.{#SNMPINDEX}]	1h	7d	0d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]	1m	7d	0	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]	5m	7d	0d	SNMP agent	Yes

Auslöser-Prototypen

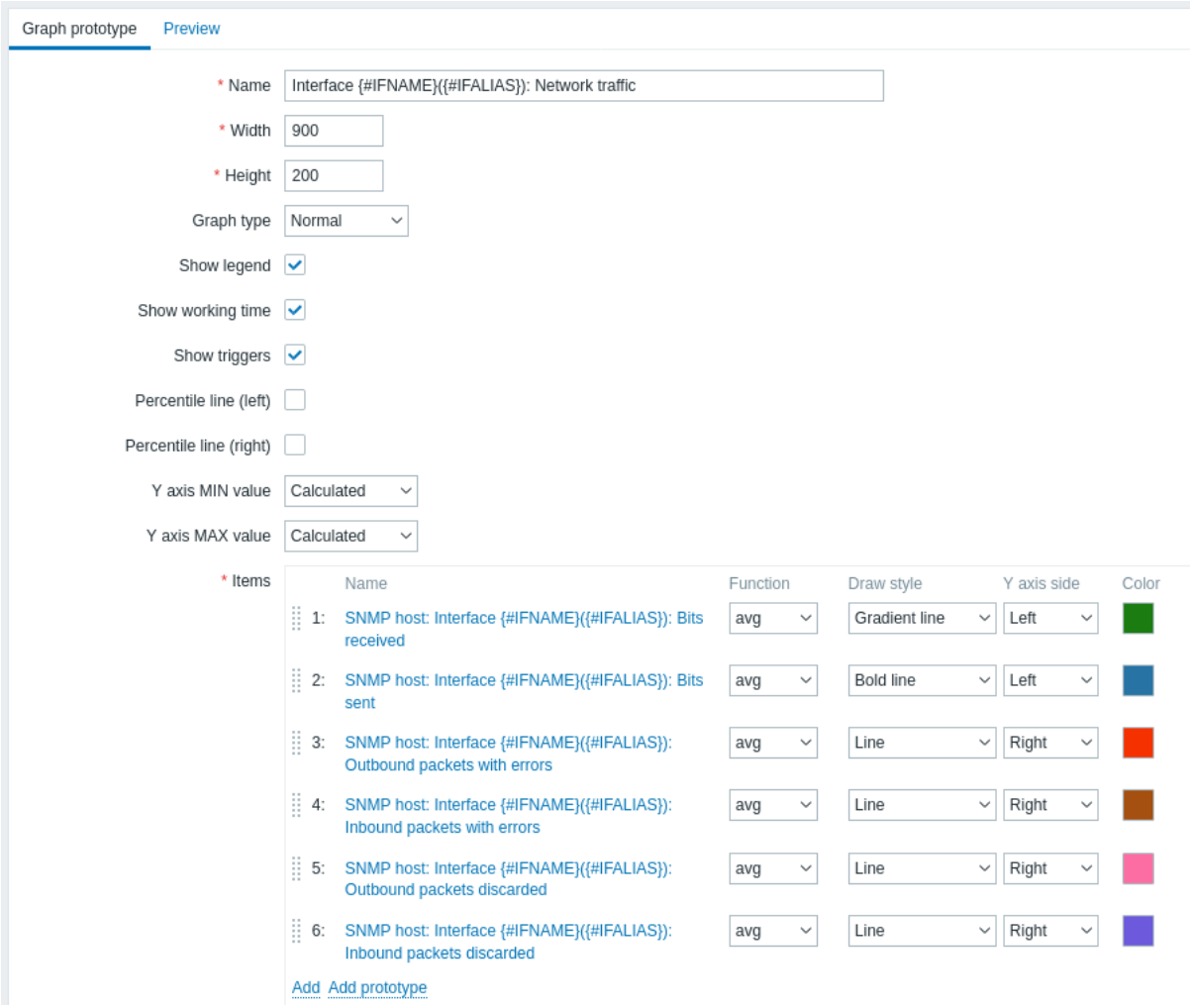
Der folgende Screenshot veranschaulicht, wie wir diese Makros in Auslöser-Prototypen verwenden können:

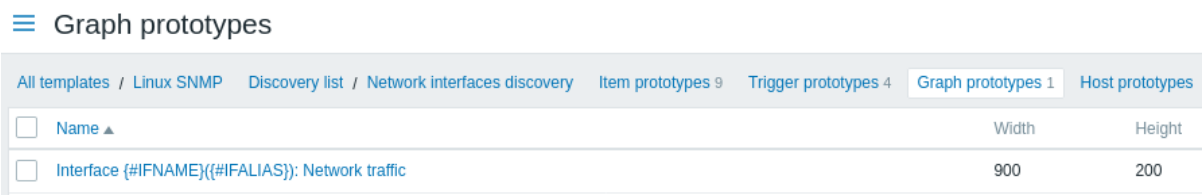




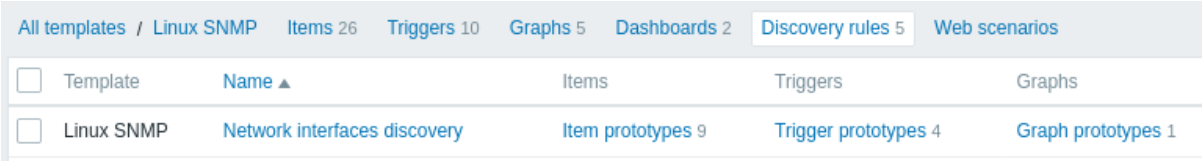
Graph-Prototypen

Der folgende Screenshot veranschaulicht, wie wir diese Makros in Graph-Prototypen verwenden können:





Eine Zusammenfassung unserer Discovery-Regel:



Erkannte Entitäten

Wenn der Server läuft, erstellt er auf Grundlage der Werte, die die SNMP-Erkennungsregel zurückgibt, reale Datenpunkte, Auslöser und Diagramme. In der Host-Konfiguration sind sie mit einem orangefarbenen Link zu der Erkennungsregel versehen, aus der sie stammen.

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Severity	Value	Name ▲	Operational data	Expression
Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before **Depends on:** SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: **change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**<0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and (**last(**/SNMP host/net.if.type[ifType.2]**)**=6 or **last(**/SNMP host/net.if.type[ifType.2]**)**=7 or **last(**/SNMP host/net.if.type[ifType.2]**)**=11 or **last(**/SNMP host/net.if.type[ifType.2]**)**=62 or **last(**/SNMP host/net.if.type[ifType.2]**)**=69 or **last(**/SNMP host/net.if.type[ifType.2]**)**=117) and (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2) Recovery: (**change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2],#2**)**>0) or (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2)
Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage **Depends on:** SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (**avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** or **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 Recovery: **avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** and **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**
Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate **Depends on:** SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: **min(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} or **min(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} Recovery: **max(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8 and **max(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8
Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {$IFCONTROL:"enp4s0"}=1 and **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2 and (**last(**/SNMP host/net.if.status[ifOperStatus.2],#1**)**<>**last(**/SNMP host/net.if.status[ifOperStatus.2],#2**)**) Recovery: **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2 or {$IFCONTROL:"enp4s0"}=0

Graphs

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲
Mounted filesystem discovery: /: Disk space usage
Linux SNMP: CPU jumps
CPU discovery: CPU usage
CPU discovery: CPU utilization
Network interfaces discovery: Interface enp4s0(): Network traffic

6 Erkennung von JMX-Objekten

Übersicht

Es ist möglich, alle JMX-MBeans oder MBean-Attribute zu entdecken oder ein Muster für die Erkennung dieser Objekte anzugeben.

Für die Konfiguration der Discovery-Regel ist es zwingend erforderlich, den Unterschied zwischen einem MBean und MBean-Attributen zu verstehen. Ein MBean ist ein Objekt, das ein Gerät, eine Anwendung oder eine beliebige Ressource repräsentieren kann, die verwaltet werden muss.

Zum Beispiel gibt es ein MBean, das einen Webserver repräsentiert. Seine Attribute sind die Anzahl der Verbindungen, die Anzahl der Threads, das Request-Timeout, der HTTP-Datei-Cache, die Speicherauslastung usw. In einer für Menschen verständlichen Sprache ausgedrückt, können wir eine Kaffeemaschine als ein MBean definieren, das die folgenden zu überwachenden Attribute hat: Wassermenge pro Tasse, durchschnittlicher Wasserverbrauch über einen bestimmten Zeitraum, Anzahl der Kaffeebohnen pro Tasse, Nachfüllzeit für Kaffeebohnen und Wasser usw.

Datenpunktschlüssel

Wählen Sie in der Konfiguration der Discovery-Regel im Feld *Typ* **JMX agent** aus.

Für die Erkennung von JMX-Objekten werden zwei Datenpunktschlüssel unterstützt: jmx.discovery[] und jmx.get[]:

Datenpunktschlüssel			
	Rückgabewert	Parameter	Kommentar
jmx.discovery[<discovery mode>,<object name>,<unique short description>]	Dieser Datenpunkt gibt ein JSON-Array mit LLD-Makros zurück, die MBean-Objekte oder deren Attribute beschreiben.	**discovery mode** - einer der folgenden Werte: *attributes* (JMX-MBean-Attribute abrufen, Standard) oder *beans* (JMX-MBeans abrufen) **object name** - Objektname-Muster (siehe documentation), das die abzurufenden MBean-Namen identifiziert (standardmäßig leer, wodurch alle registrierten Beans abgerufen werden) **unique short description** - eine eindeutige Beschreibung, die mehrere JMX-Datenpunkte mit demselben Discovery-Modus und Objektnamen auf dem Host ermöglicht (optional)	Beispiele: → jmx.discovery - alle JMX-MBean-Attribute abrufen → jmx.discovery[beans] - alle JMX-MBeans abrufen → jmx.discovery[attributes,"*:type=GarbageCollector,name - alle Attribute der Garbage Collector abrufen → jmx.discovery[beans,"*:type=GarbageCollector,name=* - alle Garbage Collector abrufen Es gibt einige Einschränkungen hinsichtlich der MBean-Eigenschaften, die dieser Datenpunkt zurückgeben kann, da bei der Generierung von Makronamen nur eine begrenzte Anzahl von Zeichen unterstützt wird (unterstützte Zeichen können durch den folgenden regulären Ausdruck beschrieben werden: `A-Z0-9_\.`). Um beispielsweise MBean-Eigenschaften mit einem Wort mit Bindestrich oder Nicht-ASCII-Zeichen zu erkennen, müssen Sie `jmx.get[]` verwenden.
jmx.get[<discovery mode>,<object name>,<unique short description>]			

Datenpunktschlüssel		
Dieser Datenpunkt gibt ein JSON-Array mit MBean-Objekten oder deren Attributen zurück. Im Vergleich zu `jmx.discovery[]` definiert er keine LLD-Makros.	**discovery mode** - einer der folgenden Werte: *attributes* (JMX-MBean-Attribute abrufen, Standard) oder *beans* (JMX-MBeans abrufen) **object name** - Objektname-Muster (siehe documentation), das die abzurufenden MBean-Namen identifiziert (standardmäßig leer, wodurch alle registrierten Beans abgerufen werden) **unique short description** - eine eindeutige Beschreibung, die mehrere JMX-Datenpunkte mit demselben Discovery-Modus und Objektnamen auf dem Host ermöglicht (optional)	Bei Verwendung dieses Datenpunkts müssen benutzerdefinierte Low-Level-Discovery-Makros definiert werden, die auf Werte verweisen, die mit JSONPath aus dem zurückgegebenen JSON extrahiert werden.

Attention:
Wenn keine Parameter übergeben werden, werden alle MBean-Attribute aus JMX angefordert. Wenn für die JMX-Erkennung keine Parameter angegeben werden oder versucht wird, alle Attribute für einen weiten Bereich wie `*:type=*,name=*` abzurufen, kann dies zu potenziellen Leistungsproblemen führen.

Verwendung von jmx.discovery

Dieser Datenpunkt gibt ein JSON-Objekt mit Low-Level-Discovery-Makros zurück, die MBean-Objekte oder -Attribute beschreiben. Zum Beispiel bei der Discovery von MBean-Attributen (zur besseren Übersichtlichkeit neu formatiert):


```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXTYPE}":"javax.management.ObjectName",
```

```
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

Bei der Discovery von MBeans (zur besseren Übersichtlichkeit neu formatiert):

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

Unterstützte Makros

Die folgenden Makros werden zur Verwendung im Filter der Discovery-Regel und in Prototypen von Datenpunkten, Auslösern und Diagrammen unterstützt:

Makro	Beschreibung
Discovery von MBean-Attributen	
{#JMXVALUE}	Attributwert.
{#JMXTYPE}	Attributtyp.
{#JMXOBJ}	Objektname.
{#JMXDESC}	Objektname einschließlich Attributname.
{#JMXATTR}	Attributname.
Discovery von MBeans	
{#JMXDOMAIN}	MBean-Domain. (*Von Zabbix reservierter Name*)
{#JMXOBJ}	Objektname. (*Von Zabbix reservierter Name*)
{#JMX<key property>}	MBean-Eigenschaften (wie {#JMXTYPE}, {#JMXNAME}) (siehe Einschränkungen unten).

Einschränkungen

Es gibt einige Einschränkungen im Zusammenhang mit dem Algorithmus zur Erstellung von LLD-Makronamen aus MBean-Eigenschaftsnamen:

- Attributnamen werden in Großbuchstaben umgewandelt
- Attributnamen werden ignoriert (es werden keine LLD-Makros erzeugt), wenn sie aus für LLD-Makronamen nicht unterstützten Zeichen bestehen. Unterstützte Zeichen können durch den folgenden regulären Ausdruck beschrieben werden: `A-Z0-9_\.`
- wenn ein Attribut „obj“ oder „domain“ heißt, wird es ignoriert, da es sich mit den Werten der reservierten Zabbix-Eigenschaften {#JMXOBJ} und {#JMXDOMAIN} überschneidet

Bitte beachten Sie dieses Beispiel für jmx.discovery (mit dem Modus „beans“). Für das MBean sind die folgenden Eigenschaften definiert (einige davon werden ignoriert; siehe unten):

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

Als Ergebnis der JMX-Erkennung werden die folgenden LLD-Makros erzeugt:

- {#JMXDOMAIN} - Zabbix-intern, beschreibt die Domain des MBean
- {#JMXOBJ} - Zabbix-intern, beschreibt das MBean-Objekt
- {#JMXNAME} - aus der Eigenschaft „name“ erstellt

Ignorierte Eigenschaften sind:

- тип : sein Name enthält nicht unterstützte Zeichen (nicht ASCII)
- attributes[] : sein Name enthält nicht unterstützte Zeichen (eckige Klammern werden nicht unterstützt)
- Name : ist bereits definiert (name=test)
- domAin : ist ein von Zabbix reservierter Name

Beispiele

Sehen wir uns zwei weitere praktische Beispiele für die Erstellung einer LLD-Regel unter Verwendung von MBean an. Um den Unterschied zwischen einer LLD-Regel, die MBeans erfasst, und einer LLD-Regel, die MBean-Attribute erfasst, besser zu verstehen, werfen Sie bitte einen Blick auf die folgende Tabelle:

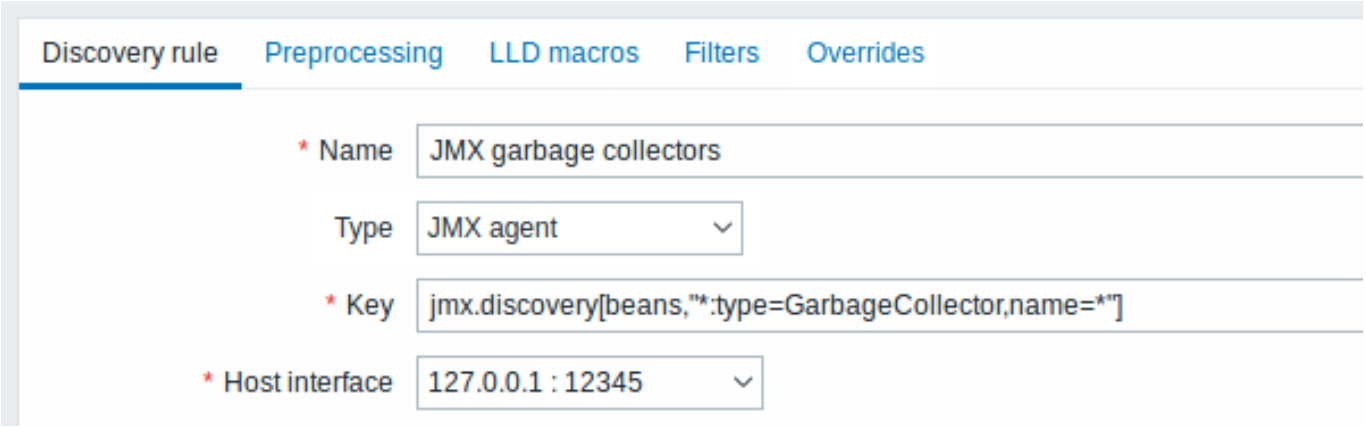
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

Beispiel 1: Erkennung von MBeans

Diese Regel gibt 3 Objekte zurück: die oberste Zeile der Spalte: MBean1, MBean2, MBean3.

Weitere Informationen zu Objekten finden Sie in der Tabelle unterstützte Makros, Abschnitt *Erkennung von MBeans*.

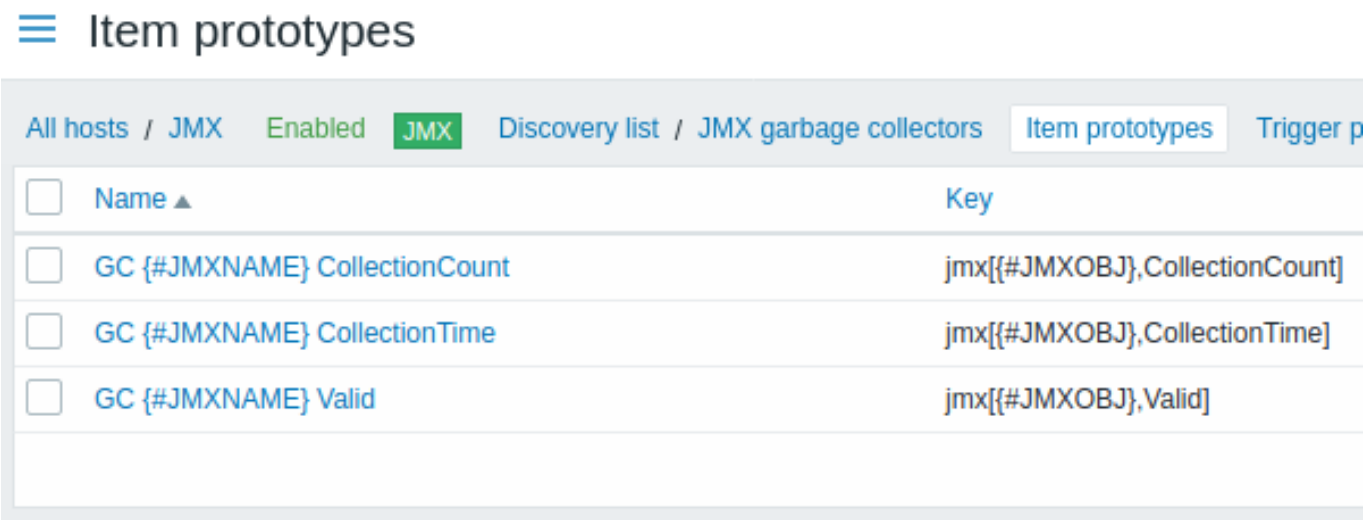
Die Konfiguration der Discovery-Regel zum Sammeln von MBeans (ohne die Attribute) sieht wie folgt aus:



Der hier verwendete Schlüssel:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

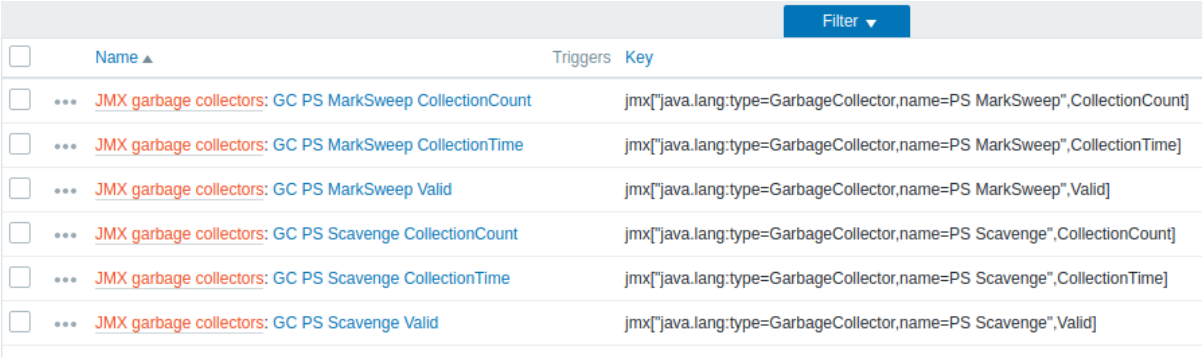
Alle Garbage Collectors ohne Attribute werden erkannt. Da Garbage Collectors denselben Attributsatz haben, können wir die gewünschten Attribute in Datenpunkt-Prototypen auf folgende Weise verwenden:



Die hier verwendeten Schlüssel:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

Die LLD-Discovery-Regel führt zu einem Ergebnis, das in etwa so aussieht (Datenpunkte werden für zwei Garbage Collectors erkannt):

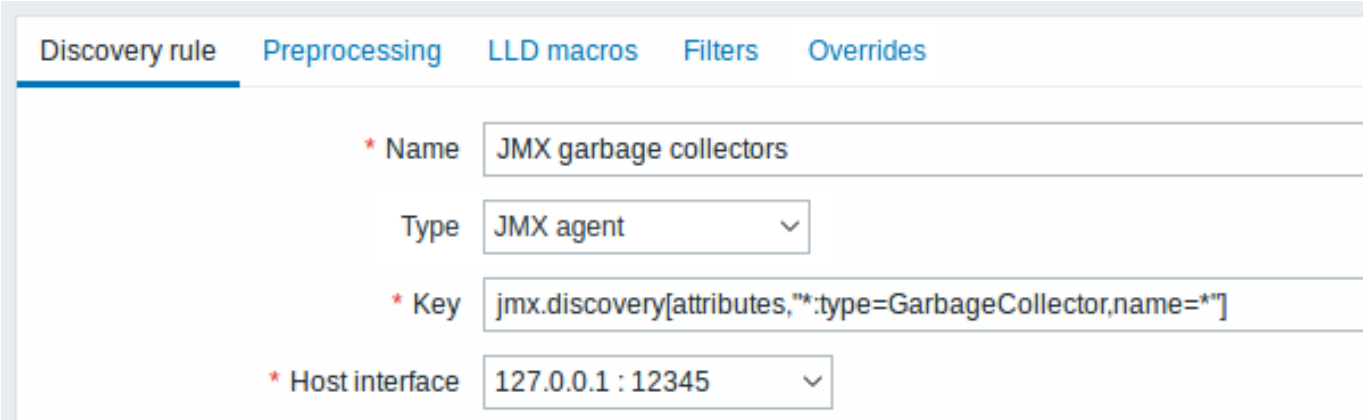


Beispiel 2: Erkennung von MBean-Attributen

Diese Regel gibt 9 Objekte mit den folgenden Feldern zurück: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attribute2,MBe MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Weitere Informationen zu Objekten finden Sie in der Tabelle unterstützte Makros, Abschnitt *Erkennung von MBean-Attributen*.

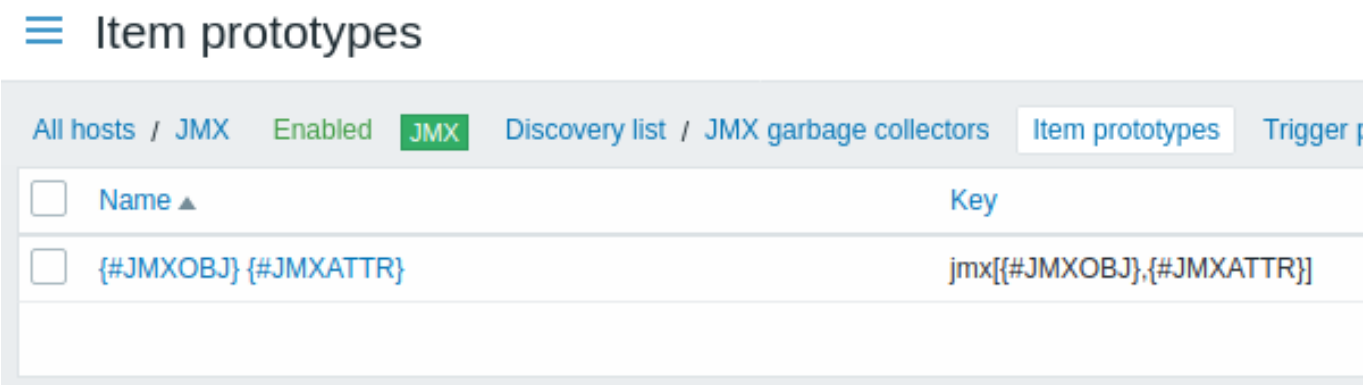
Die Konfiguration der Discovery-Regel zum Sammeln von MBean-Attributen sieht wie folgt aus:



Der hier verwendete Schlüssel:

```
jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]
```

Alle Garbage Collectors mit einem einzelnen Datenpunkt-Attribut werden erkannt.



In diesem speziellen Fall wird für jedes MBean-Attribut ein Datenpunkt aus dem Prototyp erstellt. Der Hauptnachteil dieser Konfiguration besteht darin, dass die Erstellung von Auslösern aus Auslöser-Prototypen nicht möglich ist, da es nur einen Datenpunkt-Prototyp für alle Attribute gibt. Daher kann diese Einrichtung zur Datenerfassung verwendet werden, wird jedoch nicht für die automatische Überwachung empfohlen.

Verwendung von jmx.get

`jmx.get[]` ist ähnlich wie der Datenpunkt `jmx.discovery[]`, wandelt jedoch Java-Objekteigenschaften nicht in Makronamen für Low-Level-Discovery um und kann daher Werte ohne Einschränkungen zurückgeben, die mit der Generierung von LLD-Makronamen verbunden sind, wie z. B. Bindestriche oder Nicht-ASCII-Zeichen.

Bei der Verwendung von `jmx.get[]` für Discovery können Low-Level-Discovery-Makros separat im benutzerdefinierten Tab LLD-Makro der Discovery-Regelkonfiguration definiert werden, wobei JSONPath verwendet wird, um auf die erforderlichen Werte zu verweisen.

Erkennen von MBeans

Discovery-Datenpunkt: `jmx.get[beans,"com.example:type=*,*"]`

Antwort:

```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```

Erkennen von MBean-Attributen

Discovery-Datenpunkt: `jmx.get[attributes,"com.example:type=*,*"]`

Antwort:

```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```

7 Erkennung von IPMI-Sensoren

Übersicht

Es ist möglich, IPMI-Sensoren automatisch zu erkennen.

Dazu können Sie eine Kombination aus Folgendem verwenden:

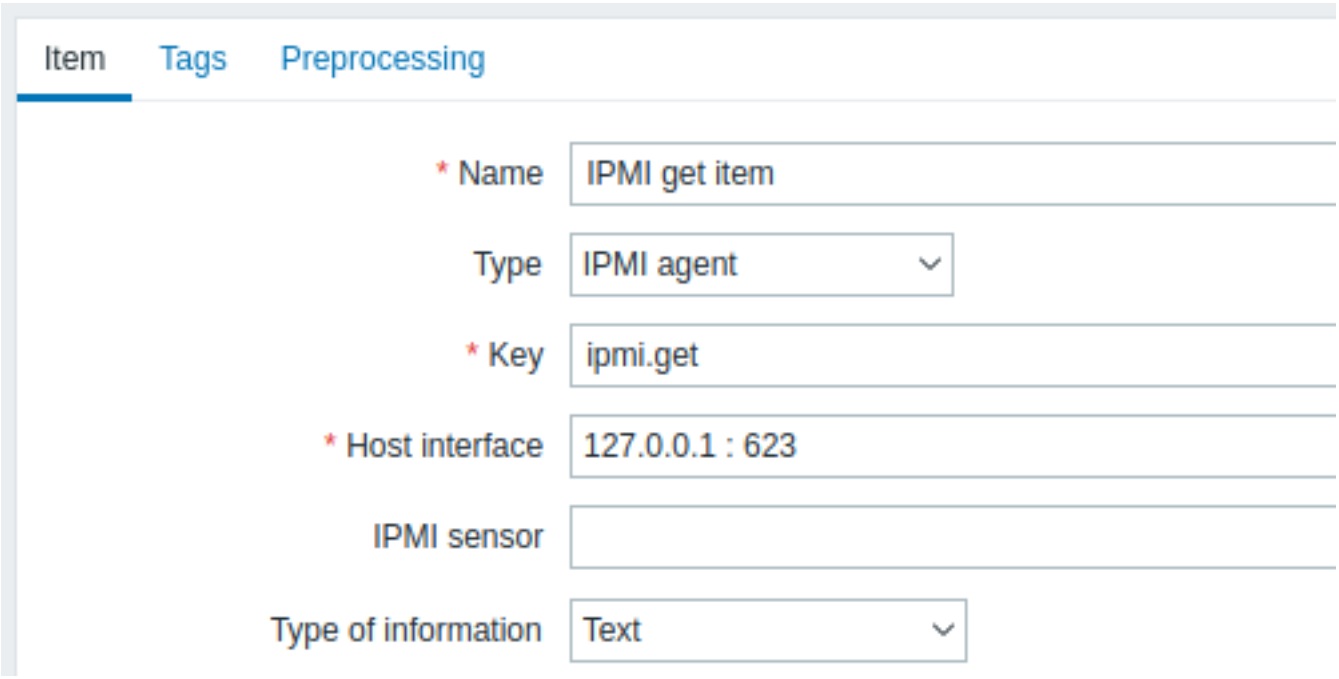
- den IPMI-Datenpunkt `ipmi.get` als Master-Datenpunkt
- abhängige Low-Level-Discovery-Regel und Datenpunkt-Prototypen

Konfiguration

Master-Datenpunkt

Erstellen Sie einen IPMI-Datenpunkt mit folgendem Schlüssel:

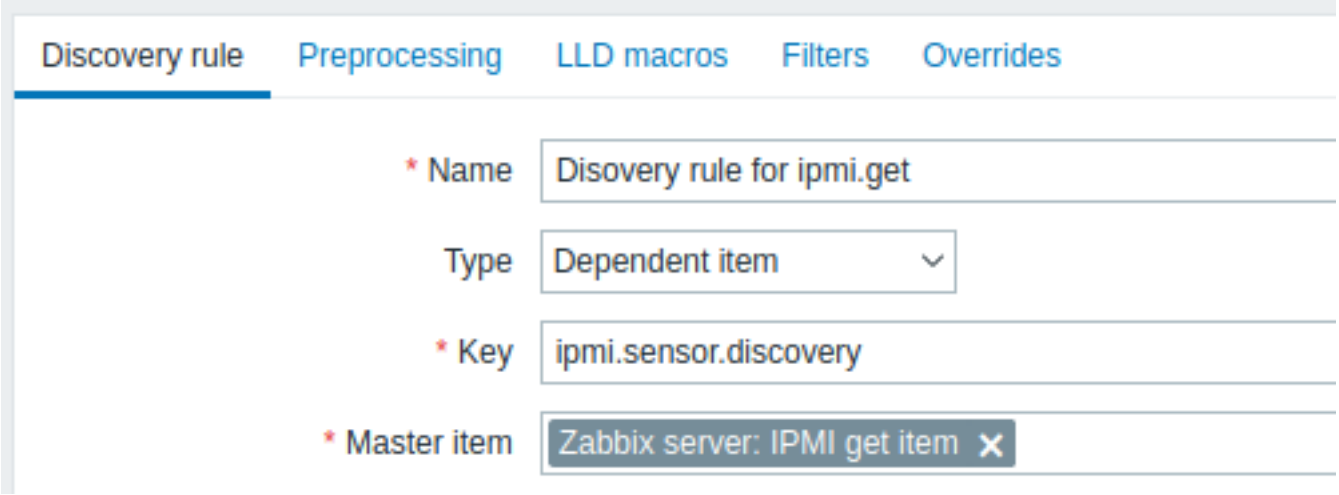
`ipmi.get`



Setzen Sie den Informationstyp auf „Text“ für möglicherweise große JSON-Daten.

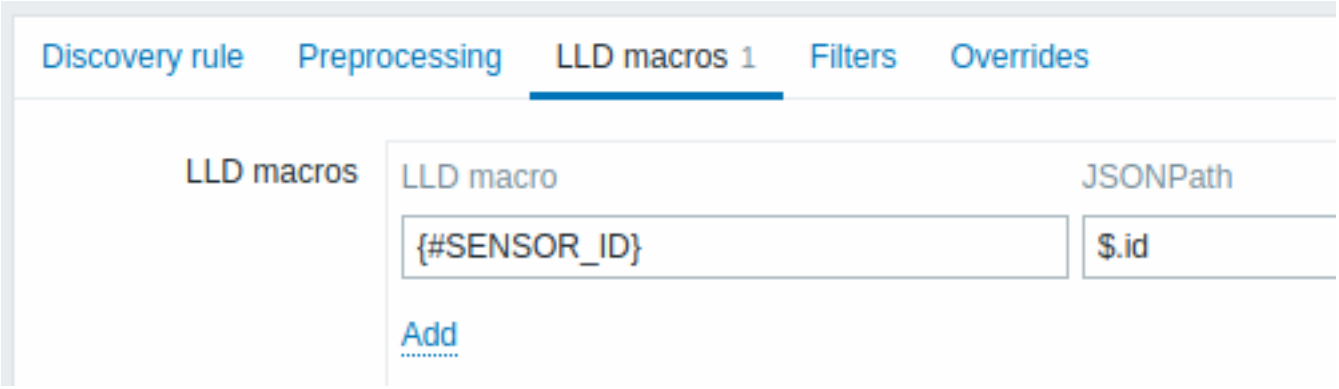
Abhängige LLD-Regel

Erstellen Sie eine Low-Level-Discovery-Regel vom Typ „Abhängiger Datenpunkt“:



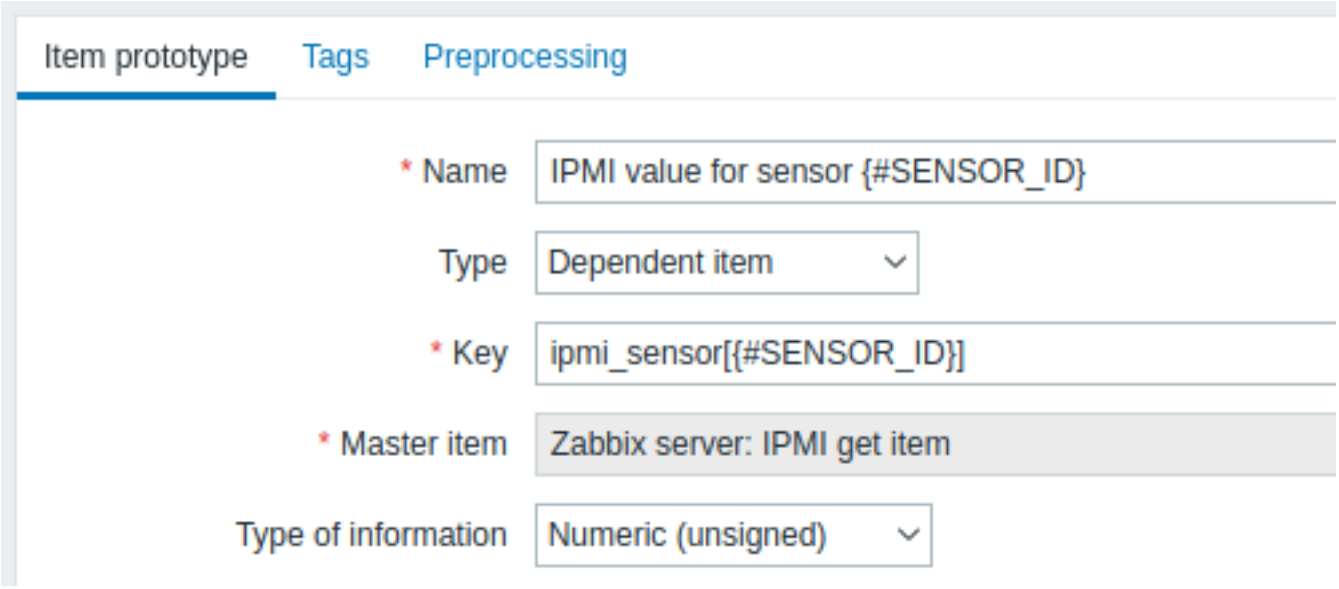
Wählen Sie als Master-Datenpunkt den zuvor erstellten Datenpunkt `ipmi.get` aus.

Definieren Sie auf der Registerkarte „LLD-Makros“ ein benutzerdefiniertes Makro mit dem entsprechenden JSONPath:



Prototyp eines abhängigen Datenpunkts

Erstellen Sie in dieser LLD-Regel ein Datenpunkt-Prototyp vom Typ „Abhängiger Datenpunkt“. Wählen Sie als Master-Datenpunkt für diesen Prototyp den Datenpunkt `ipmi.get` aus, den wir erstellt haben.



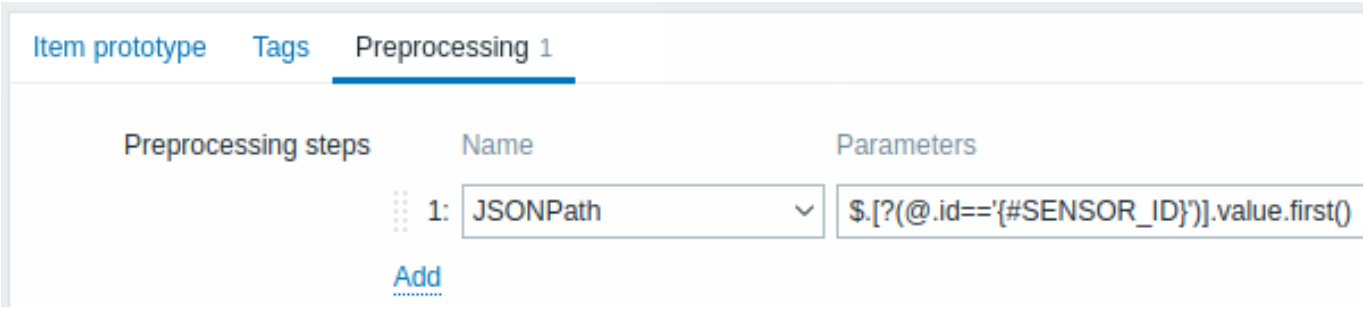
Beachten Sie die Verwendung des Makros {#SENSOR_ID} im Namen und Schlüssel des Datenpunkt-Prototyps:

- *Name*: IPMI-Wert für Sensor {#SENSOR_ID}
- *Schlüssel*: ipmi_sensor[{#SENSOR_ID}]

Als Informationstyp wählen Sie *Numerisch (ohne Vorzeichen)*.

Wählen Sie im Reiter „Vorverarbeitung“ des Datenpunkt-Prototyps JSONPath aus und verwenden Sie den folgenden JSONPath-Ausdruck als Parameter:

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



Wenn die Discovery startet, wird für jeden IPMI-Sensor ein Datenpunkt erstellt. Dieser Datenpunkt gibt den Ganzzahlwert des jeweiligen Sensors zurück.

8 Erkennung von systemd-Diensten

Übersicht

Es ist möglich, systemd-Units (standardmäßig Services) mit Zabbix zu entdecken.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunkt ist

```
systemd.unit.discovery
```

Attention:
Dieser Datenpunkt-Schlüssel wird nur von Zabbix Agent 2 unterstützt.

Dieser Datenpunkt gibt ein JSON mit Informationen über systemd-Units zurück, zum Beispiel:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "MySQL Server",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
```

```
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

Erkennung deaktivierter systemd-Units

Es ist auch möglich, **deaktivierte** systemd-Units zu erkennen. In diesem Fall werden im resultierenden JSON drei Makros zurückgegeben:

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:
Damit Datenpunkte und Auslöser aus Prototypen für deaktivierte systemd-Units erstellt werden, stellen Sie sicher, dass Sie einschränkende LLD-Filter für {#UNIT.ACTIVESTATE} und {#UNIT.UNITFILESTATE} anpassen (oder entfernen).

Unterstützte Makros

Die folgenden Makros werden zur Verwendung im Filter der Discovery-Regel und in Prototypen von Datenpunkten, Auslösern und Diagrammen unterstützt:

Makro	Beschreibung
{#UNIT.NAME}	Name der primären Unit.
{#UNIT.DESCRIPTION}	Menschenlesbare Beschreibung.
{#UNIT.LOADSTATE}	Ladezustand (d. h. ob die Unit-Datei erfolgreich geladen wurde)
{#UNIT.ACTIVESTATE}	Aktivzustand (d. h. ob die Unit derzeit gestartet ist oder nicht)
{#UNIT.SUBSTATE}	Unterzustand (eine feinere Version des Aktivzustands, die spezifisch für den Unit-Typ ist, was auf den Aktivzustand nicht zutrifft)
{#UNIT.FOLLOWED}	Unit, deren Zustand von dieser Unit verfolgt wird, falls vorhanden; andernfalls eine leere Zeichenfolge.
{#UNIT.PATH}	Objektpfad der Unit.
{#UNIT.JOBID}	Numerische Job-ID, wenn für die Job-Unit ein Job in die Warteschlange eingereiht ist; andernfalls 0.
{#UNIT.JOBTYPE}	Job-Typ.
{#UNIT.JOBPATH}	Objektpfad des Jobs.
{#UNIT.UNITFILESTATE}	Installationszustand der Unit-Datei.
{#UNIT.SERVICETYPE}	Typ der Service-Unit (z. B. `simple`, `forking`, `oneshot`, `idle` usw.). Dieses Makro wird nur zurückgegeben, wenn die Unit ein Service ist.

Datenpunkt-Prototypen

Datenpunkt-Prototypen, die auf Basis der systemd-Service-Erkennung erstellt werden können, umfassen zum Beispiel:

- Datenpunktname: `{#UNIT.DESCRIPTION} active state info`; Datenpunktschlüssel: `systemd.unit.info["{#UNIT.NAME}"`
- Datenpunktname: `{#UNIT.DESCRIPTION} load state info`; Datenpunktschlüssel: `systemd.unit.info["{#UNIT.NAME}",L`

9 Erkennung von Windows-Diensten

Übersicht

Ähnlich wie Dateisysteme erkannt werden, ist es auch möglich, Windows-Dienste zu erkennen.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunkt ist

`service.discovery`

Unterstützte Makros

Die folgenden Makros werden zur Verwendung in der Discovery-Regel filter und in Prototypen von Datenpunkten, Auslösern und Graphen unterstützt:

Macro	Description
{#SERVICE.NAME}	Dienstname.
{#SERVICE.DISPLAYNAME}	Angezeigter Dienstname.
{#SERVICE.DESCRIPTION}	Dienstbeschreibung.
{#SERVICE.STATE}	Numerischer Wert des Dienststatus. Siehe den Datenpunkt service.info für Details.
{#SERVICE.STATENAME}	Name des Dienststatus. Siehe den Datenpunkt service.info für Details.
{#SERVICE.PATH}	Dienstpfad.
{#SERVICE.USER}	Dienstbenutzer.
{#SERVICE.STARTUP}	Numerischer Wert des Dienststarttyps. Siehe den Datenpunkt service.info für Details.
{#SERVICE.STARTUPNAME}	Name des Dienststarttyps. Siehe den Datenpunkt service.info für Details.
{#SERVICE.STARTUPTRIGGER}	Numerischer Wert, der angibt, ob der Dienststarttyp Folgendes hat: 0 - keine Startauslöser 1 - hat Startauslöser Dies ist nützlich, um solche Dienststarttypen wie *Automatisch (Auslöserstart)*, *Automatisch verzögert (Auslöserstart)* und *Manuell (Auslöserstart)* zu erkennen.

Basierend auf der Windows-Diensterkennung können Sie einen Datenpunkt-Prototyp wie folgt erstellen:

`service.info[{#SERVICE.NAME},<param>]`

wobei `param` die folgenden Werte akzeptiert: *state*, *displayname*, *path*, *user*, *startup* oder *description*.

Um beispielsweise den Anzeigenamen eines Dienstes abzurufen, können Sie einen Datenpunkt "service.info[{#SERVICE.NAME},displayname]" verwenden. Wenn der Wert von `param` nicht angegeben ist ("service.info[{#SERVICE.NAME}]"), wird standardmäßig der Parameter *state* verwendet.

10 Erkennung von Windows-Leistungsindikatorinstanzen

Übersicht

Es ist möglich, Objektinstanzen von Windows-Leistungsindikatoren zu entdecken. Dies ist nützlich für Leistungsindikatoren mit mehreren Instanzen.

Datenpunktschlüssel

Um die Discovery-Regel zu konfigurieren, verwenden Sie den folgenden Datenpunkt:

- `perf_instance.discovery[object]`

Beachten Sie, dass der Objektname lokalisiert sein kann. Zum Beispiel:

```
perf_instance.discovery[Processor] # Der Objektname ist auf Englisch.
perf_instance.discovery[Processador] # Der Objektname ist auf Portugiesisch.
```

Alternativ können Sie den folgenden Datenpunkt verwenden, um sicherzustellen, dass der Objektname unabhängig von der Betriebssystemlokalisierung auf Englisch angegeben wird:

- `perf_instance_en.discovery[object]`

Zum Beispiel:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

Unterstützte Makros

Der Discovery-Prozess gibt alle Instanzen des angegebenen Objekts im Makro `{#INSTANCE}` zurück:

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

Dieses Makro kann in den Prototypen von `perf_counter[]`- und `perf_counter_en[]`-Datenpunkten verwendet werden.

Wenn beispielsweise der im Discovery-Regel verwendete Datenpunktschlüssel `perf_instance.discovery[Processor]` ist, können Sie den folgenden Datenpunktprototyp erstellen:

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

Hinweis:

- Wenn das angegebene Objekt nicht gefunden wird oder keine variablen Instanzen unterstützt, wird der Discovery-Datenpunkt zu NOTSUPPORTED.
- Wenn das angegebene Objekt variable Instanzen unterstützt, derzeit aber keine Instanzen hat, wird ein leeres JSON-Array zurückgegeben.
- Doppelte Instanzen werden übersprungen.

11 Erkennung mithilfe von WMI-Abfragen

Übersicht

WMI ist eine leistungsstarke Schnittstelle in Windows, die zum Abrufen verschiedener Informationen über Windows-Komponenten, Dienste, Zustände und installierte Software verwendet werden kann.

Sie kann für die Erkennung physischer Datenträger und die Erfassung ihrer Leistungsdaten, die Erkennung von Netzwerkschnittstellen, die Erkennung von Hyper-V-Gästen, die Überwachung von Windows-Diensten und viele andere Aufgaben im Windows-Betriebssystem verwendet werden.

Diese Art der Low-Level- Erkennung wird mithilfe von WQL- Abfragen durchgeführt, deren Ergebnisse automatisch in ein für die Low-Level-Erkennung geeignetes JSON-Objekt umgewandelt werden.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunkt ist

`wmi.getall[<namespace>,<query>]`

Dieser Datenpunkt wandelt das Abfrageergebnis in ein JSON-Array um. Zum Beispiel:

`select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'`

kann etwa Folgendes zurückgeben:

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
```

```
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
    },
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```


Low-level-Discovery-Makros

Auch wenn im zurückgegebenen JSON keine Low-level-Discovery-Makros erstellt werden, können diese Makros vom Benutzer als zusätzlicher Schritt definiert werden, indem die Funktion benutzerdefiniertes LLD-Makro mit JSONPath verwendet wird, das auf die erkannten Werte im zurückgegebenen JSON verweist.

Die Makros können dann verwendet werden, um Datenpunkt-, Auslöser- usw. Prototypen zu erstellen.

12 Discovery mit ODBC-SQL-Abfragen

Übersicht

Diese Art der Low-Level-Discovery wird mithilfe von SQL-Abfragen durchgeführt, deren Ergebnisse automatisch in ein JSON-Objekt umgewandelt werden, das für die Low-Level-Discovery geeignet ist.

Datenpunktschlüssel

SQL-Abfragen werden mit einem Datenpunkt-Typ „Database monitor“ ausgeführt. Daher gelten die meisten Anweisungen auf der Seite ODBC monitoring, um eine funktionierende Discovery-Regel vom Typ „Database monitor“ zu erhalten.

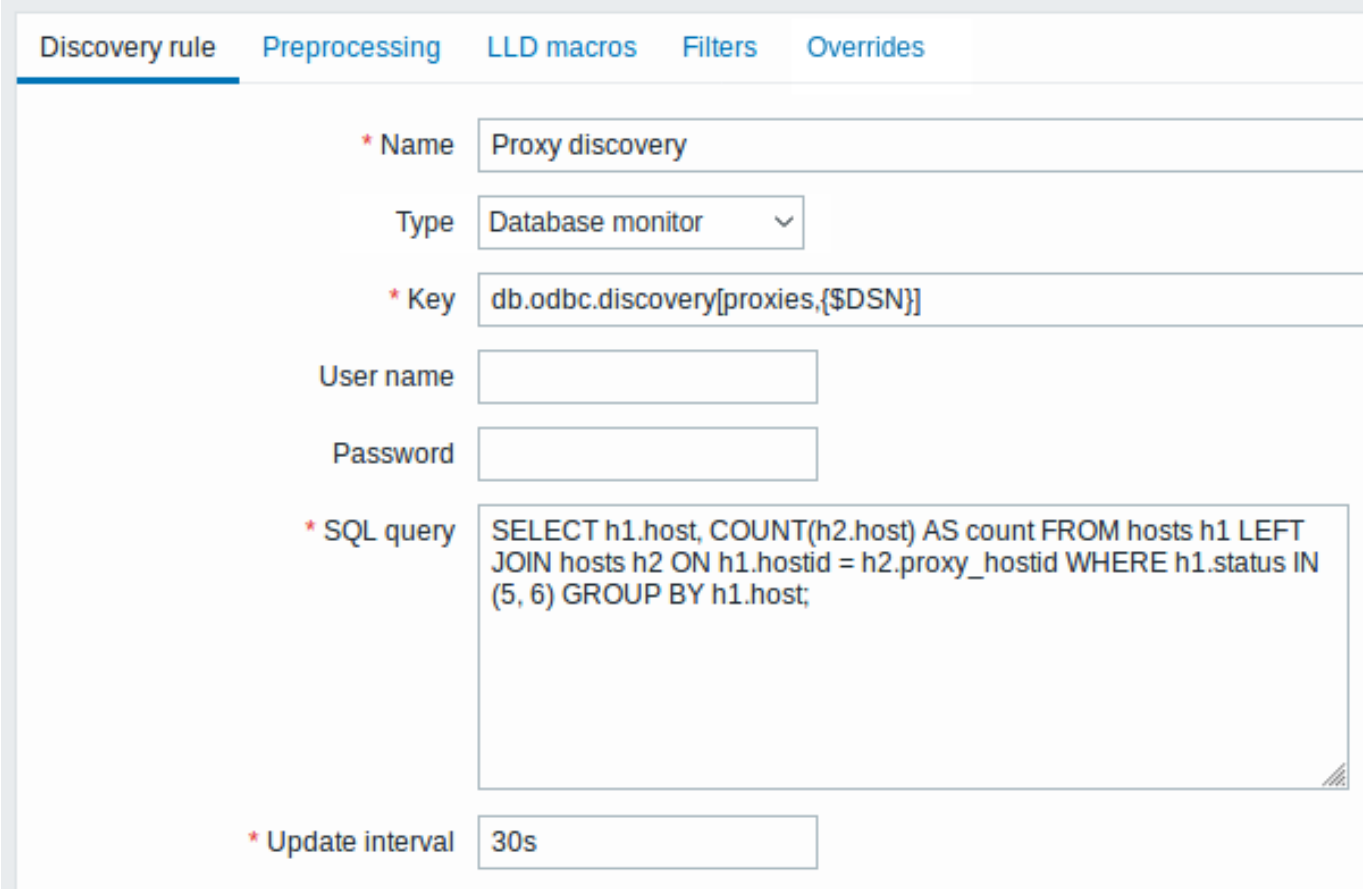
In Discovery-Regeln vom Typ „Database monitor“ können zwei Datenpunktschlüssel verwendet werden:

- **db.odbc.discovery**[<unique short description>,<dsn>,<connection string>] - dieser Datenpunkt wandelt das Ergebnis der SQL-Abfrage in ein JSON-Array um, wobei die Spaltennamen aus dem Abfrageergebnis in Low-Level-Discovery-Makronamen umgewandelt und mit den erkannten Feldwerten verknüpft werden. Diese Makros können beim Erstellen von Prototypen für Datenpunkte, Auslöser usw. verwendet werden. Siehe auch: Using db.odbc.discovery.
- **db.odbc.get**[<unique short description>,<dsn>,<connection string>] - dieser Datenpunkt wandelt das Ergebnis der SQL-Abfrage in ein JSON-Array um, wobei die ursprünglichen Spaltennamen aus dem Abfrageergebnis als Feldnamen im JSON beibehalten und mit den erkannten Werten verknüpft werden. Im Vergleich zu `db.odbc.discovery[]` erstellt dieser Datenpunkt keine Low-Level-Discovery-Makros im zurückgegebenen JSON, daher muss nicht geprüft werden, ob die Spaltennamen gültige Makronamen sein können. Die Low-Level-Discovery-Makros können bei Bedarf in einem zusätzlichen Schritt definiert werden, indem die Funktion custom LLD macro mit JSONPath verwendet wird, das auf die erkannten Werte im zurückgegebenen JSON verweist. Siehe auch: Using db.odbc.get.

Verwendung von db.odbc.discovery

Das folgende Beispiel zeigt, wie eine SQL-Abfrage mithilfe der Low-Level-Discovery von Zabbix-Proxys auf Basis einer ODBC-Abfrage an die Zabbix-Datenbank in JSON umgewandelt wird. Dies ist nützlich für die automatische Erstellung von „zabbix[proxy,<name>,lastaccess]“-internen Datenpunkten, um zu überwachen, welche Proxys aktiv sind.

Beginnen Sie mit der Konfiguration der Discovery-Regel:



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Hier wird die folgende direkte Abfrage an die Zabbix-Datenbank verwendet, um alle Zabbix-Proxys zusammen mit der Anzahl der Hosts auszuwählen, die sie überwachen. Die Anzahl der Hosts kann beispielsweise verwendet werden, um leere Proxys herauszufiltern:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

Durch die interne Funktionsweise des Datenpunkts „db.odbc.discovery[,{$DSN}]“ wird das Ergebnis dieser Abfrage automatisch in das folgende JSON umgewandelt:

```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
```

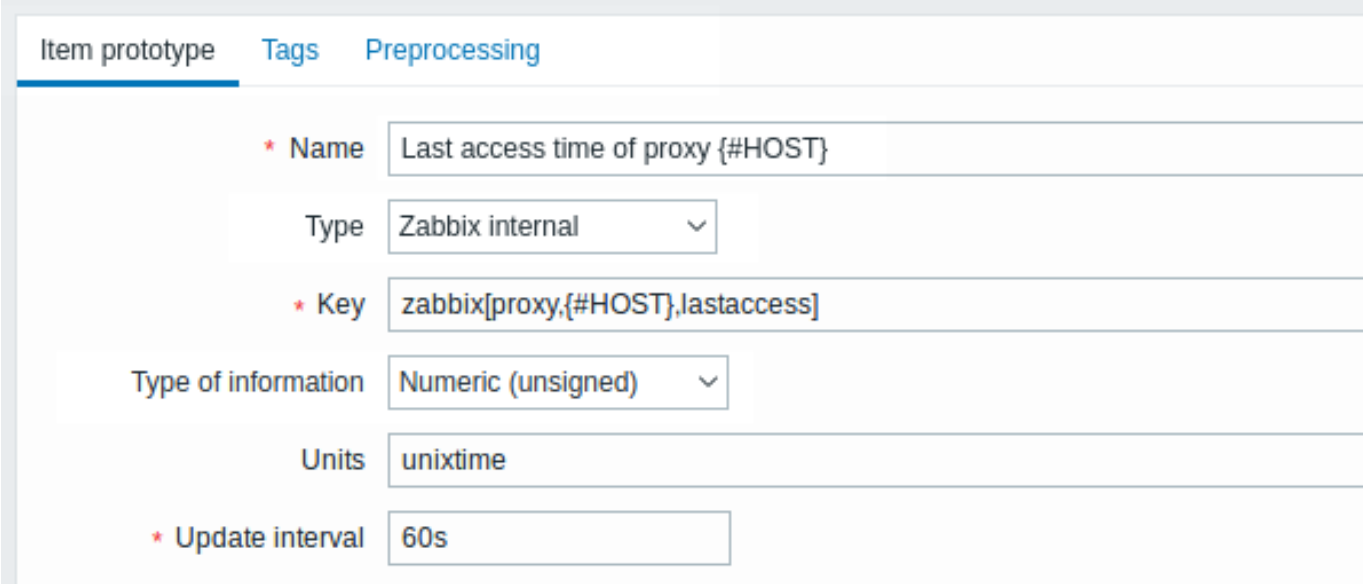
```
    }
]
```

Es ist zu erkennen, dass Spaltennamen zu Makronamen werden und ausgewählte Zeilen zu den Werten dieser Makros werden.

Note:
Wenn nicht offensichtlich ist, wie ein Spaltenname in einen Makronamen umgewandelt wird, wird empfohlen, Spaltenaliase wie „COUNT(h2.host) AS count“ im obigen Beispiel zu verwenden.
Falls ein Spaltenname nicht in einen gültigen Makronamen umgewandelt werden kann, wird die Discovery-Regel nicht unterstützt; die Fehlermeldung enthält dann die Nummer der betreffenden Spalte. Falls zusätzliche Hilfe gewünscht ist, werden die ermittelten Spaltennamen bei DebugLevel=4 in der Zabbix-Server-Logdatei ausgegeben:

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 L
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

Nachdem wir nun verstanden haben, wie eine SQL-Abfrage in ein JSON-Objekt umgewandelt wird, können wir das Makro {#HOST} in Datenpunkt-Prototypen verwenden:



Sobald die Discovery durchgeführt wurde, wird für jeden Proxy ein Datenpunkt erstellt:

Verwendung von db.odbc.get

Bei Verwendung von `db.odbc.get[,{$DSN}]` und des folgenden SQL-Beispiels:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
```

```
+---------+-------+
3 rows in set (0.01 sec)
```

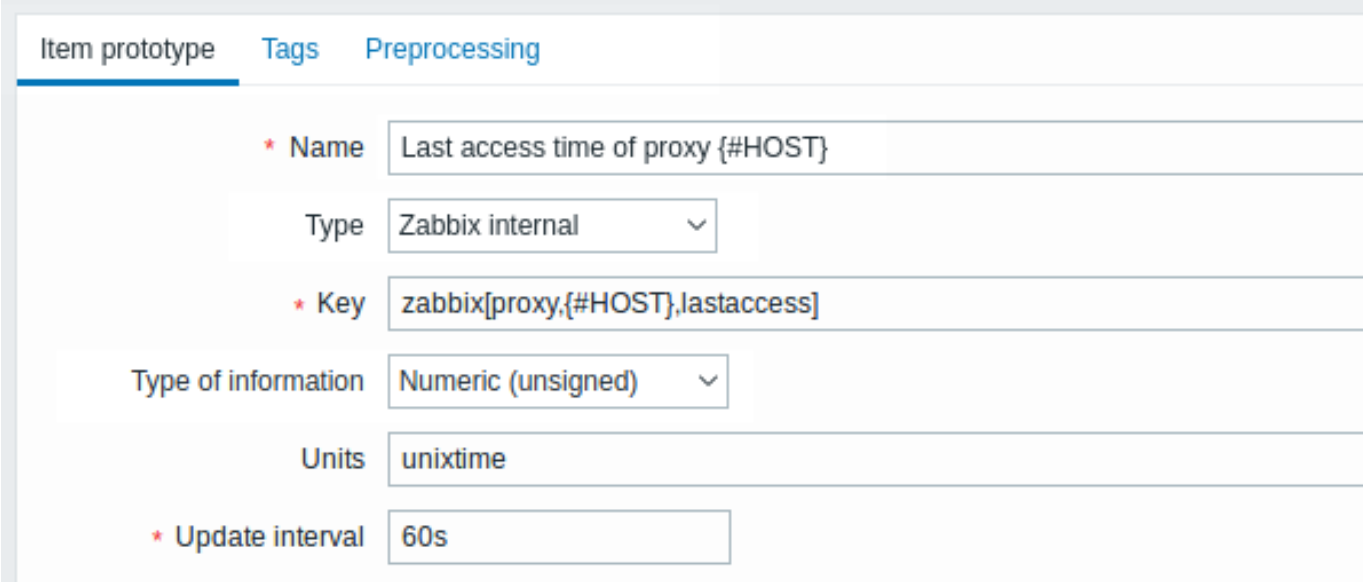
wird dieses JSON zurückgegeben:

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

Wie Sie sehen können, gibt es dort keine Low-Level-Discovery-Makros. Benutzerdefinierte Low-Level-Discovery-Makros können jedoch im Reiter LLD macros einer Discovery-Regel mithilfe von JSONPath erstellt werden, zum Beispiel:

```
{#HOST} → $.host
```

Dieses Makro {#HOST} kann nun in Datenpunkt-Prototypen verwendet werden:



13 Discovery mit Prometheus-Daten

Übersicht

Daten, die im Prometheus-Zeilenformat bereitgestellt werden, können für die Low-Level-Discovery verwendet werden.

Siehe Prometheus checks für Details dazu, wie die Abfrage von Prometheus-Daten in Zabbix implementiert ist.

Konfiguration

Die Low-Level-Discovery-Regel sollte als abhängiger Datenpunkt zum HTTP-Master-Datenpunkt erstellt werden, der Prometheus-Daten erfasst.

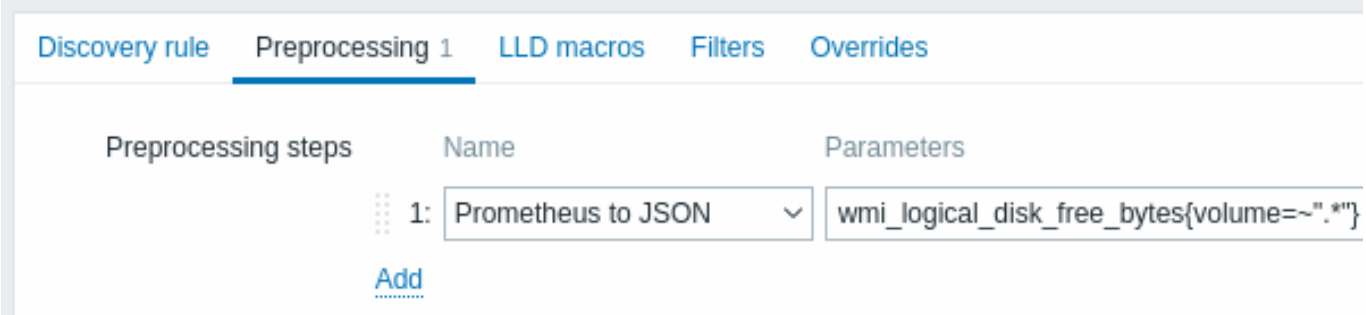
Prometheus zu JSON

Gehen Sie in der Discovery-Regel auf den Reiter Vorverarbeitung und wählen Sie die Vorverarbeitungsoption *Prometheus zu JSON* aus. Daten im JSON-Format werden für die Discovery benötigt, und die Vorverarbeitungsoption *Prometheus zu JSON* liefert genau diese mit den folgenden Attributen:

- Metrikname
- Metrikwert
- Hilfe (falls vorhanden)

- Typ (falls vorhanden)
- Labels (falls vorhanden)
- Rohzeile

Zum Beispiel bei einer Abfrage von `wmi_logical_disk_free_bytes`:



aus diesen Prometheus-Zeilen:

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

wird Folgendes zurückgegeben:

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

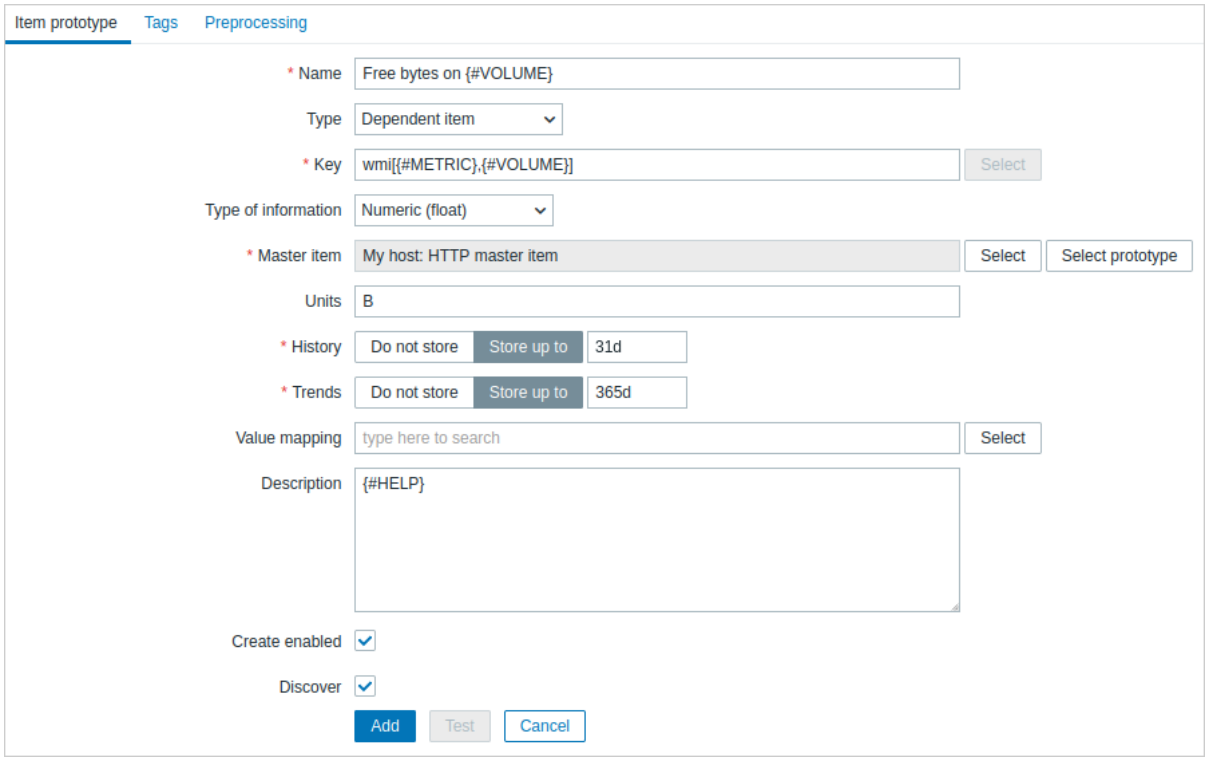
LLD-Makros zuordnen

Als Nächstes müssen Sie zur Registerkarte „LLD-Makros“ wechseln und die folgenden Zuordnungen vornehmen:

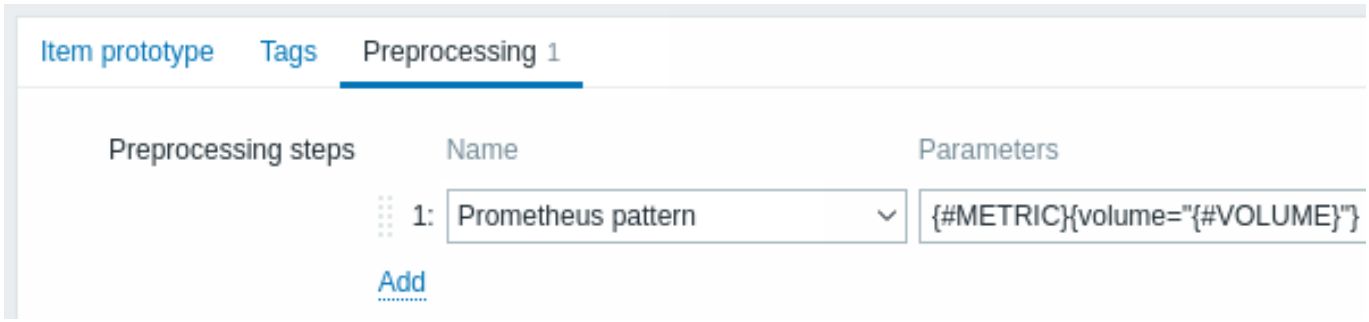
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

Datenpunkt-Prototyp

Möglicherweise möchten Sie einen Datenpunkt-Prototyp wie diesen erstellen:



mit Vorverarbeitungsoptionen:



14 Erkennung von Blockgeräten

Ähnlich wie Dateisysteme erkannt werden, ist es auch möglich, Blockgeräte und deren Typ zu erkennen.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunktschlüssel ist

`vfs.dev.discovery`

Dieser Datenpunkt wird nur auf Linux-Plattformen unterstützt.

Sie können Discovery-Regeln mit diesem Discovery-Datenpunkt erstellen und:

- filtern: **{#DEVNAME} entspricht** `sd[\D]$` – um Geräte mit den Namen „sd0“, „sd1“, „sd2“, ... zu erkennen
- filtern: **{#DEVTYPE} entspricht** `disk` **UND {#DEVNAME} entspricht nicht** `^loop.*` – um Geräte vom Typ „disk“ zu erkennen, deren Name nicht mit „loop“ beginnt

Unterstützte Makros

Dieser Discovery-Schlüssel gibt zwei Makros zurück – {#DEVNAME} und {#DEVTYPE} –, die jeweils den Namen und den Typ des Blockgeräts identifizieren, z. B.:

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
```

```
    },
    {
        "{#DEVNAME}":"dm-0",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partition"
    }
]
```

Die Discovery von Blockgeräten ermöglicht die Verwendung der Datenpunkte `vfs.dev.read[]` und `vfs.dev.write[]`, um Datenpunktprototypen mithilfe des Makros {#DEVNAME} zu erstellen, zum Beispiel:

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} ist für die Gerätefilterung vorgesehen.

15 Erkennung von Host-Schnittstellen in Zabbix

Übersicht

Es ist möglich, alle im Zabbix Frontend für einen Host konfigurierten Schnittstellen zu entdecken.

Datenpunktschlüssel

Der im Discovery-Regel zu verwendende Datenpunkt ist der interne Zabbix-Datenpunkt:

`zabbix[host,discovery,interfaces]`

Dieser Datenpunkt gibt ein JSON mit der Beschreibung von Schnittstellen zurück, einschließlich:

- IP-Adresse/DNS-Hostname (abhängig von der Host-Einstellung „Verbinden mit")
- Portnummer
- Schnittstellentyp (Zabbix Agent, SNMP, JMX, IPMI)
- Ob es sich um die Standardschnittstelle handelt oder nicht
- Ob die Funktion für Massenabfragen aktiviert ist – nur für SNMP-Schnittstellen.

Zum Beispiel:

```
[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG
```

Bei mehreren Schnittstellen werden deren Einträge im JSON wie folgt sortiert:

- Schnittstellentyp,
- Standard – die Standardschnittstelle wird vor Nicht-Standard- Schnittstellen aufgeführt,
- Schnittstellen-ID (in aufsteigender Reihenfolge).

Unterstützte Makros

Die folgenden Makros werden zur Verwendung im Filter der Discovery-Regel sowie in Prototypen von Datenpunkten, Auslösern und Diagrammen unterstützt:

Makro	Beschreibung
{#IF.CONN}	IP-Adresse der Schnittstelle oder DNS-Host-Name.
{#IF.IP}	IP-Adresse der Schnittstelle.
{#IF.DNS}	DNS-Host-Name der Schnittstelle.
{#IF.PORT}	Portnummer der Schnittstelle.
{#IF.TYPE}	Schnittstellentyp („AGENT", „SNMP", „JMX" oder „IPMI").
{#IF.DEFAULT}	Standardstatus der Schnittstelle: 0 - keine Standardschnittstelle 1 - Standardschnittstelle

Makro	Beschreibung
{#IF.SNMP.BULK}	Status der SNMP-Bulk-Verarbeitung für die Schnittstelle: 0 - deaktiviert 1 - aktiviert Dieses Makro wird nur zurückgegeben, wenn der Schnittstellentyp „SNMP“ ist.

8 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 bei der 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 Dateisystemnamen als auch den Typ enthält. Eine Möglichkeit, es zu verwenden, 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`. Dadurch wird ein solches Skript möglich:

```
####!/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"  }
]
```

Dann 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. Falls 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.

14 Verteiltes Monitoring

Übersicht Zabbix bietet eine effektive und zuverlässige Möglichkeit zur Überwachung einer verteilten IT-Infrastruktur mithilfe von Zabbix-Proxys.

Proxys können verwendet werden, um Daten lokal im Auftrag eines zentralisierten Zabbix-Servers zu erfassen und diese anschließend an den Server zu melden.

Proxy-Funktionen

Bei der Entscheidung für oder gegen die Verwendung eines Proxy 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**
Einseitige TCP-Verbindungen	**Ja**
Zentralisierte Konfiguration	**Ja**
Erzeugt Benachrichtigungen	Nein

[1] Die Funktion zur automatischen DB-Erstellung funktioniert nur mit SQLite. Für andere unterstützte Datenbanken ist eine manuelle Einrichtung erforderlich.

Attention:
Der Zabbix Proxy berücksichtigt keine Wartungszeiträume; siehe Berechnung von Warteschlangen während der Wartung für Details.

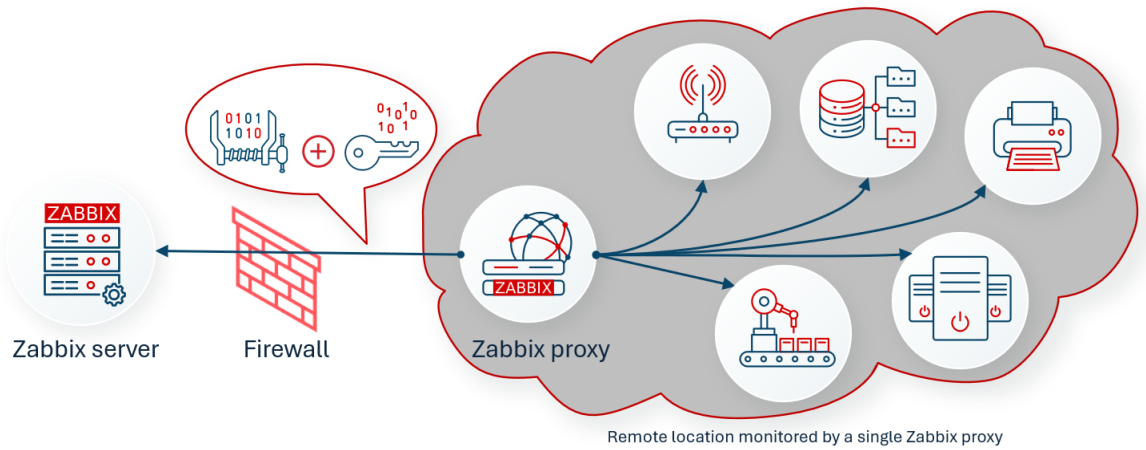
1 Proxys

Überblick Ein Zabbix Proxy kann Leistungs- und Verfügbarkeitsdaten im Auftrag des Zabbix Server erfassen. Auf diese Weise kann ein Proxy einen Teil der Last der Datenerfassung übernehmen und den Zabbix Server entlasten.

Außerdem ist die Verwendung eines Proxy der einfachste Weg, zentrales und verteiltes Monitoring zu implementieren, wenn alle Agents und Proxys an einen Zabbix Server berichten und alle Daten zentral erfasst werden.

Ein Zabbix Proxy kann verwendet werden, um:

- Entfernte Standorte zu überwachen
- Standorte mit unzuverlässiger Kommunikation zu überwachen
- Den Zabbix Server bei der Überwachung von Tausenden von Geräten zu entlasten
- Die Wartung eines verteilten Monitorings zu vereinfachen



Der Proxy benötigt nur eine TCP-Verbindung zum Zabbix Server. Dadurch ist es einfacher, eine Firewall zu umgehen, da Sie nur eine Firewall-Regel konfigurieren müssen.

Attention:
Zabbix Proxy muss eine separate Datenbank verwenden. Wenn er auf die Datenbank des Zabbix Server verweist, wird die Konfiguration beschädigt.

Alle vom Proxy erfassten Daten werden lokal gespeichert, bevor sie an den Server übertragen werden. Dadurch gehen bei vorübergehenden Kommunikationsproblemen mit dem Server keine Daten verloren. Die Parameter *ProxyLocalBuffer* und *ProxyOfflineBuffer* in der Proxy-Konfigurationsdatei steuern, wie lange die Daten lokal aufbewahrt werden.

Attention:
Es kann vorkommen, dass ein Proxy, der die neuesten Konfigurationsänderungen direkt aus der Zabbix-Server-Datenbank erhält, eine aktuellere Konfiguration hat als der Zabbix Server, dessen Konfiguration aufgrund des Werts von CacheUpdateFrequency möglicherweise nicht so schnell aktualisiert wird. Infolgedessen kann der Proxy beginnen, Daten zu sammeln und an den Zabbix Server zu senden, der diese Daten ignoriert.

Zabbix Proxy ist ein Datensammler. Er berechnet keine Auslöser, verarbeitet keine Ereignisse und sendet keine Benachrichtigungen. Einen Überblick über die Funktionalität des Proxy bietet die folgende Tabelle:

Funktion		Vom Proxy unterstützt
Datenpunkte		
	Zabbix-Agent-Prüfungen	**Ja**
	Zabbix-Agent-Prüfungen (aktiv)	**Ja** [1]
	Einfache Prüfungen	**Ja**
	Trapper-Datenpunkte	**Ja**
	SNMP-Prüfungen	**Ja**
	SNMP-Traps	**Ja**
	IPMI-Prüfungen	**Ja**
	JMX-Prüfungen	**Ja**
	Logdateiüberwachung	**Ja**
	Interne Prüfungen	**Ja**
	SSH-Prüfungen	**Ja**
	Telnet-Prüfungen	**Ja**
	Externe Prüfungen	**Ja**
	Abhängige Datenpunkte	**Ja**
	Skript-Datenpunkte	**Ja**
	Browser-Datenpunkte	**Ja**
Integriertes Web-Monitoring		**Ja**
Vorverarbeitung von Datenpunktwerten		**Ja**
Netzwerkerkennung		**Ja**
Autoregistrierung aktiver Agents		**Ja**
Low-Level-Discovery		**Ja** [2]
Remote-Befehle		**Ja**
Berechnung von Auslösern		*Nein*
Verarbeitung von Ereignissen		*Nein*
Ereigniskorrelation		*Nein*
Senden von Benachrichtigungen		*Nein*

Note:
[1] Um sicherzustellen, dass 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 dann zur weiteren Verarbeitung an den Zabbix Server.

Schutz vor Überlastung

Wenn der Zabbix Server einige Zeit nicht verfügbar war und Proxys viele Daten gesammelt haben und der Server dann startet, kann er überlastet werden (die Nutzung des History-Cache bleibt für einige Zeit bei 95–100 %). Diese Überlastung kann zu Leistungseinbußen führen, wobei Prüfungen langsamer verarbeitet werden als vorgesehen. Zum Schutz vor diesem Szenario wurde ein Mechanismus implementiert, um Probleme zu vermeiden, die durch eine Überlastung des History-Cache entstehen.

Wenn der History-Cache des Zabbix Servers voll ist, wird der Schreibzugriff auf den History-Cache gedrosselt, wodurch die Prozesse zur Datenerfassung des Servers angehalten 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 Nutzung des History-Cache 80 % erreicht, wechselt der Zabbix Server in den Drosselungsmodus. Im Drosselungsmodus akzeptiert der Server Proxy-Daten nur, wenn die Nutzung des History-Cache unter 60 % liegt, wobei die akzeptierten Proxys rotiert werden. Sobald die Nutzung des History-Cache unter 20 % fällt, wechselt der Server wieder in den normalen Modus.

Zusätzlich drosselt der Zabbix Server im normalen Modus einzelne Proxys, die sehr große Pakete (10.000+ Datensätze) senden, wenn die Nutzung des History-Cache 60 % überschreitet. Diese Entscheidung wird in dem Moment getroffen, in dem der Server einen Proxy-Upload bewertet, und wird daher möglicherweise nicht immer sofort in den Diagrammen zur Nutzung des History-Cache angezeigt (der interne Datenpunkt `zabbix[wcache,history,pused]` und sein Aktualisierungsintervall erfassen kurze Spitzen möglicherweise nicht).

Dieser Drosselungsmodus bleibt bestehen, bis entweder die Cache-Nutzung erneut 80 % erreicht oder auf 20 % fällt oder die Drosselungsliste leer ist. Im ersten Fall akzeptiert der Server erneut 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-Modus	Aktion des Zabbix Servers
Erreicht 80 %	Warten	Akzeptiert keine Proxy-Daten mehr, führt jedoch weiterhin eine *Drosselungsliste* (priorisierte Liste von Proxys, die später kontaktiert werden sollen).
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 Drosselungsliste und beginnt wieder mit der normalen Annahme von Proxy-Daten.

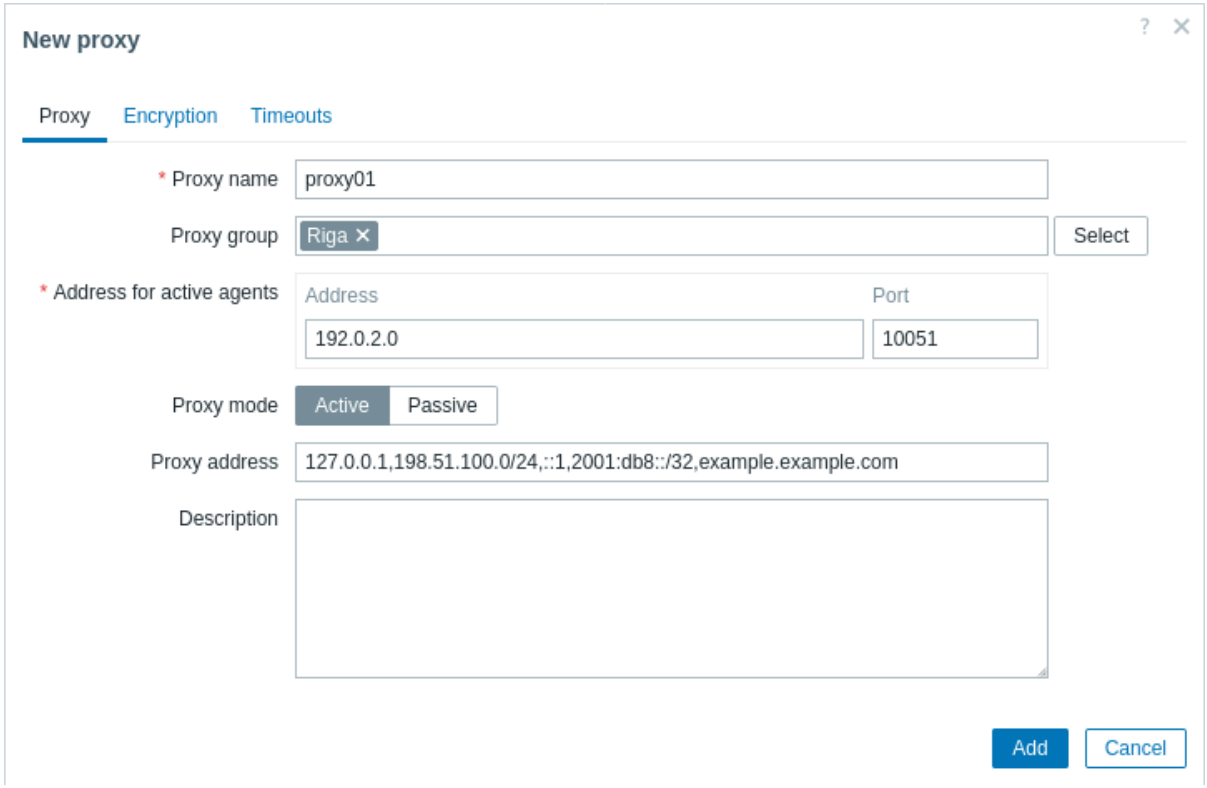
Sie können den internen Datenpunkt zabbix[wcache,history,pused] verwenden, um dieses Verhalten des Zabbix Servers mit einer Metrik zu korrelieren.

Konfiguration Sobald Sie einen Proxy installiert und konfiguriert haben, ist es an der Zeit, ihn im Zabbix Frontend zu konfigurieren.

Proxys hinzufügen

Um einen Proxy im Zabbix Frontend zu konfigurieren:

- Gehen Sie zu: *Administration* > *Proxies*
- Klicken Sie auf *Proxy erstellen*

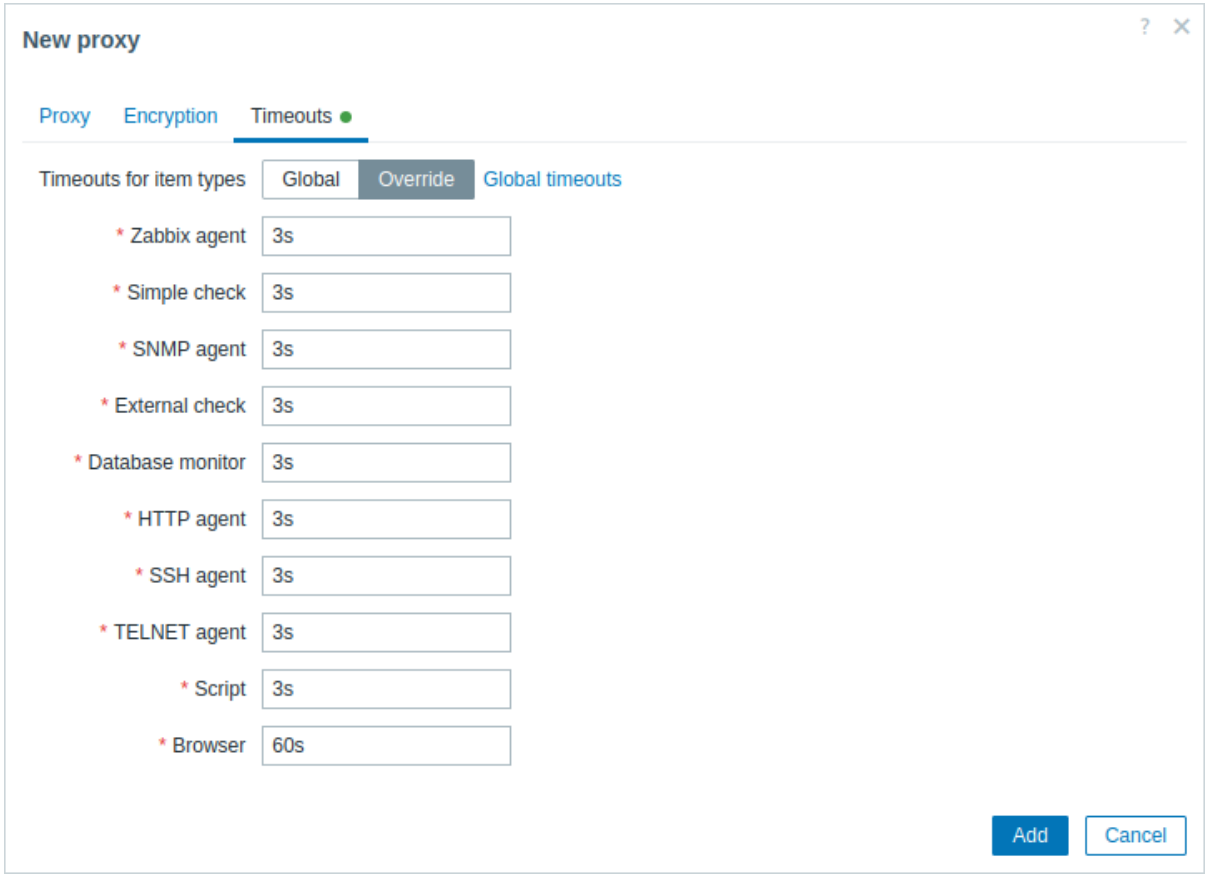


Parameter	Beschreibung
Proxy-Name	Geben Sie den Proxy-Namen ein. Er muss mit dem Namen im Parameter *Hostname* in der Proxy-Konfigurationsdatei übereinstimmen.
Proxy-Gruppe	Wählen Sie eine Proxy-Gruppe für Proxy-Lastverteilung/Hochverfügbarkeit aus.
Adresse für aktive Agents	Geben Sie die Adresse ein, mit der sich die überwachten aktiven Agents oder Sender verbinden müssen. Unterstützt **nur** für Zabbix 7.0 Agents oder neuer. Diese Adresse wird für die Verbindung zu aktiven und passiven Proxys verwendet. Dieses Feld ist nur verfügbar, wenn im Feld *Proxy-Gruppe* eine Proxy-Gruppe ausgewählt ist.
Adresse	IP-Adresse/DNS-Name für die Verbindung.
Port	TCP-Portnummer für die Verbindung (standardmäßig 10051). Benutzermakros werden unterstützt.
Proxy-Modus	Wählen Sie den Proxy-Modus aus. **Aktiv** - der Proxy verbindet sich mit dem Zabbix Server und fordert Konfigurationsdaten an **Passiv** - der Zabbix Server verbindet sich mit dem Proxy Beachten Sie, dass ohne verschlüsselte Kommunikation sensible 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 stattfindet oder Proxy-Adressen im Feld *Proxy-Adresse* nicht eingeschränkt sind.
Proxy-Adresse	Falls angegeben, werden Anfragen aktiver Proxys nur aus dieser durch Kommas getrennten Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen aktiver Zabbix-Proxys akzeptiert. Dieses Feld ist nur verfügbar, wenn im Feld *Proxy-Modus* ein aktiver Proxy ausgewählt ist. Makros werden nicht unterstützt.
Schnittstelle	Geben Sie die Schnittstellendetails für einen passiven Proxy ein. Dieses Feld ist nur verfügbar, wenn im Feld Proxy-Modus ein passiver Proxy ausgewählt ist.
Adresse	IP-Adresse/DNS-Name des passiven Proxys.
Port	TCP-Portnummer des passiven Proxys (standardmäßig 10051). Benutzermakros werden unterstützt.
Beschreibung	Geben Sie die Proxy-Beschreibung ein.

Die Registerkarte **Verschlüsselung** ermöglicht es Ihnen, verschlüsselte Verbindungen mit dem Proxy zu verlangen.

Parameter	Beschreibung
Verbindungen zum Proxy	Wie sich der Server mit dem passiven Proxy verbindet: keine Verschlüsselung (Standard), mit PSK (vorab geteilter Schlüssel) oder Zertifikat.
Verbindungen vom Proxy	Wählen Sie aus, welche Verbindungstypen vom aktiven Proxy erlaubt sind. Mehrere Verbindungstypen können gleichzeitig ausgewählt werden (nützlich zum Testen und zum Wechsel auf einen anderen Verbindungstyp). Standard ist „Keine Verschlüsselung“.
Aussteller	Erlaubter Aussteller des Zertifikats. Das Zertifikat wird zuerst mit der CA (Zertifizierungsstelle) validiert. Ist es gültig und von der CA signiert, kann das Feld *Aussteller* verwendet werden, um die erlaubte CA weiter einzuschränken. Dieses Feld ist optional und für den Einsatz gedacht, wenn Ihre Zabbix-Installation Zertifikate von mehreren CAs verwendet.
Betreff	Erlaubter Betreff des Zertifikats. Das Zertifikat wird zuerst mit der CA validiert. Ist es gültig und von der CA signiert, kann das Feld *Betreff* verwendet werden, um nur einen Wert der Zeichenfolge *Betreff* zuzulassen. Wenn dieses Feld leer ist, wird jedes gültige Zertifikat akzeptiert, das von der konfigurierten CA signiert wurde.
PSK-Identität	Identitätszeichenfolge des vorab geteilten Schlüssels. Geben Sie keine sensiblen Informationen in die PSK-Identität ein, da sie unverschlüsselt über das Netzwerk übertragen wird, um dem Empfänger mitzuteilen, welchen PSK er verwenden soll.
PSK	Vorab geteilter Schlüssel (Hex-Zeichenfolge). Maximale Länge: 512 Hex-Ziffern (256-Byte-PSK), wenn Zabbix die Bibliothek GnuTLS oder OpenSSL verwendet, 64 Hex-Ziffern (32-Byte-PSK), wenn Zabbix die Bibliothek mbed TLS (PolarSSL) verwendet. Beispiel: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Die Registerkarte **Timeouts** ermöglicht es Ihnen, globale Timeouts für Datenpunkttypen zu überschreiben, die dies unterstützen.



Parameter	Beschreibung
Timeouts für Datenpunkttypen	Legen Sie das Datenpunkt-Timeout fest (basierend auf seinem Typ): **Global** - globales Timeout verwenden (im ausgegrauten Feld *Timeout* pro Datenpunkttyp angezeigt); **Überschreiben** - ein benutzerdefiniertes Timeout festlegen (im Feld *Timeout* für jeden Datenpunkttyp). Zulässiger Bereich: 1 - 600s (Standard: von globalen Timeouts geerbt). Zeitsuffixe, z. B. 30s, 1m, und Benutzermakros werden unterstützt. Durch Klicken auf den Link *Globale Timeouts* können Sie globale Timeouts konfigurieren. Beachten Sie, dass der Link *Globale Timeouts* nur für Benutzer vom Typ *Super admin* sichtbar ist, die Berechtigungen für den Frontend-Bereich *Administration* > *General* haben. Beachten Sie, dass Timeouts auf Proxy-Ebene zwar die globalen überschreiben, aber von individuellen Datenpunkt-Timeouts überschrieben werden, falls diese konfiguriert sind.

Note:
Wenn die Hauptversion des Proxys nicht mit der Hauptversion des Servers übereinstimmt, wird neben *Timeouts für Datenpunkttypen* das Symbol mit der Hover-Meldung „Timeouts disabled because the proxy and server versions do not match“ angezeigt. In solchen Fällen verwendet der Proxy den Parameter `Timeout` aus der Proxy-Konfigurationsdatei.

Das Bearbeitungsformular eines vorhandenen Proxys enthält die folgenden zusätzlichen Schaltflächen:

- *Konfiguration aktualisieren* - Konfiguration des Proxys aktualisieren
- *Klonen* - einen neuen Proxy auf Basis der Eigenschaften des vorhandenen Proxys erstellen
- *Löschen* - den Proxy löschen

Host-Konfiguration

Sie können im Formular Host-Konfiguration über das Feld *Überwacht durch* 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.

1 Synchronisierung der Monitoring-Konfiguration

Übersicht

Diese Seite enthält Details zur Aktualisierung der Monitoring-Konfiguration für den Proxy, d. h. wie Änderungen an der Monitoring-Konfiguration 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 (das heißt, wenn keine Entitäten geändert wurden, wird nichts gesendet). Dieser Ansatz ermöglicht es, 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 sind als die Revision der Proxy-Konfiguration, werden in die an den Proxy gesendeten Konfigurationsdaten aufgenommen.

Die Entitäten für eine Konfigurationssynchronisierung sind wie folgt:

Entität	Details
Autoregistrierungs-TLS-Daten	Alle TLS-Daten der Autoregistrierung.
Ausdrücke	Alle Ausdrücke (reguläre Ausdrücke, Ausdruckstabellen).
globale Konfiguration	Globale Konfiguration, die in der Tabelle ’config’ definiert ist

Entität	Details
Host	Alle Eigenschaften, Schnittstellen, Inventar, Datenpunkte, Datenpunkt-Präprozessierung, Datenpunkt-Parameter, Webszenarien eines Hosts.
Host-Makros	Alle auf einem Host definierten Makros und alle mit ihm verknüpften Vorlagen-IDs.
Proxy-Erkennungsregel	Einem Proxy zugewiesene Discovery-Regeln und Prüfungen.

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 eine vollständige Konfigurationssynchronisierung beim Start/Neustart eines Proxy, bei einem HA-Failover, wenn sich das Session-Token geändert hat oder wenn die Konfigurationsaktualisierung auf dem Proxy fehlgeschlagen ist, erfolgt, zum Beispiel wenn die Verbindung während des Empfangs von Konfigurationsdaten unterbrochen wurde.

Konfigurationsparameter

Der Parameter **ProxyConfigFrequency** legt fest, wie oft die Proxy-Konfiguration mit dem Server synchronisiert wird (standardmäßig alle 10 Sekunden).

Beachten Sie, dass ProxyConfigFrequency ist:

- ein Server-Parameter für passive Proxys
- ein Proxy-Parameter für aktive Proxys

Bei aktiven Proxys ist ProxyConfigFrequency seit Zabbix 6.4 ein neuer Parameter und muss anstelle von ConfigFrequency verwendet werden, das inzwischen veraltet ist.

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 kennt keine Wartungszeiträume; siehe Berechnung von Warteschlangen während der Wartung für Details.

2 Proxy-Lastverteilung und Hochverfügbarkeit

Übersicht

Zabbix-Proxys können in Proxy-Gruppen organisiert werden, um Proxy-Lastverteilung und Hochverfügbarkeit zu ermöglichen.

Proxy-Lastverteilung und Hochverfügbarkeit ist 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 eine deutlich höhere/niedrigere Anzahl von Hosts als andere Proxys hat, werden seine Hosts auf andere Proxys verschoben, um die Proxy-Last auszugleichen.

Die Host-Umverteilung funktioniert nur zwischen Proxys in einer Gruppe, die die folgenden Bedingungen erfüllen:

- Proxys führen Zabbix 7.0 oder höher aus.
- Die Proxy-Version entspricht der Zabbix-Server-Version. Bei Verwendung von Zabbix Agent (passiv) muss die Proxy-Version der Agent-Version entsprechen. Aktive Agents erfordern nur Zabbix 7.0 oder höher.
- Die Proxy-Gruppe hat den Status online.
- Hosts sind so konfiguriert, dass sie von einer Proxy-Gruppe statt von einzelnen Proxys überwacht werden.

Note:
Der Zustand der Proxy-Gruppe kann mit internen Prüfungen durch jeden 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

Proxy-Lastverteilung und Hochverfügbarkeit 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. Eine Proxy-Lastverteilung findet ebenfalls statt, da Hosts Proxys mit den wenigsten zugewiesenen Hosts neu zugewiesen werden.

Zusätzlich wird die Proxy-Lastverteilung 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). Wenn das Ungleichgewicht nach einer Schonfrist (10 x Failover-Verzögerung) bestehen bleibt, wird die Proxy-Gruppe für die Host-Umverteilung in die Warteschlange gestellt.

Der Proxy-Gruppen-Manager verteilt Hosts nach folgender Logik um:

1. Berechnen Sie die durchschnittliche Anzahl von Hosts pro Proxy.
2. Bei Proxys mit Host-Überschuss werden die überschüssigen Hosts in den Pool nicht zugewiesener Proxys verschoben.
3. Bei Proxys mit Host-Defizit wird berechnet, wie viele Hosts benötigt werden, um ein Gleichgewicht zu erreichen.
4. Die erforderliche Anzahl von Hosts wird von den Proxys mit den meisten Hosts entfernt.
5. Nicht zugewiesene Hosts werden zu den Proxys mit den wenigsten Hosts verschoben.

Beispiele für die Host-Umverteilung:

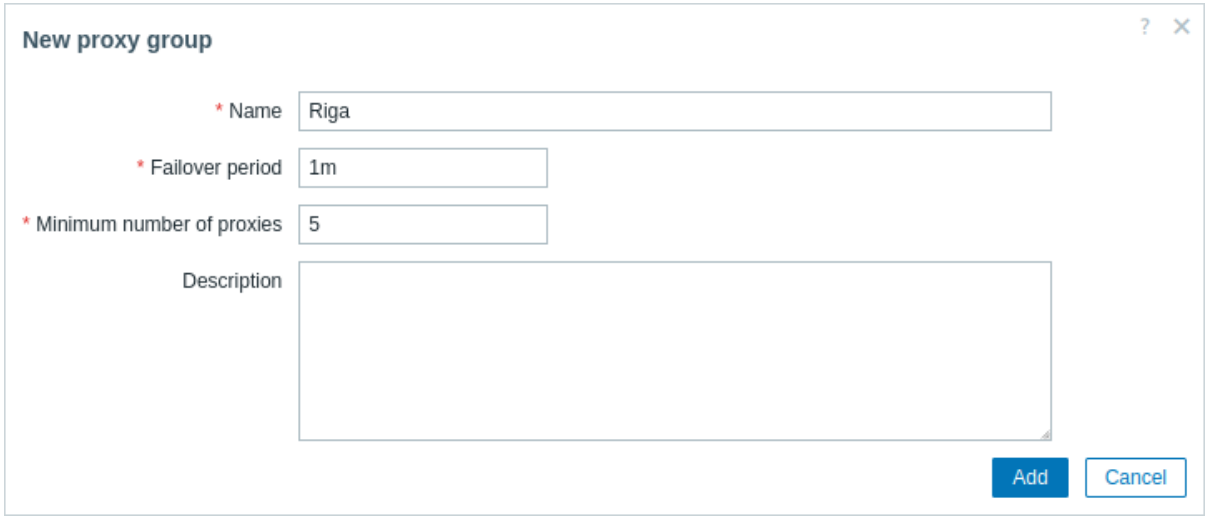
Hosts auf Proxy	Gruppendurchschnitt	Host-Neuzuweisung
100	50	Ja
60	50	Nein
40	50	Nein
25	50	Ja
15	5	Ja
10	5	Nein

Attention:
Wenn weniger als 10 Hosts von einer Proxy-Gruppe überwacht werden, kann dies zu einer ungleichmäßigen Host-Verteilung unter 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*



Parameter	Beschreibung
Name	Name der Proxy-Gruppe.
Failover period	Zeitraum in Sekunden, innerhalb dessen ein Proxy in der Proxy-Gruppe mit dem Zabbix Server kommunizieren muss, um als online zu gelten (Standard: 1m; Bereich: 10s–15m). Wenn der Proxy innerhalb dieses Zeitraums nicht kommuniziert, wird sein Status auf *Offline* geändert, und seine Hosts werden sofort auf andere Proxys verteilt. Der Lastausgleich des Proxys beginnt nach dem 10-Fachen dieses Zeitraums. Unterstützt Zeitsuffixe (z. B. 30s, 1m) und Benutzermakros.
Minimum number of proxies	Mindestanzahl an online proxies, die erforderlich ist, damit die proxy group online bleibt (Standard: 1; Bereich: 1–1000). Unterstützt Benutzermakros. Dieser Wert sollte kleiner sein als die Gesamtzahl der Proxys in der Gruppe. In einer Gruppe mit 10 Proxys führt beispielsweise die Einstellung des Minimums auf 10 dazu, dass die Gruppe offline geht, wenn ein Proxy ausfällt. Beachten Sie, dass online Proxys in einer offline Gruppe weiterhin normal funktionieren, Lastausgleich/High Availability jedoch nicht stattfindet.
Description	Beschreibung der Proxy-Gruppe.
Proxies	Zeigt beim Bearbeiten einer Gruppe mit Proxys eine Liste von bis zu fünf Proxys an (als Links oder als Klartext, abhängig von den Benutzerberechtigungen für Proxys).

Konfigurieren des Proxy-Load-Balancing

Um Proxy-Load-Balancing zu verwenden, müssen Sie eine Proxy-Gruppe im Zabbix Frontend konfigurieren (siehe oben) und sicherstellen, dass Hosts von einer Proxy-Gruppe überwacht werden, nicht von einzelnen Proxys (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 Zabbix Agent 7.0 (oder höher) mit Proxy-Gruppen im aktiven Modus funktioniert.

Wenn der Parameter `ServerActive` nur einen Proxy aus einer Proxy-Gruppe (oder den Zabbix Server) enthält, kann der Agent dennoch eine Verbindung zum richtigen Proxy herstellen. Wenn der Agent-Dienst startet und sich mit dem angegebenen Proxy verbindet, empfängt und zwischenspeichert der Agent die vollständige Liste der Proxy-IPs und deren aktuelle Last innerhalb der Gruppe. Anschließend werden aktive Prüfungen basierend auf der aktuellen Proxy-Host-Zuweisung innerhalb der Proxy-Gruppe an den richtigen online verfügbaren Proxy für den Host umgeleitet.

Warning:
Wenn im Parameter `ServerActive` des Zabbix Agent nur ein einzelner Proxy angegeben ist, kann dies zu verlorenen Monitoring-Daten führen, falls 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 online verfügbaren Proxy für den Host umgeleitet. 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 in der Lage sein, sich durch die Firewall mit allen Proxys in der Proxy-Gruppe zu verbinden. Andernfalls können aktive Prüfungen während der Umleitung oder des Failover hängen bleiben oder fehlschlagen. Zum Beispiel:

- Während aktiver Prüfungen kann ein Proxy den Agent an einen anderen Proxy umleiten. Wenn dieser Proxy durch eine Firewall blockiert ist, bleibt die Kommunikation hängen, während auf eine Antwort gewartet wird.
- In stabilen Hochverfügbarkeits-Setups ohne kürzlich erfolgte Neuverteilung kontaktieren Agents möglicherweise niemals Backup-Proxys. Wenn sich Firewall-Regeln geändert haben und nicht getestet wurden, kann das Failover fehlschlagen.

Testen des Proxy-Load-Balancing

So testen Sie das Proxy-Load-Balancing:

1. Konfigurieren Sie eine Proxy-Gruppe.
2. Stellen Sie sicher, dass die Proxy-Gruppe den Status online hat.
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 externer Konfiguration abhängen (z. B. Skripte für externe Prüfungen oder ODBC-Konfiguration für Datenbankprüfungen), müssen auf allen Proxys in der Proxy-Gruppe gleich konfiguriert sein.
- Datenbankprüfungen erfordern erweiterte Berechtigungen für das Datenbankobjekt bzw. den Server.
- 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, was eine zusätzliche Last auf vCenter verursacht.
- Hosts, die auf Grundlage von Autoregistrierungsdaten von einem Proxy in einer Proxy-Gruppe erstellt werden, werden so festgelegt, dass sie von dieser Proxy-Gruppe überwacht werden. Hosts, die jedoch auf Grundlage von Netzwerk-Erkennungsdaten von einem Proxy in einer Proxy-Gruppe erstellt werden, werden so festgelegt, dass sie von diesem Proxy überwacht werden.

15 Verschlüsselung

Übersicht Zabbix unterstützt verschlüsselte Kommunikation zwischen Zabbix-Komponenten unter Verwendung des Transport Layer Security (TLS)-Protokolls v.1.2 und 1.3 (abhängig von der Kryptobibliothek). Zertifikatsbasierte und auf vorinstallierten Schlüsseln basierende Verschlüsselung werden unterstützt.

Verschlüsselung kann für Verbindungen konfiguriert werden:

- Zwischen Zabbix Server, Zabbix Proxy, Zabbix Agent, Zabbix-Webservice sowie den Dienstprogrammen zabbix_sender und zabbix_get
- 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 Proxys und Agents können so konfiguriert werden, dass sie zertifikatsbasierte Verschlüsselung mit dem Server verwenden, während andere auf vorinstallierten Schlüsseln basierende Verschlüsselung nutzen und wieder andere weiterhin unverschlüsselte Kommunikation verwenden (wie bisher).
- Server (Proxy) kann unterschiedliche Verschlüsselungskonfigurationen für verschiedene Hosts verwenden.

Zabbix-Daemon-Programme verwenden einen Listening-Port sowohl für verschlüsselte als auch für unverschlüsselte eingehende Verbindungen. Das Hinzufügen von Verschlüsselung erfordert nicht das Öffnen neuer Ports in Firewalls.

Einschränkungen

- Private Schlüssel werden im Klartext in Dateien gespeichert, die beim Start von Zabbix-Komponenten lesbar sind.
- Pre-shared Keys werden im Zabbix Frontend eingegeben und im Klartext in der Zabbix-Datenbank 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 aufgebaut; Session-Caching und Tickets sind nicht implementiert.
- Das Hinzufügen von Verschlüsselung erhöht abhängig von der Netzwerklatenz die Zeit für Datenpunkt-Prüfungen und Aktionen:
  - Wenn die Paketverzögerung beispielsweise 100 ms beträgt, dauert das Öffnen einer TCP-Verbindung und das Senden einer unverschlüsselten Anfrage etwa 200 ms. Mit Verschlüsselung kommen für den Aufbau der TLS-Verbindung etwa 1000 ms hinzu.
  - Möglicherweise müssen Timeouts erhöht werden, andernfalls können einige Datenpunkte und Aktionen, die entfernte Skripte auf Agents ausführen, mit unverschlüsselten Verbindungen funktionieren, mit verschlüsselten jedoch wegen eines Timeouts fehlschlagen.
- Verschlüsselung wird von der Netzwerkerkennung nicht unterstützt. Von der Netzwerkerkennung durchgeführte Zabbix-Agent-Prüfungen sind unverschlüsselt, und wenn der Zabbix-Agent so konfiguriert ist, dass er unverschlüsselte Verbindungen ablehnt, werden solche Prüfungen nicht erfolgreich sein.

Kompilieren von Zabbix mit Verschlüsselungsunterstützung Um Verschlüsselung zu unterstützen, 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

- 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 PSK nicht verwenden, es können nur Zertifikate genutzt werden.

Note:
Weitere Informationen zur Einrichtung von SSL für das Zabbix Frontend finden Sie in diesen Best Practices.

Die Bibliothek wird ausgewählt, indem die entsprechende Option für das Skript "configure" angegeben 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 etwa 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*, ein Agent mit *GnuTLS*).

Attention:
Wenn Sie die Verwendung von Pre-Shared Keys (PSK) planen, 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-Chiffriersuiten mit Perfect Forward Secrecy. Ältere Versionen der *OpenSSL*-Bibliothek (1.0.1, 1.0.2c) unterstützen ebenfalls PSKs, aber die verfügbaren PSK-Chiffriersuiten bieten keine Perfect Forward Secrecy.

Verwaltung der Verbindungsverschlüsselung Verbindungen in Zabbix können Folgendes verwenden:

- keine Verschlüsselung (Standard)
- RSA-zertifikatbasierte Verschlüsselung
- PSK-basierte Verschlüsselung

Es gibt zwei wichtige Parameter, mit denen die Verschlüsselung zwischen Zabbix-Komponenten festgelegt wird:

- TLSConnect - gibt an, welche Verschlüsselung für ausgehende Verbindungen verwendet werden soll (unverschlüsselt, PSK oder Zertifikat)
- TLSAccept - gibt an, welche Verbindungstypen für eingehende Verbindungen erlaubt sind (unverschlüsselt, PSK oder Zertifikat). Es können ein oder mehrere Werte angegeben werden.

`TLSConnect` wird in den Konfigurationsdateien für Zabbix Proxy (im aktiven Modus gibt dies nur Verbindungen zum Server an) und Zabbix Agent (für aktive Prüfungen) verwendet. Im Zabbix Frontend ist das Äquivalent zu TLSConnect das Feld *Verbindungen zum Host* im Reiter *Datenerfassung → Hosts → <ein Host> → Verschlüsselung* sowie das Feld *Verbindungen zum Proxy* im Reiter *Administration → Proxys → <ein Proxy> → Verschlüsselung*. Wenn der konfigurierte Verschlüsselungstyp für die Verbindung fehlschlägt, werden keine anderen Verschlüsselungstypen ausprobiert.

`TLSAccept` wird in den Konfigurationsdateien für Zabbix Proxy (im passiven Modus gibt dies nur Verbindungen vom Server an) und Zabbix Agent (für passive Prüfungen) verwendet. Im Zabbix Frontend ist das Äquivalent zu TLSAccept das Feld *Verbindungen vom Host* im Reiter *Datenerfassung → Hosts → <ein Host> → Verschlüsselung* sowie das Feld *Verbindungen vom Proxy* im Reiter *Administration → Proxys → <ein Proxy> → Verschlüsselung*.

Normalerweise konfigurieren Sie nur einen Verschlüsselungstyp für eingehende Verbindungen. Möglicherweise möchten Sie den Verschlüsselungstyp jedoch wechseln, z. B. von unverschlüsselt zu zertifikatbasiert, mit minimaler Ausfallzeit und der Möglichkeit zum Rollback. Gehen Sie dazu wie folgt vor:

- Setzen Sie `TLSAccept=unencrypted,cert` in der Konfigurationsdatei des Agent und starten Sie den Zabbix Agent neu.
- Testen Sie die Verbindung zum Agent mit zabbix_get unter Verwendung eines Zertifikats. Wenn dies funktioniert, können Sie die Verschlüsselung für diesen Agent im Zabbix Frontend im Reiter *Datenerfassung → Hosts → <ein Host> → Verschlüsselung* neu konfigurieren, indem Sie *Verbindungen zum Host* auf „Zertifikat" setzen.
- Wenn der Konfigurations-Cache des Server aktualisiert wird (und die Proxy-Konfiguration aktualisiert wird, falls der Host von einem Proxy überwacht wird), werden die Verbindungen zu diesem Agent verschlüsselt.
- Wenn alles wie erwartet funktioniert, können Sie `TLSAccept=cert` in der Konfigurationsdatei des Agent setzen und den Zabbix Agent neu starten. Der Agent akzeptiert dann nur noch verschlüsselte zertifikatbasierte Verbindungen. Unverschlüsselte und PSK-basierte Verbindungen werden abgelehnt.

Auf ähnliche Weise funktioniert dies bei Server und 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 zertifikatbasiert verschlüsselte Verbindungen akzeptiert.

Sehr wahrscheinlich werden Sie eingehende und ausgehende Verbindungen so konfigurieren, dass derselbe Verschlüsselungstyp verwendet wird oder überhaupt keine Verschlüsselung. Technisch ist jedoch auch eine asymmetrische Konfiguration möglich, z. B. zertifikatbasierte Verschlüsselung für eingehende und PSK-basierte für ausgehende Verbindungen.

Die Verschlüsselungskonfiguration für jeden Host wird im Zabbix Frontend unter *Datenerfassung* → *Hosts* in der Spalte *Agent-Verschlüsselung* angezeigt. Zum Beispiel:

Beispiel	Verbindungen zum Host	Erlaubte Verbindungen vom Host	Abgelehnte Verbindungen vom Host
NONE	Unverschlüsselt	Unverschlüsselt	Verschlüsselt, zertifikatbasiert und PSK-basiert verschlüsselt
CERT NONE PSK CERT	Verschlüsselt, zertifikatbasiert	Verschlüsselt, zertifikatbasiert	Unverschlüsselt und PSK-basiert verschlüsselt
PSK NONE PSK CERT	Verschlüsselt, PSK-basiert	Verschlüsselt, PSK-basiert	Unverschlüsselt und zertifikatbasiert verschlüsselt
PSK NONE PSK CERT	Verschlüsselt, PSK-basiert	Unverschlüsselt und PSK-basiert verschlüsselt	Zertifikatbasiert verschlüsselt
CERT NONE PSK CERT	Verschlüsselt, zertifikatbasiert	Unverschlüsselt, PSK- oder zertifikatbasiert 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 zu zabbix_get und zabbix_sender für Informationen zur Verwendung mit Verschlüsselung.

Cipher-Suites Cipher-Suites werden standardmäßig intern beim Start von Zabbix konfiguriert.

Auch benutzerkonfigurierte Cipher-Suites werden für GnuTLS und OpenSSL unterstützt. Benutzer können Cipher-Suites entsprechend ihren Sicherheitsrichtlinien konfigurieren. Die Verwendung dieser Funktion ist optional (die integrierten Standard-Cipher-Suites funktionieren weiterhin).

Bei Kryptobibliotheken, die mit Standardeinstellungen kompiliert wurden, führen die integrierten Zabbix-Regeln typischerweise zu den folgenden Cipher-Suites (in der Reihenfolge von höherer zu niedrigerer Priorität):

Library	Zertifikat-Cipher-Suites	PSK-Cipher-Suites
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	Zertifikat-Cipher-Suites	PSK-Cipher-Suites
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

Benutzerdefinierte Ciphersuites Die integrierten Auswahlkriterien für Ciphersuites können durch benutzerdefinierte Ciphersuites überschrieben werden.

Attention:
Benutzerdefinierte Ciphersuites sind eine Funktion für fortgeschrittene Benutzer, die TLS-Ciphersuites, deren Sicherheit und die Folgen von Fehlern verstehen und mit der Fehlerbehebung bei TLS vertraut sind.

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).	Auswahlkriterien für zertifikatbasierte Ciphersuites für TLS 1.3 Nur OpenSSL 1.1.1 oder neuer.
	TLSCipherCert	Gültige OpenSSL-cipher strings für TLS 1.2 oder gültige GnuTLS-priority strings. Ihre Werte werden jeweils an die Funktionen `SSL_CTX_set_cipher_list()` oder `gnutls_priority_init()` übergeben.	Auswahlkriterien für zertifikatbasierte 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).	Auswahlkriterien für PSK-basierte Ciphersuites für TLS 1.3 Nur OpenSSL 1.1.1 oder neuer.
	TLSCipherPSK	Gültige OpenSSL-cipher strings für TLS 1.2 oder gültige GnuTLS-priority strings. Ihre Werte werden jeweils an die Funktionen `SSL_CTX_set_cipher_list()` oder `gnutls_priority_init()` übergeben.	Auswahlkriterien für PSK-basierte 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-priority strings. Ihre Werte werden jeweils an die Funktionen `SSL_CTX_set_cipher_list()` oder `gnutls_priority_init()` übergeben.	Auswahlkriterien für Ciphersuites für TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Um die Ciphersuite-Auswahl 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, werden 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 Befehlszeilenparameter `--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-Verbindungseinrichtung	Cipher-Konfiguration
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**:

Verbindungseinrichtung	Cipher-Konfiguration
Ausgehende Verbindungen mit PSK	TLSCipherPSK, TLSCipherPSK13
Eingehende Verbindungen mit Zertifikaten	TLSCipherAll, TLSCipherAll13
Eingehende Verbindungen mit PSK, wenn der Server kein Zertifikat hat	TLSCipherPSK, TLSCipherPSK13
Eingehende Verbindungen mit PSK, wenn der Server ein Zertifikat hat	TLSCipherAll, TLSCipherAll13

In den beiden obigen Tabellen ist ein bestimmtes Muster zu erkennen:

- TLSCipherAll und TLSCipherAll13 können nur angegeben werden, wenn eine kombinierte Liste von zertifikat- **und** PSK-basierten Cipher-Suites verwendet wird. Dafür gibt es zwei Fälle: Server (Proxy) mit einem konfigurierten Zertifikat (PSK-Cipher-Suites werden auf Server und Proxy immer konfiguriert, wenn die Kryptobibliothek PSK unterstützt), Agent so konfiguriert, dass sowohl zertifikat- als auch PSK-basierte eingehende Verbindungen akzeptiert werden
- in anderen Fällen sind TLSCipherCert* und/oder TLSCipherPSK* ausreichend

Die folgenden Tabellen zeigen die integrierten Standardwerte von `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	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128

Parameter	OpenSSL 1.1.1d [1]
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

Nachfolgend finden Sie 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 nur von PFS-Cipher-Suites

Um zu sehen, welche Cipher-Suites ausgewählt wurden, müssen Sie `DebugLevel=4` in der Konfigurationsdatei setzen oder die Option `-vv` für zabbix_sender verwenden.

Möglicherweise ist etwas Experimentieren mit den Parametern `TLSCipher*` erforderlich, bevor Sie die gewünschten Cipher-Suites erhalten. Es ist unpraktisch, den Zabbix Server, Proxy oder Agent mehrfach neu zu starten, nur um die Parameter `TLSCipher*` anzupassen. Bequemer ist die Verwendung von zabbix_sender oder des Befehls `openssl`. Sehen wir uns beides an.

1. Verwendung von zabbix_sender.

Erstellen wir eine Test-Konfigurationsdatei, 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 Zertifikats- und PSK-Dateipfade und -namen an Ihre Umgebung an.

Wenn Sie keine Zertifikate, sondern nur PSK verwenden, können Sie eine einfachere Testdatei erstellen:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Die ausgewählten Cipher-Suites 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 Cipher-Suites. Diese Standardwerte wurden so gewählt, dass die Interoperabilität mit Zabbix Agents auf Systemen mit älteren OpenSSL-Versionen (ab 1.0.1) gewährleistet ist.

Auf neueren Systemen können Sie die Sicherheit erhöhen, indem Sie nur wenige Cipher-Suites zulassen, z. B. nur Cipher-Suites mit PFS (Perfect Forward Secrecy). Versuchen wir, mit den Parametern `TLSCipher*` nur Cipher-Suites mit PFS zuzulassen.

Attention:
Das Ergebnis ist nicht interoperabel mit Systemen, die OpenSSL 1.0.1 und 1.0.2 verwenden, wenn PSK verwendet wird. Zertifikatbasierte 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 wie erwartet nur TLS 1.3-Cipher-Suites und TLS 1.2 ECDHE-*-Cipher-Suites.

2. TLSCipherAll und TLSCipherAll13 können nicht mit zabbix_sender getestet werden; sie beeinflussen nicht den Wert „certificate and PSK ciphersuites“, der im obigen Beispiel angezeigt wird. Um TLSCipherAll und TLSCipherAll13 anzupassen, müssen Sie mit dem Agent, Proxy oder Server experimentieren.

Um also nur PFS-Cipher-Suites zuzulassen, müssen Sie möglicherweise bis zu drei Parameter

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

zu zabbix_agentd.conf, zabbix_proxy.conf und zabbix_server_conf hinzufügen, wenn für jeden von ihnen ein Zertifikat konfiguriert ist und der Agent zusätzlich PSK verwendet.

Wenn Ihre Zabbix-Umgebung nur PSK-basierte Verschlüsselung und keine Zertifikate verwendet, dann nur diesen einen:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Nachdem Sie nun verstanden haben, wie es funktioniert, können Sie die Auswahl der Cipher-Suites sogar 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` für eine ausführlichere Ausgabe:

```
$ 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 Prioritäts-Strings 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 als integrierten Standard für Daten. Nehmen wir an, Sie verwenden Zertifikate und möchten unter OpenSSL 1.1.1 auf AES256 umstellen.

Dies kann durch Hinzufügen der entsprechenden Parameter in `zabbix_server.conf` erreicht 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, sind auch die Parameter `TLSCipherPSK*` definiert, um deren Standardwerte zu vermeiden, die für eine breitere Interoperabilität weniger sichere Chiffren 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
```

1 Verwendung von Zertifikaten

Übersicht

Zabbix kann RSA-Zertifikate im PEM-Format verwenden, die von einer öffentlichen oder einer internen Zertifizierungsstelle (CA) signiert sind.

Die Zertifikatsprüfung wird anhand eines vorkonfigurierten CA-Zertifikats durchgeführt. 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 Erstellen und Signieren von Zertifikatsanfragen sowie zum Widerrufen von Zertifikaten finden Sie in Anleitungen 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.

Konfigurationsparameter für Zertifikate

Die folgenden Konfigurationsparameter werden für die Einrichtung von Zertifikaten auf Zabbix-Komponenten unterstützt.

Parameter	Pflichtfeld	Beschreibung
TLSCAFile	ja	Vollständiger Pfadname zu einer Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält. Wenn eine Zertifikatskette mit mehreren Gliedern verwendet wird, ordnen Sie die Zertifikate so an, dass die Zertifikate der CA(s) niedrigerer Ebene zuerst stehen, gefolgt von den Zertifikaten der CA(s) höherer Ebene. Zertifikate von mehreren CAs können in einer einzigen Datei enthalten sein.
TLSCRLFile	nein	Vollständiger Pfadname zu einer Datei, die Certificate Revocation Lists (CRL) enthält.
TLSCertFile	ja	Vollständiger Pfadname zu einer Datei, die das Zertifikat enthält. Wenn eine Zertifikatskette mit mehreren Gliedern verwendet wird, ordnen Sie die Zertifikate so an, dass das Zertifikat des Servers, Proxys oder Agents zuerst steht, gefolgt von den Zertifikaten der CA(s) niedrigerer Ebene und abschließend von den Zertifikaten der CA(s) höherer Ebene.

Parameter	Pflichtfeld	Beschreibung
TLSKeyFile	ja	Vollständiger Pfadname zu einer Datei, die den privaten Schlüssel enthält. Stellen Sie sicher, dass diese Datei nur für den Zabbix-Benutzer lesbar ist, indem Sie entsprechende Zugriffsrechte setzen.
TLSServerCertIssuer	nein	Zulässiger Aussteller des Serverzertifikats.
TLSServerCertSubject	nein	Zulässiger Betreff des Serverzertifikats.

Konfigurationsbeispiele

Nach dem Einrichten der erforderlichen Zertifikate konfigurieren Sie die Zabbix-Komponenten so, dass sie zertifikatbasierte 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 verifizieren, 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 sie 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, zum Beispiel unter `/home/zabbix/zabbix_server.crt`.
Das erste Zertifikat ist das Zabbix-Server-Zertifikat, gefolgt vom Zwischen-CA-Zertifikat:

```
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/wUtcVIvWXdMIFJMOHw==
-----END CERTIFICATE-----
```

Note:
Verwenden Sie sowohl für Client- als auch für Server-Zertifikate nur die oben genannten Attribute, um den Zertifikatsverifizierungsprozess nicht zu beeinflussen. Beispielsweise kann OpenSSL 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, zum Beispiel unter `/home/zabbix/zabbix_server.key`:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Bearbeiten Sie die TLS-Konfigurationsparameter in der Zabbix-Server-Konfigurationsdatei:

```
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 CA-Zertifikaten der obersten Ebene, dem Zertifikat/der Zertifikatskette des Zabbix Proxy sowie dem privaten Schlüssel vor, wie im Abschnitt Zabbix Server beschrieben. Bearbeiten Sie dann die Parameter `TLSCAFile`, `TLSCertFile` und `TLSKeyFile` entsprechend in der Zabbix Proxy-Konfigurationsdatei.

2. Bearbeiten Sie zusätzliche TLS-Parameter in der Zabbix Proxy-Konfigurationsdatei:

- Für aktiven Proxy: `TLSConnect=cert`
- Für passiven Proxy: `TLSAccept=cert`

Note:
Um die Sicherheit des Proxy zu verbessern, können Sie auch die Parameter `TLSServerCertIssuer` und `TLSServerCertSubject` festlegen. Weitere Informationen finden Sie unter Einschränken des zulässigen Zertifikat-ausstellers und -betreffs.

Die TLS-Parameter in der endgültigen Proxy-Konfigurationsdatei können beispielsweise wie folgt aussehen:

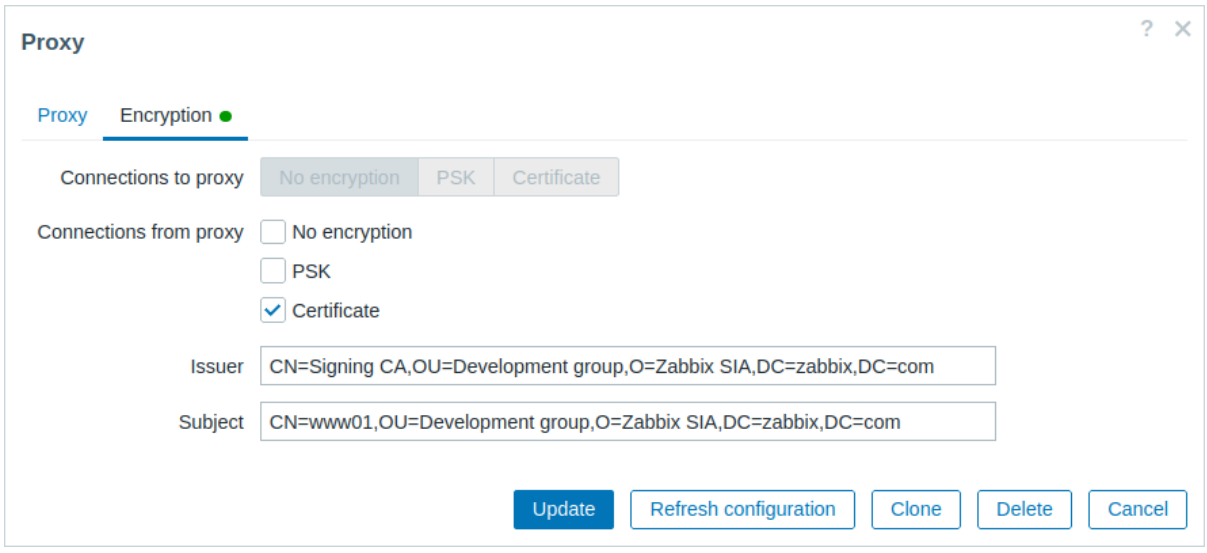
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Konfigurieren Sie die Verschlüsselung für diesen Proxy im Zabbix Frontend:

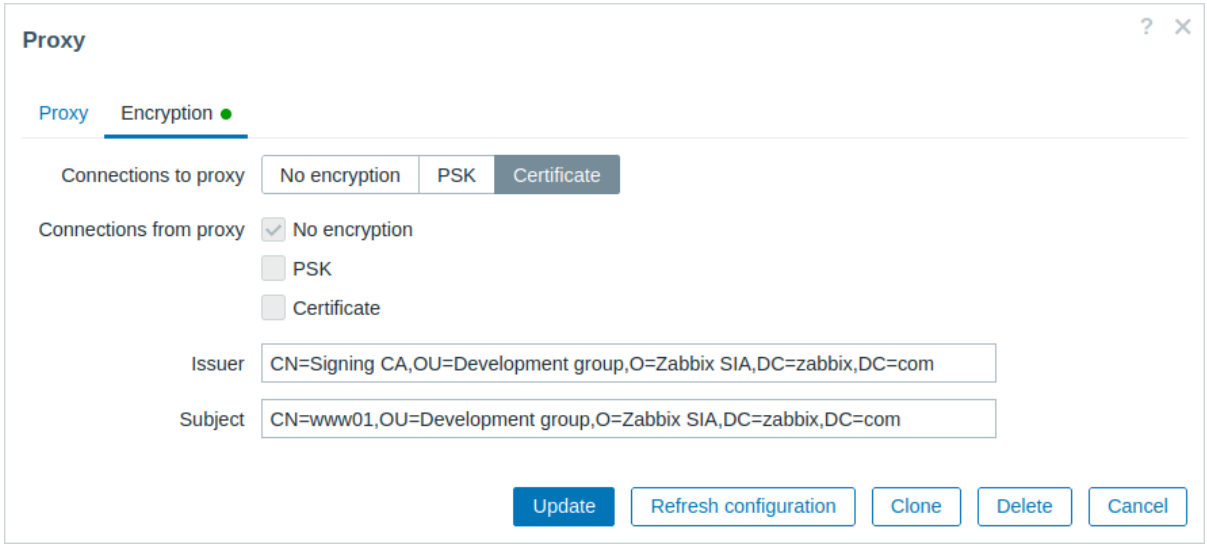
- Gehen Sie zu: *Verwaltung* → *Proxys*.
- Wählen Sie den Proxy aus und klicken Sie auf die Registerkarte *Verschlüsselung*.

In den folgenden Beispielen sind die Felder *Issuer* und *Subject* ausgefüllt. Weitere Informationen dazu, warum und wie diese Felder verwendet werden, finden Sie unter Einschränken des zulässigen Zertifikatsausstellers und -betreffs.

Für aktiven Proxy:



Für passiven Proxy:



Zabbix Agent

1. Bereiten Sie Dateien mit den CA-Zertifikaten der obersten Ebene, dem Zabbix-Agent-Zertifikat/Zertifikatskette und dem privaten Schlüssel vor, wie im Abschnitt Zabbix Server beschrieben. Bearbeiten Sie dann die Parameter `TLSCAFile`, `TLSCertFile` und `TLSKeyFile` in der Zabbix-Agent-Konfigurationsdatei entsprechend.

2. Bearbeiten Sie zusätzliche TLS-Parameter in der Zabbix-Agent-Konfigurationsdatei:

- Für aktiven Agent: `TLSConnect=cert`
- Für passiven Agent: `TLSAccept=cert`

Note:
Um die Sicherheit des Agent zu verbessern, können Sie die Parameter `TLSServerCertIssuer` und `TLSServerCertSubject` festlegen. Weitere Informationen finden Sie unter Einschränken des zulässigen Zertifikat-ausstellers und -betreffs.

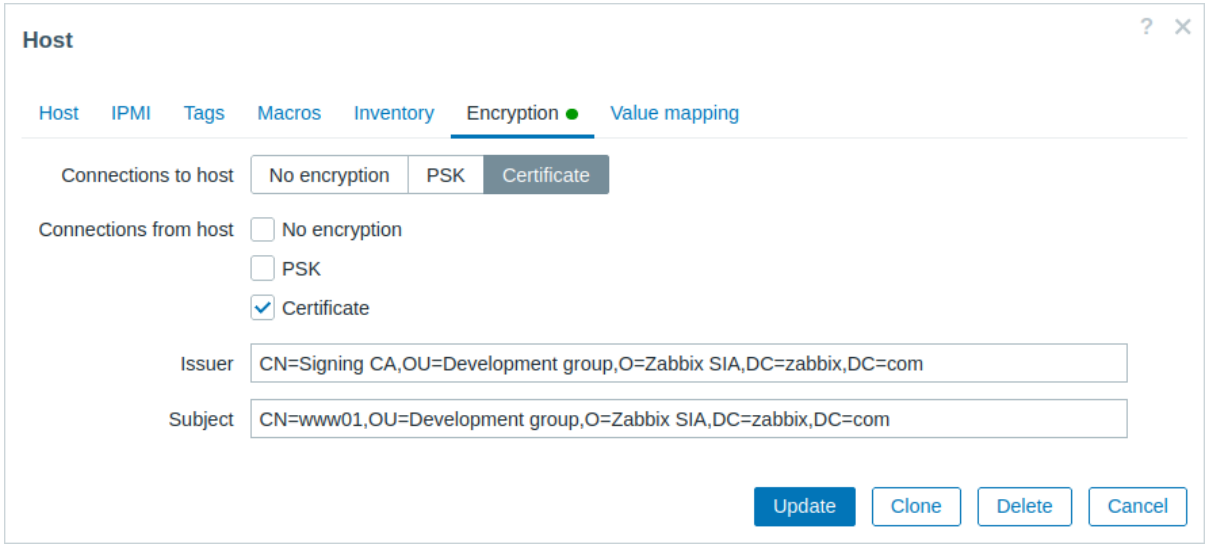
Die TLS-Parameter in der endgültigen Agent-Konfigurationsdatei können beispielsweise wie folgt aussehen. Beachten Sie, dass das Beispiel davon ausgeht, dass der Host von einem Proxy überwacht wird; daher ist dieser als Zertifikatsbetreff angegeben:

```
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: *Datenerfassung* → *Hosts*.
- Wählen Sie den Host aus und klicken Sie auf die Registerkarte *Verschlüsselung*.

Im folgenden Beispiel sind die Felder *Aussteller* und *Betreff* ausgefüllt. Weitere Informationen dazu, warum und wie diese Felder verwendet werden, finden Sie unter Einschränken des zulässigen Zertifikatsausstellers und -betreffs.



Zabbix-Webservice

1. Bereiten Sie Dateien mit den CA-Zertifikaten der obersten Ebene, dem Zertifikat/der Zertifikatskette der Zabbix-Webservice und dem privaten Schlüssel vor, wie im Abschnitt Zabbix-Server beschrieben. Bearbeiten Sie dann die Parameter `TLSCAFile`, `TLSCertFile` und `TLSKeyFile` entsprechend in der Zabbix-Webservice-Konfigurationsdatei.

2. Bearbeiten Sie einen zusätzlichen TLS-Parameter in der Zabbix-Webservice-Konfigurationsdatei: `TLSAccept=cert`

Die TLS-Parameter in der endgültigen Webservice-Konfigurationsdatei 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 zur TLS-konfigurierten Zabbix-Webservice herstellt, indem Sie den Parameter `WebServiceURL` in der Zabbix-Server-Konfigurationsdatei bearbeiten:

```
WebServiceURL=https://example.com:443/report
```

Einschränkung zulässiger Zertifikatsaussteller und -subjekte

Wenn zwei Zabbix-Komponenten (zum Beispiel Server und Agent) eine TLS-Verbindung aufbauen, prüfen sie die Zertifikate der jeweils anderen Seite. Wenn ein Zertifikat der Gegenstelle von einer vertrauenswürdigen CA signiert ist (mit einem vorkonfigurierten Stammzertifikat in `TLSCAFile`), gültig ist, nicht abgelaufen ist und andere Prüfungen besteht, kann die Kommunikation zwischen den Komponenten fortgesetzt werden. In diesem einfachsten Fall werden Zertifikatsaussteller und -subjekt nicht überprüft.

Dies birgt jedoch ein Risiko: Jeder mit einem gültigen Zertifikat kann sich als jemand anderes ausgeben (zum Beispiel 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, reicht jedoch in größeren oder sicherheitssensibleren Umgebungen möglicherweise nicht aus.

Wenn Ihre Stamm-CA Zertifikate ausstellt, die von Zabbix nicht akzeptiert werden sollen, oder wenn Sie das Risiko einer Identitätsvortäuschung verringern möchten, können Sie zulässige Zertifikate einschränken, indem Sie deren Aussteller und Subjekt angeben.

Zum Beispiel könnten Sie in der Zabbix-Proxy-Konfigurationsdatei 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 hat. Ebenso akzeptiert ein passiver Proxy keine Anfragen von einem solchen Server.

Regeln für den Abgleich von `Issuer`- und `Subject`-Zeichenfolgen

Die Regeln für den Abgleich von `Issuer`- und `Subject`-Zeichenfolgen sind wie folgt:

- `Issuer`- und `Subject`-Zeichenfolgen werden unabhängig voneinander geprüft. Beide sind optional.
- Eine nicht angegebene Zeichenfolge bedeutet, dass jede Zeichenfolge akzeptiert wird.
- Zeichenfolgen werden *unverändert* verglichen und müssen exakt übereinstimmen.
- UTF-8-Zeichen werden unterstützt. Platzhalter (`*`) oder reguläre Ausdrücke werden jedoch nicht unterstützt.

- Die folgenden Anforderungen aus RFC 4514 sind implementiert – Zeichen, die maskiert werden müssen (mit einem Backslash '\', U+005C):
  - an beliebiger Stelle in der Zeichenfolge: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - am Anfang der Zeichenfolge: Leerzeichen (' ', U+0020) oder Nummernzeichen ('#', U+0023);
  - am Ende der Zeichenfolge: Leerzeichen (' ', U+0020).
- Nullzeichen (U+0000) werden nicht unterstützt. Wenn ein Nullzeichen erkannt wird, schlägt der Abgleich fehl.
- Die Standards RFC 4517 und RFC 4518 werden nicht unterstützt.

Wenn beispielsweise die Organisationszeichenfolgen (O) von `Issuer` und `Subject` nachgestellte Leerzeichen enthalten und die Zeichenfolge der Organisationseinheit (OU) 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, das eine „umgekehrte" Reihenfolge für diese Felder festlegt, beginnend mit den Feldern der niedrigsten Ebene (CN), weiter zu den Feldern der mittleren Ebene (OU, O) und abschließend mit 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* für Zabbix korrekt 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-Zertifikaterweiterungen

Bei der Implementierung von X.509-v3-Zertifikaten in Zabbix werden bestimmte Erweiterungen möglicherweise nicht vollständig unterstützt oder können zu inkonsistentem Verhalten führen.

Erweiterung „Subject Alternative Name"

Zabbix unterstützt die Erweiterung *Subject Alternative Name* nicht, die zur Angabe alternativer DNS-Namen wie IP-Adressen oder E-Mail-Adressen verwendet wird. Zabbix kann nur den Wert im Feld *Subject* des Zertifikats validieren (siehe Einschränken des zulässigen Zertifikatausstellers und Subject). Wenn Zertifikate das Feld `subjectAltName` enthalten, kann das Ergebnis der Zertifikatsvalidierung je nach den spezifischen Kryptografie-Toolkits variieren, mit denen die Zabbix-Komponenten kompiliert wurden. Daher kann Zabbix Zertifikate abhängig von diesen Kombinationen entweder akzeptieren oder ablehnen.

Erweiterung „Extended Key Usage"

Zabbix unterstützt die Erweiterung *Extended Key Usage*. Wenn sie jedoch verwendet wird, ist es im Allgemeinen erforderlich, dass sowohl die Attribute *clientAuth* (für TLS-WWW-Client-Authentifizierung) als auch *serverAuth* (für TLS-WWW-Server-

Authentifizierung) angegeben werden. Zum 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 bei Verstößen gegen die Schlüsselnutzung möglicherweise eine Warnung ausgibt, wird die Kommunikation trotz dieser Warnungen in der Regel zugelassen.

Erweiterung „Name Constraints“

Die Unterstützung für die Erweiterung *Name Constraints* variiert zwischen Kryptografie-Toolkits. Stellen Sie sicher, dass das von Ihnen gewählte Toolkit diese Erweiterung unterstützt. Je nach verwendetem Toolkit kann diese Erweiterung Zabbix daran hindern, CA-Zertifikate zu laden, wenn dieser Abschnitt als kritisch markiert ist.

Zertifikatsperrlisten (CRL)

Wenn ein Zertifikat kompromittiert ist, kann die Zertifizierungsstelle (CA) es widerrufen, indem sie das Zertifikat in eine Zertifikatsperrliste (CRL) aufnimmt. CRLs werden über Konfigurationsdateien verwaltet und können mit dem Parameter `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 von mehreren CAs enthalten und könnte wie folgt 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 werden und CRLs verwendet werden, stellen Sie sicher, dass für jede Stamm-CA und Zwischen-CA in den Zertifikatsketten eine entsprechende CRL (auch wenn sie leer ist) in `TLSCRLFile` enthalten ist.

2 Verwendung von Pre-Shared Keys

Übersicht

Jeder Pre-Shared Key (PSK) in Zabbix ist tatsächlich ein Paar aus:

- einer nicht geheimen PSK-Identitätszeichenfolge,
- einem geheimen PSK-Zeichenfolgenwert.

Die PSK-Identitätszeichenfolge ist eine nicht leere UTF-8-Zeichenfolge. Zum Beispiel „PSK ID 001 Zabbix agentd“. Es handelt sich um einen eindeutigen Namen, mit dem dieser spezifische PSK von Zabbix-Komponenten referenziert wird. Geben Sie keine sensiblen Informationen in der PSK-Identitätszeichenfolge an – sie wird unverschlüsselt über das Netzwerk übertragen.

Der PSK-Wert ist eine schwer zu erratende Zeichenfolge aus hexadezimalen Ziffern, zum Beispiel „e560cb0d918d26d31b4f642181f5f570ad89a

Größenbeschränkungen

In Zabbix gibt es Größenbeschränkungen für PSK-Identität und -Wert; in einigen Fällen kann eine Kryptobibliothek niedrigere Grenzwerte 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 hexadezimale Ziffern)	2048 Bit (256-Byte-PSK, eingegeben als 512 hexadezimale Ziffern)
GnuTLS	128 Byte (kann UTF-8-Zeichen enthalten)	-	2048 Bit (256-Byte-PSK, eingegeben als 512 hexadezimale Ziffern)
OpenSSL 1.0.x, 1.1.0	127 Byte (kann UTF-8-Zeichen enthalten)	-	2048 Bit (256-Byte-PSK, eingegeben als 512 hexadezimale Ziffern)
OpenSSL 1.1.1	127 Byte (kann UTF-8-Zeichen enthalten)	-	512 Bit (64-Byte-PSK, eingegeben als 128 hexadezimale Ziffern)
OpenSSL 1.1.1a und höher	127 Byte (kann UTF-8-Zeichen enthalten)	-	2048 Bit (256-Byte-PSK, eingegeben als 512 hexadezimale Ziffern)

Attention:
Das Zabbix Frontend erlaubt die Konfiguration einer bis zu 128 Zeichen langen PSK-Identitätszeichenfolge und eines bis zu 2048 Bit langen PSK, unabhängig von den verwendeten Kryptobibliotheken.
Wenn einige Zabbix-Komponenten niedrigere Grenzwerte unterstützen, liegt es in der Verantwortung des Benutzers, PSK-Identität und -Wert mit einer für diese Komponenten zulässigen Länge zu konfigurieren.
Das Überschreiten der Längenbeschränkungen führt zu Kommunikationsfehlern zwischen Zabbix- Komponenten.

Bevor der Zabbix Server sich mit dem Agent über PSK verbindet, sucht der Server die für diesen Agent in der Datenbank konfigurierte PSK-Identität und den PSK-Wert nach (tatsächlich im Konfigurationscache). 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 sein.

Attention:
Jede PSK-Identität darf nur mit genau 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 dies zu unvorhersehbaren Fehlern oder Unterbrechungen der Kommunikation zwischen Zabbix- Komponenten führen, die PSKs mit dieser PSK-Identitätszeichenfolge verwenden.

PSK generieren

Zum Beispiel kann ein 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 erzeugt. Zabbix erwartet in der PSK-Datei jedoch nur eine PSK, daher sollten die Identitätszeichenfolge und der Doppelpunkt (':') aus der Datei entfernt werden.

PSK für die Server-Agent-Kommunikation konfigurieren (Beispiel)

Schreiben Sie auf dem Agent-Host den PSK-Wert in eine Datei, zum Beispiel `/home/zabbix/zabbix_agentd.psk`. Die Datei muss den PSK in der ersten Textzeile enthalten, zum Beispiel:

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

Setzen Sie die Zugriffsrechte für die PSK-Datei so, dass sie nur vom Benutzer Zabbix lesbar ist.

Bearbeiten Sie die TLS-Parameter in der Agent-Konfigurationsdatei `zabbix_agentd.conf`, zum Beispiel setzen Sie:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

Der Agent wird sich mit dem Server verbinden (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:

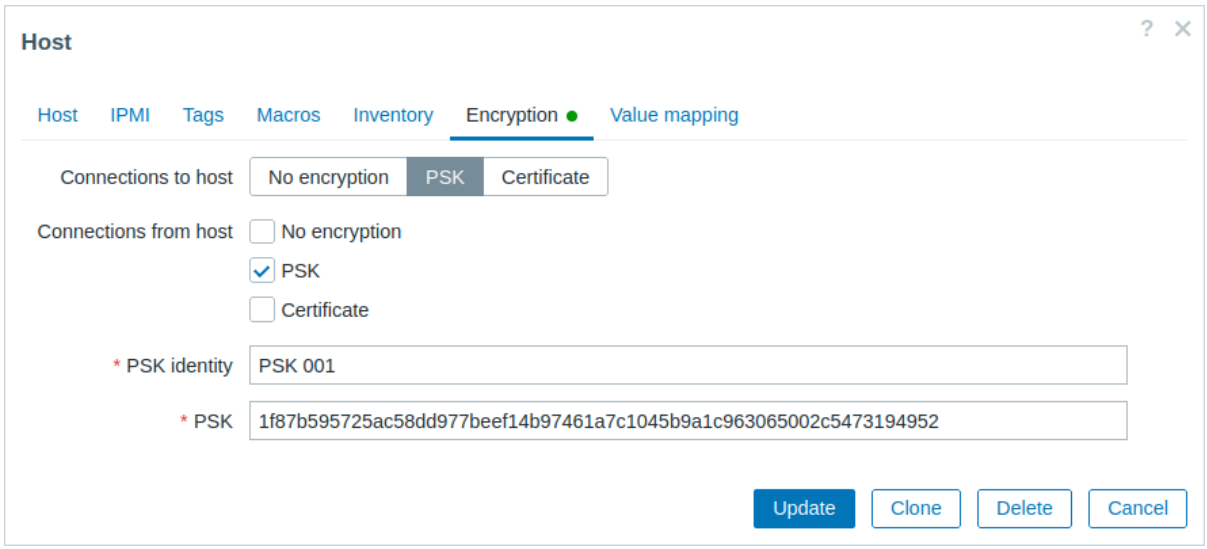
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Um Ausfallzeiten zu minimieren, lesen Sie nach, wie der Verbindungstyp in Connection encryption management geändert wird.)

Konfigurieren Sie die PSK-Verschlüsselung für diesen Agent im Zabbix Frontend:

- Gehen Sie zu: *Datenerfassung* → *Hosts*
- Wählen Sie den Host aus und klicken Sie auf den Reiter **Encryption**

Beispiel:



Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Wenn der Konfigurations-Cache mit der Datenbank synchronisiert ist, verwenden die neuen Verbindungen PSK. Prüfen Sie die Logdateien von Server und Agent auf Fehlermeldungen.

Konfiguration von PSK für die Kommunikation zwischen Server und aktivem Proxy (Beispiel)

Schreiben Sie auf dem Proxy den PSK-Wert in eine Datei, zum Beispiel `/home/zabbix/zabbix_proxy.psk`. Die Datei muss den PSK in der ersten Textzeile enthalten, zum Beispiel:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Setzen Sie die Zugriffsrechte für die PSK-Datei so, dass sie nur vom Zabbix-Benutzer lesbar ist.

Bearbeiten Sie die TLS-Parameter in der Proxy-Konfigurationsdatei `zabbix_proxy.conf`, und setzen Sie zum Beispiel:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Der Proxy wird sich mit dem Server unter Verwendung von PSK verbinden. Die PSK-Identität wird "PSK 002" sein.

(Um Ausfallzeiten zu minimieren, lesen Sie, wie der Verbindungstyp in der Verwaltung der Verbindungsverschlüsselung geändert wird.)

Konfigurieren Sie PSK für diesen Proxy im Zabbix Frontend. Gehen Sie zu *Administration→Proxies*, wählen Sie den Proxy aus und öffnen Sie den Reiter "Encryption". Markieren Sie unter "Connections from proxy" PSK. Fügen Sie in das Feld "PSK identity" "PSK 002" und "in das Feld "PSK" "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9" ein. Klicken Sie auf "Update".

Starten Sie den Proxy neu. Er wird dann PSK-basierte verschlüsselte Verbindungen zum Server verwenden. Prüfen Sie die Logdateien von Server und Proxy auf Fehlermeldungen.

Für einen passiven Proxy ist das Verfahren sehr ähnlich. Der einzige Unterschied: Setzen Sie `TLSAccept=psk` in der Proxy-Konfigurationsdatei und setzen Sie im Zabbix Frontend "Connections to proxy" auf `PSK`.

3 Fehlerbehebung

Allgemeine Empfehlungen

- Beginnen Sie damit zu verstehen, welche Komponente im Problemfall als TLS-Client und welche als TLS-Server fungiert.
  Zabbix Server, Proxys und Agenten können je nach Interaktion zwischen ihnen alle sowohl als TLS-Server als auch als TLS-Clients arbeiten.
  Zum Beispiel fungiert der Zabbix Server, der sich für eine passive Prüfung mit dem Agent verbindet, als TLS-Client. Der Agent übernimmt die Rolle des TLS-Servers.
  Ein Zabbix Agent, der von einem Proxy eine Liste aktiver Prüfungen anfordert, fungiert als TLS-Client. Der Proxy übernimmt die Rolle des TLS-Servers.
  Die Dienstprogramme `zabbix_get` und `zabbix_sender` fungieren immer als TLS-Clients.
- Zabbix verwendet gegenseitige Authentifizierung.
  Jede Seite überprüft ihr Gegenüber und kann die Verbindung verweigern.
  Zum Beispiel kann der Zabbix Server, der sich mit dem Agent verbindet, die Verbindung sofort schließen, wenn das Zertifikat des Agent ungültig ist. Und umgekehrt kann ein Zabbix Agent, der eine Verbindung vom Server akzeptiert, die Verbindung schließen, wenn der Server vom Agent nicht als vertrauenswürdig eingestuft wird.
- Untersuchen Sie die Protokolldateien auf beiden Seiten – beim TLS-Client und beim TLS-Server.
  Die Seite, die die Verbindung verweigert, protokolliert möglicherweise einen genauen Grund für die Verweigerung. Die andere Seite meldet oft nur einen eher allgemeinen Fehler (z. B. „Verbindung durch Gegenstelle geschlossen", „Verbindung wurde nicht ordnungsgemäß beendet").
- Manchmal führt eine falsch konfigurierte Verschlüsselung zu irreführenden Fehlermeldungen, die in keiner Weise auf die tatsächliche 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 könnte.
  Bitte beachten Sie, dass verschiedene Kryptografie-Toolkits (OpenSSL, GnuTLS) in denselben Problemsituationen oft unterschiedliche Fehlermeldungen erzeugen.
  Manchmal hängen Fehlermeldungen sogar von der jeweiligen Kombination der Kryptografie-Toolkits auf beiden Seiten ab.

1 Probleme mit Verbindungstyp oder Berechtigungen

Server ist so konfiguriert, dass er sich mit PSK mit dem Agent verbindet, aber der Agent akzeptiert nur unverschlüsselte Verbindungen

Im Server- oder Proxy-Protokoll (mit *GnuTLS* 3.3.16)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

Im Server- oder Proxy-Protokoll (mit *OpenSSL* 1.0.2c)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

Eine Seite verbindet sich mit Zertifikat, aber die andere Seite akzeptiert nur PSK oder umgekehrt

In einem beliebigen Log (mit *GnuTLS*):

```
Fehler beim Akzeptieren einer eingehenden Verbindung: von 127.0.0.1: zbx_tls_accept(): gnutls_handshake()
    -21 Es konnte keine unterstützte Cipher-Suite ausgehandelt werden.
```

In einem beliebigen Log (mit *OpenSSL* 1.0.2c):

```
Fehler beim Akzeptieren einer eingehenden Verbindung: 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 schrieb fatalen Alert "handshake failure"
```

Versuch, Zabbix sender mit TLS-Unterstützung zu verwenden, um Daten an einen Zabbix Server/Proxy zu senden, der ohne TLS kompiliert wurde

Im Log auf der Verbindungsseite:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL-Bibliothek (Version OpenSSL 1.1.1  11 Sep 2018) initialisiert
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FEHLER error:'Verbindung vom Peer geschlossen'
...send value error: TCP erfolgreich, TLS kann zu [[localhost]:10051] nicht aufgebaut werden: Verbindung v
```

Windows:

```
...OpenSSL-Bibliothek (Version OpenSSL 1.1.1a  20 Nov 2018) initialisiert
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() hat die PSK-Identität "PSK test sender" angefordert
...End of zbx_tls_connect():FEHLER error:'SSL_connect() E/A-Fehler: [0x00000000] Der Vorgang wurde erfolgr
...send value error: TCP erfolgreich, TLS kann zu [[192.168.1.2]:10051] nicht aufgebaut werden: SSL_connec
```

Im Log der akzeptierenden Seite:

```
...eingehende Verbindung konnte nicht akzeptiert werden: von 127.0.0.1: Unterstützung für TLS wurde nicht
```

Eine Seite verbindet sich mit PSK, aber die andere Seite verwendet LibreSSL oder wurde ohne Unterstützung für Verschlüsselung kompiliert

LibreSSL unterstützt kein PSK.

Im Log der verbindenden Seite:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

Im Log der akzeptierenden Seite:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Im Zabbix Frontend:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

Eine Seite verbindet sich mit PSK, aber auf der anderen Seite wird OpenSSL mit deaktivierter PSK-Unterstützung verwendet

Im Log der verbindenden Seite:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERR
```

Im Log der annehmenden Seite:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 Zertifikatsprobleme

OpenSSL wird mit CRLs verwendet, und für einige CAs in der Zertifikatskette ist ihre CRL nicht in `TLSCRLFile` enthalten

Im TLS-Server-Log 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-Server-Log 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 Server-Betriebs ab

OpenSSL, im Server-Log:

- vor dem Ablauf:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- nach dem Ablauf:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

Der Punkt hierbei ist, dass bei einer gültigen 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:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

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, wenn das Server-Zertifikat versehentlich dieselbe Issuer- und Subject-Zeichenfolge hatte, obwohl es von einer CA signiert war. Issuer und Subject sind im CA-Zertifikat der obersten Ebene gleich, dürfen jedoch im Server-Zertifikat 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
```

Es ist zulässig, dass das Root-Zertifikat (oberste Ebene) identische Werte für Issuer und Subject hat.

3 PSK-Probleme

PSK enthält eine ungerade Anzahl von Hex-Ziffern

Proxy oder Agent startet nicht, Meldung im Proxy- oder Agent-Log:

```
ungültiger PSK in Datei "/home/zabbix/zabbix_proxy.psk"
```

PSK-Identitätszeichenfolge länger als 128 Byte wird an GnuTLS übergeben

Im TLS-Client-seitigen Log:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

Im TLS-Server-seitigen Log:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Zu langer PSK-Wert bei Verwendung von OpenSSL 1.1.1

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: Grenzwerte für die Wertgröße

16 Weboberfläche

Übersicht Für einen einfachen Zugriff auf Zabbix von überall und von jeder Plattform aus steht eine webbasierte Oberfläche zur Verfügung.

> **Note:**
> Wenn mehr als eine Frontend-Instanz verwendet wird, stellen Sie sicher, dass die Locales und Bibliotheken (LDAP, SAML usw.) auf allen Frontends identisch installiert und konfiguriert sind.

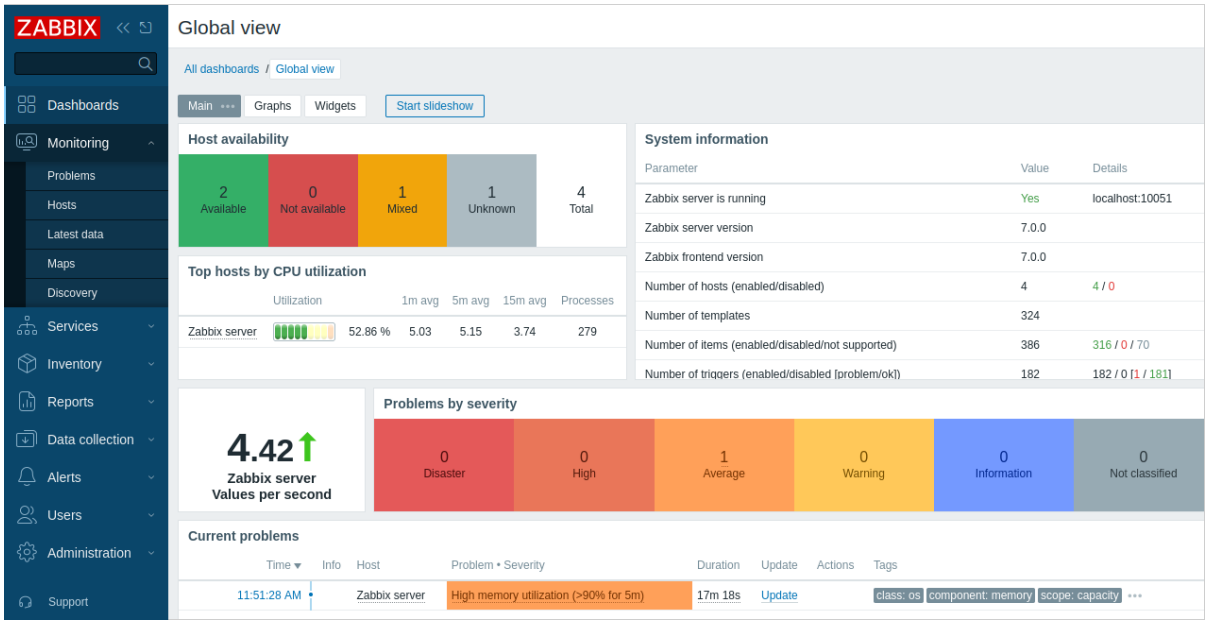
Frontend-Hilfe Ein Hilfe-Link ? wird in Formularen des Zabbix Frontends bereitgestellt und enthält direkte Links zu den entsprechenden Teilen der Dokumentation.

1 Menü

Übersicht

Ein vertikales Menü in einer Seitenleiste bietet Zugriff auf verschiedene Bereiche des Zabbix Frontend.

Im Standard-Theme ist das Menü dunkelblau.

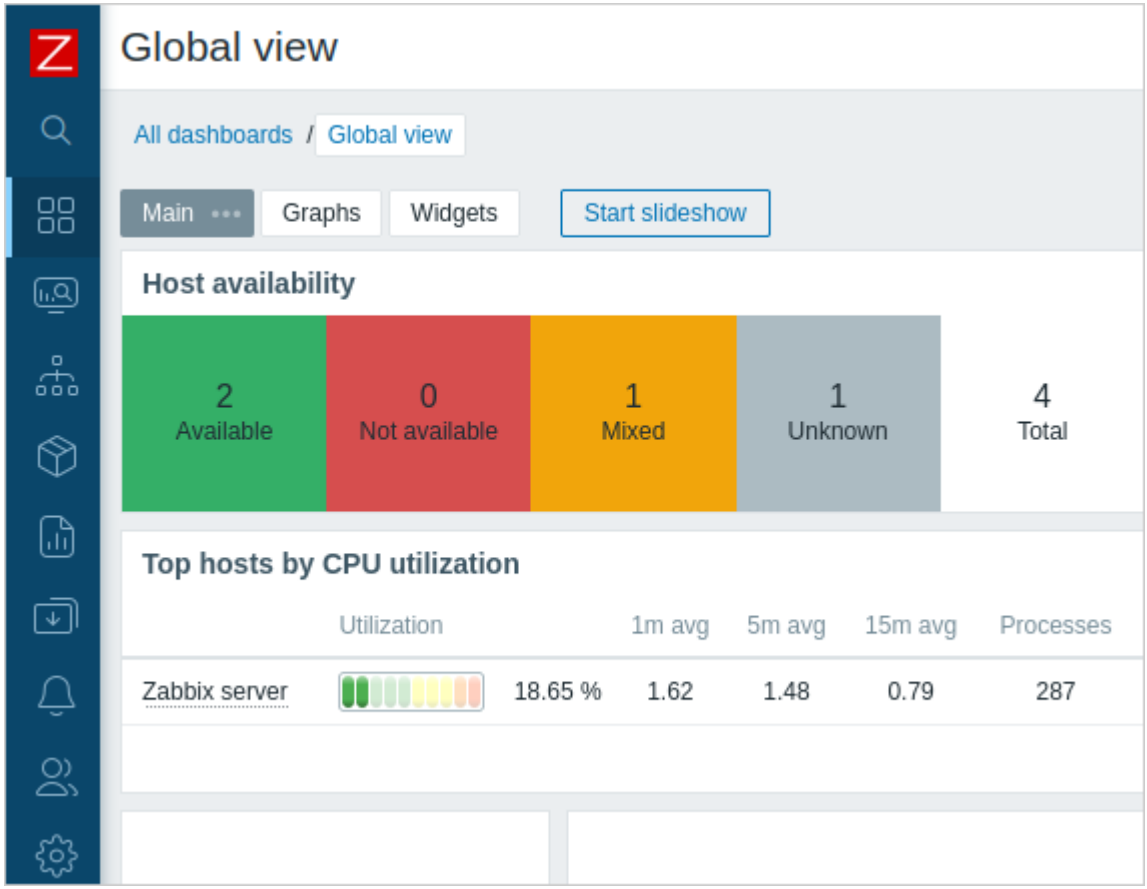


Arbeiten mit dem Menü

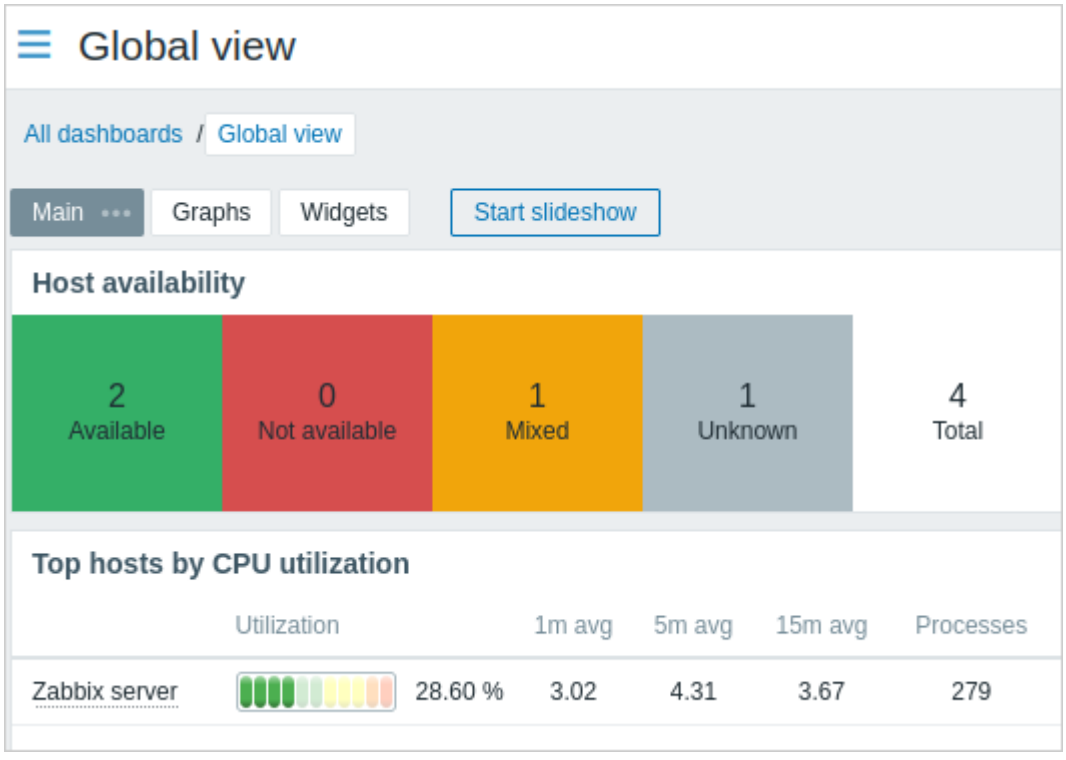
Ein Feld für die globale Suche befindet sich unterhalb des Zabbix-Logos.

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 nur auf Symbole reduziert eingeklappt ist, erscheint das vollständige Menü erneut, sobald der Mauszeiger darauf bewegt 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. Wenn der Mauszeiger anschließend wieder außerhalb des vollständigen Menüs platziert wird, klappt das Menü nach zwei Sekunden erneut ein.

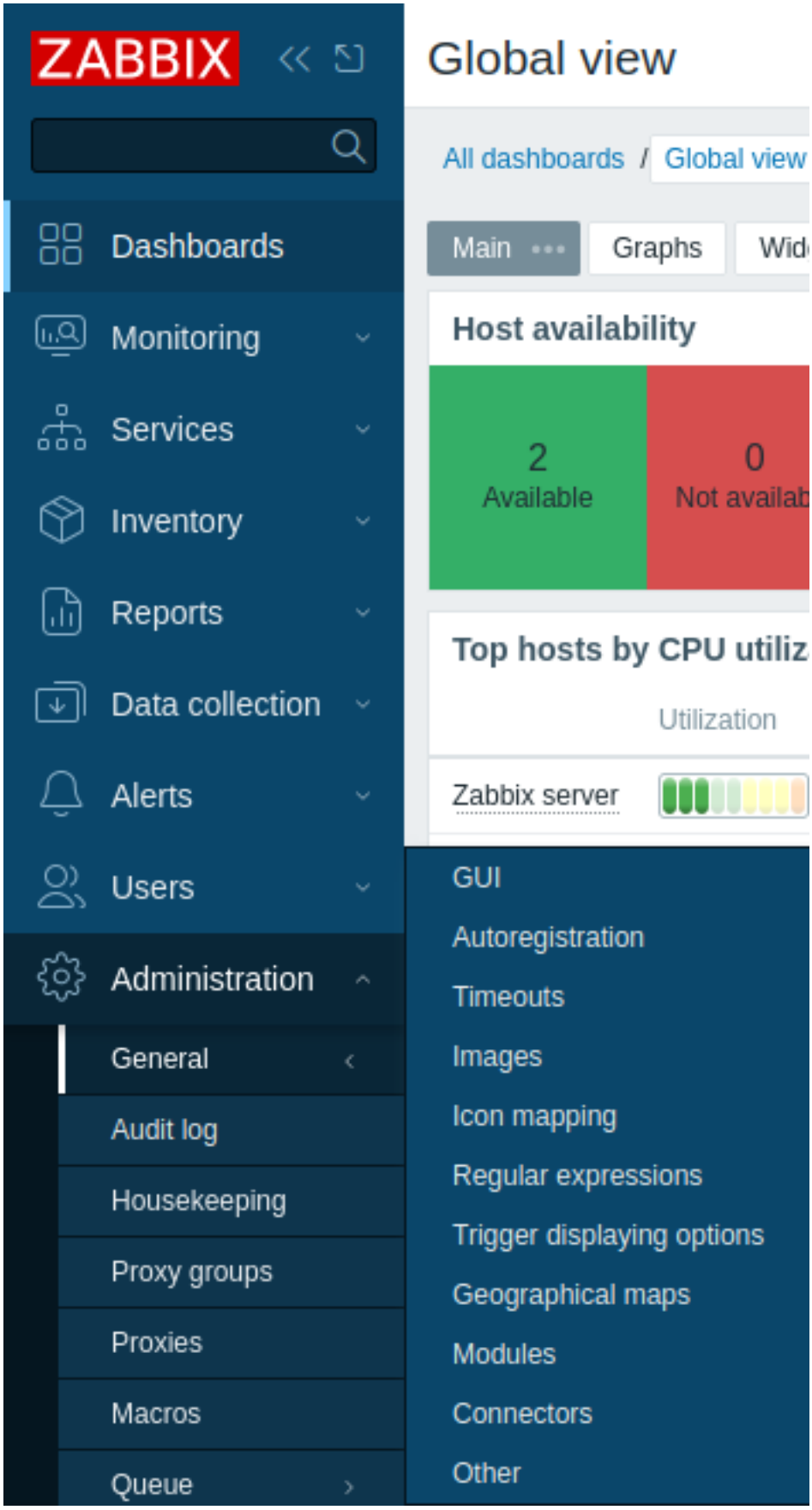
Sie können ein eingeklapptes Menü auch wieder vollständig anzeigen lassen, indem Sie die Tabulatortaste drücken. Durch wiederholtes Drücken der Tabulatortaste können Sie das nächste Menüelement fokussieren.

Ausgeblendetes Menü

Auch wenn das Menü vollständig ausgeblendet ist, ist das vollständige Menü nur einen Mausklick entfernt, indem Sie auf das Burger-Symbol  klicken. Beachten Sie, dass es über dem Seiteninhalt wieder erscheint; um den Seiteninhalt nach rechts zu verschieben, müssen Sie das Menü wieder einblenden, indem Sie auf die Schaltfläche zum Anzeigen der Seitenleiste klicken.

Menüebenen

Es gibt bis zu drei Ebenen im Menü.



Kontextmenüs

Zusätzlich zum Hauptmenü bietet Zabbix Kontextmenüs für Host, Datenpunkt und Ereignis für den schnellen Zugriff auf häufig verwendete Elemente wie die letzten Werte, ein einfaches Diagramm, das Konfigurationsformular, zugehörige Skripte oder externe Links.

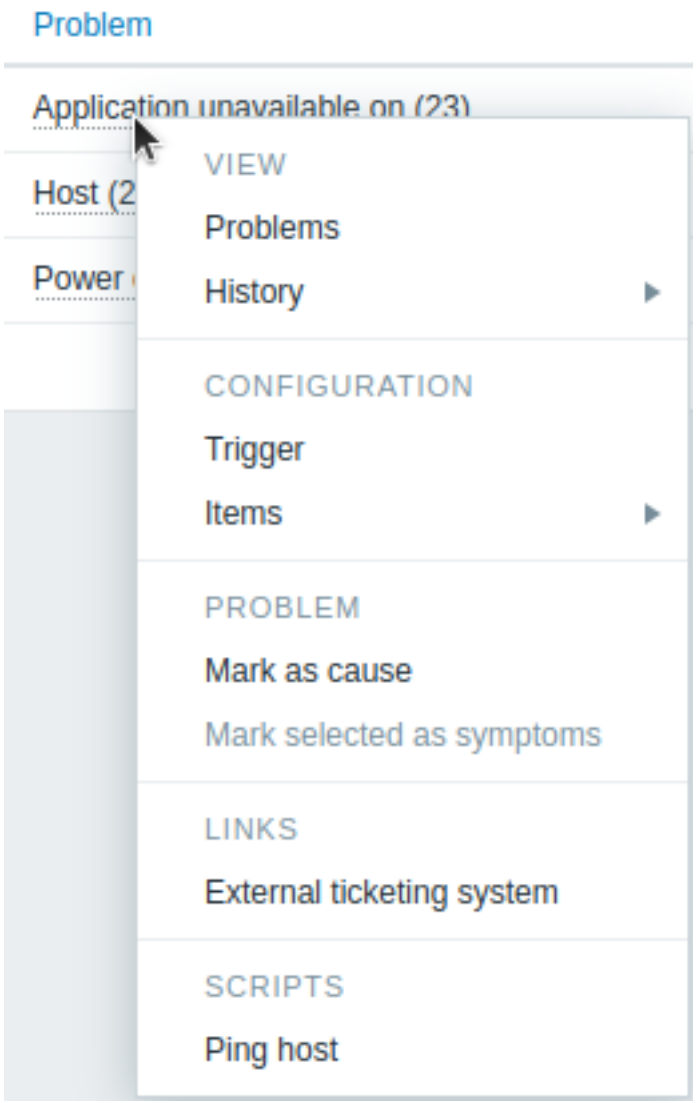
Die Kontextmenüs sind zugänglich, indem Sie an unterstützten Stellen auf den Namen des Hosts, Datenpunkts oder Problems/Auslösers klicken.

1 Menü „Ereignisse“

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 Kontextmenü des Ereignisses hat sechs Abschnitte: *Ansicht*, *Aktionen*, *Konfiguration*, *Problem*, *Links* und *Skripte*. Für die nicht konfigurierten Entitäten sind die Links deaktiviert und werden grau dargestellt. Die Abschnitte *Skripte* und *Links* werden angezeigt, wenn ihre Entitäten konfiguriert sind.

Der Abschnitt *Ansicht* enthält Links zu:

- **Problemen** - öffnet die Liste der ungelösten Probleme des zugrunde liegenden Auslösers;
- **Verlauf** - führt zum Graphen/zur Historie der *Letzten Daten* des zugrunde liegenden Datenpunkts bzw. der zugrunde liegenden Datenpunkte. Wenn ein Auslöser mehrere Datenpunkte verwendet, sind für jeden von ihnen Links verfügbar.

Der Abschnitt *Aktionen* ist nur in Widgets der *Auslöserübersicht* verfügbar. Er enthält einen Link zu:

- **Problem aktualisieren** - öffnet den Bildschirm zum Aktualisieren von Problemen.

Der Abschnitt *Konfiguration* enthält Links zur Konfiguration von:

- **Auslöser**, der das Problem ausgelöst hat;
- **Datenpunkten**, die im Auslöserausdruck verwendet werden.

> **Note:**
> Beachten Sie, dass der Konfigurationsabschnitt nur für Benutzer mit den Rollen Admin und Super admin verfügbar ist.

Der Abschnitt *Problem* enthält die Optionen zum:

- **Als Ursache markieren** - das Problem als Ursache markieren;
- **Ausgewählte als Symptome markieren** - die ausgewählten Probleme als Symptome dieses Problems markieren.

Der Abschnitt *Links* enthält Links zum:

- Zugriff auf eine konfigurierte Auslöser-URL;
- Zugriff auf benutzerdefinierte Links, die in Global scripts konfiguriert sind (mit dem Geltungsbereich 'Manual event action' und dem Typ 'URL');
- Zugriff auf ein konfiguriertes externes Ticket für das Problem (siehe die Option *Include event menu entry* bei der Konfiguration von webhooks).

Der Abschnitt *Skripte* enthält Links zum Ausführen eines globalen Skripts (mit dem Geltungsbereich *Manual event action*). Diese Funktion kann nützlich sein, um Skripte auszuführen, die zur Verwaltung von Problem-Tickets in externen Systemen verwendet werden.

Unterstützte Orte

Das Ereignis-Kontextmenü ist durch Klicken auf einen Problem- oder Ereignisnamen in verschiedenen Frontend-Bereichen zugänglich, zum Beispiel:

- Dashboards Widgets, wie z. B. das Widget *Probleme*, das Widget *Auslöserübersicht* usw.
- Überwachung → Probleme
- Überwachung → Probleme → Ereignisdetails
- Berichte → Top 100 Auslöser (globale Skripte und der Zugriff auf externe Tickets werden an diesem Ort nicht unterstützt)

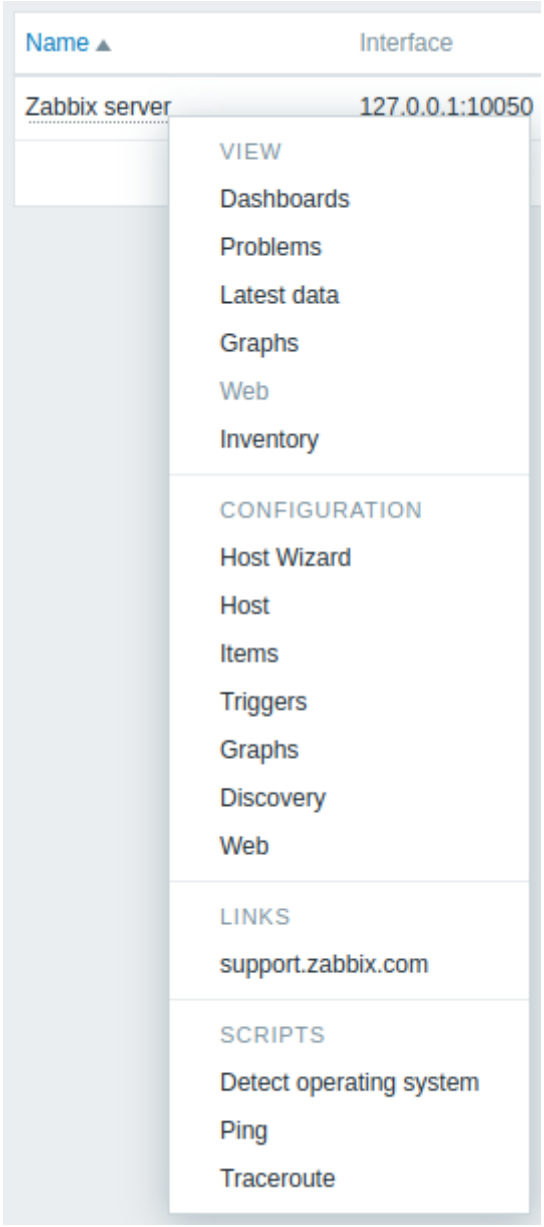
2 Host-Menü

Übersicht

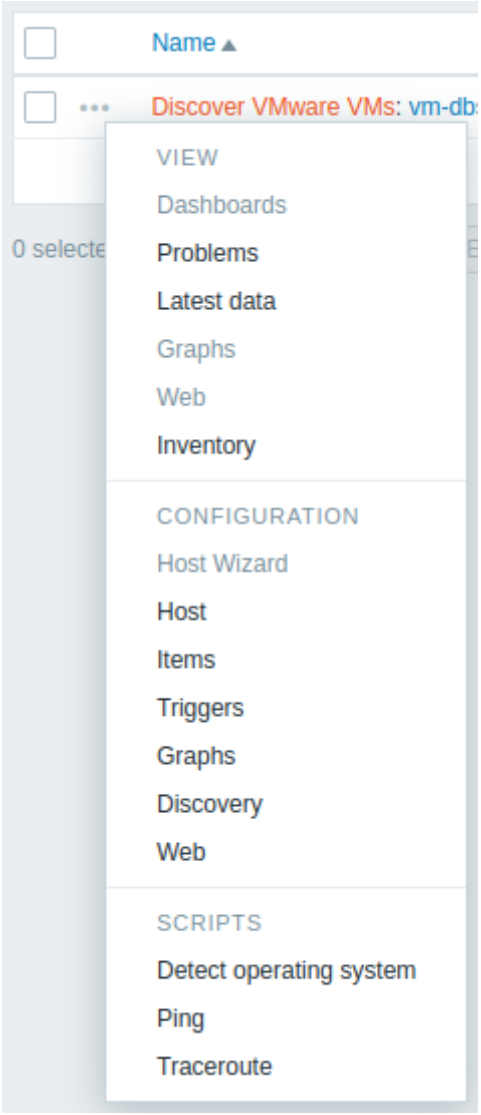
Das Host-Menü enthält Verknüpfungen zu Aktionen oder Frontend-Bereichen, die für einen Host häufig benötigt werden.

Das Host-Menü kann je nach Frontend-Bereich durch Klicken auf den Host-Namen oder das Symbol mit den drei Punkten geöffnet werden, zum Beispiel:

Monitoring → *Neueste Daten*



Datenerfassung → *Hosts*



Inhalt

Das Kontextmenü des Hosts hat vier Abschnitte: *Ansicht*, *Konfiguration*, *Links* und *Skripte*. Für die Entitäten, die nicht konfiguriert sind, sind die Links deaktiviert und werden grau dargestellt. Die Abschnitte *Skripte* und *Links* werden angezeigt, wenn ihre Entitäten konfiguriert sind.

Der Abschnitt *Ansicht* enthält Links zu:

- **Dashboards** - öffnet Widgets und Graphen.
- **Probleme** - öffnet den Abschnitt *Probleme* mit der Liste der nicht behobenen Probleme des zugrunde liegenden Auslösers.
- **Letzte Daten** - öffnet den Abschnitt *Letzte Daten* mit der Liste aller neuesten Daten des aktuellen Hosts.
- **Graphen** - öffnet einfache Graphen des aktuellen Hosts.
- **Web** - öffnet den Link zu den konfigurierten Webszenarien.
- **Inventar** - öffnet den Link zum Inventar des aktuellen Hosts.

Der Abschnitt *Konfiguration* enthält Links zu:

- **Host-Assistent** - öffnet den Host Wizard für den aktuellen Host (für entdeckte Hosts deaktiviert).
- **Host** - Konfigurationsformular des aktuellen Hosts.
- **Datenpunkte** - die Liste der Datenpunkte des aktuellen Hosts.
- **Auslöser** - die Liste der Auslöser des aktuellen Hosts.
- **Graphen** - einfache Graphen des aktuellen Hosts.
- **Discovery** - die Liste der Low-Level-Discovery-Regeln des aktuellen Hosts.

- **Web** - die Liste der Webszenarien des aktuellen Hosts.

> **Note:**
> Beachten Sie, dass der Abschnitt „Konfiguration“ nur für Benutzer mit den Rollen Admin und Super admin verfügbar ist.

Der Abschnitt *Links* enthält Links zu:

- Zugriff auf eine konfigurierte Auslöser-URL.
- Zugriff auf benutzerdefinierte Links, die in Global scripts konfiguriert sind (mit dem Geltungsbereich *Manual host action* und dem Typ 'URL').

Der Abschnitt *Skripte* ermöglicht die Ausführung von global scripts, die für den aktuellen Host konfiguriert sind. Damit diese Skripte im Host-Menü verfügbar sind, muss ihr Geltungsbereich als *Manual host action* definiert sein.

Unterstützte Orte

Das Host-Menü ist durch Klicken auf einen Host-Namen in verschiedenen Frontend-Bereichen zugänglich, zum Beispiel:

- Dashboards-Widgets wie Probleme, Top-Datenpunkte, Auslöser-Übersicht usw.
- Überwachung → Probleme
- Überwachung → Probleme → Ereignisdetails
- Überwachung → Hosts
- Überwachung → Hosts → Web-Überwachung
- Überwachung → Neueste Daten
- Überwachung → Karten
- Inventar → Hosts
- Berichte → Top 100 Auslöser

Das Host-Menü ist durch Klicken auf das Symbol mit den drei Punkten unter Datensammlung → Hosts zugänglich.

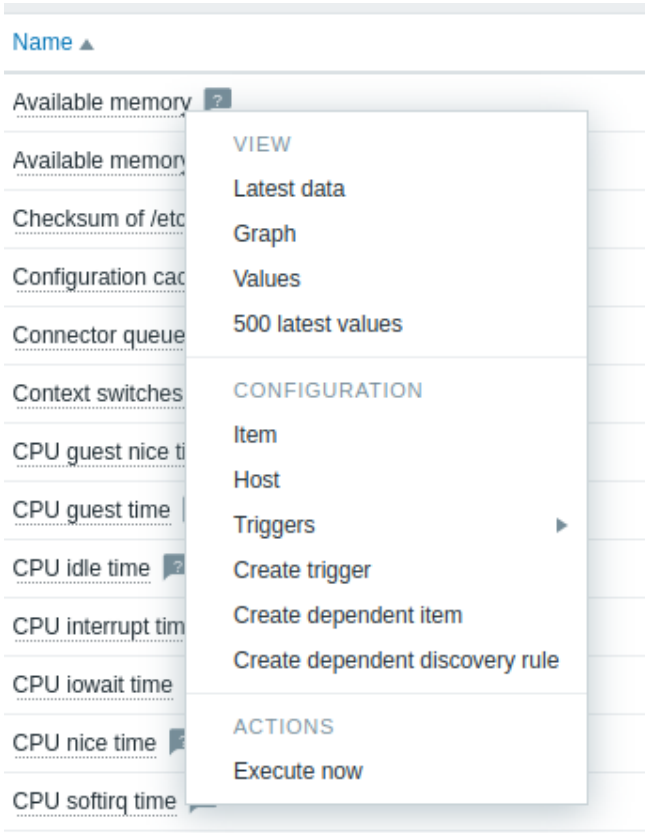
3 Menü „Datenpunkt“

Übersicht

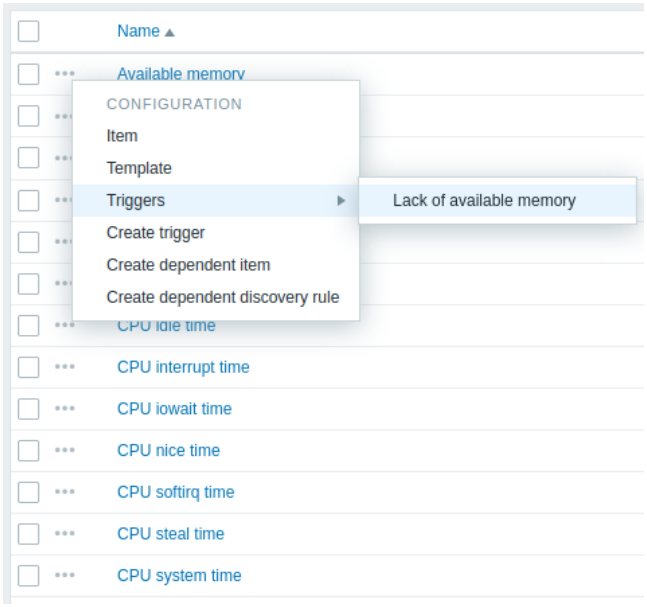
Das Datenpunkt-Menü enthält Verknüpfungen zu Aktionen oder Frontend-Bereichen, die für einen Datenpunkt häufig benötigt werden.

Das Datenpunkt-Menü kann je nach Frontend-Bereich durch Klicken auf den Namen des Datenpunkts oder auf das Symbol mit den drei Punkten geöffnet werden, zum Beispiel:

Monitoring → *Letzte Daten*



Datensammlung → *Vorlagen* → *Datenpunkte*



Inhalt

Das Menü des Datenpunkts hat drei Abschnitte: *Ansicht*, *Konfiguration* und *Aktionen*.

Der Abschnitt *Ansicht* enthält die folgenden Optionen:

- **Letzte Daten** - öffnet den Abschnitt *Letzte Daten*, gefiltert nach dem aktuellen Host und Datenpunkt;
- **Graph** - öffnet einen einfachen Graphen des aktuellen Datenpunkts;
- **Werte** - öffnet die Liste aller Werte, die für den aktuellen Datenpunkt in den letzten 60 Minuten empfangen wurden;
- **500 letzte Werte** - öffnet die Liste der 500 letzten Werte für den aktuellen Datenpunkt.

Der Abschnitt *Konfiguration* (nur verfügbar für Benutzer des Typs *Admin* und *Super admin*) enthält die folgenden Optionen:

- **Datenpunkt** - öffnet das Konfigurationsformular des Datenpunkts des aktuellen Datenpunkts;
- **Vorlage** - öffnet das Konfigurationsformular der Vorlage der Vorlage, zu der der Datenpunkt gehört (nur verfügbar, wenn auf das Menü des Datenpunkts über *Datensammlung* → *Vorlagen* → *Datenpunkte* zugegriffen wird);
- **Host** - öffnet das Konfigurationsformular des Hosts des Hosts, zu dem der Datenpunkt gehört;
- **Auslöser** - öffnet beim Überfahren mit der Maus eine Liste der Auslöser des Datenpunkts, falls 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 Konfigurationsformular des Datenpunkts, um einen abhängigen Datenpunkt zu erstellen, wobei der aktuelle Datenpunkt als Master-Datenpunkt verwendet wird;
- **Abhängige Discovery-Regel erstellen** - öffnet das Konfigurationsformular der Discovery-Regel, 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 aus, um einen neuen Wert für den Datenpunkt abzurufen.

Unterstützte Positionen

Das Datenpunkt-Menü ist durch Klicken auf einen Datenpunkt-Namen in verschiedenen Frontend-Bereichen zugänglich, zum Beispiel:

- Überwachung → Letzte Daten
- Datensammlung → Hosts → Datenpunkte
- Datensammlung → Hosts → Discovery-Regeln → Datenpunkt-Prototypen

Das Datenpunkt-Menü ist durch Klicken auf einen Datenpunkt-Wert im Dashboard-Widget Top items zugänglich.

2 Frontend-Bereiche

Menüstruktur Das Frontend-Menü von Zabbix hat die folgende Struktur:

- Dashboards
- Überwachung
  - Probleme
  - Hosts
  - Letzte Daten
  - Karten
  - Discovery
- Services
  - Services
  - SLA
  - SLA-Bericht
- Inventar
  - Übersicht
  - Hosts
- Berichte
  - Systeminformationen
  - Geplante Berichte
  - Verfügbarkeitsbericht
  - Top 100 Auslöser
  - Audit-Log
  - Aktionsprotokoll
  - Benachrichtigungen
- Datenerfassung
  - Vorlagengruppen
  - Host-Gruppen
  - Vorlagen
  - Hosts
  - Wartung
  - Ereigniskorrelation
  - Discovery
- Warnungen
  - Aktionen
    - Auslöser-Aktionen
    - Service-Aktionen
    - Discovery-Aktionen
    - Autoregistrierungs-Aktionen
    - Interne Aktionen
  - Medientypen
  - Skripte
- Benutzer
  - Benutzergruppen
  - Benutzerrollen
  - Benutzer
  - API-Tokens
  - Authentifizierung
- Administration
  - Allgemein
    - GUI
    - Autoregistrierung
    - Timeouts
    - Bilder
    - Symbolzuordnung
    - Reguläre Ausdrücke
    - Optionen zur Auslöseranzeige
    - Geografische Karten

    - Module
    - Konnektoren
    - Sonstiges
- Audit-Log
- Bereinigung
- Proxy-Gruppen
- Proxys
- Makros
- Warteschlange
    - Warteschlangenübersicht
    - Warteschlangenübersicht nach Proxy
    - Warteschlangendetails

1 Dashboards

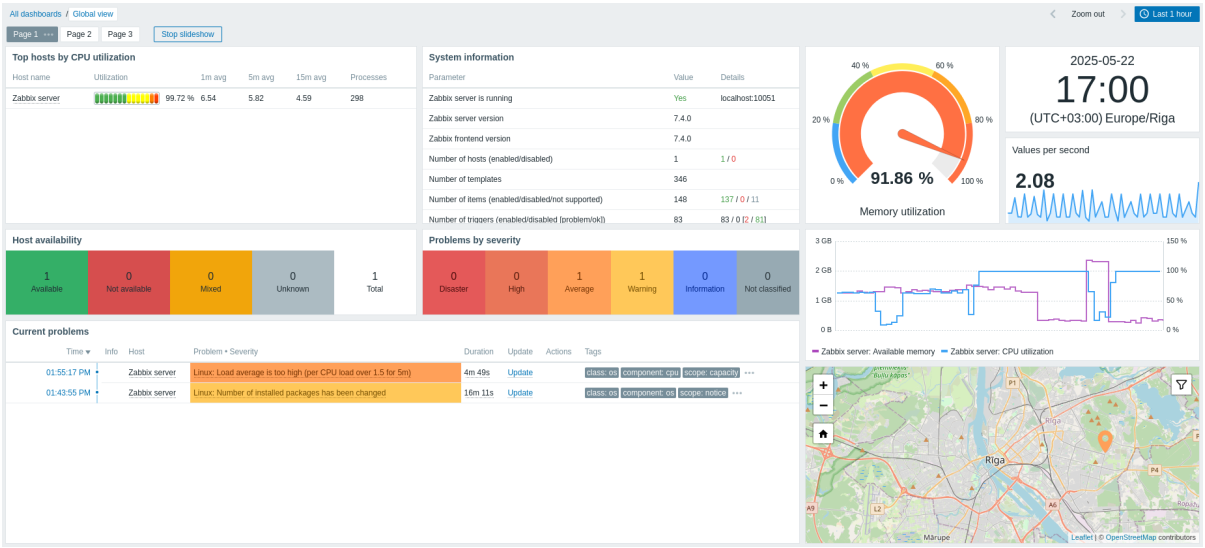
Übersicht

Der Abschnitt *Dashboards* dient dazu, Zusammenfassungen aller wichtigen Informationen in einem **Dashboard** anzuzeigen.

Während jeweils nur ein Dashboard gleichzeitig angezeigt werden kann, ist es möglich, mehrere Dashboards zu konfigurieren. Jedes Dashboard kann eine oder mehrere Seiten enthalten, die in einer Diashow rotiert werden können.

Eine Dashboard-Seite besteht aus Widgets, und jedes Widget ist dafür ausgelegt, Informationen einer bestimmten Art und Quelle anzuzeigen, zum Beispiel eine Zusammenfassung, eine Karte, ein Diagramm, die Uhr usw.

Der Zugriff auf Hosts in den Widgets hängt von den Berechtigungen für Hosts ab.



Seiten und Widgets werden dem Dashboard hinzugefügt und im Dashboard-Bearbeitungsmodus bearbeitet. Seiten können im Ansichtsmodus angezeigt und rotiert werden.

Der in Diagramm-Widgets angezeigte Zeitraum wird über den Zeitraumauswahl gesteuert, die sich oberhalb der Widgets befindet. Ihre Beschriftung zeigt den aktuell ausgewählten Zeitraum an. Durch Klicken auf die Beschriftung kann die Zeitraumauswahl ein- und ausgeklappt werden.

Dashboard-Größe

Die Mindestbreite eines Dashboards beträgt 1200 Pixel. Das Dashboard wird nicht unter diese Breite verkleinert; stattdessen wird eine horizontale Bildlaufleiste angezeigt, wenn das Browserfenster kleiner ist.

Die maximale Breite eines Dashboards entspricht der Breite des Browserfensters. Dashboard-Widgets werden horizontal gestreckt, damit sie in das Fenster passen. Gleichzeitig kann ein Dashboard-Widget horizontal nicht über die Grenzen des Fensters hinaus gestreckt werden.

Horizontal besteht das Dashboard aus 72 Spalten mit stets gleicher Breite, die dynamisch gestreckt/verkleinert werden (jedoch nicht auf insgesamt weniger als 1200 Pixel).

Vertikal kann das Dashboard maximal 64 Zeilen enthalten; jede Zeile hat eine feste Höhe von 70 Pixeln.

Ein Widget kann daher bis zu 72 Spalten breit und 64 Zeilen hoch sein.

Dashboards anzeigen

Um alle konfigurierten Dashboards anzuzeigen, klicken Sie direkt unter dem Abschnittstitel auf *Alle Dashboards*.

Dashboards werden mit einem Freigabe-Tag angezeigt:

- *Meine* - kennzeichnet ein Dashboard, das dem aktuellen Benutzer gehört
- *Geteilt* - kennzeichnet ein öffentliches Dashboard oder ein privates Dashboard, das für einen beliebigen Benutzer oder eine Benutzergruppe freigegeben wurde

Beachten Sie, dass die Freigabe-Tags nur für die Dashboards angezeigt werden, die dem aktuellen Benutzer gehören. Die Dashboards anderer Benutzer werden ohne Freigabe-Tags aufgelistet.

Mit dem Filter rechts oberhalb der Liste können Dashboards nach Namen sowie nach den vom aktuellen Benutzer erstellten Dashboards gefiltert werden.

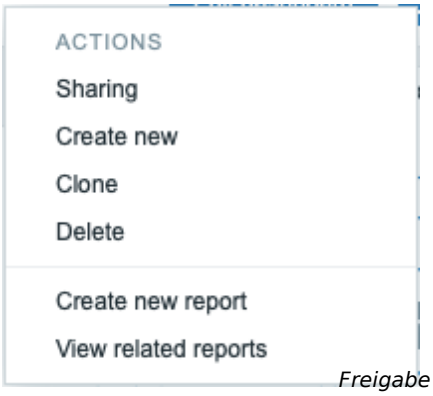
Um ein oder mehrere Dashboards zu löschen, markieren Sie die Kontrollkästchen der entsprechenden Dashboards und klicken Sie unterhalb der Liste auf *Löschen*.

Anzeigen eines Dashboards

Um ein einzelnes Dashboard anzuzeigen, klicken Sie in der Dashboard-Liste auf seinen Namen.

Beim **Anzeigen** eines Dashboards sind die folgenden Optionen verfügbar:

Edit dashboard	In den Dashboard-**Bearbeitungs**modus wechseln. Der Bearbeitungsmodus wird auch geöffnet, wenn ein neues Dashboard erstellt wird und wenn Sie auf die Schaltfläche zum Bearbeiten eines Widgets klicken.
	Das Aktionsmenü öffnen (siehe Aktionsbeschreibungen unten).

	*Freigabe* - die Freigabeeinstellungen für das Dashboard bearbeiten. *Neu erstellen* - ein neues Dashboard erstellen. *Klonen* - ein neues Dashboard erstellen, indem die Eigenschaften des vorhandenen Dashboards kopiert werden. Zuerst werden Sie aufgefordert, Dashboard-Parameter einzugeben. Anschließend wird das neue Dashboard im Bearbeitungsmodus mit allen Widgets des ursprünglichen Dashboards geöffnet. *Löschen* - das Dashboard löschen. *Neuen Bericht erstellen* - ein Pop-up-Fenster mit dem Konfigurationsformular für Berichte öffnen. Deaktiviert, wenn der Benutzer keine Berechtigung zum Verwalten geplanter Berichte hat. *Zugehörige Berichte anzeigen* - ein Pop-up-Fenster mit einer Liste vorhandener Berichte öffnen, die auf dem aktuellen Dashboard basieren. Deaktiviert, wenn keine zugehörigen Berichte vorhanden sind oder der Benutzer keine Berechtigung zum Anzeigen geplanter Berichte hat.
	Nur den Seiteninhalt anzeigen (Kiosk-Modus). Der Kiosk-Modus kann auch mit den folgenden URL-Parametern aufgerufen werden: `/zabbix.php?action=dashboard.view&kiosk=1`. Zur Rückkehr in den normalen Modus: `/zabbix.php?action=dashboard.view&kiosk=0`.

Freigabe

Dashboards können öffentlich oder privat gemacht werden.

Öffentliche Dashboards sind für alle Benutzer sichtbar. Private Dashboards sind nur für ihren Eigentümer sowie für die in den Freigabeeinstellungen hinzugefügten Benutzer/Benutzergruppen sichtbar.

Um den Freigabestatus eines Dashboards zu bearbeiten, klicken Sie beim Anzeigen eines einzelnen Dashboards im Aktionsmenü auf die Option *Freigabe*:

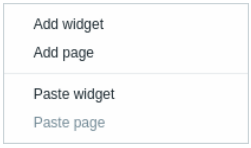
Parameter	Beschreibung
Type	Wählen Sie den Dashboard-Typ aus: **Private** - das Dashboard ist nur für ausgewählte Benutzergruppen und Benutzer sichtbar. **Public** - das Dashboard ist für alle sichtbar.
List of user group shares	Wählen Sie die Benutzergruppen aus, für die das Dashboard zugänglich ist. Sie können schreibgeschützten oder Lese-/Schreibzugriff erlauben.
List of user shares	Wählen Sie die Benutzer aus, für die das Dashboard zugänglich ist. Sie können schreibgeschützten oder Lese-/Schreibzugriff erlauben.

Siehe Berechtigungen für Dashboards, um zu erfahren, wie Freigabeoptionen die für Benutzer verfügbaren Aktionen einschränken.

Bearbeiten eines Dashboards

Beim **Bearbeiten** eines Dashboards sind die folgenden Optionen verfügbar:

	Allgemeine Dashboard-Parameter bearbeiten.
	Ein neues Widget hinzufügen. Ein Klick auf die Pfeilschaltfläche öffnet das Aktionsmenü (siehe Aktionsbeschreibungen unten).

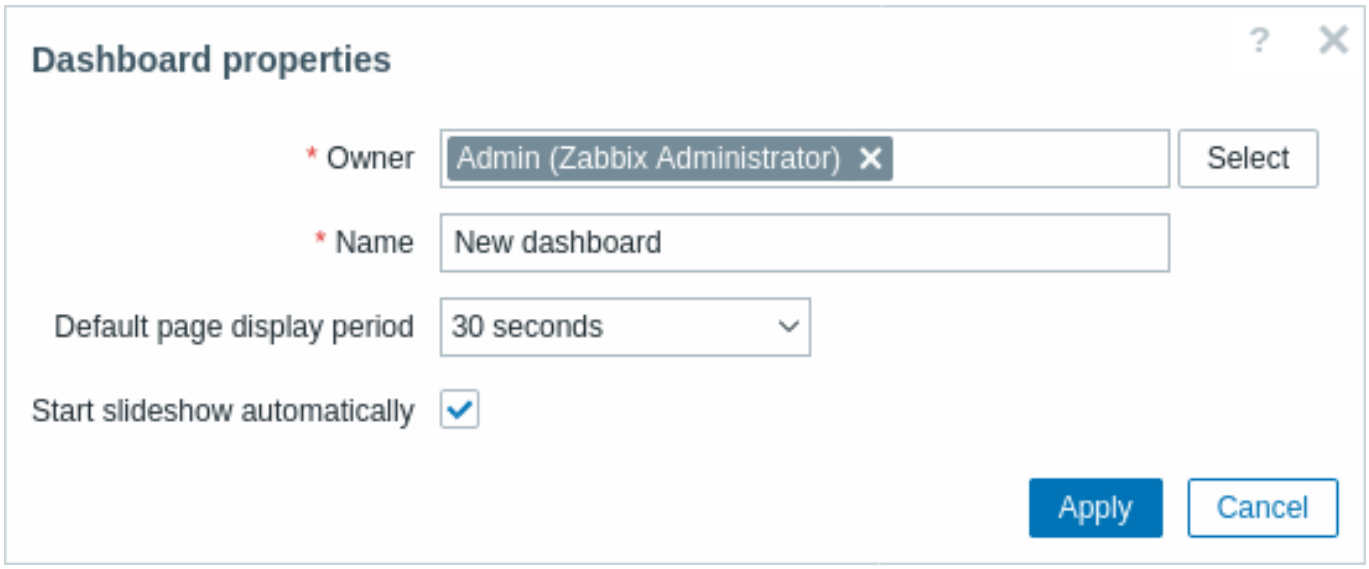
	*Widget hinzufügen* - ein neues Widget hinzufügen. *Seite hinzufügen* - eine neue Seite hinzufügen. *Widget einfügen* - ein kopiertes Widget einfügen. Diese Option ist ausgegraut, wenn kein Widget kopiert wurde. Es kann jeweils nur eine Entität (Widget oder Seite) gleichzeitig kopiert werden. *Seite einfügen* - eine kopierte Seite einfügen. Diese Option ist ausgegraut, wenn keine Seite kopiert wurde.
	Dashboard-Änderungen speichern.
	Dashboard-Änderungen verwerfen.

Erstellen eines Dashboards

Es ist möglich, ein neues Dashboard auf zwei Arten zu erstellen:

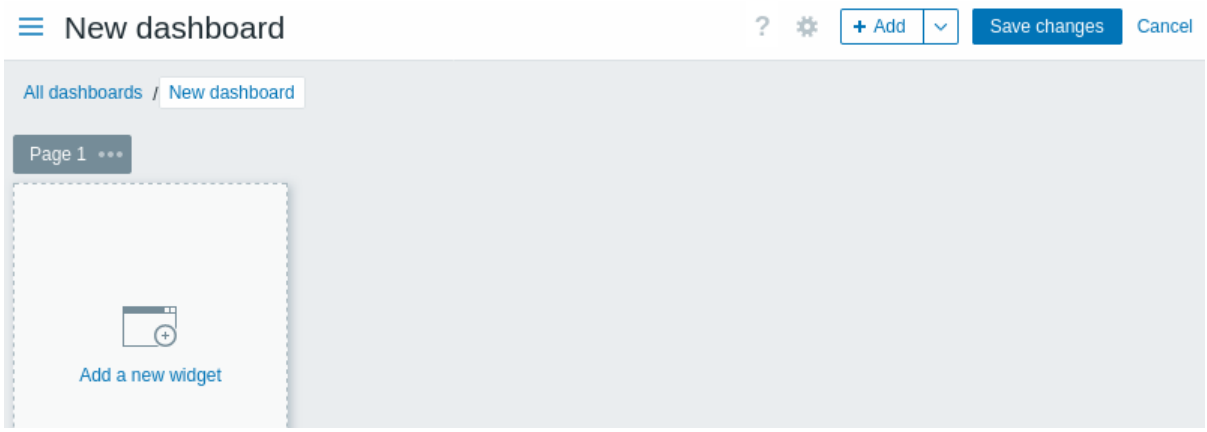
- Klicken Sie auf *Dashboard erstellen*, wenn Sie alle Dashboards anzeigen
- Wählen Sie *Neu erstellen* aus dem Aktionsmenü, wenn Sie ein einzelnes Dashboard anzeigen

Zunächst werden Sie aufgefordert, allgemeine Dashboard-Parameter einzugeben:



Parameter	Beschreibung
Eigentümer	Wählen Sie den Systembenutzer aus, der Eigentümer des Dashboards sein soll.
Name	Geben Sie den Namen des Dashboards ein.
Standard-Anzeigezeitraum der Seite	Wählen Sie den Zeitraum aus, wie lange eine Dashboard-Seite angezeigt wird, bevor in einer Diashow zur nächsten Seite gewechselt wird.
Diashow automatisch starten	Aktivieren Sie dieses Kontrollkästchen, um eine Diashow automatisch auszuführen, wenn mehr als eine Dashboard-Seite vorhanden ist.

Wenn Sie auf *Anwenden* klicken, wird ein leeres Dashboard geöffnet:



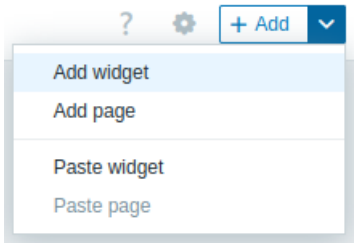
Um das Dashboard zu befüllen, können Sie Widgets und Seiten hinzufügen.

Klicken Sie auf die Schaltfläche *Änderungen speichern*, um das Dashboard zu speichern. Wenn Sie auf *Abbrechen* klicken, wird das Dashboard nicht erstellt.

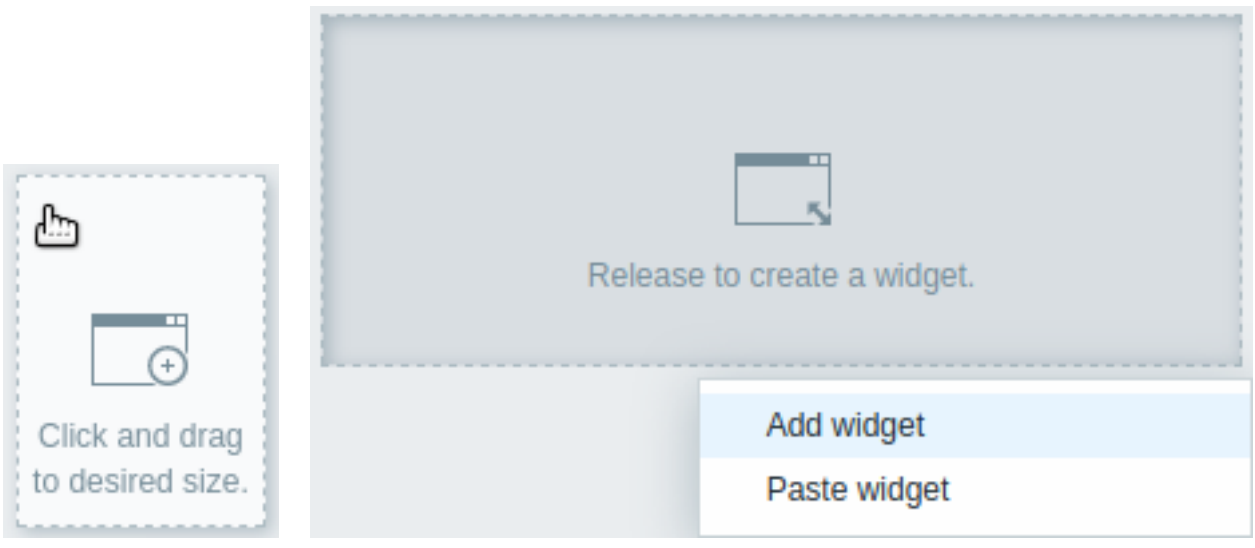
Widgets hinzufügen

Sie können einem Dashboard verschiedene Widgets (Aktionsprotokoll, Uhr, Ermittlungsstatus usw.) hinzufügen. Widgets können auf zwei Arten hinzugefügt werden:

- Klicken Sie auf die Schaltfläche *Hinzufügen* oder auf den Pfeil daneben und wählen Sie dann *Widget hinzufügen* aus dem Aktionsmenü. Nach der Konfiguration des Widgets wird es in seiner Standardgröße hinzugefügt und nach allen bereits vorhandenen Widgets platziert.

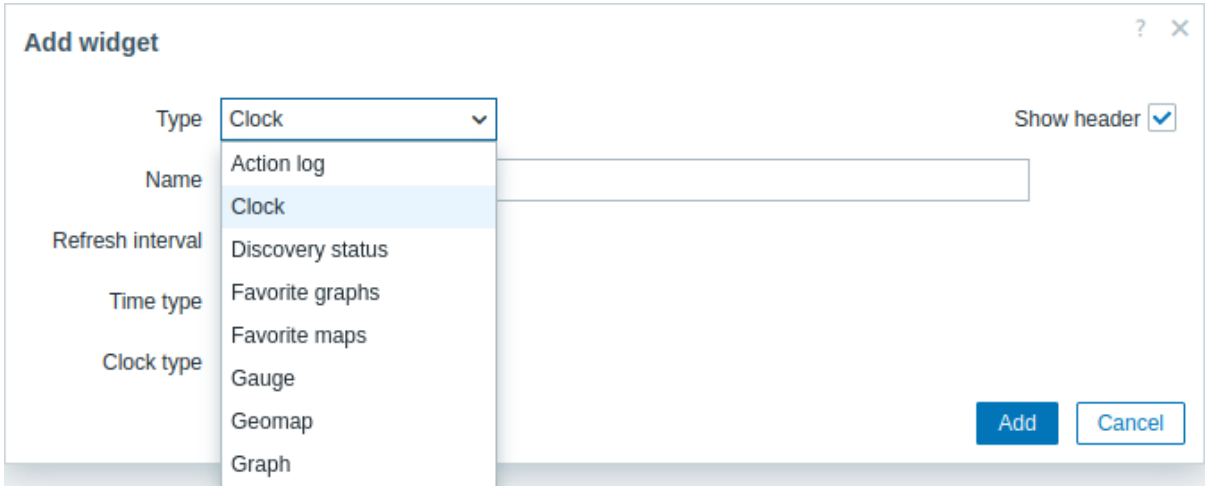


- Bewegen Sie den Mauszeiger auf eine freie Stelle im Dashboard. Es erscheint ein Platzhalter; klicken Sie darauf, um das Formular zur Widget-Konfiguration zu öffnen. Nach der Konfiguration des Widgets wird es in seiner Standardgröße hinzugefügt oder an den verfügbaren Platz angepasst. Alternativ können Sie den Platzhalter ziehen, um die Widget-Größe festzulegen, und ihn dann loslassen, um das Formular zur Widget-Konfiguration zu öffnen (wenn ein Widget in die Zwischenablage kopiert wurde, werden Sie zunächst aufgefordert, zwischen *Widget hinzufügen* und *Widget einfügen* zu wählen).



Im Formular zur Widget-Konfiguration:

1. Wählen Sie den Widget-Typ aus.
2. Konfigurieren Sie die Widget-Parameter.
3. Klicken Sie auf *Hinzufügen*, um das Widget zum Dashboard hinzuzufügen.



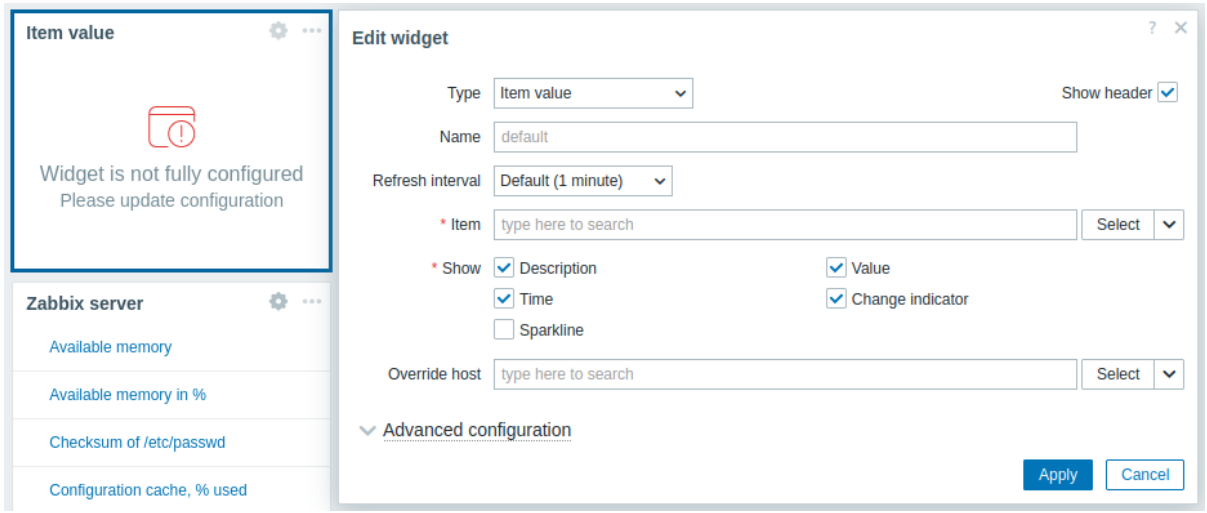
Widgets bearbeiten

Widgets können im Bearbeitungsmodus des Dashboards durch Ziehen ihrer Titelleiste in der Größe geändert und neu positioniert werden. Sie können auch das Formular zur Widget-Konfiguration durch Ziehen seiner Titelleiste neu positionieren.

Jedes Widget enthält Bedienelemente in seiner oberen rechten Ecke:

- - das Widget bearbeiten
- - das Widget-Menü öffnen

Während der Bearbeitung wird das Widget hervorgehoben und Änderungen werden in Echtzeit in der Vorschau angezeigt. Wenn erforderliche Widget-Parameter nicht konfiguriert sind, wechselt das Widget in den Status *Widget ist nicht vollständig konfiguriert*.



Klicken Sie nach der Bearbeitung im Widget auf *Anwenden* und dann oben rechts im Dashboard auf *Änderungen speichern*, um Ihre Änderungen zu übernehmen.

Widgets kopieren/einfügen

Sie können Dashboard-Widgets kopieren und einfügen, um schnell neue Widgets mit derselben Konfiguration zu erstellen. Widgets können innerhalb desselben Dashboards oder zwischen Dashboards kopiert und eingefügt werden, die in verschiedenen Tabs geöffnet sind.

Um ein Widget zu kopieren, verwenden Sie das Widget-Menü. So fügen Sie ein kopiertes Widget ein (verfügbar im Dashboard-Bearbeitungsmodus):

- Klicken Sie auf den Pfeil neben der Schaltfläche *Hinzufügen* und wählen Sie *Widget einfügen*.
- Alternativ klicken Sie auf einen leeren Bereich im Dashboard und wählen *Widget einfügen*.

Sie können ein kopiertes Widget auch über ein vorhandenes Widget einfügen, indem Sie die Option *Einfügen* in dessen Widget-Menü verwenden.

Erstellen einer Diashow

Eine Diashow wird automatisch ausgeführt, wenn das Dashboard zwei oder mehr Seiten enthält (siehe Seiten hinzufügen) und wenn eine der folgenden Bedingungen erfüllt ist:

- Die Option *Diashow automatisch starten* ist in den Dashboard-Eigenschaften aktiviert

- Die Dashboard-URL enthält den Parameter `slideshow=1`

Die Seiten wechseln entsprechend den in den Eigenschaften des Dashboards und der einzelnen Seiten festgelegten Intervallen. Klicken Sie auf:

- *Diashow stoppen* - um die Diashow zu stoppen
- *Diashow starten* - um die Diashow zu starten

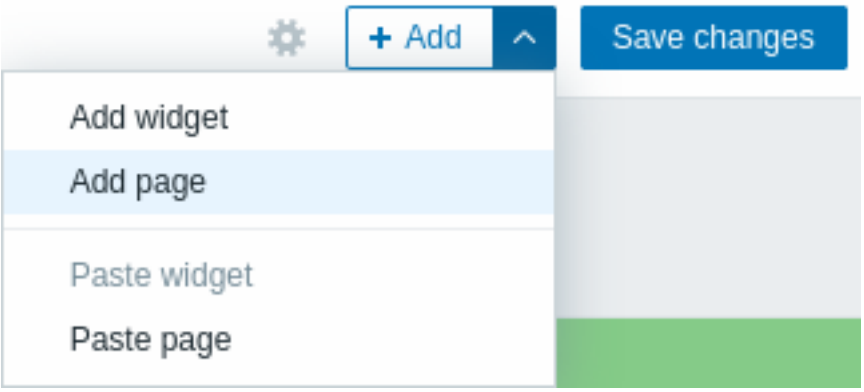
Steuerelemente für die Diashow sind auch im Kiosk-Modus verfügbar (in dem nur der Seiteninhalt angezeigt wird):

- - Diashow stoppen
- - Diashow starten
- - eine Seite zurückgehen
- - zur nächsten Seite wechseln

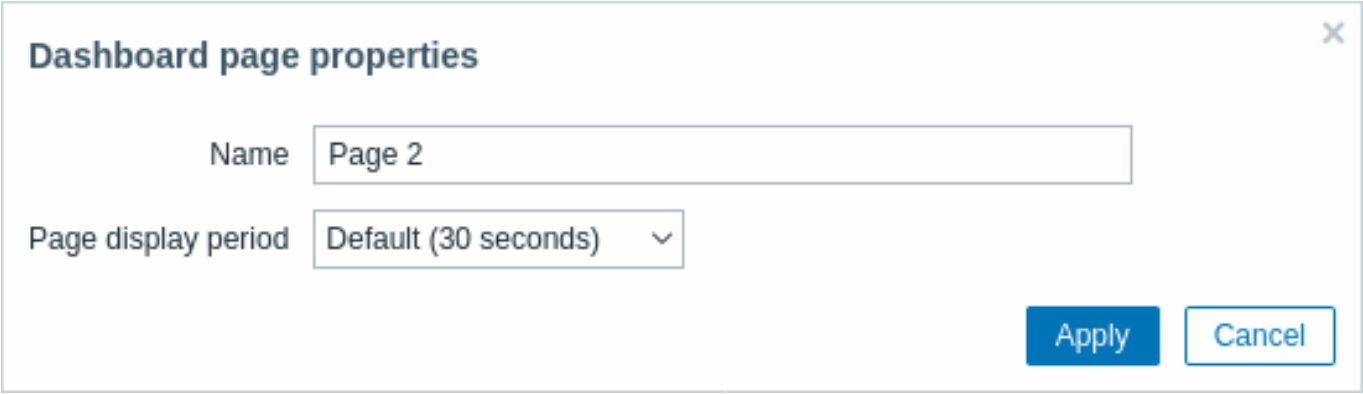
Hinzufügen von Seiten

So fügen Sie einem Dashboard eine neue Seite hinzu:

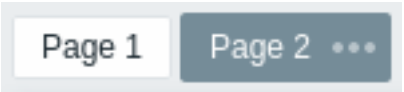
- Stellen Sie sicher, dass sich das Dashboard im Bearbeitungsmodus befindet.
- Klicken Sie auf den Pfeil neben der Schaltfläche *Hinzufügen* und wählen Sie die Option *Seite hinzufügen* aus.



- Füllen Sie die allgemeinen Seitenparameter aus und klicken Sie auf *Übernehmen*. Wenn Sie den Namen leer lassen, wird die Seite mit dem Namen `Page N` hinzugefügt, wobei „N“ die fortlaufende Nummer der Seite ist. Mit der Anzeigedauer der Seite können Sie festlegen, wie lange eine Seite in einer Diashow angezeigt wird.



Eine neue Seite wird hinzugefügt, erkennbar an einer neuen Registerkarte (*Page 2*).



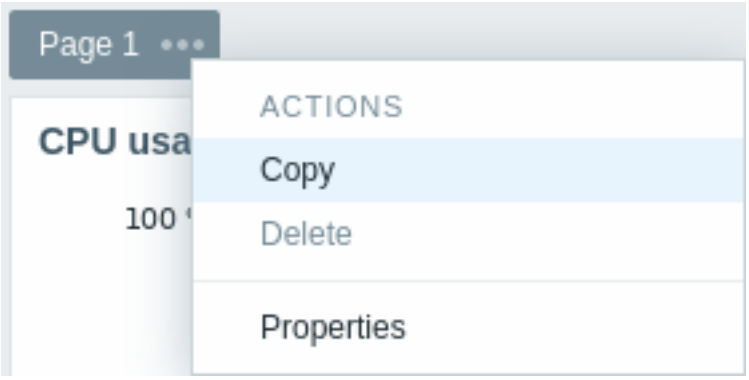
Die Seiten können durch Ziehen und Ablegen der Seitenregisterkarten neu angeordnet werden. Bei der Neuordnung bleiben die ursprünglichen Seitennamen erhalten. Sie können jederzeit zu einer Seite wechseln, indem Sie auf ihre Registerkarte klicken.

Wenn eine neue Seite hinzugefügt wird, ist sie leer. Sie können ihr wie oben beschrieben Widgets hinzufügen.

Seiten kopieren/einfügen

Dashboard-Seiten können kopiert und eingefügt werden, sodass eine neue Seite mit den Eigenschaften einer bestehenden erstellt werden kann. Sie können aus demselben Dashboard oder aus einem anderen Dashboard eingefügt werden.

Um eine bestehende Seite in das Dashboard einzufügen, kopieren Sie sie zunächst über das Seitenmenü:

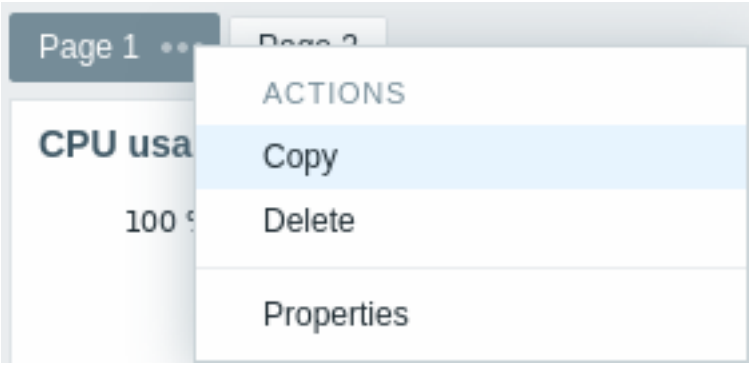


So fügen Sie die kopierte Seite ein:

- Stellen Sie sicher, dass sich das Dashboard im Bearbeitungsmodus befindet
- Klicken Sie auf den Pfeil neben der Schaltfläche *Hinzufügen* und wählen Sie die Option *Seite einfügen*

Seitenmenü

Das Seitenmenü kann durch Klicken auf die drei Punkte  neben dem Seitennamen geöffnet werden:

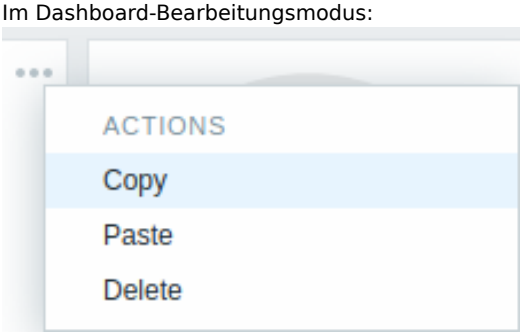


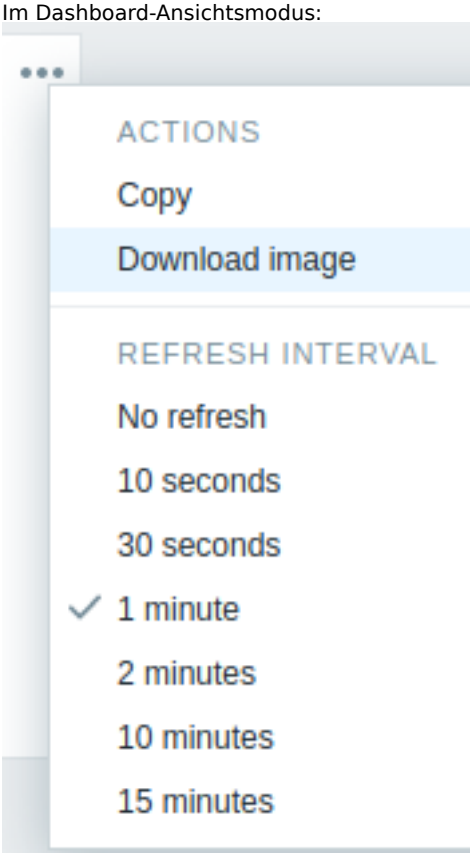
Es enthält die folgenden Optionen:

- *Kopieren* - die Seite kopieren
- *Löschen* - die Seite löschen (Seiten können nur im Dashboard-Bearbeitungsmodus gelöscht werden)
- *Eigenschaften* - die Seitenparameter anpassen (den Namen und die Anzeigedauer der Seite in einer Diashow)

Widget-Menü

Das Widget-Menü enthält je nachdem, ob sich das Dashboard im Bearbeitungs- oder Ansichtsmodus befindet, unterschiedliche Optionen:

Widget-Menü	Optionen
Im Dashboard-Bearbeitungsmodus: 	*Kopieren* - das Widget kopieren. *Einfügen* - ein kopiertes Widget über dieses Widget einfügen. Diese Option ist ausgegraut, wenn kein Widget kopiert wurde. *Löschen* - das Widget löschen.

Widget-Menü	Optionen
Im Dashboard-Ansichtsmodus: 	*Kopieren* - das Widget kopieren. *Bild herunterladen* - das Widget als PNG-Bild herunterladen. Diese Option ist für Widgets verfügbar, die diese Funktion unterstützen (siehe die einzelnen Widget-Seiten). Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner „Downloads“ gespeichert oder es wird ein Dialogfeld „Speichern“ geöffnet, in dem Sie den Speicherort auswählen können. *Aktualisierungsintervall* - die Häufigkeit der Aktualisierung des Widget-Inhalts auswählen

Berechtigungen für Dashboards

Die Berechtigungen für Dashboards sind für normale Benutzer und Benutzer vom Typ „Admin“ (vorausgesetzt, der Zugriff auf Dashboards und die Option *Dashboards erstellen und bearbeiten* sind für die ihnen zugewiesene Benutzerrolle aktiviert) auf die unten beschriebene Weise eingeschränkt.

- Sie können ein öffentliches Dashboard sehen und klonen, auch wenn in den Freigabeeinstellungen des Dashboards keine Benutzer oder Benutzergruppen hinzugefügt wurden.
- Sie können ein privates Dashboard sehen und klonen, wenn sie dafür mindestens *Lesen*-Rechte haben, die über die Freigabeeinstellungen festgelegt wurden.
- Sie können ein Dashboard nur bearbeiten und löschen, wenn sie dafür *Lesen/Schreiben*-Rechte haben, die über die Freigabeeinstellungen festgelegt wurden.
- Sie können den Eigentümer des Dashboards nicht ändern.

1 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.

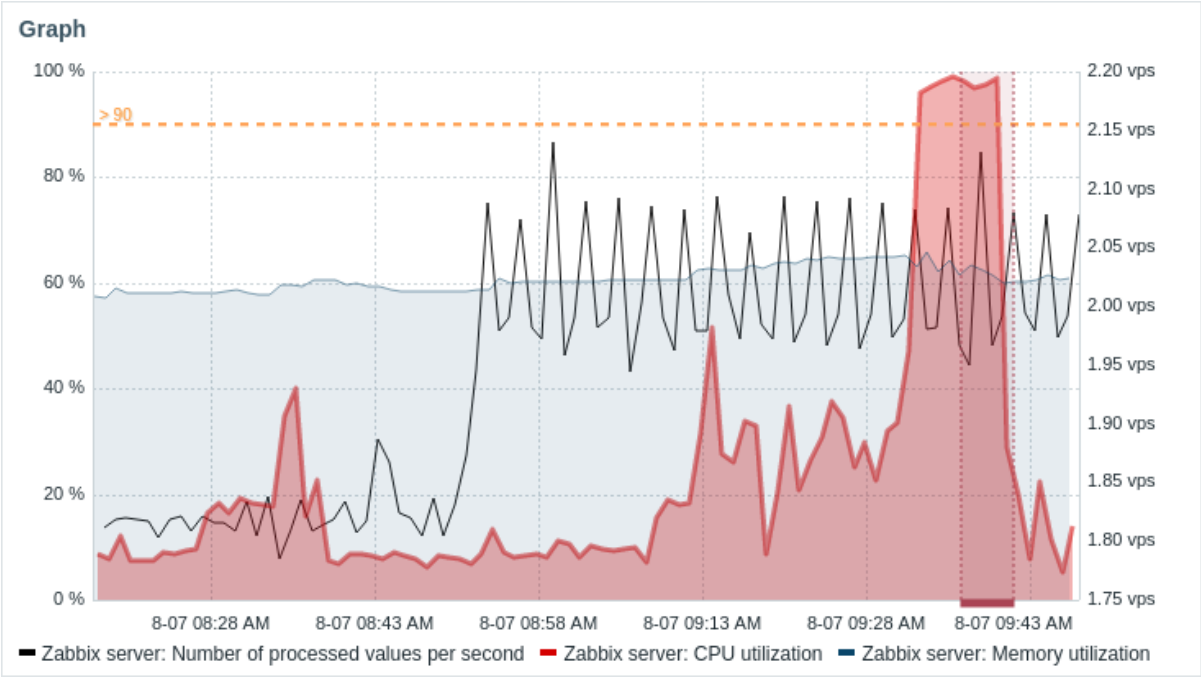
Ausführliche Informationen zur Widget-Konfiguration finden Sie unter Dashboard-Widget-Parameter.

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.

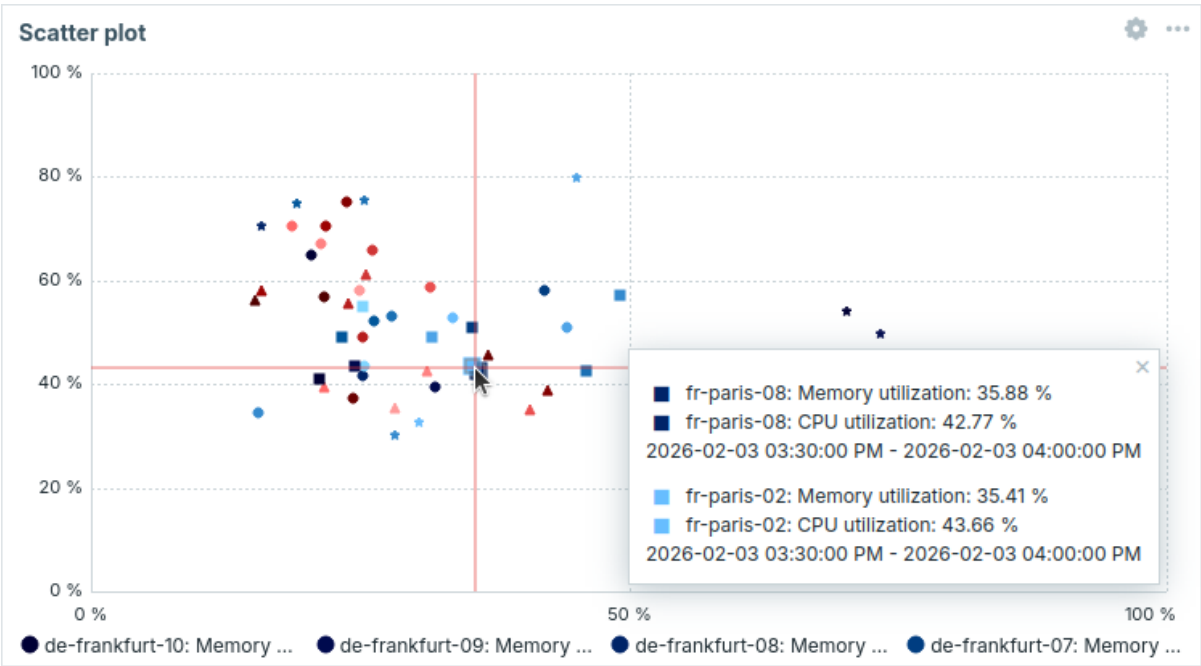


Siehe auch:

- 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.

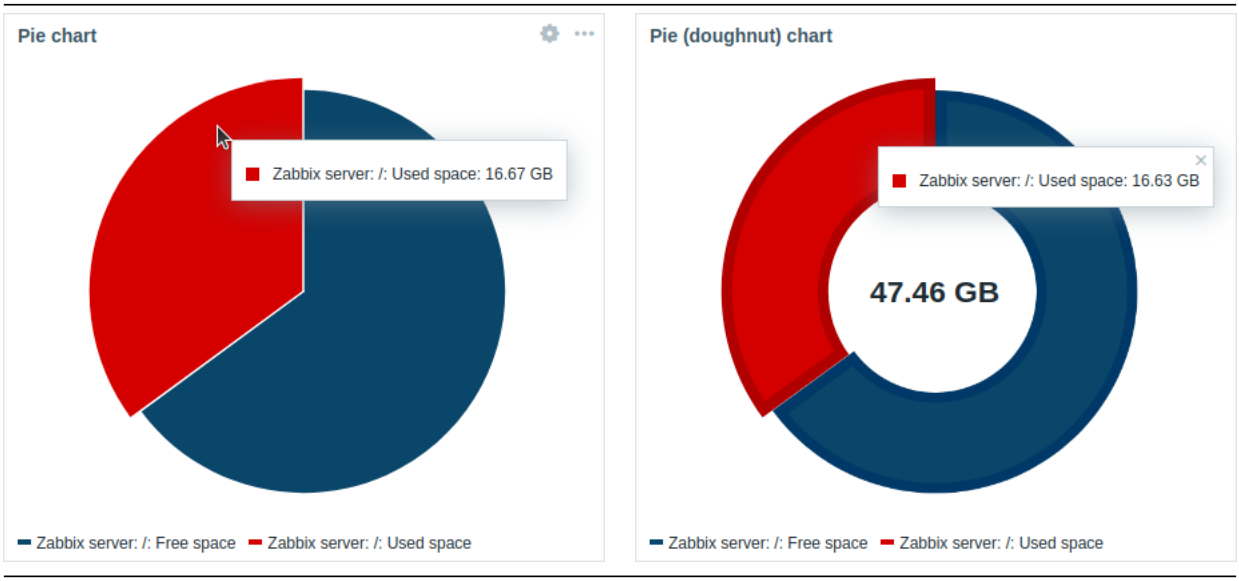
Streudiagramm

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

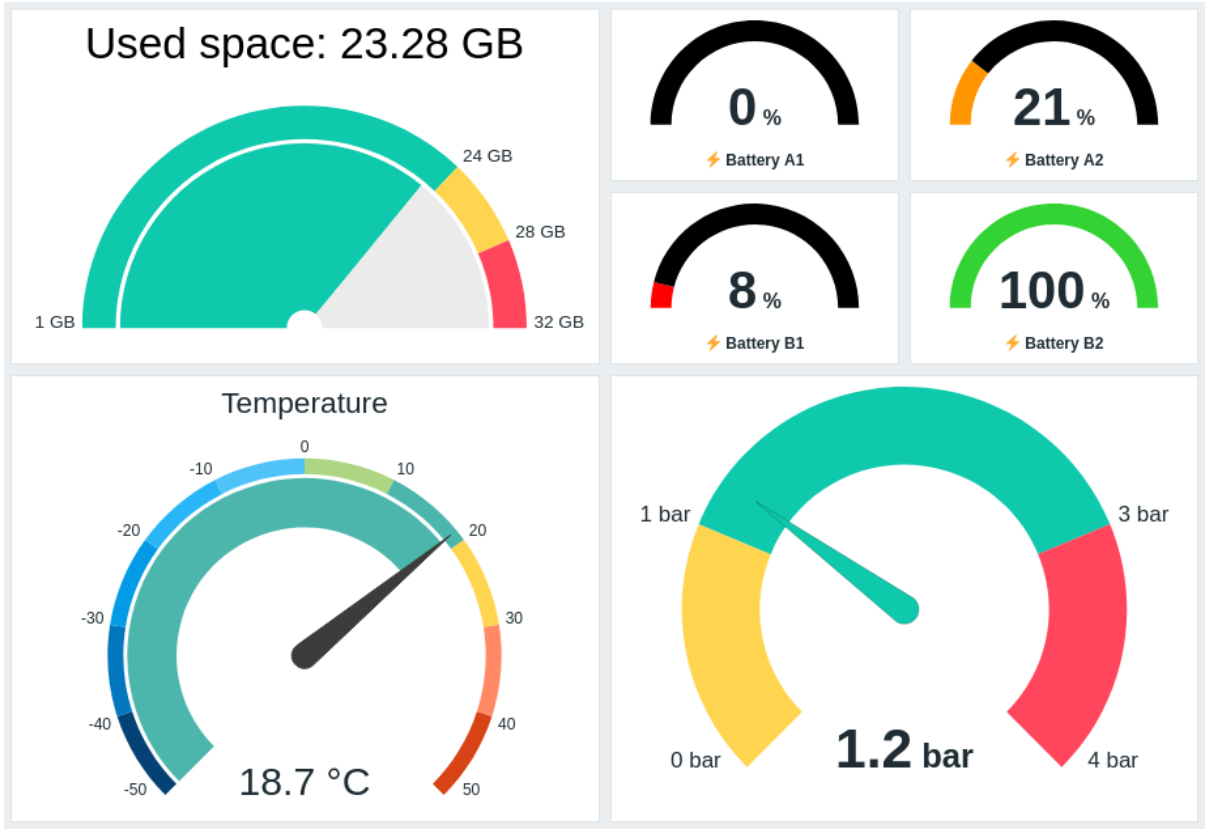
Das Widget Kreisdiagramm zeigt numerische Datenpunkt-Daten als vektorbasiertes Kreis- oder Ringdiagramm an. Dieses Widget kann Ihnen dabei helfen zu visualisieren, wie Datenpunkte oder Hosts zum gesamten Datensatz beitragen.



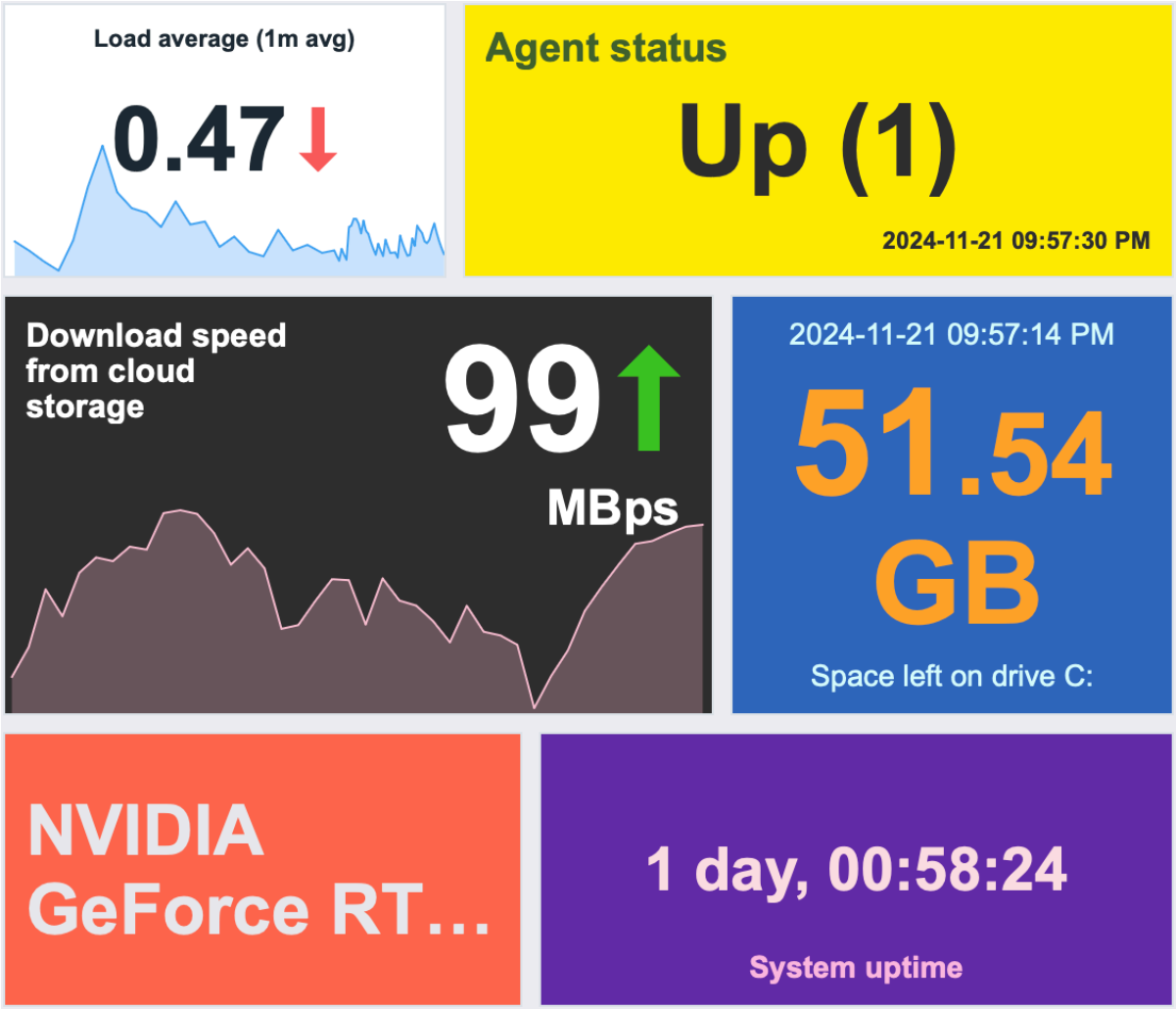
Messanzeige/Datenpunktwert

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“:

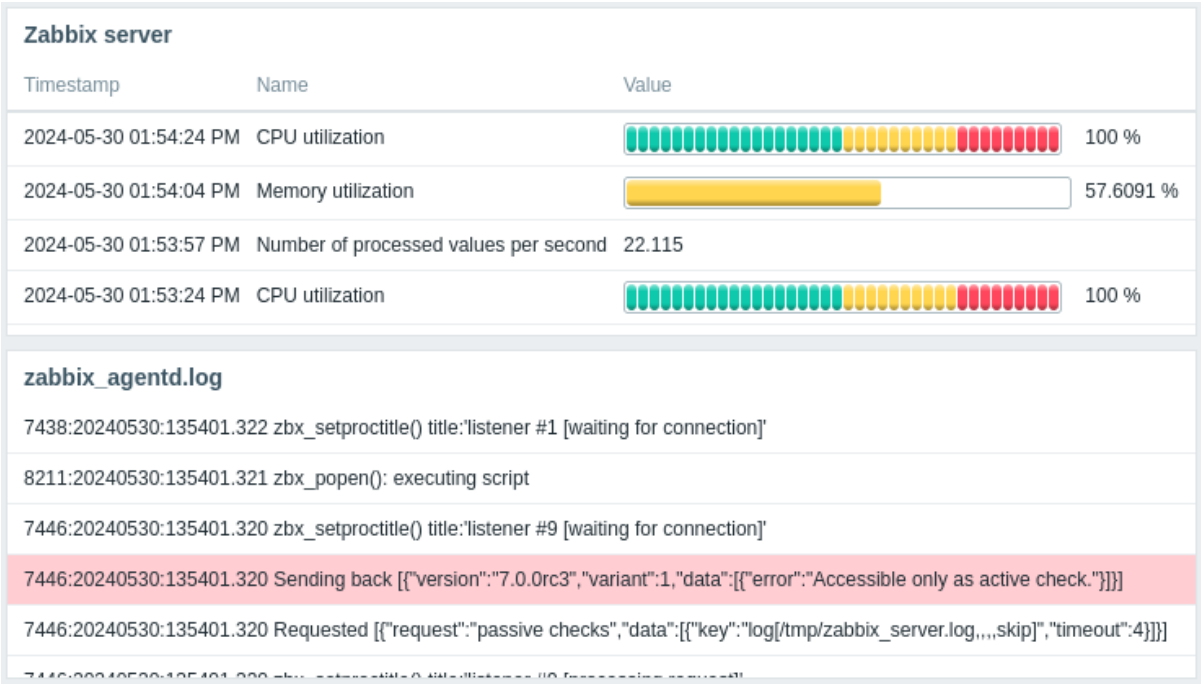


Widget „Datenpunktwert“:



Datenpunkt-Verlauf

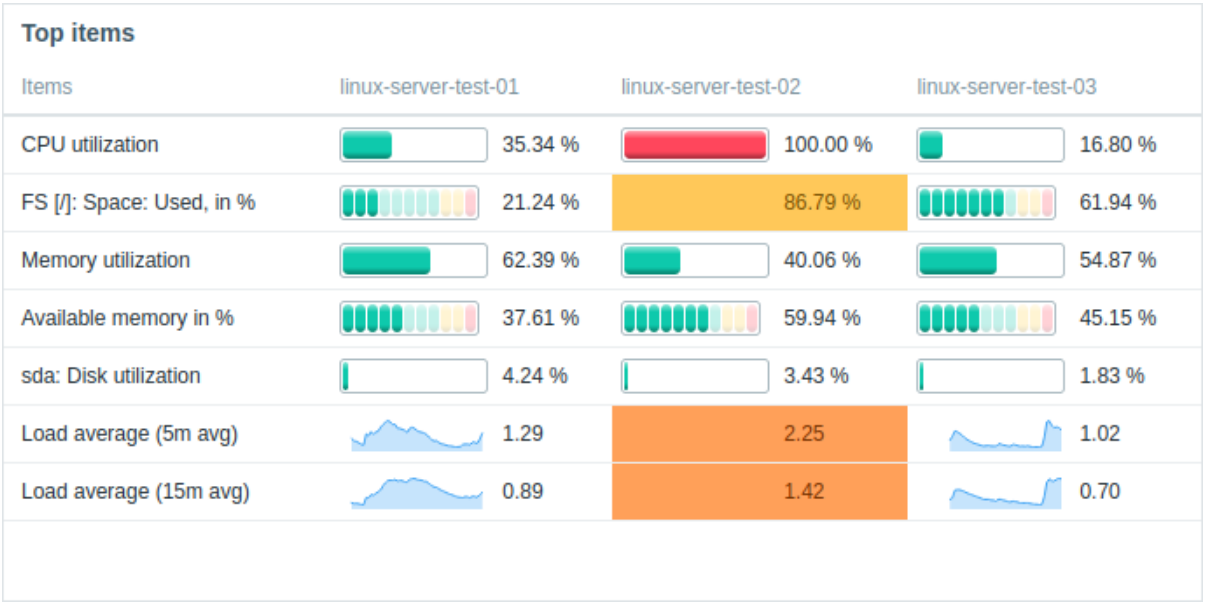
Das Widget Datenpunkt-Verlauf zeigt die neuesten Daten für alle Datenpunkt-Typen (numerisch, Text usw.) in Tabellenform an. Es kann Fortschrittsbalken und Bilder anzeigen (nützlich für Browser-Datenpunkte) und Werte hervorheben (nützlich für die Überwachung von Protokolldateien).



Top-Datenpunkte

Das Widget Top-Datenpunkte zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte an und bietet einen schnellen

Überblick über deren Leistung.

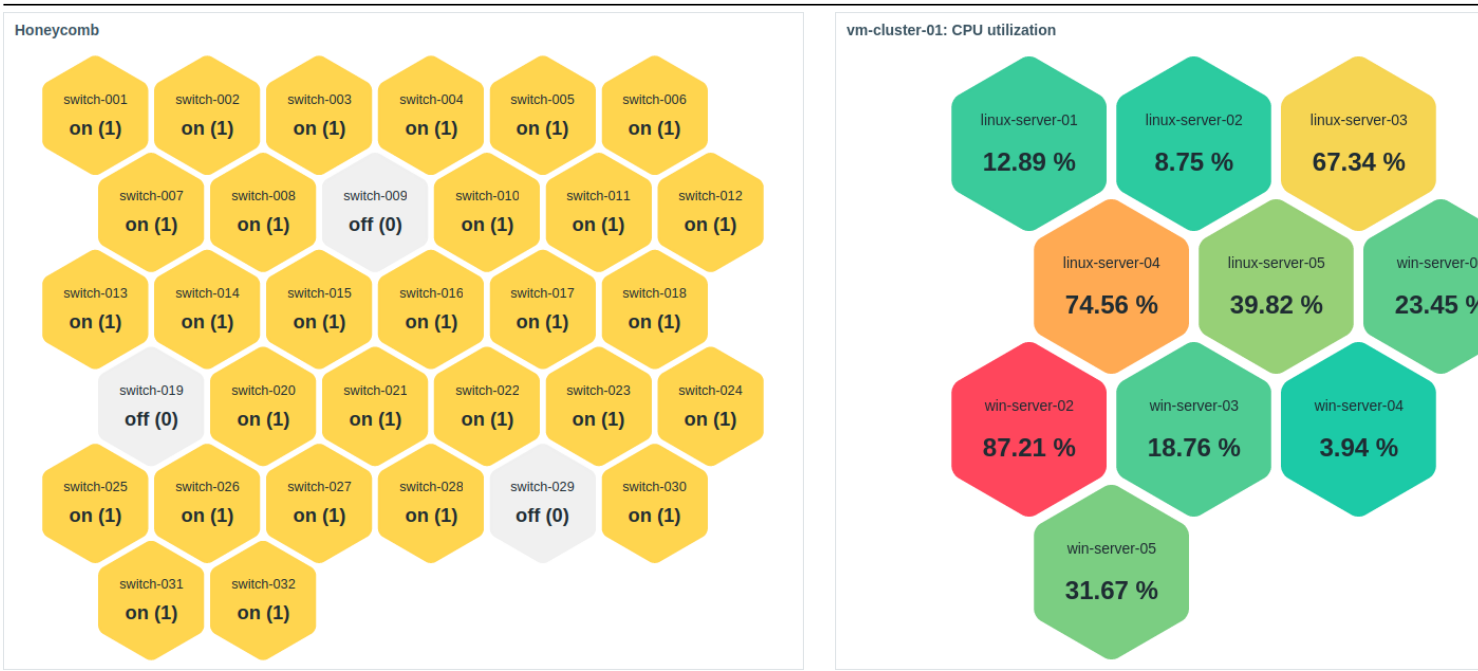


Siehe auch:

- 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.

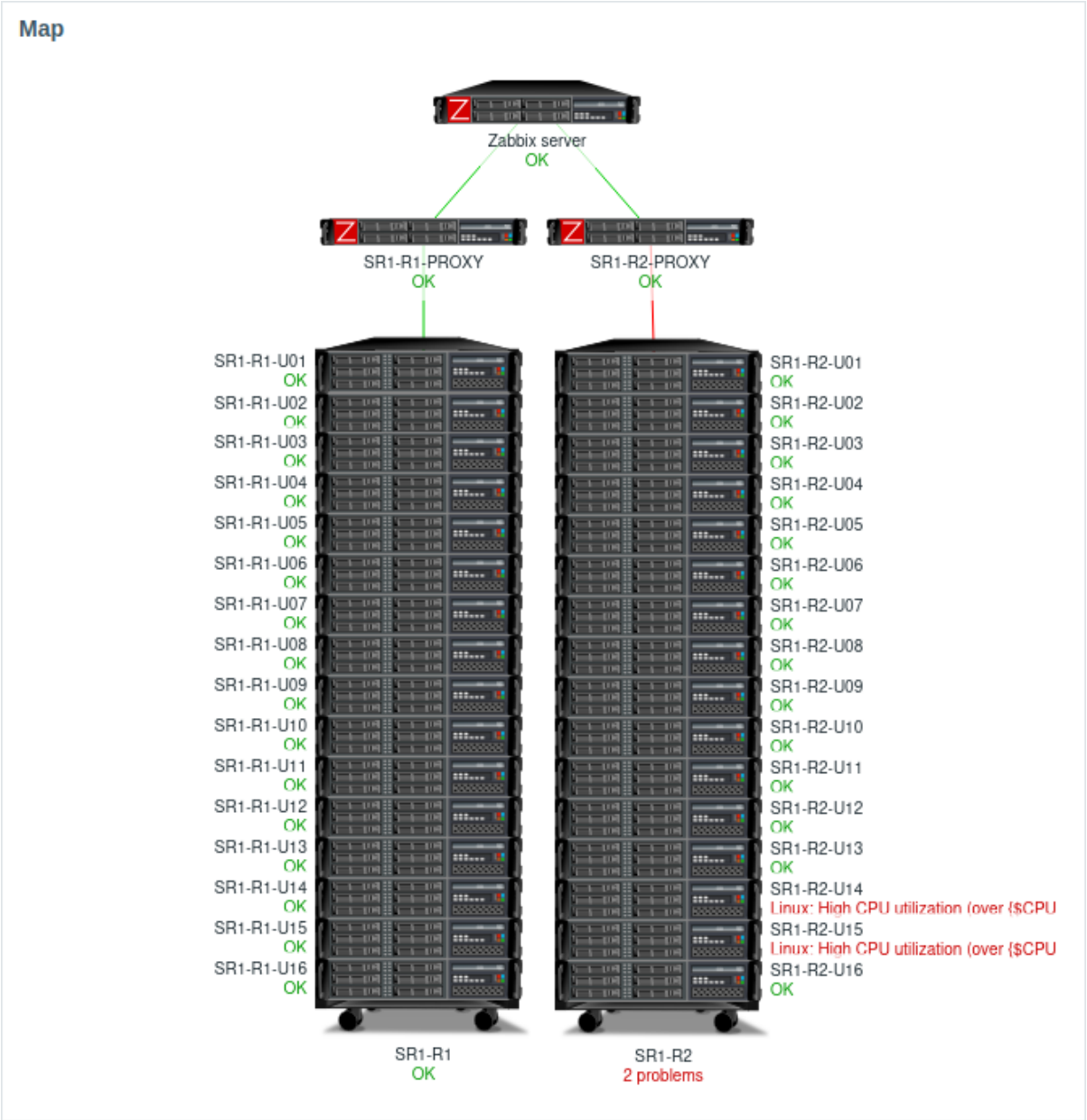


Karten

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.

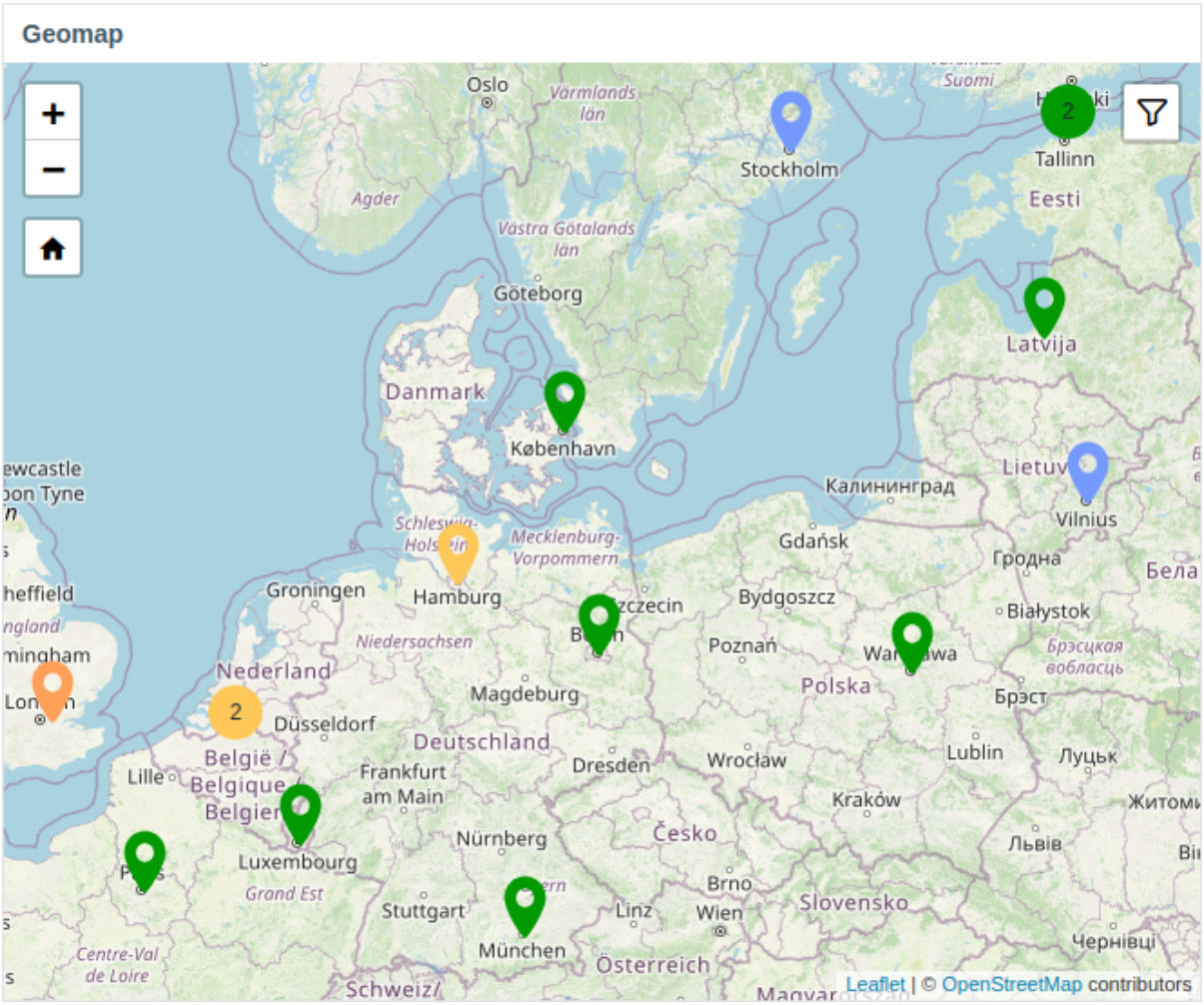


Siehe auch:

- 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.

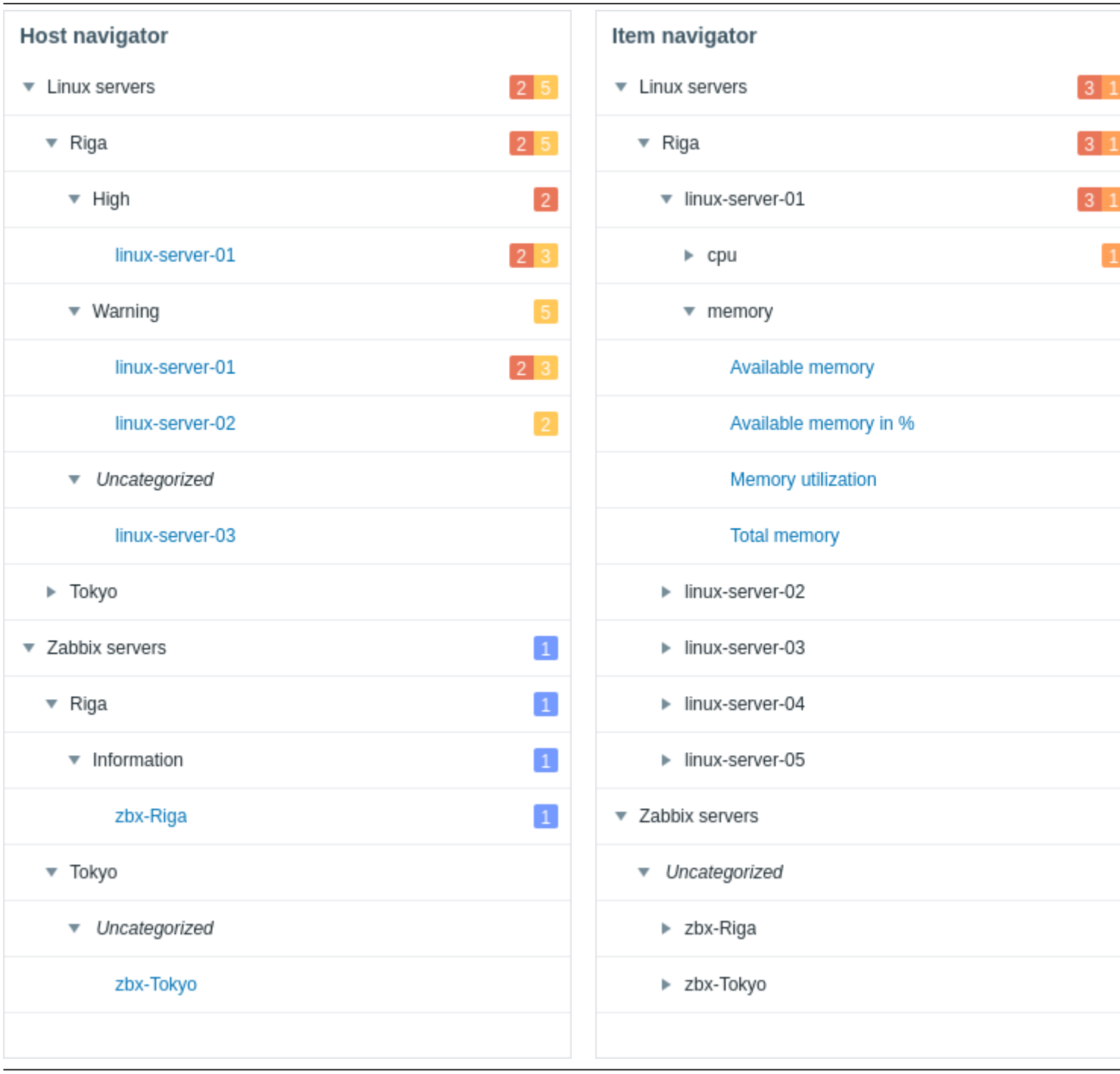


Host-/Datenpunkt-Details

Diese Widgets stellen Listen und Detailansichten von Hosts oder Datenpunkten bereit.

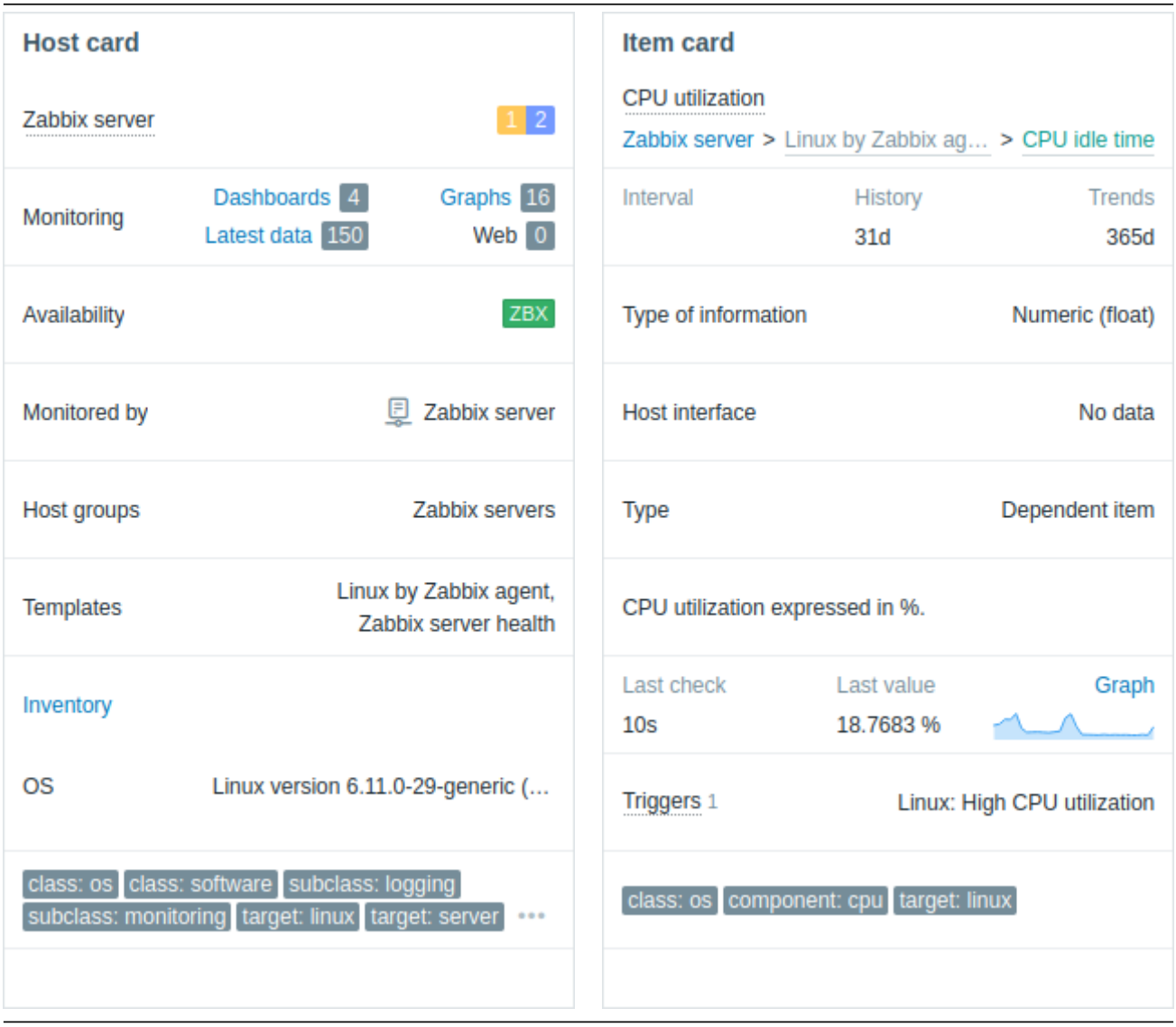
Host-/Datenpunkt-Navigator

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

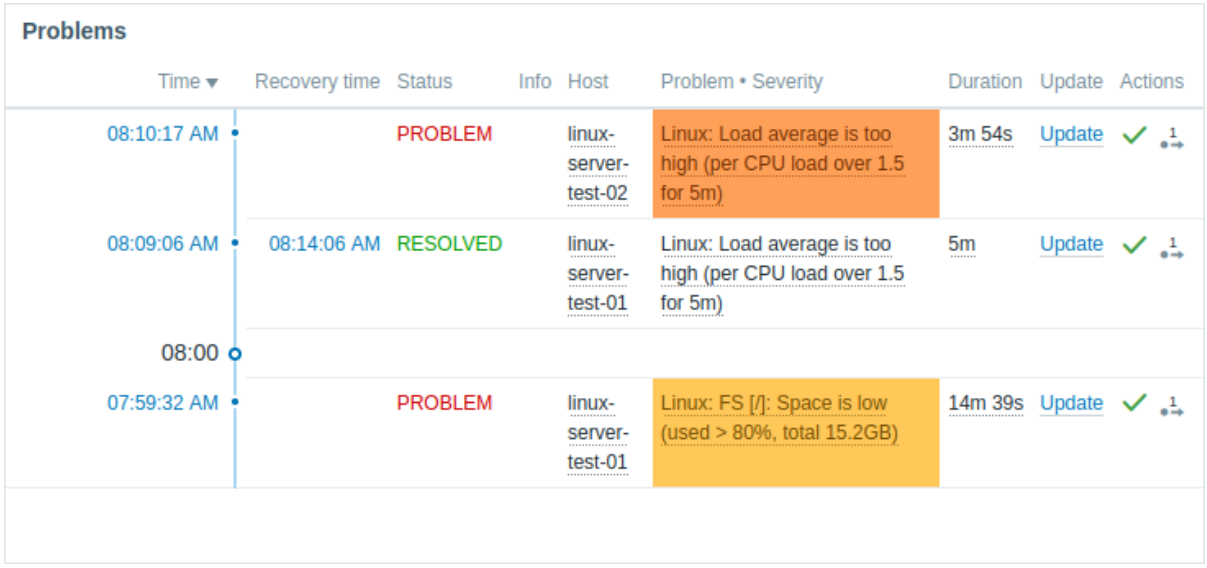
Das Widget Host-Verfügbarkeit zeigt die Verfügbarkeit von Hosts in den ausgewählten Host-Gruppen an und ermöglicht es Ihnen, die Anzahl der Hosts zu überwachen, die verfügbar oder nicht verfügbar sind.

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

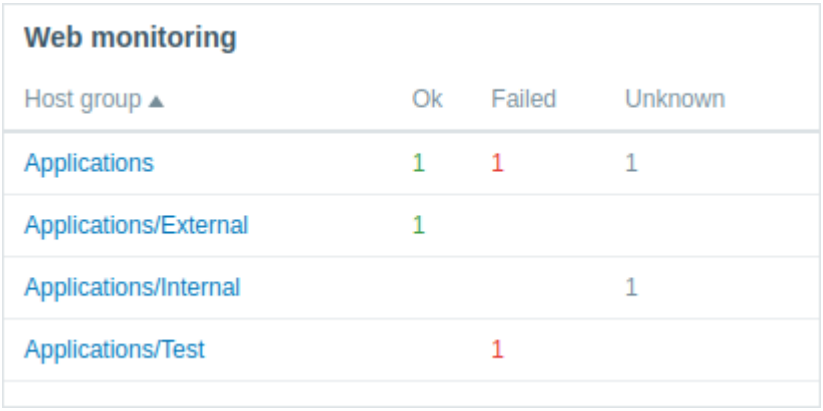
Das Widget SLA report zeigt SLA-Berichte an und hilft Ihnen, die Service-Performance anhand definierter Ziele zu überwachen.

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

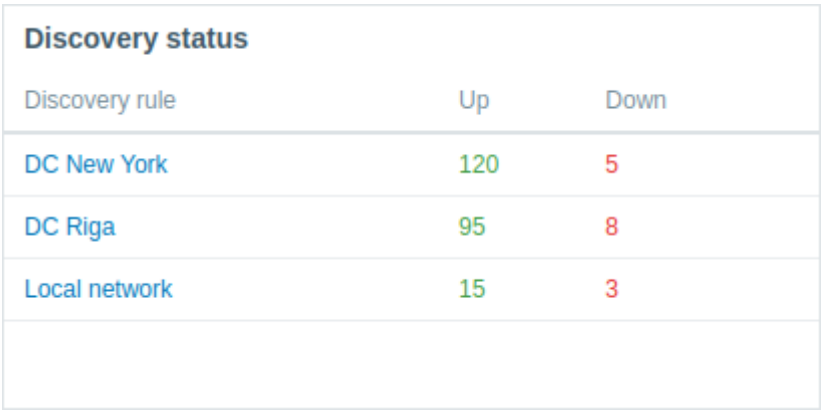
Webüberwachung

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

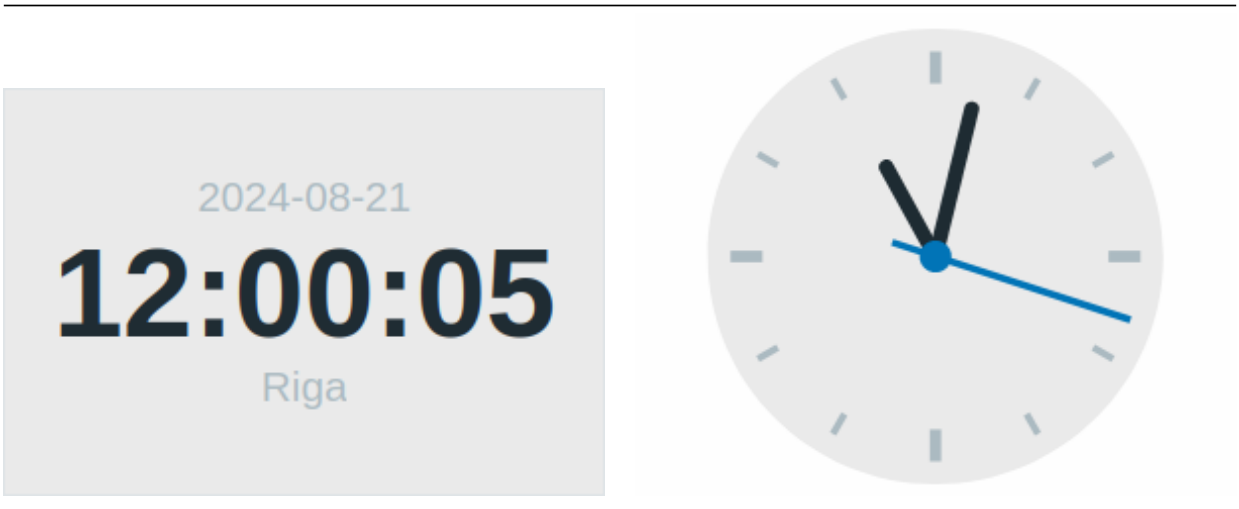
Das Widget Systeminformationen zeigt eine Zusammenfassung wichtiger Zabbix-Server- und Systemdaten oder Details zu Hochverfügbarkeitsknoten an.

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	

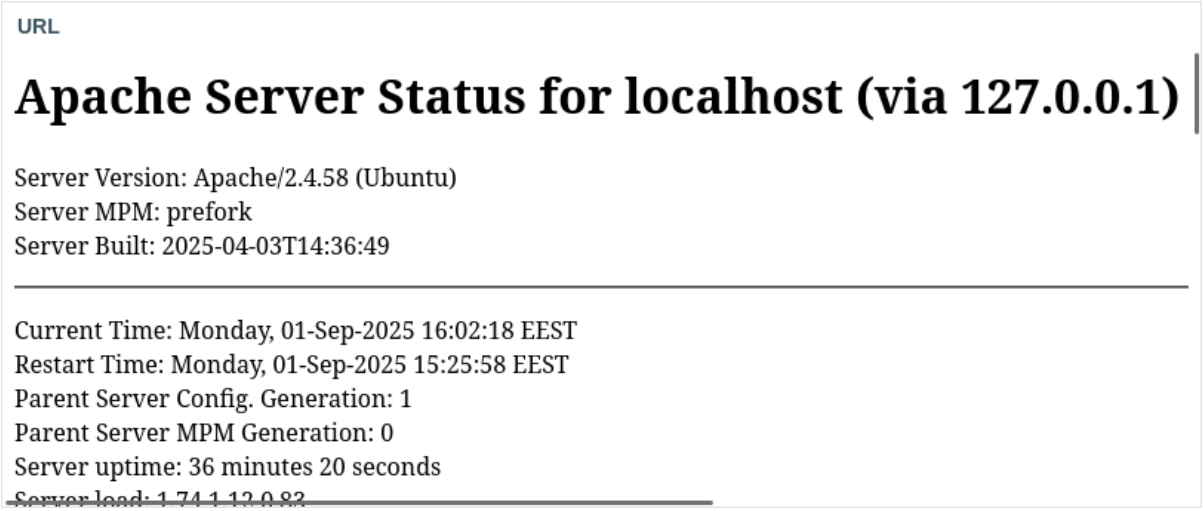
Uhr

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.



2 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 sind für jedes einzelne Widget gemeinsam:

Name	Geben Sie einen Widget-Namen ein.
Refresh interval	Konfigurieren Sie das Standard-Aktualisierungsintervall. Die Standard-Aktualisierungsintervalle für Widgets reichen je nach Widget-Typ von *No refresh* bis *15 minutes*. Zum Beispiel: - *No refresh* für das Widget *URL*; - *1 minute* für das Widget *Action log*; - *15 minutes* für das Widget *Clock*. Aktualisierungsintervalle können für alle Benutzer auf einen Standardwert gesetzt werden. Schalten 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 auch sein eigenes Widget-Aktualisierungsintervall festlegen. Klicken Sie im Dashboard im Ansichtsmodus auf die Schaltfläche mit den drei Punkten auf einem Widget und wählen Sie das gewünschte Aktualisierungsintervall aus der Dropdown-Liste aus. Beachten Sie, dass das benutzerspezifische Aktualisierungsintervall Vorrang vor der Widget-Einstellung hat und auch dann beibehalten wird, wenn die Widget-Einstellung geändert wird.
Show header	Aktivieren Sie das Kontrollkästchen, um die Widget-Kopfzeile dauerhaft anzuzeigen. Wenn es nicht aktiviert ist, wird die Kopfzeile ausgeblendet, um Platz zu sparen, und nur sichtbar, wenn sich der Mauszeiger über dem Widget befindet (sowohl im Ansichts- als auch im Bearbeitungsmodus). Die Kopfzeile ist auch halb sichtbar, wenn ein Widget an eine neue Position gezogen wird.

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:

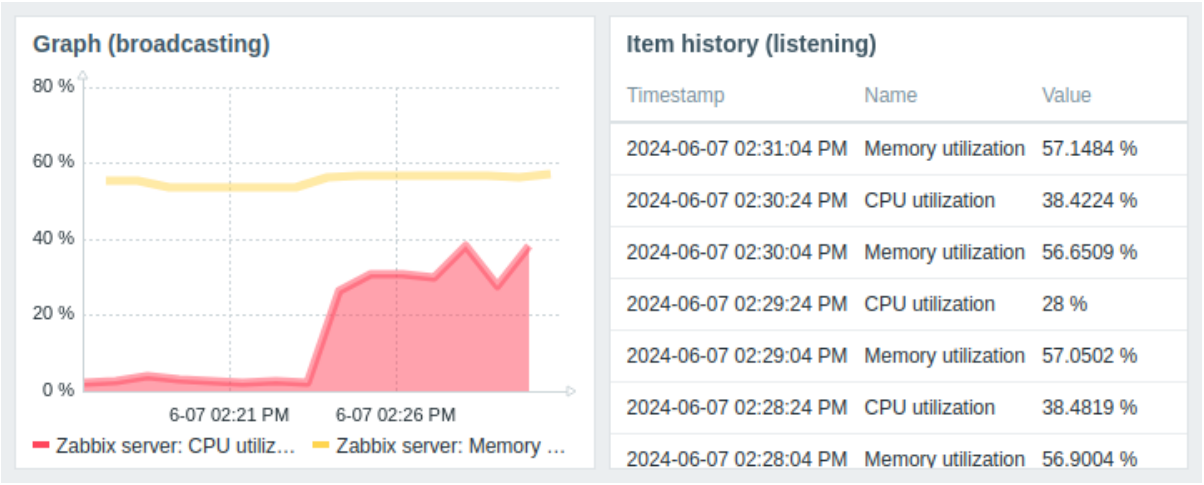
- Das Widget *Aktionsprotokoll* kann Zeitraumbereichsdaten nur von den Widgets *Graph*, *Graph (classic)* und *Graph prototype* abrufen.
- Das Widget *Geomap* kann Host-Daten an Widgets übertragen, die diese empfangen (*Honeycomb*, *Top items* usw.), und kann außerdem Hostgruppen- und Host-Daten von 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.

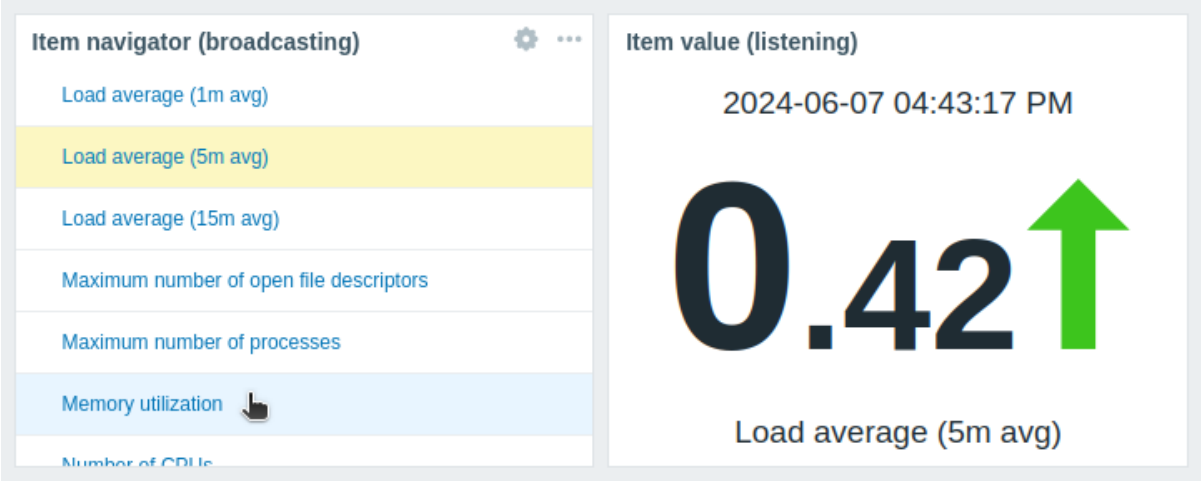
Widget	Überträgt	Empfängt
Aktionsprotokoll	-	Zeitraum
Uhr	-	-
Discovery-Status	-	-
Favorisierte Graphen	-	-
Favorisierte Karten	-	-
Messanzeige	-	Hosts, Datenpunkte
Geomap	Hosts	Hostgruppen, Hosts
Graph	Zeitraum, Datenpunkte	Hosts, Datenpunkte, Zeitraum
Graph (classic)	Zeitraum	Hosts, Datenpunkte, Graphen, 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**.

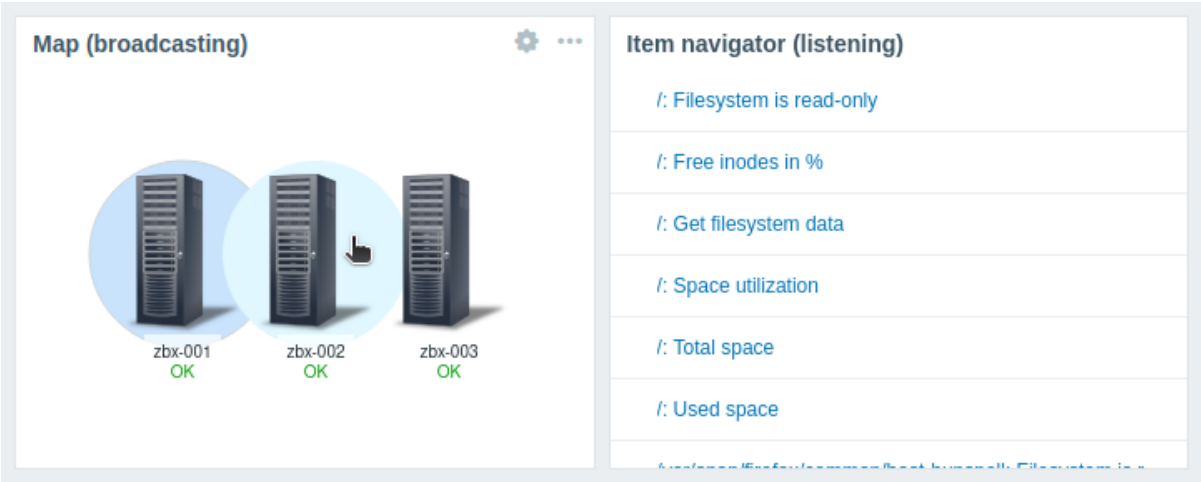
Widgets, die das Übertragen unterstützen, beginnen nach ihrer Erstellung automatisch damit. Zum Beispiel überträgt das Widget *Graph* sofort Daten zum Zeitraum an zuhörende Widgets.



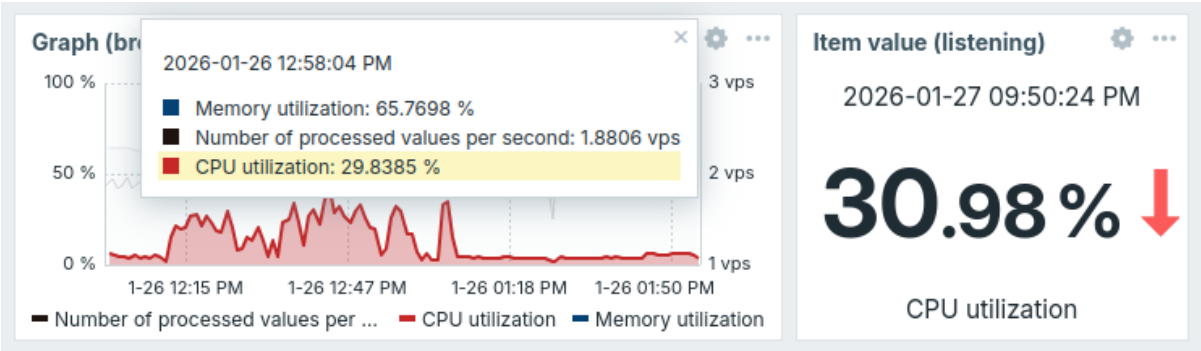
Widgets, die die Auswahl von Entitäten unterstützen, übertragen automatisch Daten für die erste verfügbare Entität. Zum Beispiel überträgt das Widget *Item navigator* Daten für den ersten Datenpunkt in seiner Datenpunktliste. Wenn ein anderer Datenpunkt ausgewählt wird, überträgt es Daten für diesen 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.

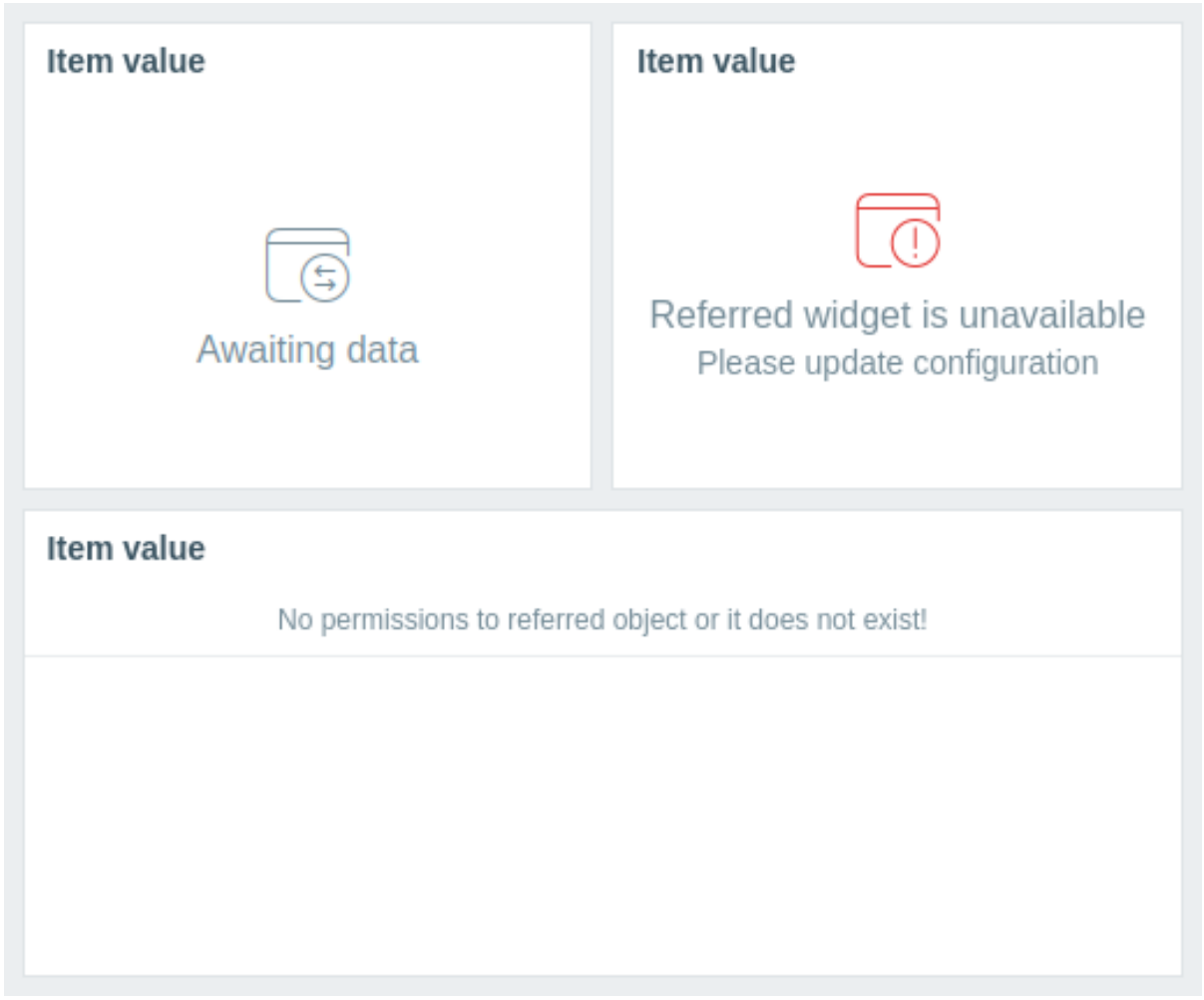


Widgets wie *Graph* und *Scatter plot* können Datenpunkte übertragen, wenn in ihren Tooltips ein Datenpunkt ausgewählt wird:



Widgets haben außerdem ein spezifisches Verhalten, wenn sie auf Daten anderer Widgets **reagieren**:

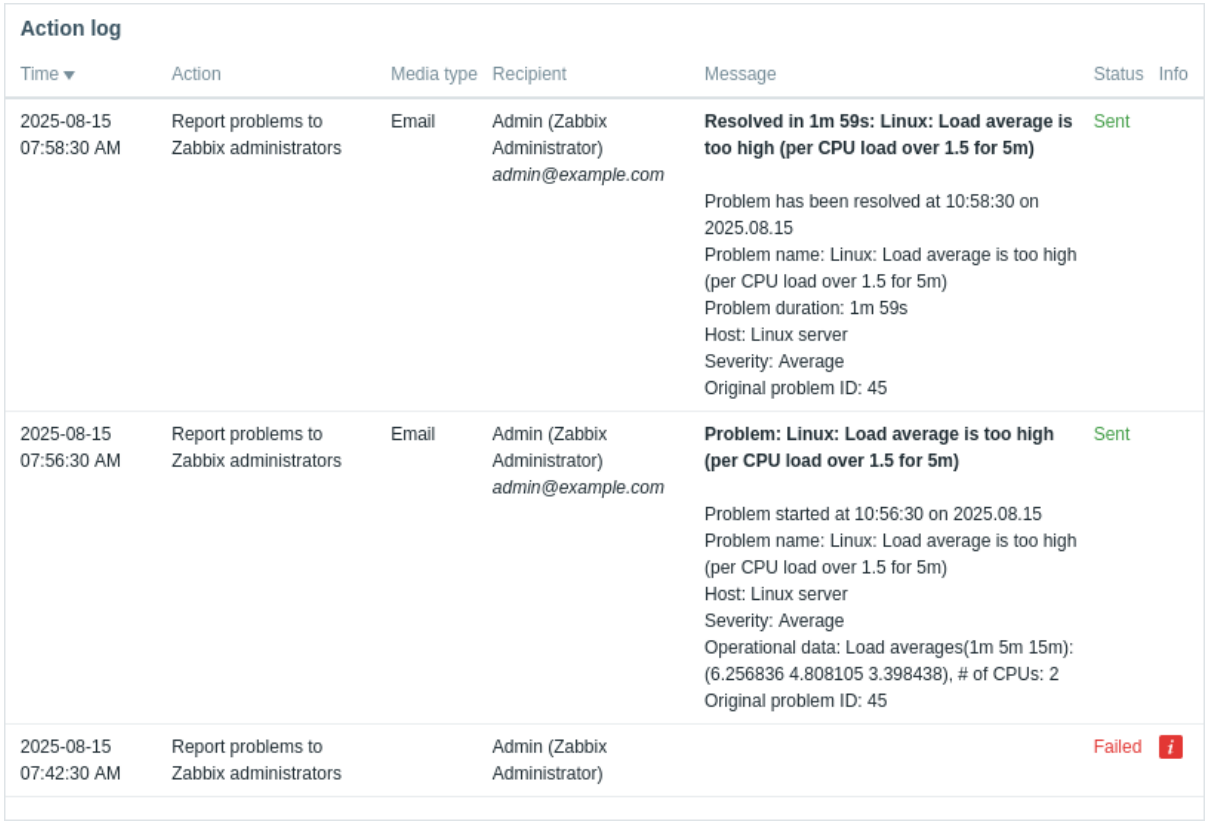
- Wenn das Widget der Datenquelle keine Daten überträgt, wechselt das zuhörende Widget in den Status *Awaiting data*.
- Wenn das Widget der Datenquelle gelöscht, durch ein inkompatibles Widget ersetzt oder auf eine andere Dashboard-Seite verschoben wurde, wechselt das zuhörende Widget in den Status *Referred 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!"*



1 Aktionsprotokoll

Übersicht

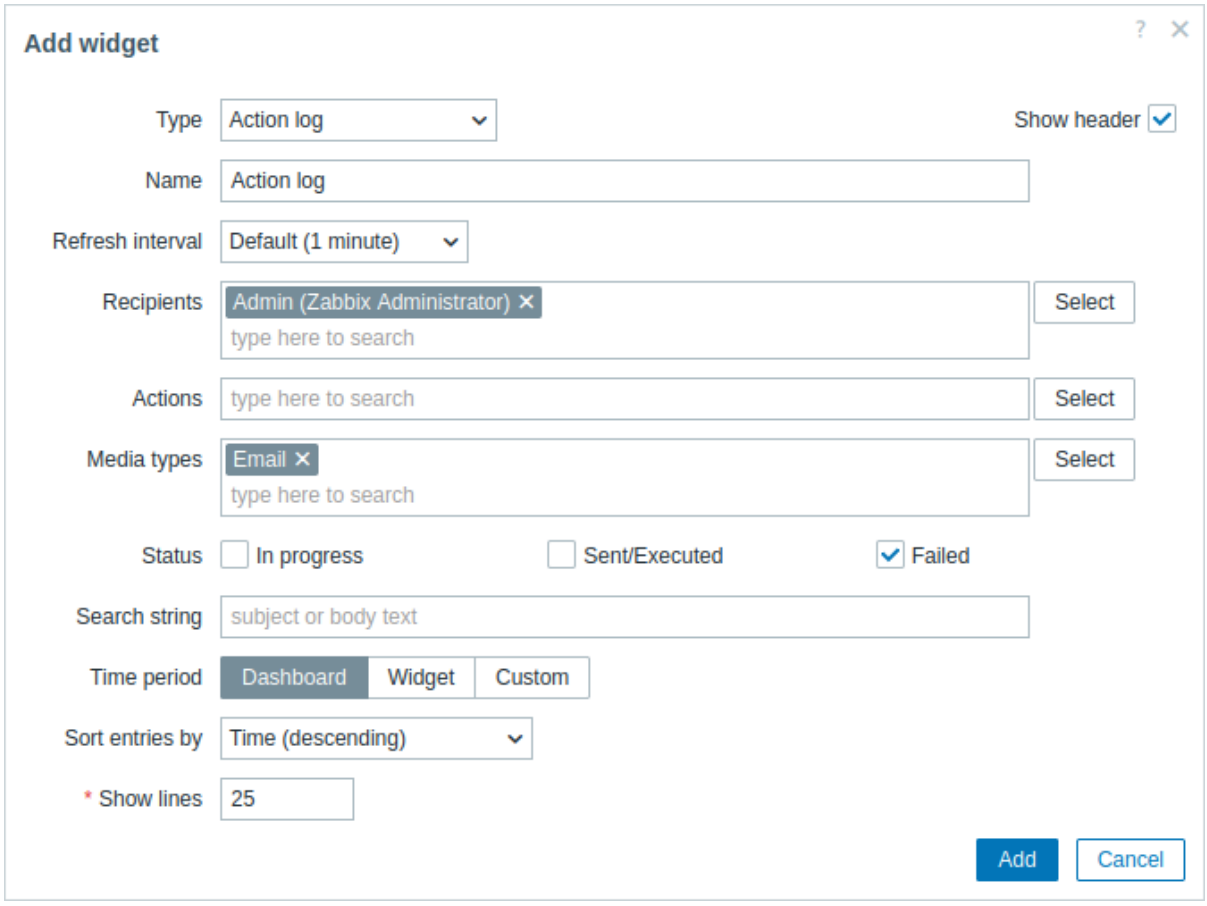
Das Widget *Aktionsprotokoll* zeigt Details zu Operationen (Benachrichtigungen, Remote-Befehle) an, die innerhalb einer Aktion ausgeführt wurden. Das detaillierte Protokoll unterstützt bei Audits, der Fehlerbehebung und der Überwachung der Ausführung jeder Aktion.



Es zeigt dieselben Daten wie *Berichte* > *Aktionsprotokoll* an und kann bis zu 1000 Einträge anzeigen.

Konfiguration

Wählen Sie zur Konfiguration *Aktionsprotokoll* als Typ aus:



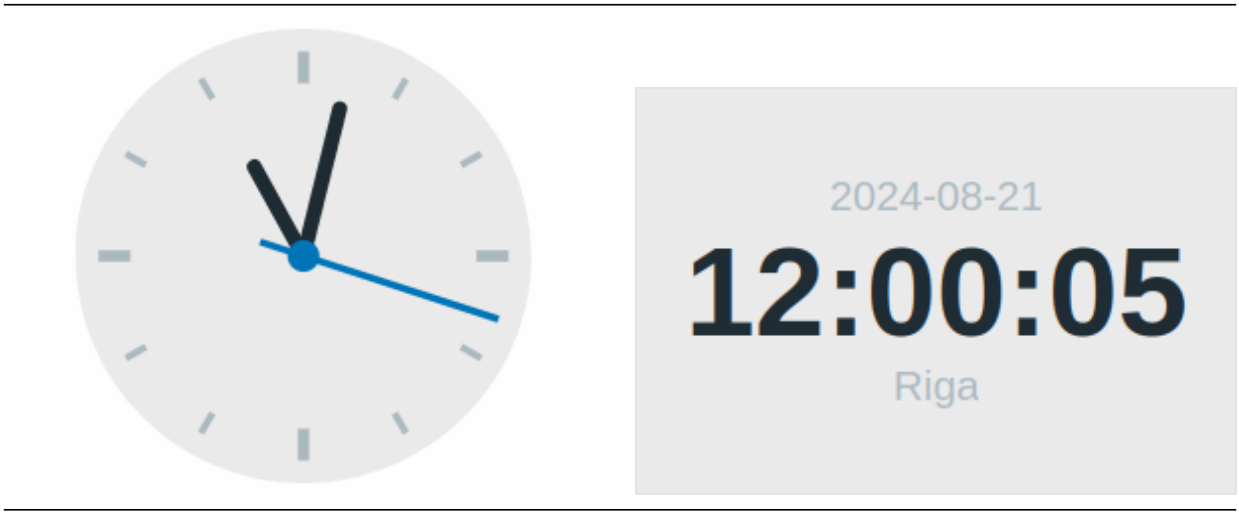
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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.

2 Uhr

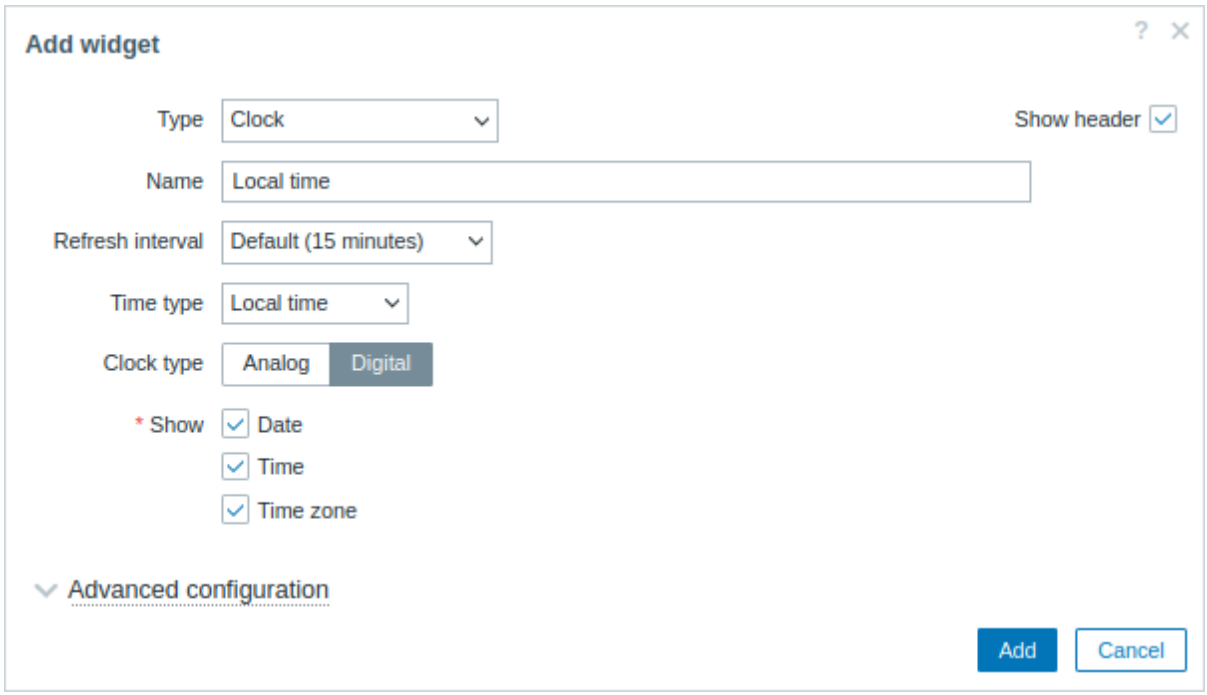
Übersicht

Das Widget *Uhr* zeigt die lokale Zeit, die Server-Zeit oder die Host-Zeit im analogen oder digitalen Format an.



Konfiguration

Um zu konfigurieren, wählen Sie *Uhr* als Typ aus:



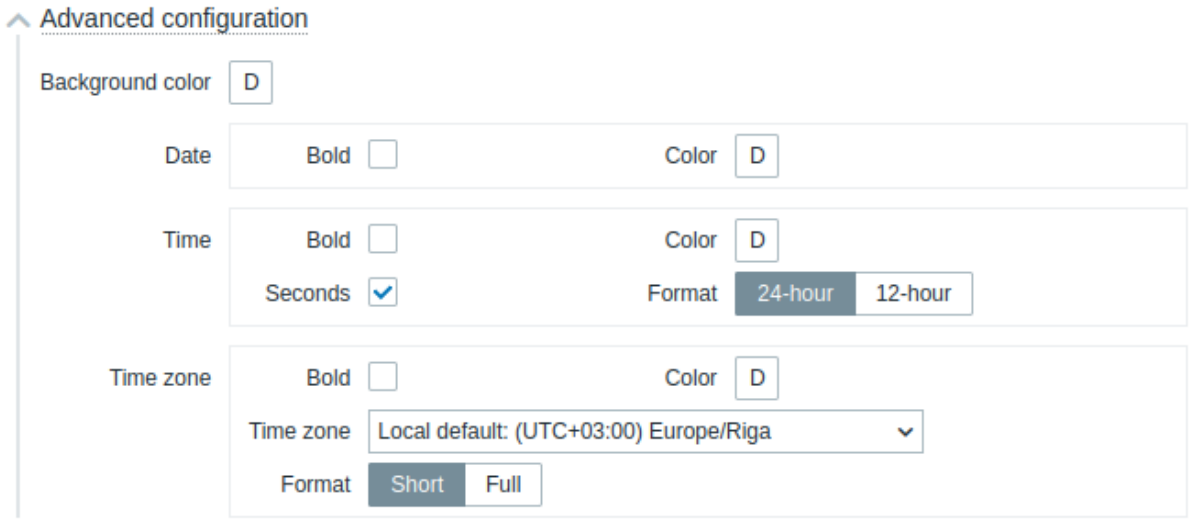
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Zeittyp	Wählen Sie die Zeitquelle aus: **Lokal** - Systemzeit des Geräts, auf dem der Browser ausgeführt wird; **Server** - Zeitzone, die global oder für den Zabbix-Benutzer festgelegt ist; **Host** - Zeit, die von einem Host-Datenpunkt abgerufen wird.
Datenpunkt	Wählen Sie den Datenpunkt für die Host-Zeit aus (verwenden Sie den Datenpunkt system.localtime[local]). Dieser Parameter ist verfügbar, wenn *Zeittyp* auf „Host-Zeit“ gesetzt ist.
Uhrtyp	Wählen Sie den Uhrtyp aus (analog oder digital).
Anzeigen	Wählen Sie eine oder mehrere Informationseinheiten (Datum, Uhrzeit, Zeitzone) aus, die in der digitalen Uhr angezeigt werden sollen. Die Größe der Einheiten passt sich automatisch an die angezeigten Einheiten, die Widget-Größe und die Anzeigeskalierung an. Dieser Parameter ist verfügbar, wenn *Uhrtyp* auf „Digital“ gesetzt ist.
Erweiterte Konfiguration	Klicken Sie auf die Bezeichnung *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration für die digitale Uhr anzuzeigen. Dieser Abschnitt ist verfügbar, wenn *Uhrtyp* auf „Digital“ gesetzt ist.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar und nur für die Elemente, die im Feld *Anzeigen* ausgewählt sind (siehe oben).

Zusätzlich ermöglicht die erweiterte Konfiguration, die Hintergrundfarbe für das gesamte Widget zu ändern.

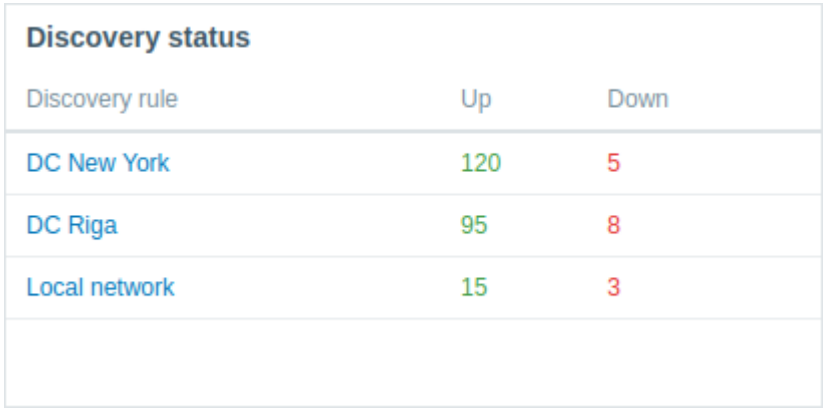


Hintergrundfarbe	Wählen Sie die Hintergrundfarbe in der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Datum	
Fett	Aktivieren Sie das Kontrollkästchen, um das Datum fett darzustellen.
Farbe	Wählen Sie die Datumsfarbe in der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Uhrzeit	
Fett	Aktivieren Sie das Kontrollkästchen, um die Uhrzeit fett darzustellen.
Farbe	Wählen Sie die Farbe der Uhrzeit in der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, 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	
Fett	Aktivieren Sie das Kontrollkästchen, um die Zeitzone fett darzustellen.
Farbe	Wählen Sie die Farbe der Zeitzone in der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Zeitzone	Wählen Sie die Zeitzone aus.
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.

3 Discovery-Status

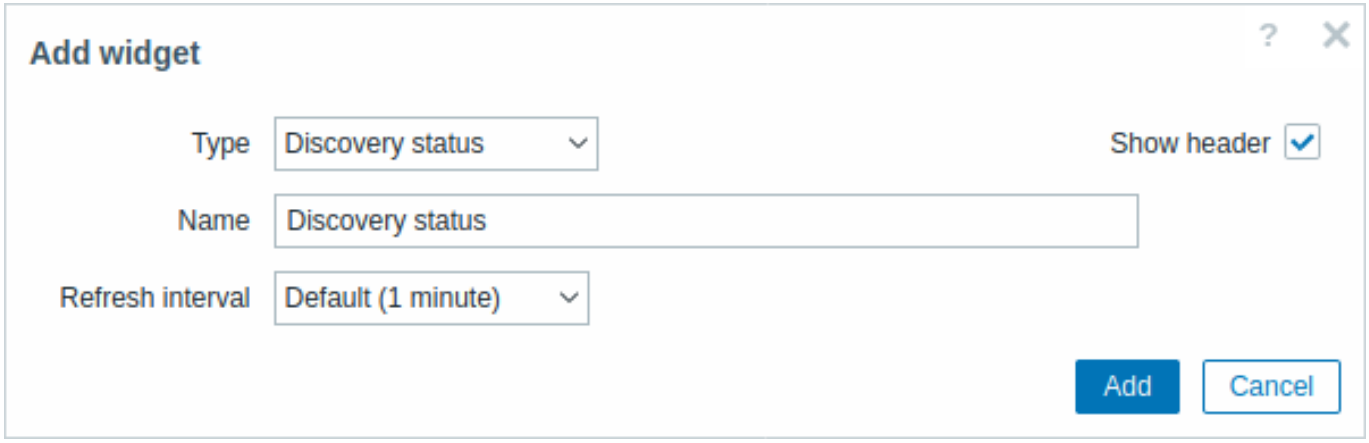
Übersicht

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

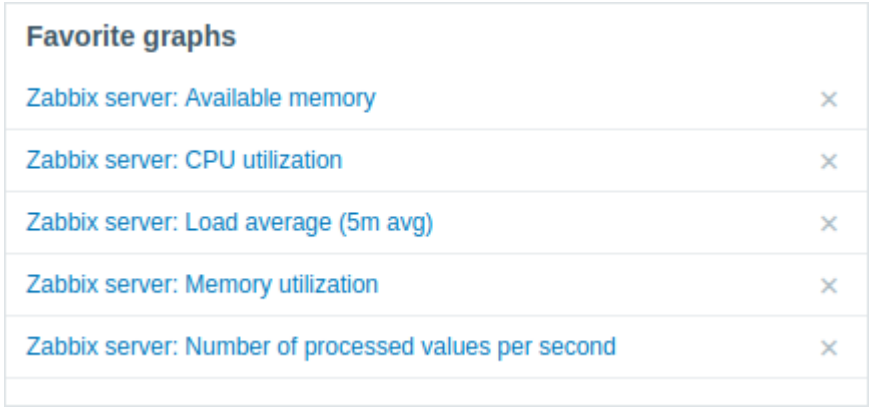
Alle Konfigurationsparameter für dieses Widget sind für alle Widgets gemeinsam.



4 Favorisierte Diagramme

Übersicht

Das Widget *Favorite graphs* zeigt eine alphabetische Liste von Verknüpfungen zu Diagrammen an, die vom aktuellen Benutzer als Favoriten markiert wurden.

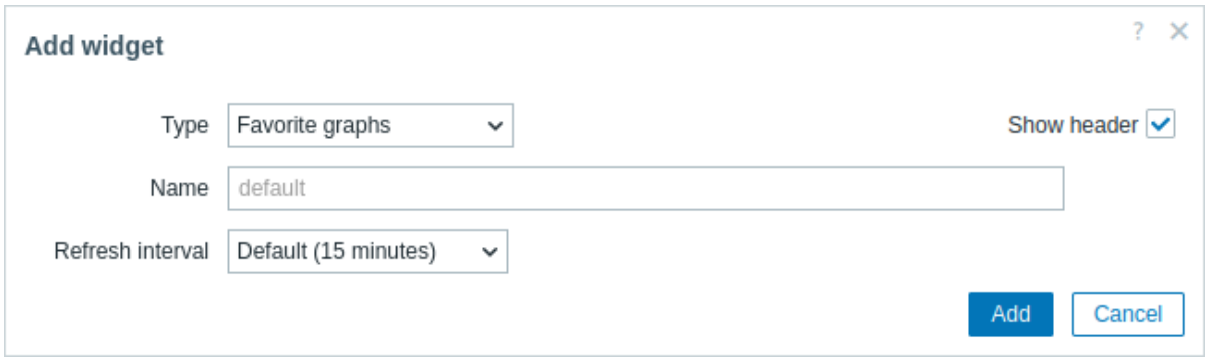


Durch Klicken auf x wird das Diagramm aus der Liste und aus den Favoriten entfernt.

Diagramme werden als Favoriten markiert, indem beim Anzeigen von Datenpunkt-Diagrammen unter *Monitoring* > *Latest data* auf die Schaltfläche  *Add to favorites* geklickt wird.

Konfiguration

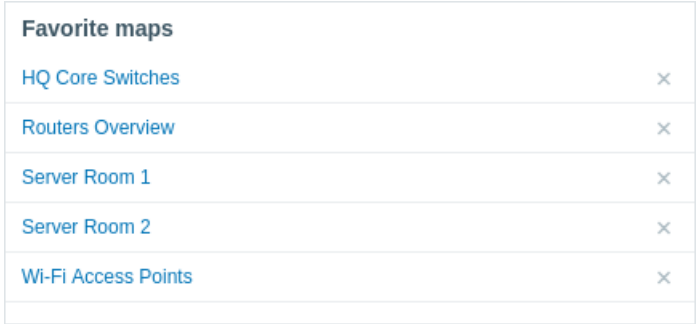
Alle Konfigurationsparameter für dieses Widget sind für alle Widgets gemeinsam.



5 Bevorzugte Karten

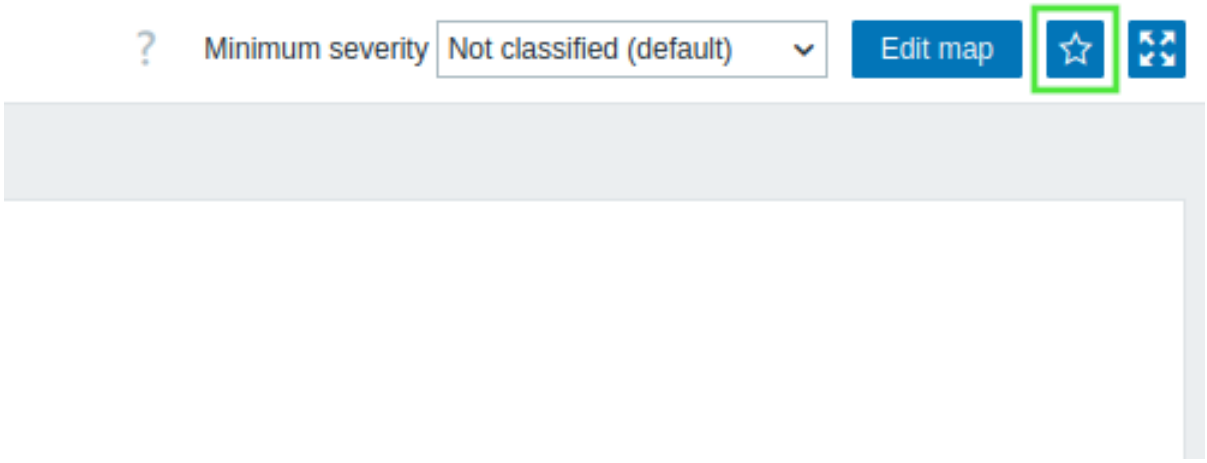
Übersicht

Das Widget *Bevorzugte Karten* zeigt eine alphabetisch sortierte Liste von Verknüpfungen zu Karten an, die vom aktuellen Benutzer als Favoriten markiert wurden.



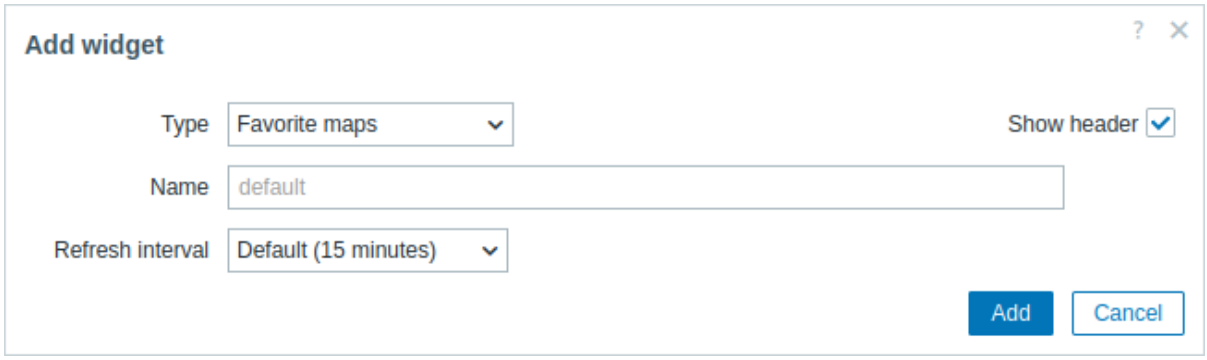
Durch Klicken auf x wird die Karte aus den Favoriten entfernt.

Karten werden als Favoriten markiert, indem beim Anzeigen von Karten in *Monitoring* > *Karten* auf die Schaltfläche  *Zu Favoriten hinzufügen* geklickt wird.



Konfiguration

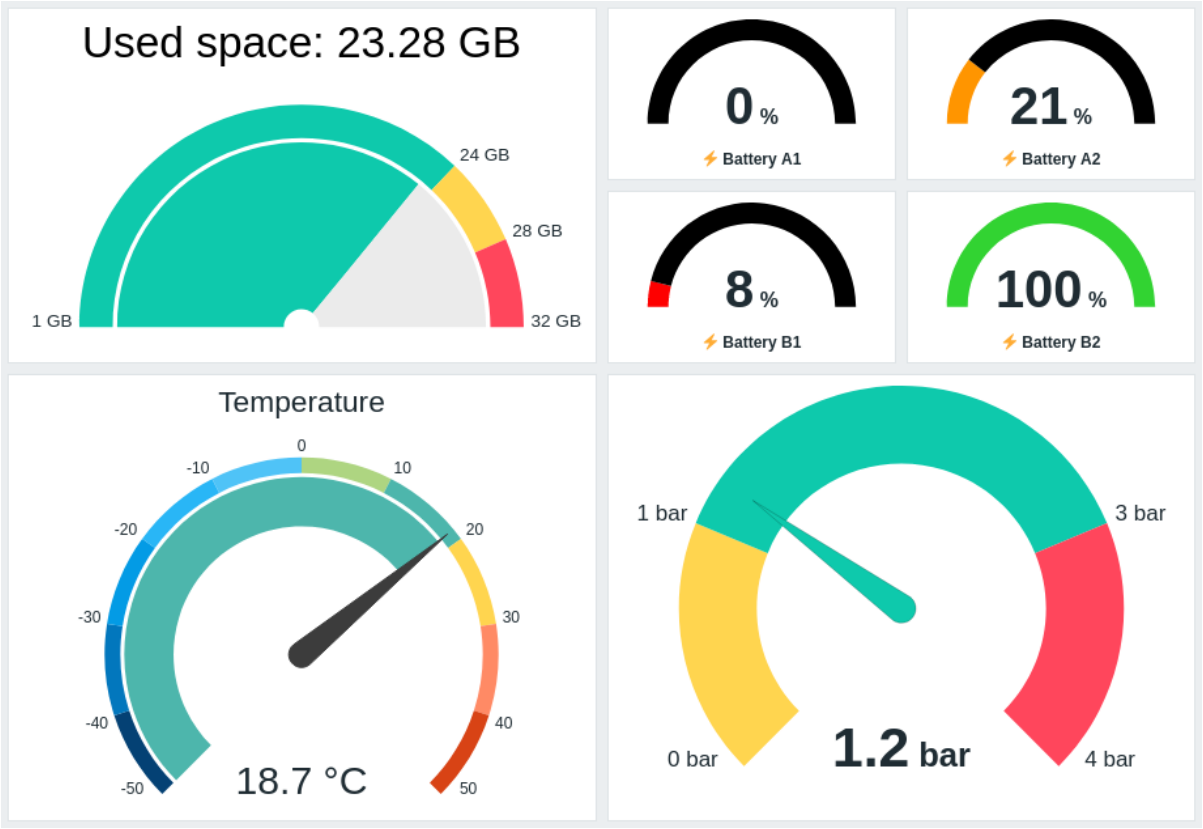
Alle Konfigurationsparameter für dieses Widget sind für alle Widgets gemeinsam.



6 Messanzeige

Übersicht

Das Widget *Gauge* zeigt den Wert eines einzelnen numerischen Datenpunkts als Messinstrument an. Es ist nützlich, um wichtige Metriken im Blick zu behalten, Schwellenwerte zu visualisieren und plötzliche Änderungen zu erkennen.



Sie können das Widget so konfigurieren, dass Folgendes angezeigt wird:

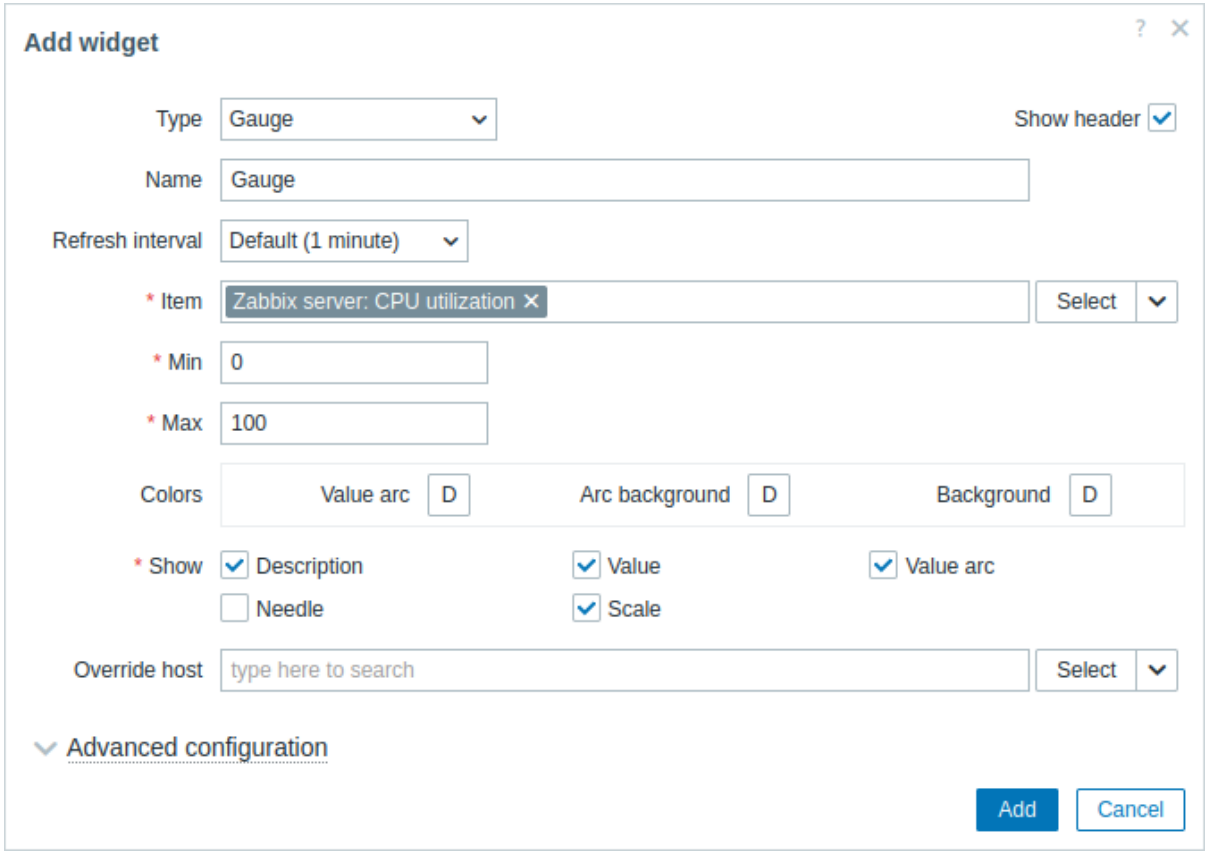
- Beschreibung des Datenpunkts (Used space: {ITEM.LASTVALUE}, Temperature)
- Datenpunktwert und Einheiten (21 %, 18.7 °C, 1.2 bar)
- Messskala (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Messbogen (Wertebogen des Messinstruments und Schwellenwertbogen des Messinstruments)
- Messnadel

Wenn Sie auf das Widget klicken, wird für den Datenpunkt ein einfaches Diagramm geöffnet.

Die im Widget *Gauge* angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem Sie im Widget-Menü die Option *Download image* auswählen.

Konfiguration

Um die Konfiguration vorzunehmen, wählen Sie *Gauge* als Typ aus:



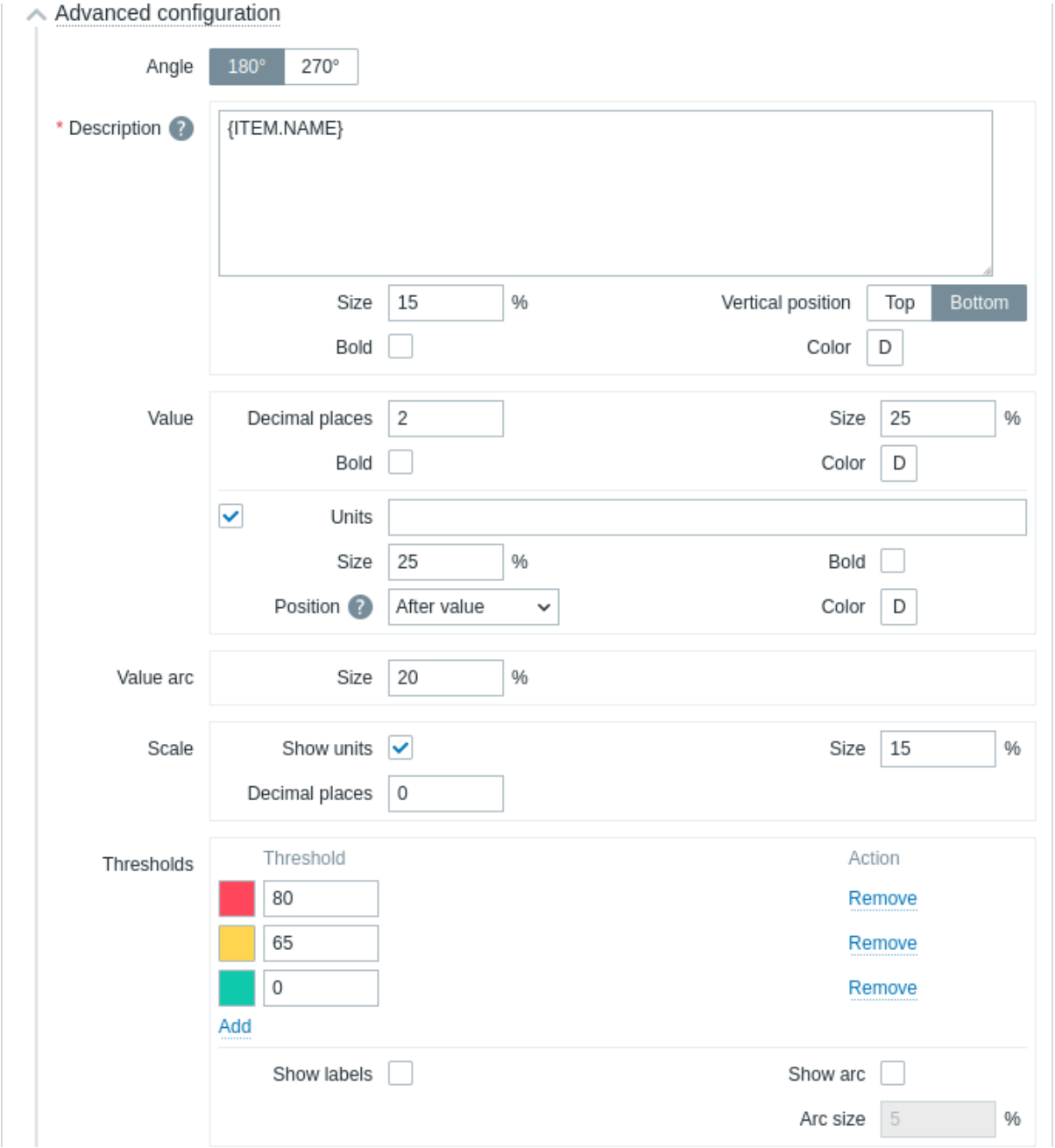
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Item	Wählen Sie den Datenpunkt aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Datenpunkte auswählen. Dieses Feld unterstützt Autovervollständigung. Wenn Sie also beginnen, den Namen eines Datenpunkts einzugeben, wird eine Dropdown-Liste mit passenden Datenpunkten angezeigt. Beachten Sie, dass Sie nur Datenpunkte auswählen können, die numerische Daten zurückgeben (außer 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.
Colors	Wählen Sie eine Farbe im Farbwähler aus: **Value arc** - wählen Sie die Farbe des Wertbogens der Anzeige aus; **Arc background** - wählen Sie die Hintergrundfarbe des Wertbogens und des Schwellenwertbogens der Anzeige aus; **Background** - wählen Sie die Hintergrundfarbe des Widgets aus. „D“ steht für die Standardfarbe, die vom Frontend-Theme abhängt. Wenn *Thresholds* festgelegt sind, hängt die Standardfarbe für **Value arc** von der Schwellenwertfarbe ab. Um zur Standardfarbe zurückzukehren, klicken Sie im Farbwähler auf die Schaltfläche *Use default*.
Show	Aktivieren Sie das Kontrollkästchen, um das jeweilige Element der Anzeige anzuzeigen – Beschreibung, Wert, Wertbogen, Zeiger, Skala (den 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.

Override host	Wählen Sie ein kompatibles Widget oder den Dashboard-Host-Selektor 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 *Advanced configuration*, um die Optionen der erweiterten Konfiguration anzuzeigen. Hier können Sie auch die im Feld *Show* ausgewählten Elemente der Anzeige anpassen.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar:

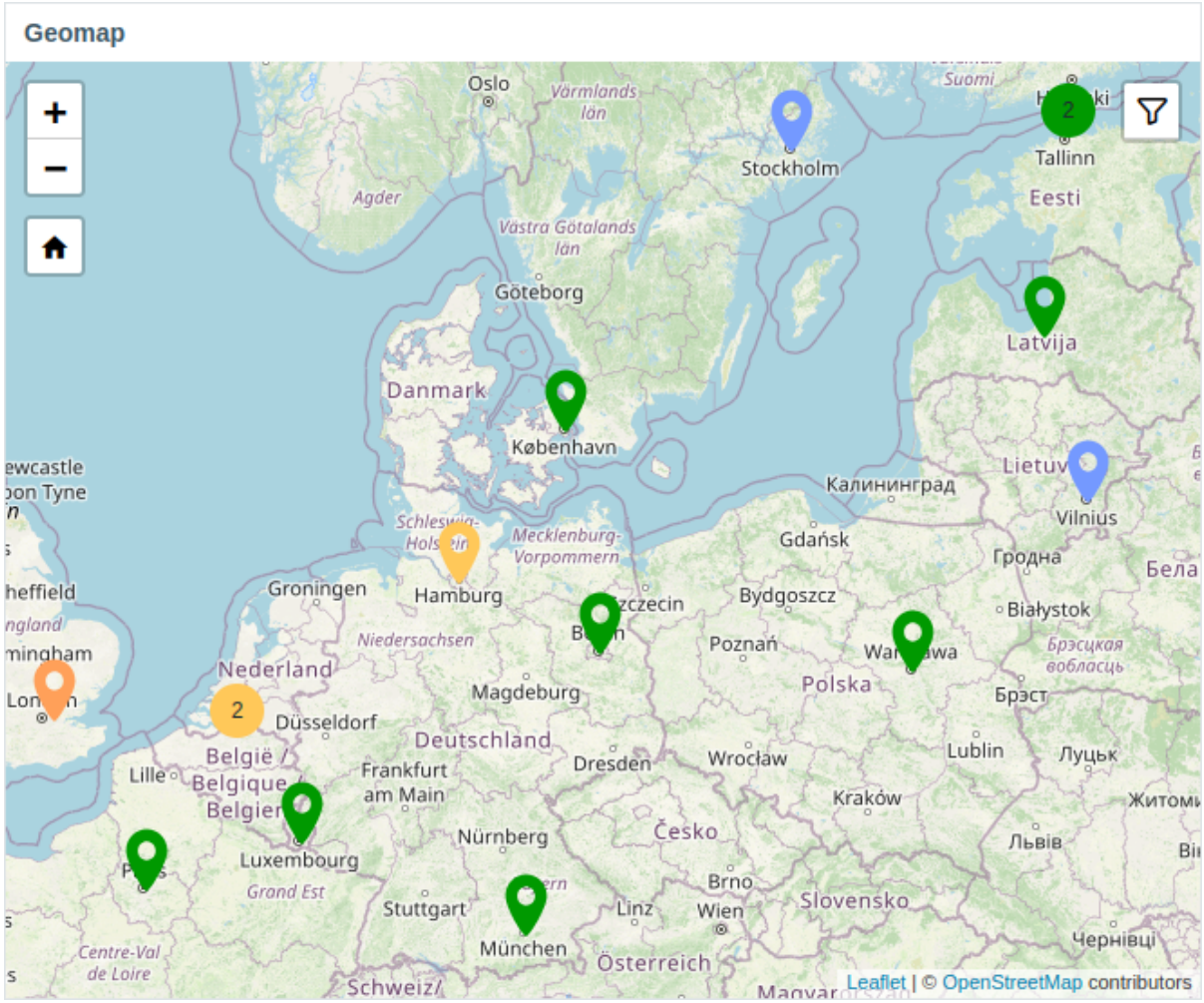


Winkel	Wählen Sie den Winkel der Anzeige (180° oder 270°).
Beschreibung	
Beschreibung	Geben Sie die Beschreibung des Datenpunkts ein. Diese Beschreibung kann den Standardnamen des Datenpunkts überschreiben. Mehrzeilige Beschreibungen werden unterstützt. Eine Kombination aus Text und unterstützten Makros 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.

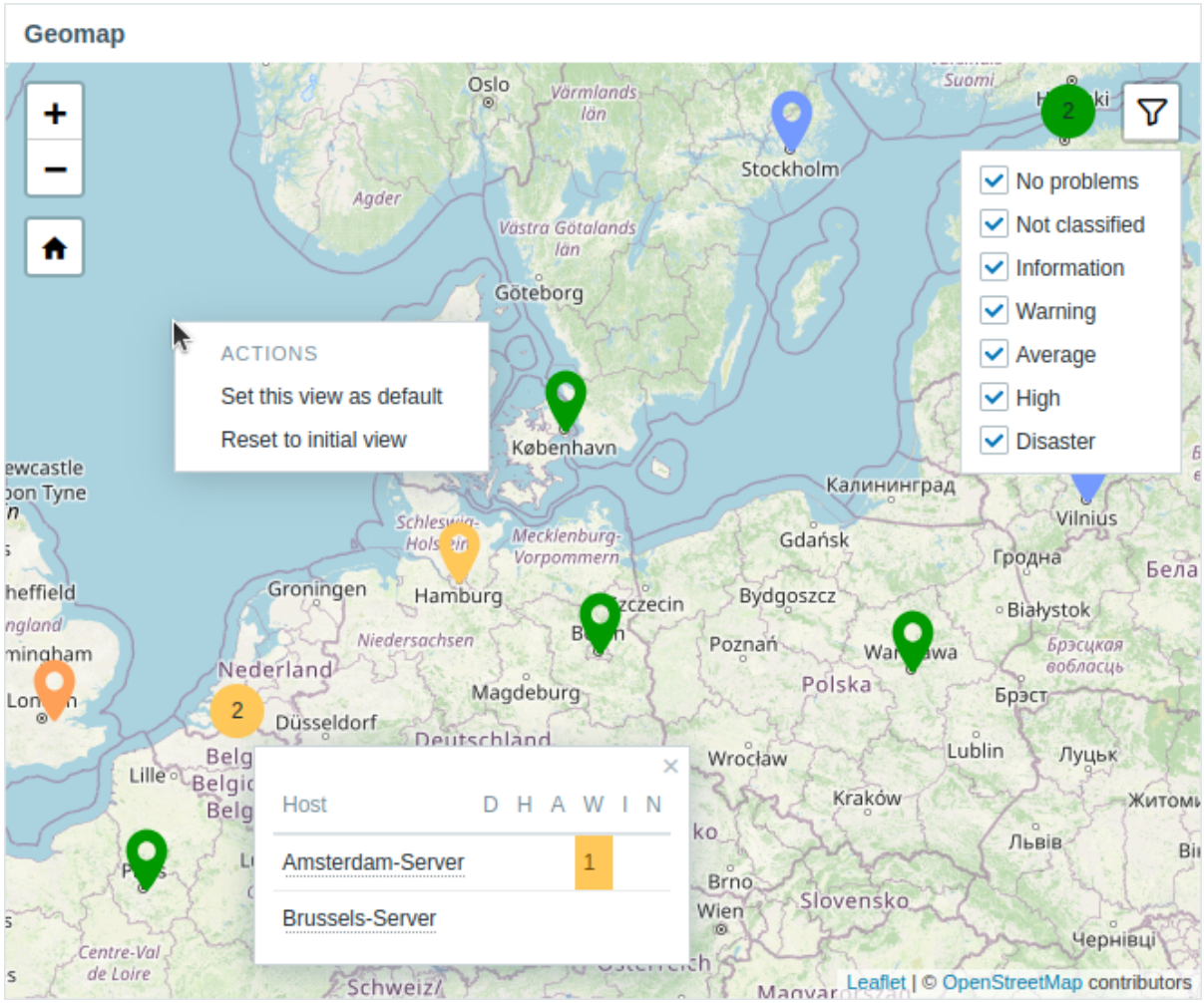


Note:
Zabbix bietet mehrere vordefinierte Anbieter für Kartenkacheln und ermöglicht es Ihnen, benutzerdefinierte Anbieter unter *Administration* > *General* > *Geographical maps* hinzuzufügen.

Standardmäßig zeigt das Widget alle aktivierten Hosts an, die in ihrem Inventar gültige Koordinaten haben (Breitengrad -90 bis 90, Längengrad -180 bis 180).

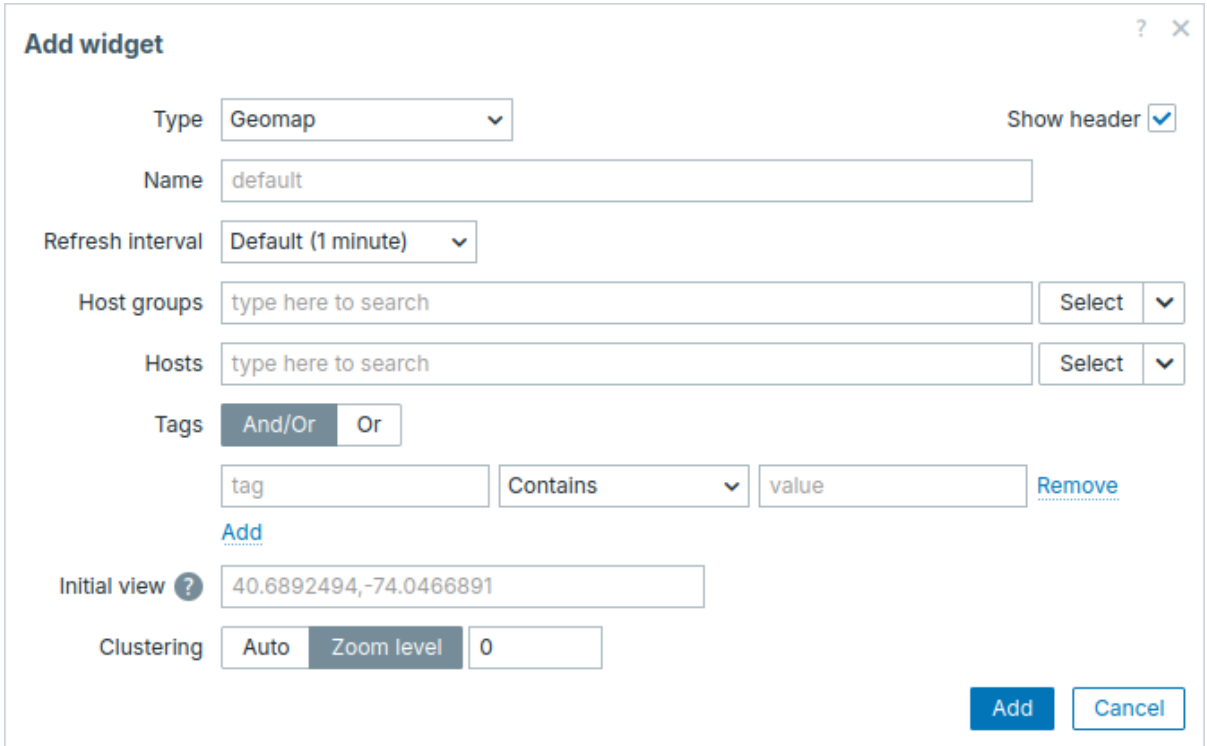
Verwenden Sie die folgenden Optionen, um mit dem Widget zu interagieren:

- 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.



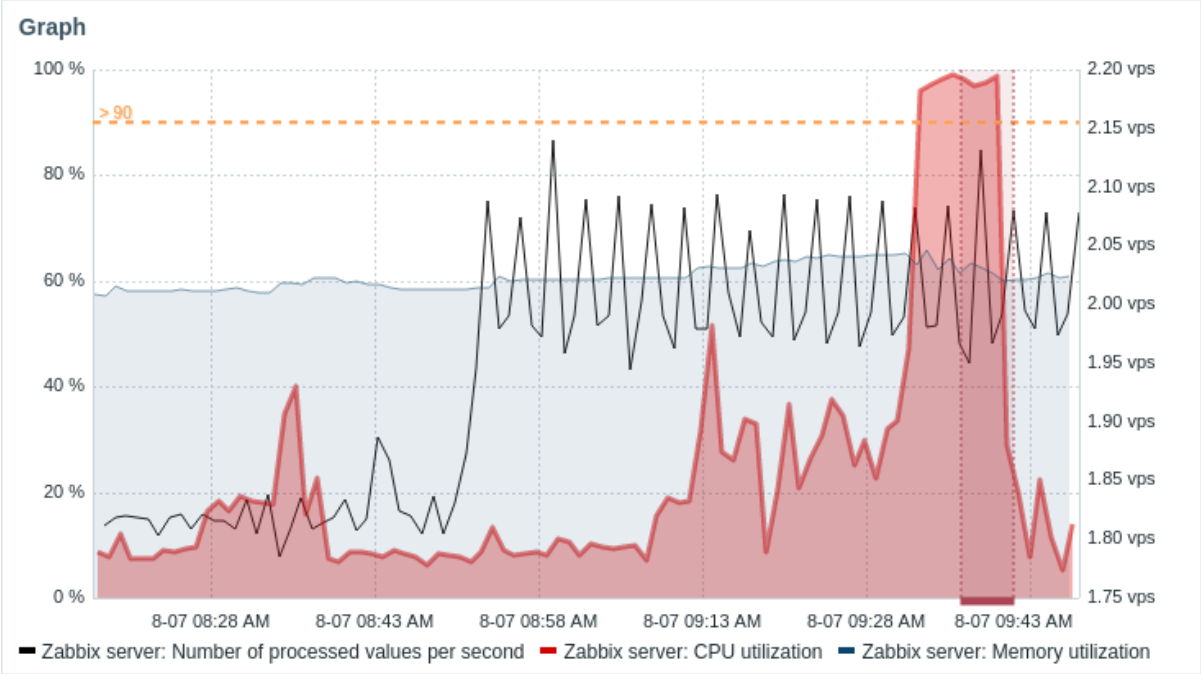
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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 verfügt über eine Autovervollständigung. 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 in 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 verfügt über eine Autovervollständigung. 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 in 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 in 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 sollen. 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: **Automatisch** - 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 auf keiner Zoomstufe zusammengefasst; bei 15 werden Markierungen vom maximalen Herauszoomen bis zur Zoomstufe 15 zusammengefasst; bei 30 werden Markierungen auf allen Zoomstufen zusammengefasst. Beachten Sie, dass die maximale Hineinzoomstufe vom Kartenkachelanbieter bestimmt wird.

8 Diagramm

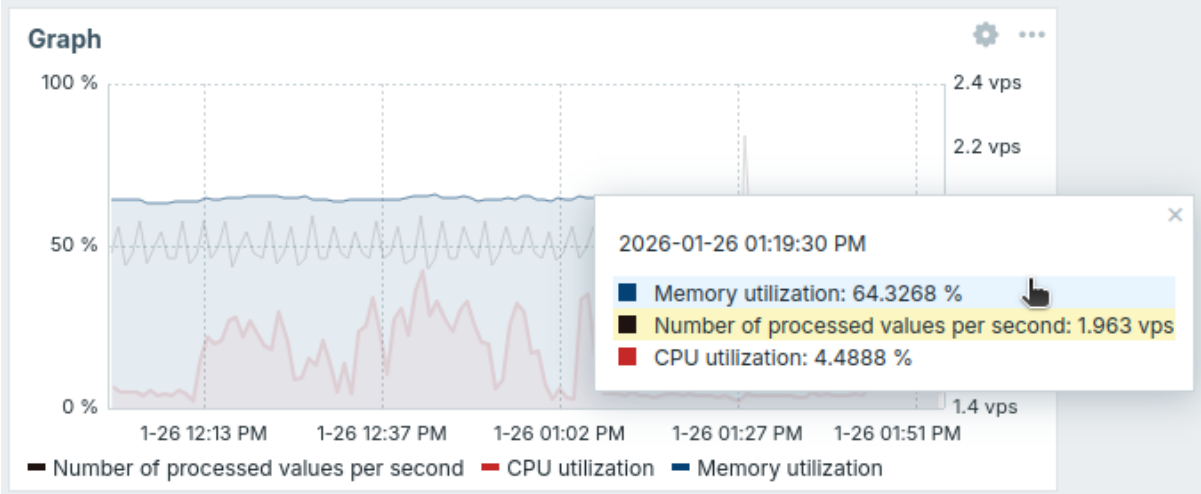
Übersicht

Das Widget *Graph* zeigt numerische Datenpunkt-Daten als vektorbasiertes Diagramm an. Es kann Ihnen helfen, Trends zu verfolgen, Probleme zu erkennen und Werte im Zeitverlauf sowie Host-übergreifend zu vergleichen.

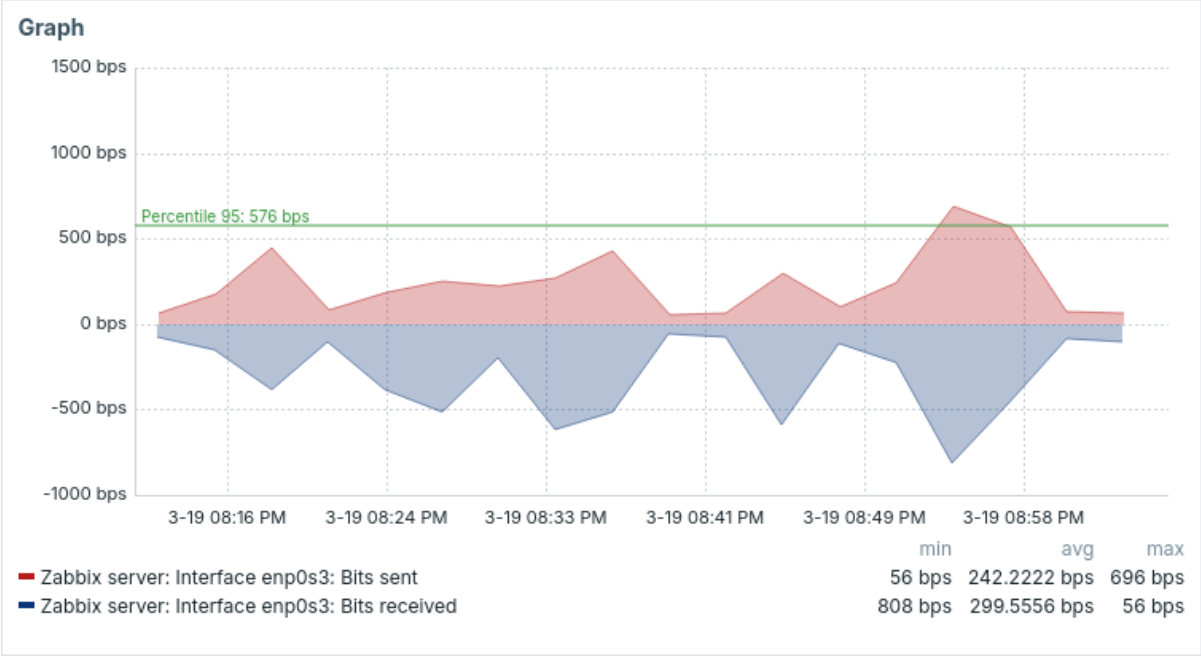


Wenn Sie den Mauszeiger über das Diagramm bewegen, wird ein Tooltip mit einer nach Wert absteigend sortierten Liste von Datenpunkten angezeigt.

Wenn Sie den Mauszeiger über einen Datenpunkt im Tooltip bewegen, wird dessen Diagramm hervorgehoben (die anderen werden abgeblendet); durch Auswahl des Datenpunkts werden seine Daten an andere Widgets übertragen.



Das Diagramm kann mit invertierten Y-Achsenwerten angezeigt werden; dies wirkt sich nur auf die visuelle Darstellung aus und verändert die ursprünglichen Werte nicht.

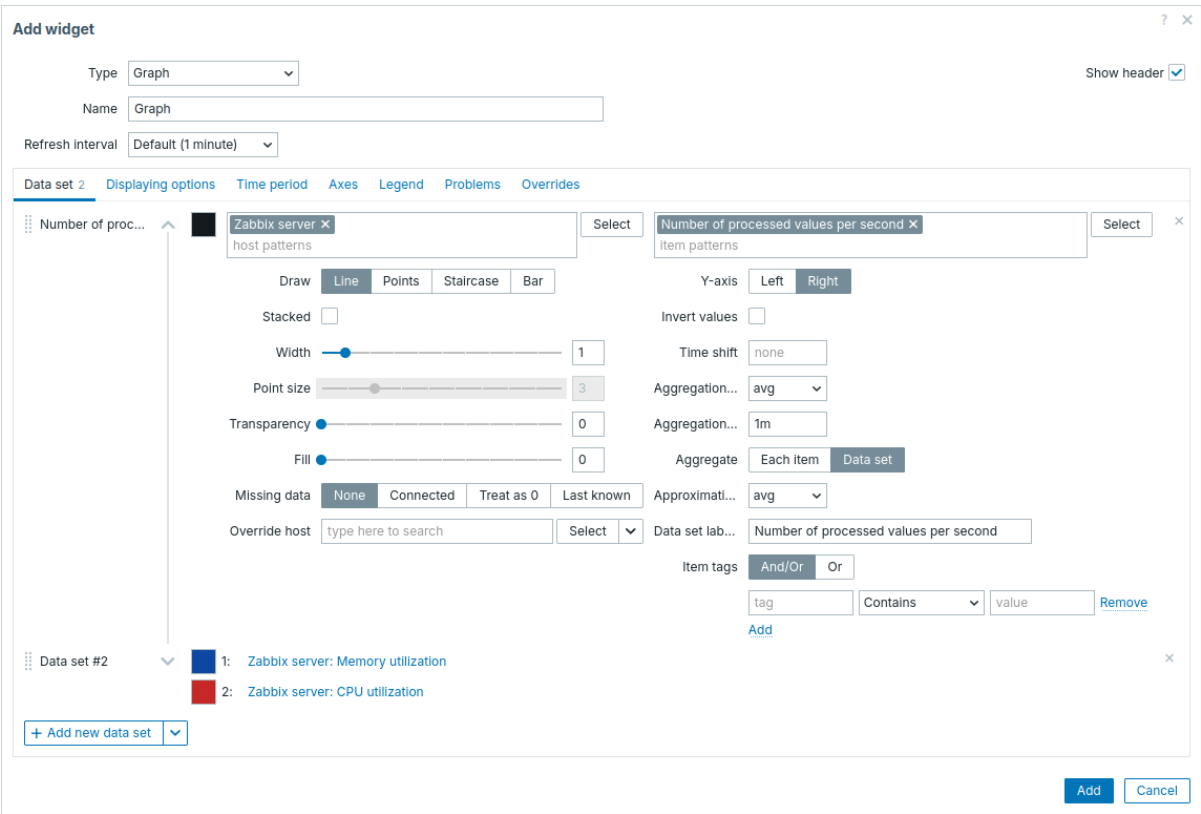


Das ältere Widget Graph (classic) ist ebenfalls verfügbar.

Die im Widget *Graph* angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem Sie im Widget-Menü die Option *Bild herunterladen* auswählen.

Konfiguration

Wählen Sie zur Konfiguration *Graph* als Typ aus:



Datensatz

Die Registerkarte **Datensatz** ermöglicht die Auswahl von Daten für das Diagramm durch Hinzufügen von Datensätzen. Es können zwei Arten von Datensätzen hinzugefügt werden:

- *Datenpunkt-Muster* - Daten von übereinstimmenden Datenpunkten werden angezeigt. Sie können eine einzelne Grundfarbe auswählen oder eine Palettenzeile wählen, um jedem übereinstimmenden Datenpunkt eine eigene Farbe zuzuweisen.
- *Datenpunkt-Liste* - Daten von ausgewählten Datenpunkten werden angezeigt. Sie können die Farbe jedes Datenpunkts einzeln im Farbwähler festlegen.

Standardmäßig wird ein Datensatz vom Typ *Datenpunkt-Muster* hinzugefügt.

Datensatz	Für den Datensatz **Datenpunkt-Muster**: Wählen Sie Host- und Datenpunkt-Muster aus oder geben Sie sie ein; Daten von Datenpunkten, die diesen Mustern entsprechen, werden im Diagramm angezeigt; es können bis zu 50 Datenpunkte angezeigt werden. Für die Auswahl können Platzhaltermuster verwendet werden (zum Beispiel liefert * Ergebnisse zurück, 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 nicht möglich, zum Beispiel einen Datenpunkt mit dem Namen *item** einzeln hinzuzufügen, wenn es andere passende Datenpunkte gibt (zum Beispiel *item2*, *item3*). Die Angabe von Host- und Datenpunkt-Mustern ist für Datensätze vom Typ "Datenpunkt-Muster" obligatorisch. Siehe auch: Details zur Datensatzkonfiguration. Für den Datensatz **Datenpunkt-Liste**: 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 "Datenpunkt-Liste" obligatorisch. Siehe auch: Details zur Datensatzkonfiguration. Beachten Sie, dass nur numerische Datenpunkttypen zulässig sind. Wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird, ist der Parameter zur Angabe von Host-Mustern nicht verfügbar, und der Parameter zur Angabe einer Datenpunkt-Liste erlaubt nur die Auswahl der in der Vorlage konfigurierten Datenpunkte.
Zeichnen	Wählen Sie den Zeichnungstyp der Metrik. Mögliche Zeichnungstypen: *Linie* (standardmäßig eingestellt), *Punkte*, *Treppenlinie* und *Balken*. Beachten Sie, dass bei nur einem Datenpunkt in einem Linien-/Treppenliniendiagramm dieser unabhängig vom Zeichnungstyp als Punkt dargestellt wird. Die Punktgröße wird aus der Linienbreite berechnet, kann jedoch nicht kleiner als 3 Pixel sein, auch wenn die Linienbreite geringer ist.
Gestapelt	Aktivieren Sie das Kontrollkästchen, um Daten gestapelt anzuzeigen (gefüllte Bereiche werden dargestellt). Diese Option ist deaktiviert, wenn der Zeichnungstyp *Punkte* ausgewählt ist.
Breite	Legen Sie die Linienbreite fest. Diese Option ist verfügbar, wenn der Zeichnungstyp *Linie* oder *Treppenlinie* ausgewählt ist.
Punktgröße	Legen Sie die Punktgröße fest. Diese Option ist verfügbar, wenn der Zeichnungstyp *Punkte* ausgewählt ist.
Transparenz	Legen Sie den Transparenzgrad fest.
Füllung	Legen Sie den Füllgrad fest. Diese Option ist verfügbar, wenn der Zeichnungstyp *Linie* oder *Treppenlinie* ausgewählt ist.
Fehlende Daten	Wählen Sie die Option zur Anzeige fehlender Daten: **Keine** - die Lücke bleibt leer; **Verbunden** - zwei Randwerte werden verbunden; **Als 0 behandeln** - die fehlenden Daten werden als Werte 0 angezeigt; **Letzter bekannter Wert** - die fehlenden Daten werden mit demselben Wert wie der letzte bekannte Wert angezeigt; nicht anwendbar für die Zeichnungstypen *Punkte* und *Balken*.
Host überschreiben	Wählen Sie ein kompatibles Widget oder den Dashboard-Host-Selektor 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, auf der die Y-Achse angezeigt wird.

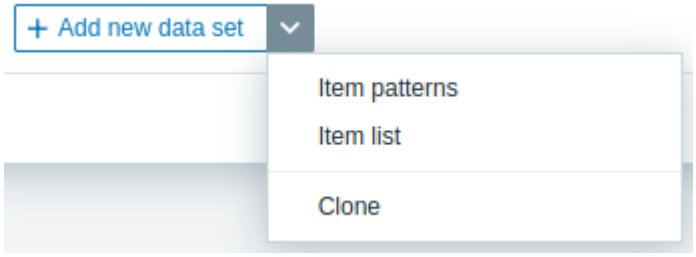
Werte invertieren	Aktivieren Sie das Kontrollkästchen, um alle Y-Achsenwerte vor der Anzeige im Diagramm mit -1 zu multiplizieren. Dies betrifft nur die visuelle Darstellung; die Originaldaten bleiben unverändert. Diese Option wirkt sich nicht auf den Tooltip, die Legende oder die Perzentillinie aus. Wenn für die Achse die Option *Einfache Auslöser* ausgewählt ist, werden die Auslöserschwellen invertiert angezeigt.
Zeitverschiebung	Geben Sie bei Bedarf eine Zeitverschiebung an. Sie können in diesem Feld Zeitsuffixe verwenden. Negative Werte sind zulässig.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion 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; **none** - alle Werte anzeigen (keine Aggregation). Die Aggregation ermöglicht die Anzeige eines aggregierten Werts für das gewählte Intervall (5 Minuten, eine Stunde, ein Tag) anstelle aller Werte. Siehe auch: Aggregation in Diagrammen.
Aggregationsintervall	Geben Sie das Intervall für die Aggregation von Werten an. Sie können in diesem Feld Zeitsuffixe verwenden. Ein numerischer Wert ohne Suffix wird als Sekunden interpretiert. Beachten Sie, dass bei einer Widget-Konfiguration zur Anzeige historischer Daten auf Basis von Trends (*Auswahl historischer Daten* ist auf *Trends* oder *Auto* 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 kann die Verwendung eines Aggregationsintervalls, das kein Vielfaches von 1 Stunde ist (z. B. 100s, 7min, 15min, 90min usw.), zu Ergebnissen führen, die schwer zu interpretieren sind.
Aggregieren	Geben Sie an, ob aggregiert werden soll: **Jeder Datenpunkt** - jeder Datenpunkt im Datensatz wird aggregiert und separat angezeigt; **Datensatz** - alle Datenpunkte des Datensatzes werden aggregiert und als ein Wert angezeigt.
Approximation	Geben Sie an, welcher Wert angezeigt werden soll, wenn pro vertikalem Diagramm-Pixel mehr als ein Wert vorhanden ist: **all** - 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 häufigem Aktualisierungsintervall angezeigt wird (zum Beispiel 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, auch diejenigen mit angegebener *Datensatzbezeichnung*. Wenn keine Bezeichnung angegeben ist, wird der Datensatz entsprechend seiner Nummer automatisch bezeichnet (z. B. ”Datensatz #2”, ”Datensatz #3” usw.). Die Nummerierung der Datensätze wird nach dem Umordnen/Ziehen von Datensätzen neu berechnet. Zu lange Datensatzbezeichnungen werden gekürzt, damit sie in den verfügbaren Platz passen (z. B. ”Anzahl der Pro...”).

Datenpunkt-Tags	Geben Sie Tags an, um die im Widget angezeigten Datenpunkte 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. 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.

Details zur Konfiguration von Datensätzen

Vorhandene Datensätze werden in einer Liste angezeigt. Sie können:

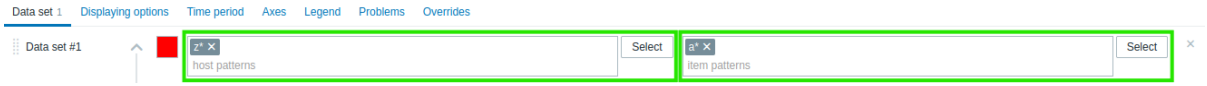
- Auf das Verschiebesymbol klicken und einen Datensatz an eine neue Position in der Liste ziehen.
- Auf das Erweitern-Symbol klicken, um die Details des Datensatzes einzublenden. Wenn es erweitert ist, wird dieses Symbol zu einem Reduzieren-Symbol .
- Auf das Farbsymbol klicken, um die Auswahl zu öffnen. Sie können einen Hex-Code eingeben, ein einfarbiges Farbfeld auswählen oder zur Registerkarte *Palette* wechseln und eine Zeile vordefinierter Farben auswählen. Die gewählte Farbe wird direkt auf Datensätze vom Typ *Item list* angewendet oder als Basis für erzeugte Farbabstufungen in *Item patterns* verwendet. Verwenden Sie die Tabulatortaste, um zwischen den Steuerelementen des Dialogs 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 Host- und Datenpunkt-Muster ausgewählt werden können. Wenn Sie auf das nach unten zeigende Symbol neben der Schaltfläche *Add new data set* klicken, wird ein Dropdown-Menü angezeigt, über das Sie einen neuen Datensatz vom Typ *Item patterns* oder *Item list* hinzufügen oder den aktuell geöffneten Datensatz *Clone* können. Wenn alle Datensätze reduziert sind, ist die Option *Clone* nicht verfügbar.



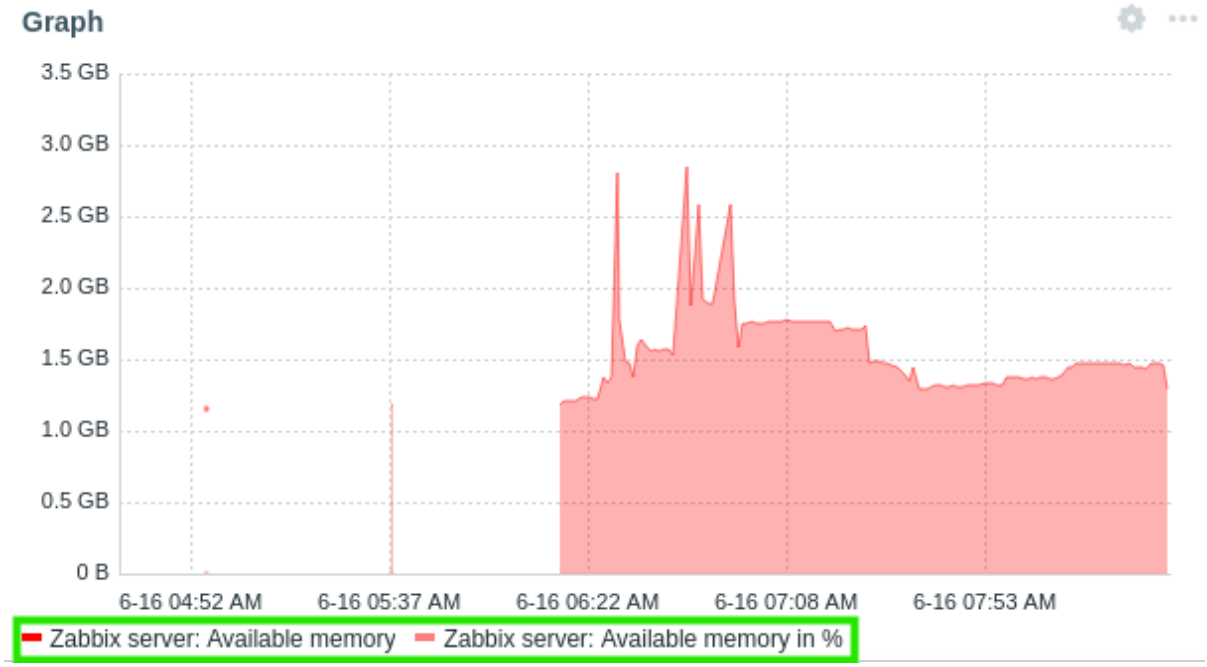
Der Datensatz **Datenpunkt-Muster** enthält die Felder *Host-Muster* und *Datenpunkt-Muster*, die beide vollständige Namen oder Muster mit einem Platzhaltersymbol (*) erkennen. Diese Funktion ermöglicht es Ihnen, alle Host-Namen und Datenpunkt-Namen auszuwählen, die das gewählte Muster enthalten. Während 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 %**.

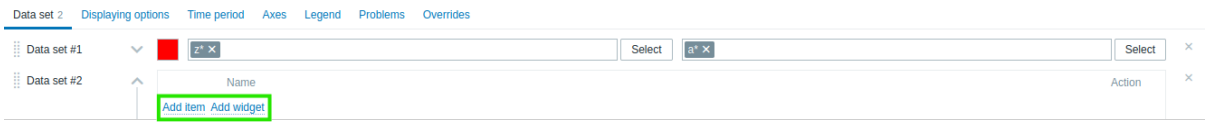
Nach dem Drücken von *Enter* wird das Muster übernommen und als **a*** angezeigt.



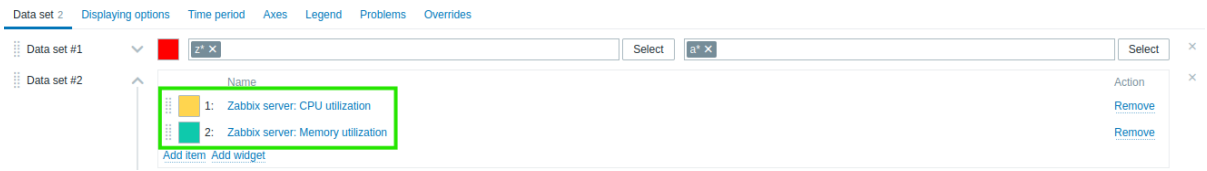
Das Diagramm zeigt dann alle Datenpunkte an, die zu den ausgewählten Host-Namen gehören.



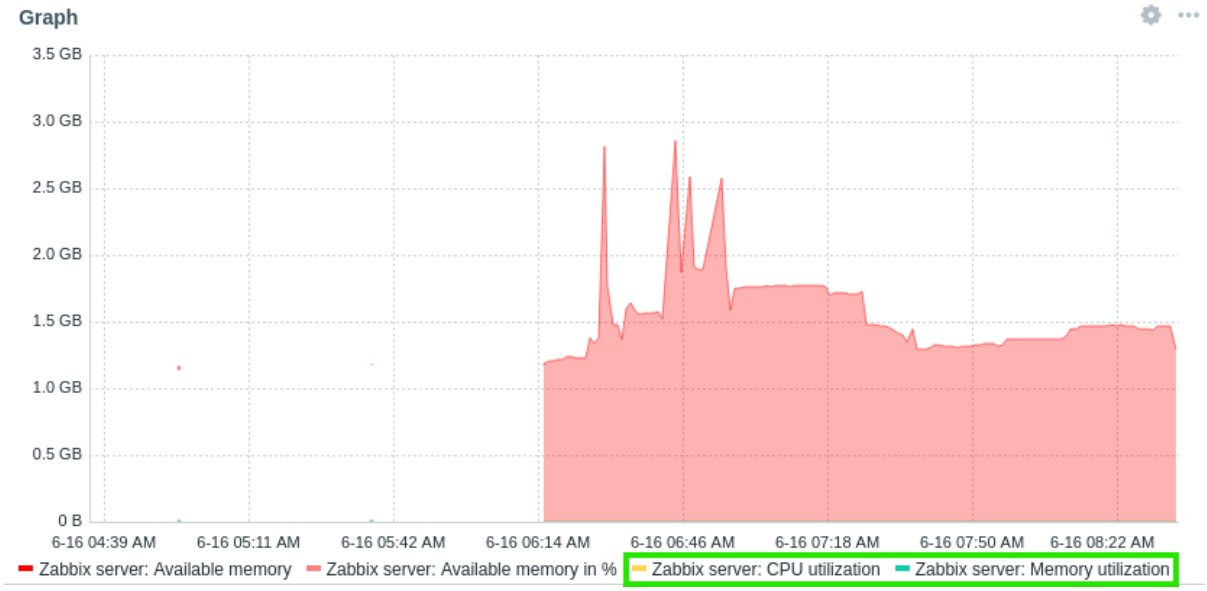
Der Datensatz **Datenpunktliste** enthält die Schaltfläche *Datenpunkt hinzufügen*, mit der Sie Datenpunkte hinzufügen können, die im Diagramm angezeigt werden sollen. Sie können auch kompatible Widgets als Datenquelle für Datenpunkte hinzufügen, indem Sie auf die Schaltfläche *Widget hinzufügen* klicken.



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.

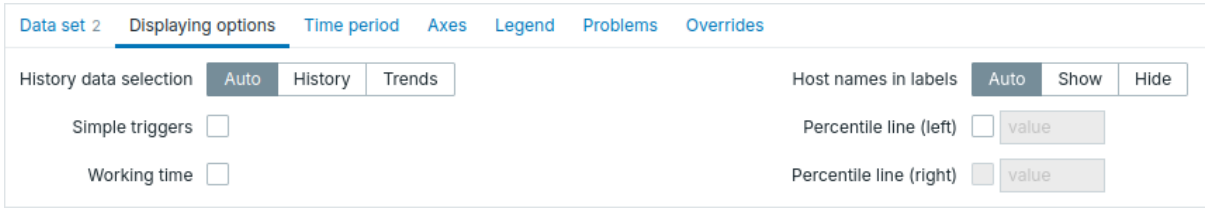


Nachdem Sie einen oder mehrere Datenpunkte ausgewählt haben, werden diese in der Datenpunktliste des Datensatzes und im Diagramm angezeigt.



Anzeigeoptionen

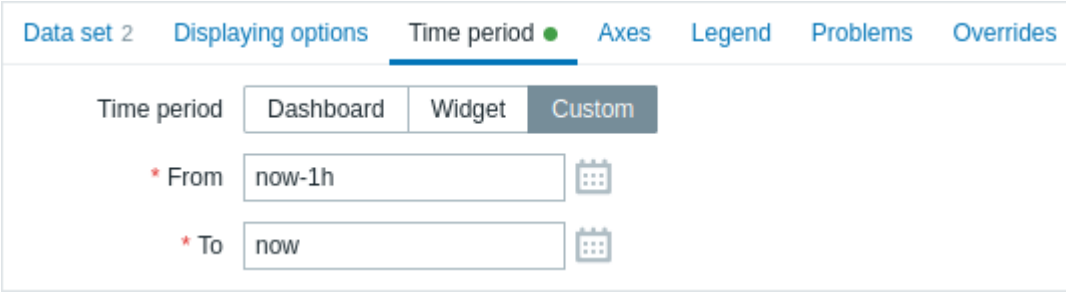
Die Registerkarte **Anzeigeoptionen** ermöglicht die Festlegung der Auswahl von Verlaufsdaten:



Auswahl der Verlaufsdaten	Legen Sie die Quelle der Diagrammdaten fest: **Auto** - Daten werden gemäß dem klassischen Diagramm-Algorithmus bezogen (Standard); **Verlauf** - Daten aus dem Verlauf; **Trends** - Daten aus Trends.
Einfache Auslöser	Aktivieren Sie das Kontrollkästchen, um die Auslöserschwellen für einfache Auslöser anzuzeigen. Die Schwellen werden als gestrichelte Linien in der Farbe des Auslöserschweregrads 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, um sichtbar zu sein.
Arbeitszeit	Aktivieren Sie das Kontrollkästchen, um die Arbeitszeit im Diagramm anzuzeigen. Die Arbeitszeit (Arbeitstage) wird in Diagrammen als weißer Hintergrund dargestellt, während die Nicht-Arbeitszeit grau dargestellt wird (mit dem Standard-Frontend-Theme *Original blue*).
Host-Namen in Beschriftungen	Wählen Sie, 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). **Anzeigen** - Host-Namen werden angezeigt. **Ausblenden** - Host-Namen werden ausgeblendet. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
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 ein 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 ein 95%-Perzentil festgelegt ist, befindet sich die Perzentillinie auf der Höhe, unter der 95 Prozent der Werte liegen.

Zeitraum

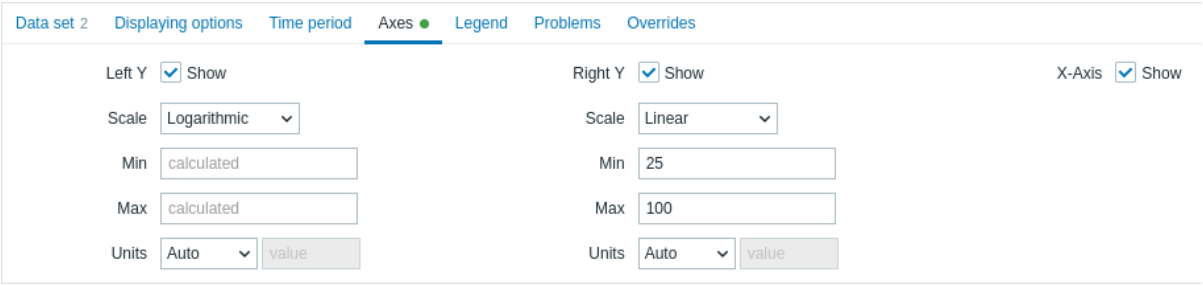
Die Registerkarte **Zeitraum** ermöglicht es, einen Zeitraum festzulegen, für den Daten im Diagramm angezeigt werden:



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 festgelegte 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.

Achsen

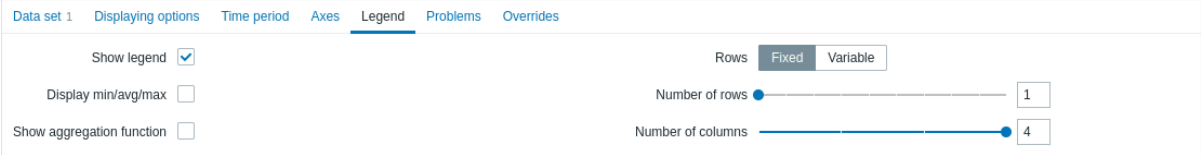
Die Registerkarte **Achsen** ermöglicht es, die Anzeige der Achsen anzupassen:



Linke Y-Achse	Aktivieren Sie dieses Kontrollkästchen, um die linke Y-Achse sichtbar zu machen. Das Kontrollkästchen kann deaktiviert sein, wenn es entweder in der Registerkarte *Datensatz* oder in der Registerkarte *Überschreibungen* nicht ausgewählt ist.
Rechte Y-Achse	Aktivieren Sie dieses Kontrollkästchen, um die rechte Y-Achse sichtbar zu machen. Das Kontrollkästchen kann deaktiviert sein, wenn es entweder in der Registerkarte *Datensatz* oder in der Registerkarte *Überschreibungen* nicht ausgewählt ist.
X-Achse	Deaktivieren Sie dieses Kontrollkästchen, um die X-Achse auszublenden (standardmäßig aktiviert).
Skalierung	Wählen Sie die Skalierung für die Achsenwerte des Diagramms aus der Dropdown-Liste: **Linear** - die Achsenwerte steigen um einen festen Betrag (z. B. 10, 20, 30); geeignet für Daten, die sich gleichmäßig ändern oder einen kleinen bis mittleren Bereich abdecken; **Logarithmisch** - die Achsenwerte steigen exponentiell an (z. B. 10, 100, 1000); geeignet für Daten, die sich schnell ändern oder einen großen Bereich abdecken.
Min	Legen Sie den Mindestwert der entsprechenden Achse fest. Der minimale sichtbare Bereichswert der Y-Achse wird angegeben.
Max	Legen Sie den Höchstwert der entsprechenden Achse fest. Der maximale sichtbare Bereichswert der Y-Achse wird angegeben.
Einheiten	Wählen Sie die Einheit für die Achsenwerte des Diagramms aus der Dropdown-Liste: **Auto** - die Achsenwerte werden mit der Einheit des ersten Datenpunkts im Datensatz angezeigt; **Statisch** - die Achsenwerte werden mit der im Eingabefeld *Wert* angegebenen Einheit angezeigt; bleibt das Feld leer, werden nur numerische Werte angezeigt.

Legende

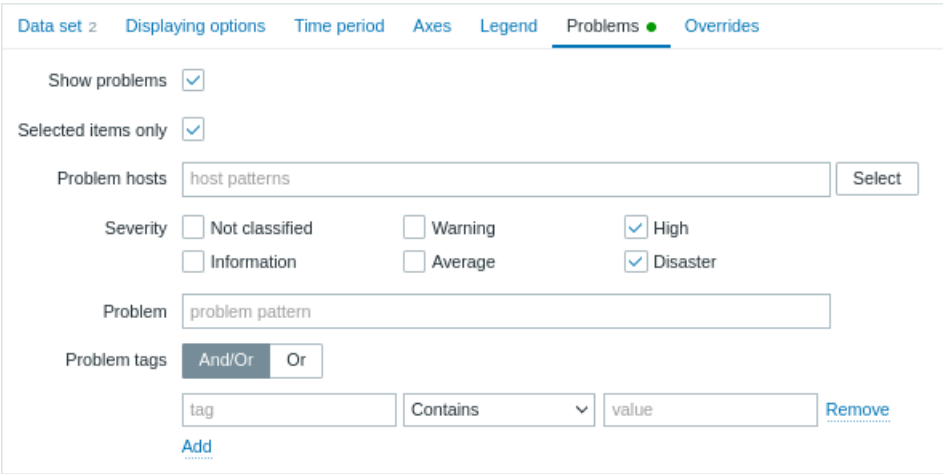
Die Registerkarte **Legende** ermöglicht die Anpassung der Diagrammlegende:



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 Legendenzeilen 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, ohne den Wert des Parameters *Maximale Anzahl der Zeilen* zu 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

Die Registerkarte **Probleme** ermöglicht die Anpassung der Problemanzeige:

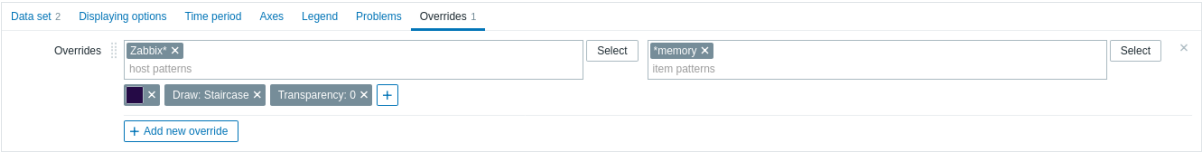


Probleme anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Anzeige von Problemen im Diagramm zu aktivieren (standardmäßig nicht markiert, d. h. deaktiviert).
Nur ausgewählte Datenpunkte	Aktivieren Sie dieses Kontrollkästchen, damit im Diagramm nur Probleme für die ausgewählten Datenpunkte angezeigt werden.
Problem-Hosts	Wählen Sie die Problem-Hosts aus, die im Diagramm angezeigt werden sollen. Platzhaltermuster können verwendet werden (zum Beispiel liefert ∗ Ergebnisse zurück, die null oder mehr Zeichen entsprechen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge einfach manuell ein und drücken Sie *Enter*. Während der Eingabe werden alle passenden Hosts in der Dropdown-Liste angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Schweregrad	Markieren Sie die Schweregrade von Problemen, um die im Diagramm angezeigten Probleme zu filtern. Wenn keine Schweregrade markiert sind, 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 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.

Überschreibungen

Die Registerkarte **Überschreibungen** ermöglicht das Hinzufügen benutzerdefinierter Überschreibungen für Datensätze:



Überschreibungen sind nützlich, wenn mehrere Datenpunkte für einen Datensatz mithilfe des Platzhalters ∗ ausgewählt werden und Sie ändern möchten, wie die Datenpunkte standardmäßig angezeigt werden sollen (z. B. die Standard-Basisfarbe 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 

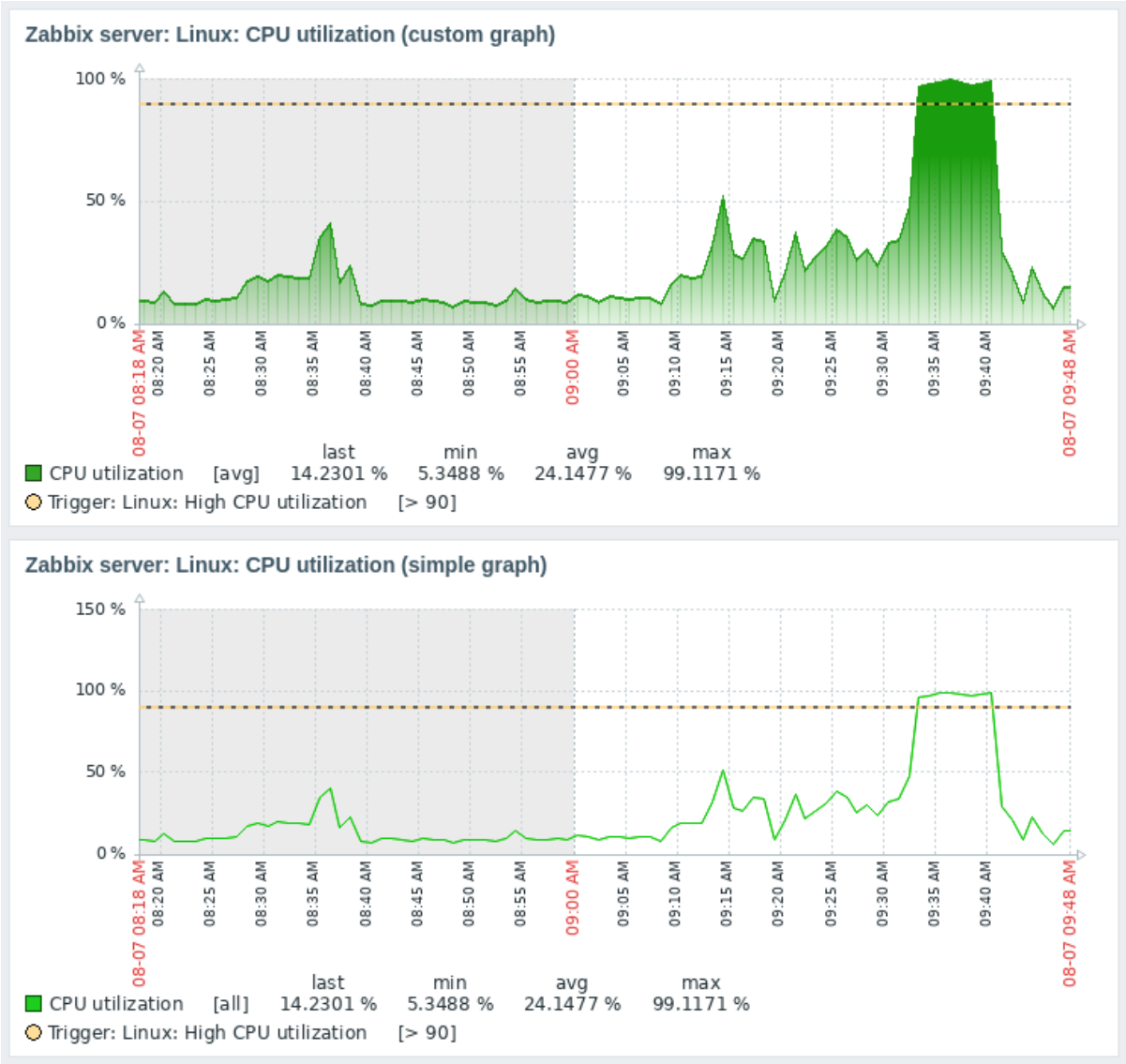
- Wählen Sie Hosts und Datenpunkte für die Überschreibung aus. Alternativ können Sie Host- und Datenpunktmuster eingeben. Platzhaltermuster können verwendet werden (zum Beispiel liefert ∗ Ergebnisse zurück, die null oder mehr Zeichen entsprechen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge einfach manuell ein und drücken Sie *Enter*. Während der Eingabe werden alle übereinstimmenden Hosts in der Dropdown-Liste angezeigt. 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). Die Parameter für Host-Muster und Datenpunktmuster sind obligatorisch. Der Parameter zur Angabe von Host-Mustern ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird. Der Parameter zur Angabe einer Datenpunktliste erlaubt bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard nur die Auswahl von auf der Vorlage konfigurierten Datenpunkten.
- Klicken Sie auf 

, um Überschreibungsparameter auszuwählen. Es sollte mindestens ein Überschreibungsparameter ausgewählt werden. Beschreibungen der Parameter finden Sie oben auf der Registerkarte *Datensatz*.

9 Graph (klassisch)

Übersicht

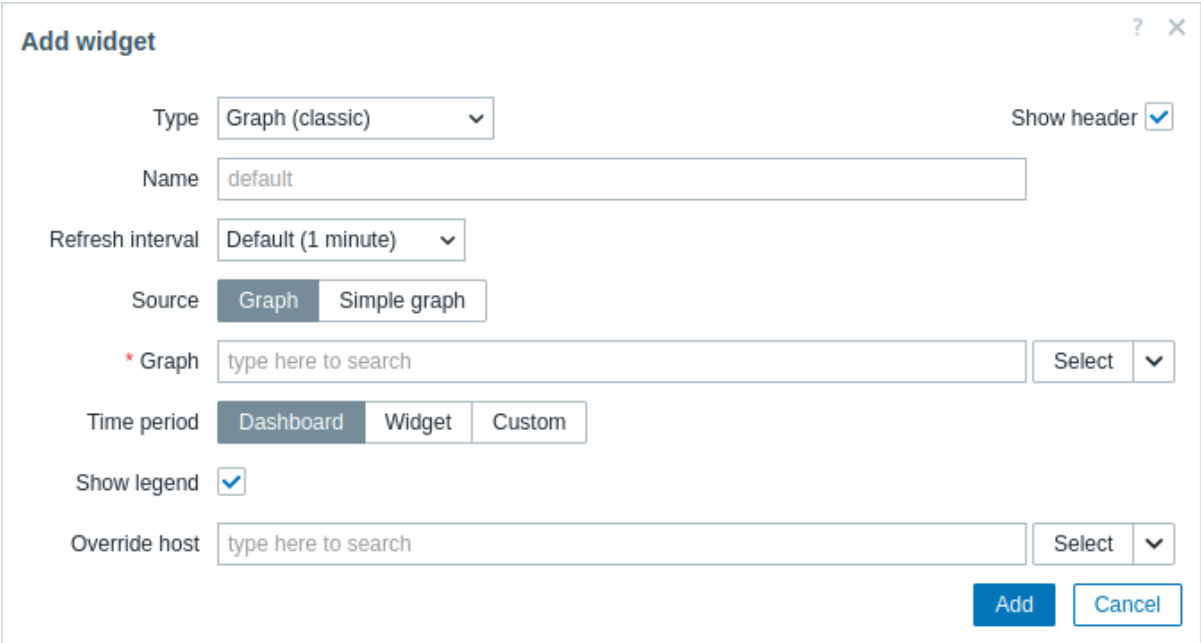
Das Widget *Graph (classic)* zeigt numerische Datenpunkt-Daten als bildbasiertes benutzerdefiniertes Diagramm oder einfaches Diagramm an. Es unterstützt die Auswahl des Zeitraums, Auslöser-Linien und die optionale Anzeige einer Legende.



Die im Widget *Graph (classic)* angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem Sie im Widget-Menü die Option *Bild herunterladen* auswählen.

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)

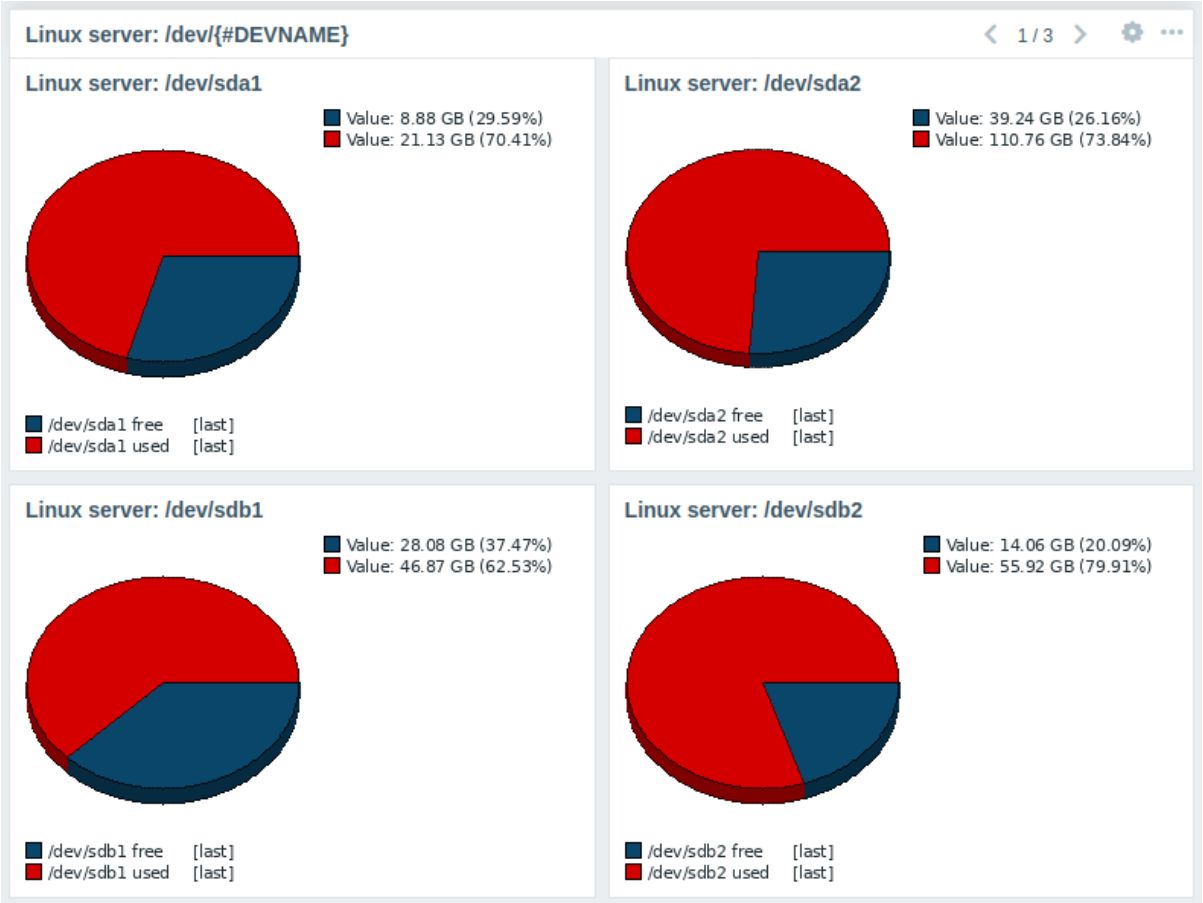
Wenn die Höhe des Graph innerhalb des Widgets nicht ausreicht, wird die Legende möglicherweise nicht oder nur teilweise angezeigt. Auslöser und Perzentile werden zuerst ausgeblendet, gefolgt von der Datenpunkt-Legende. Um die vollständige Legende anzuzeigen, erhöhen Sie die vertikale Größe des Widgets.

Note:
Es können nicht mehr als 3 Auslöser-Linien angezeigt werden. Wenn es mehr Auslöser gibt, werden die Auslöser mit niedrigerem Schweregrad bevorzugt angezeigt.

10 Graph-Prototyp

Übersicht

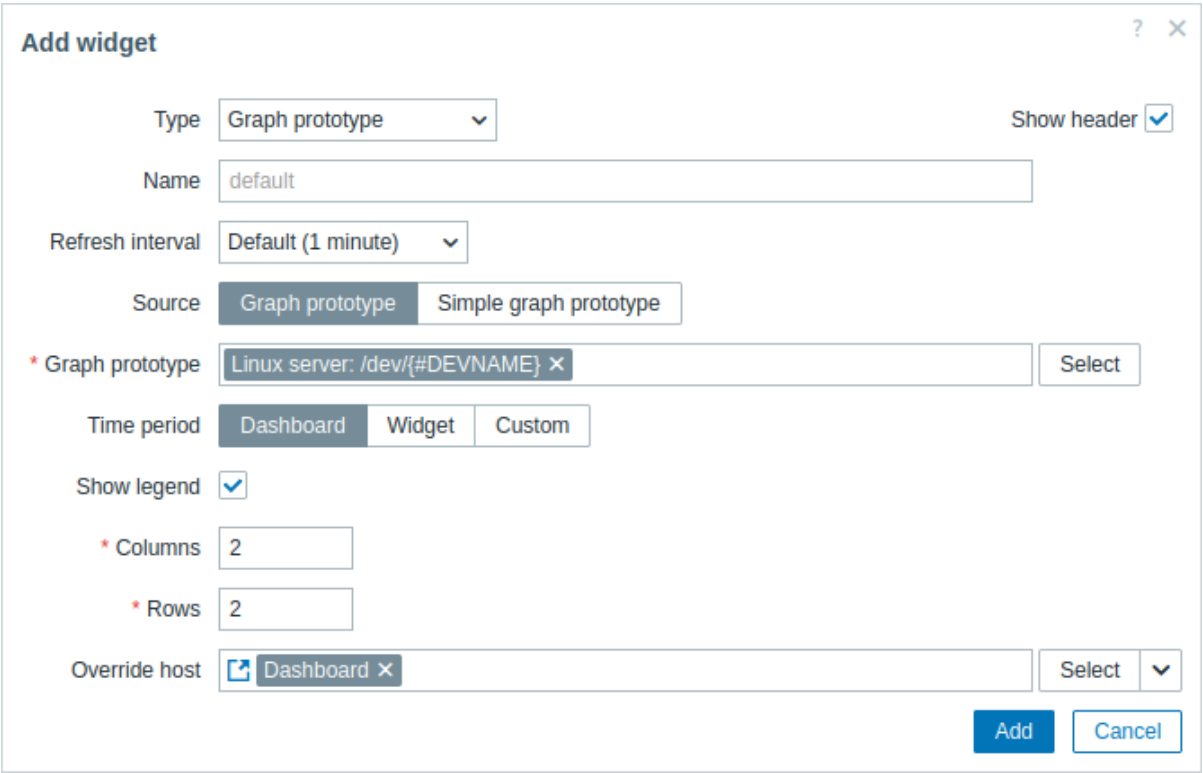
Das Widget *Graph prototype* zeigt ein Raster automatisch erstellter benutzerdefinierter Graphen basierend auf Graph-Prototypen oder Datenpunkt-Prototypen von Low-Level-Discovery-Regeln an.



Die Graphen werden in einem konfigurierbaren Rasterlayout angeordnet; wenn die Anzahl der Graphen das Layout überschreitet, wird im Widget eine Seitennavigation aktiviert.

Konfiguration

Um zu konfigurieren, wählen Sie *Graph prototype* als Widget-Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Source	Wählen Sie die Quelle der Graphen aus: **Graph prototype** oder **Simple graph prototype**.

Graph prototype	Wählen Sie einen Graph-Prototyp aus, um von diesem Graph-Prototyp entdeckte Graphen anzuzeigen. Dieser Parameter ist verfügbar, wenn *Source* auf „Graph prototype“ gesetzt ist.
Item prototype	Wählen Sie einen Datenpunkt-Prototyp aus, um einfache Graphen für Datenpunkte anzuzeigen, die durch den Datenpunkt-Prototyp entdeckt wurden. Dieser Parameter ist verfügbar, wenn *Source* auf „Simple graph prototype“ gesetzt ist.
Time period	Legen Sie einen Zeitraum fest, für den Daten in den Graphen angezeigt werden sollen. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswähler 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 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 *Time period*-Parameters des Widgets diesen weiterhin als Datenquelle für den Zeitraum verwenden 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.
Show legend	Deaktivieren Sie dieses Kontrollkästchen, um die Legende in den Graphen auszublenden (standardmäßig aktiviert).
Override host	Wählen Sie ein kompatibles Widget oder den Host-Auswähler des Dashboards als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Columns	Geben Sie die Anzahl der Spalten mit Graphen ein, die innerhalb eines Graph-Prototyp-Widgets angezeigt werden sollen.
Rows	Geben Sie die Anzahl der Zeilen mit Graphen ein, die innerhalb eines Graph-Prototyp-Widgets angezeigt werden sollen.

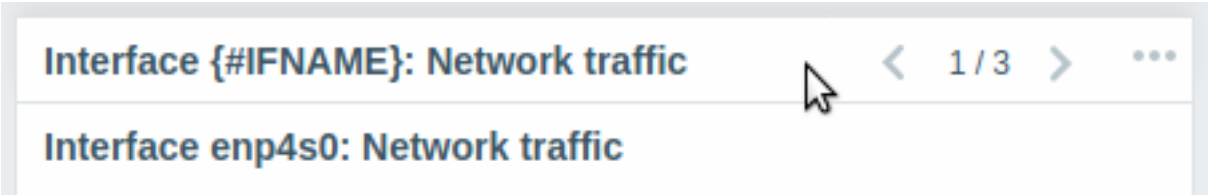
Die Legende des Graphen besteht aus drei Abschnitten:

- Datenpunkte und ihre aggregierten Werte
- Perzentile (falls konfiguriert)
- Auslöser (falls welche mit den angezeigten Datenpunkten verknüpft sind)

Wenn die Höhe des Graphen innerhalb des Widgets nicht ausreicht, wird die Legende möglicherweise nicht oder nur teilweise angezeigt. Zuerst werden Auslöser und Perzentile ausgeblendet, danach die 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.

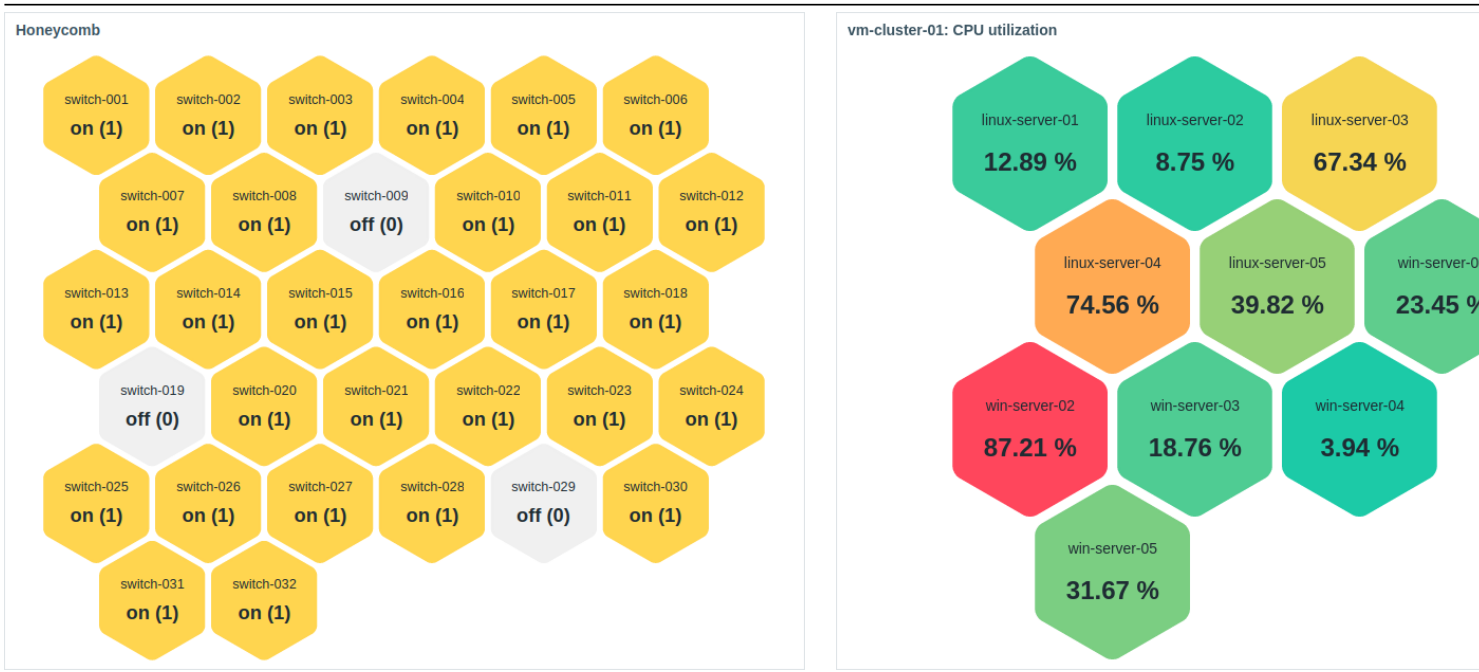
Auch wenn die Parameter *Columns* und *Rows* es ermöglichen, mehr als einen Graphen im Widget unterzubringen, kann es dennoch mehr entdeckte Graphen geben, als Spalten/Zeilen im Widget vorhanden sind. In diesem Fall wird im Widget eine Seitennavigation verfügbar, und eine aufklappbare Kopfzeile ermöglicht das Wechseln zwischen den Seiten mit den linken und rechten Pfeilen:



11 Honeycomb

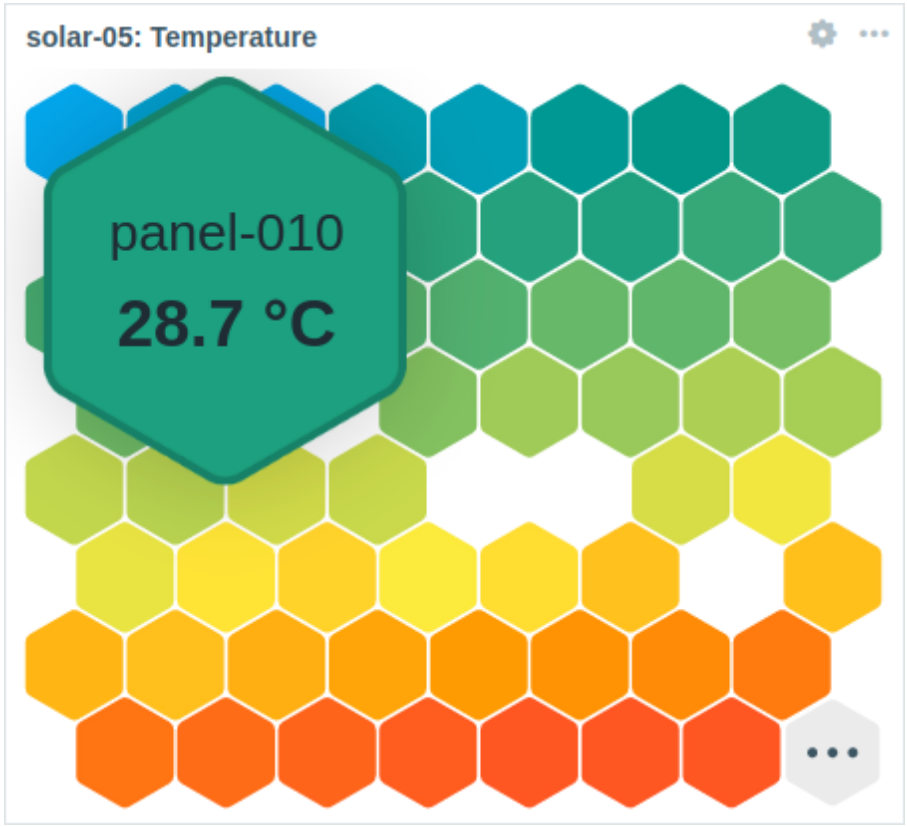
Übersicht

Das *Honeycomb*-Widget zeigt Hosts oder Datenpunkte als sechseckige Zellen in einem Raster an. Dieses Layout erleichtert es, problematische Hosts/Datenpunkte zu erkennen, Häufungen von Problemen zu identifizieren, Gruppen auf einen Blick zu vergleichen und mehr.



Die Anzahl der angezeigten Zellen ist durch die Widget-Größe und die minimale Zellgröße (32px) begrenzt. Wenn nicht alle Zellen hineinpassen, zeigt die letzte Zelle Auslassungspunkte an. Die Größe des Widgets kann geändert werden, wobei Anzahl, Größe und Positionierung der Zellen dynamisch angepasst werden.

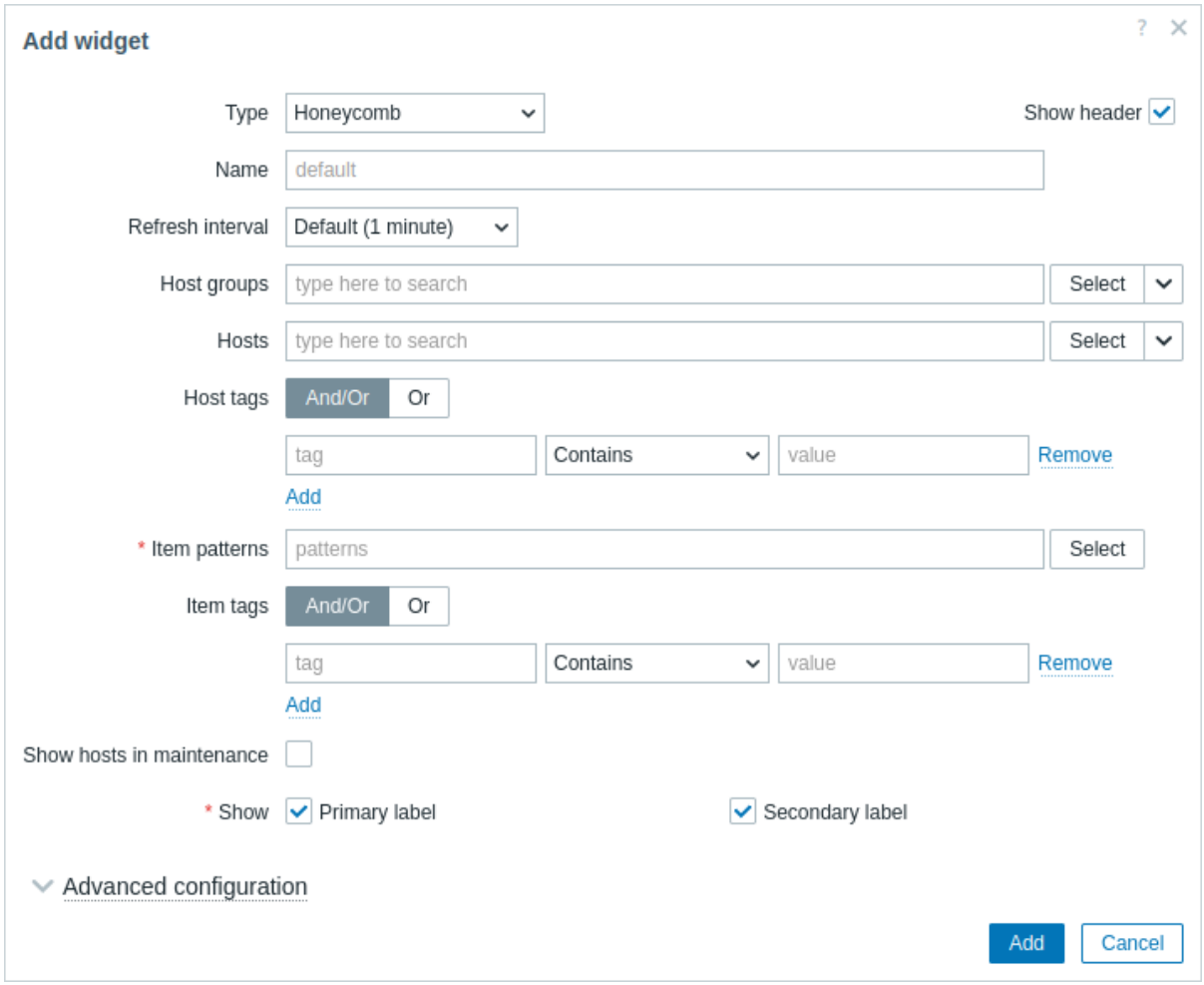
Beim Überfahren mit der Maus wird die fokussierte Zelle zur besseren Sichtbarkeit vergrößert. Beim Klicken auf eine Zelle wird ihr Rand hervorgehoben, bis eine andere Zelle ausgewählt wird.



Die im *Honeycomb*-Widget angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem im Widget-Menü die Option *Bild herunterladen* ausgewählt wird.

Konfiguration

Wählen Sie zur Konfiguration *Honeycomb* als Typ aus:



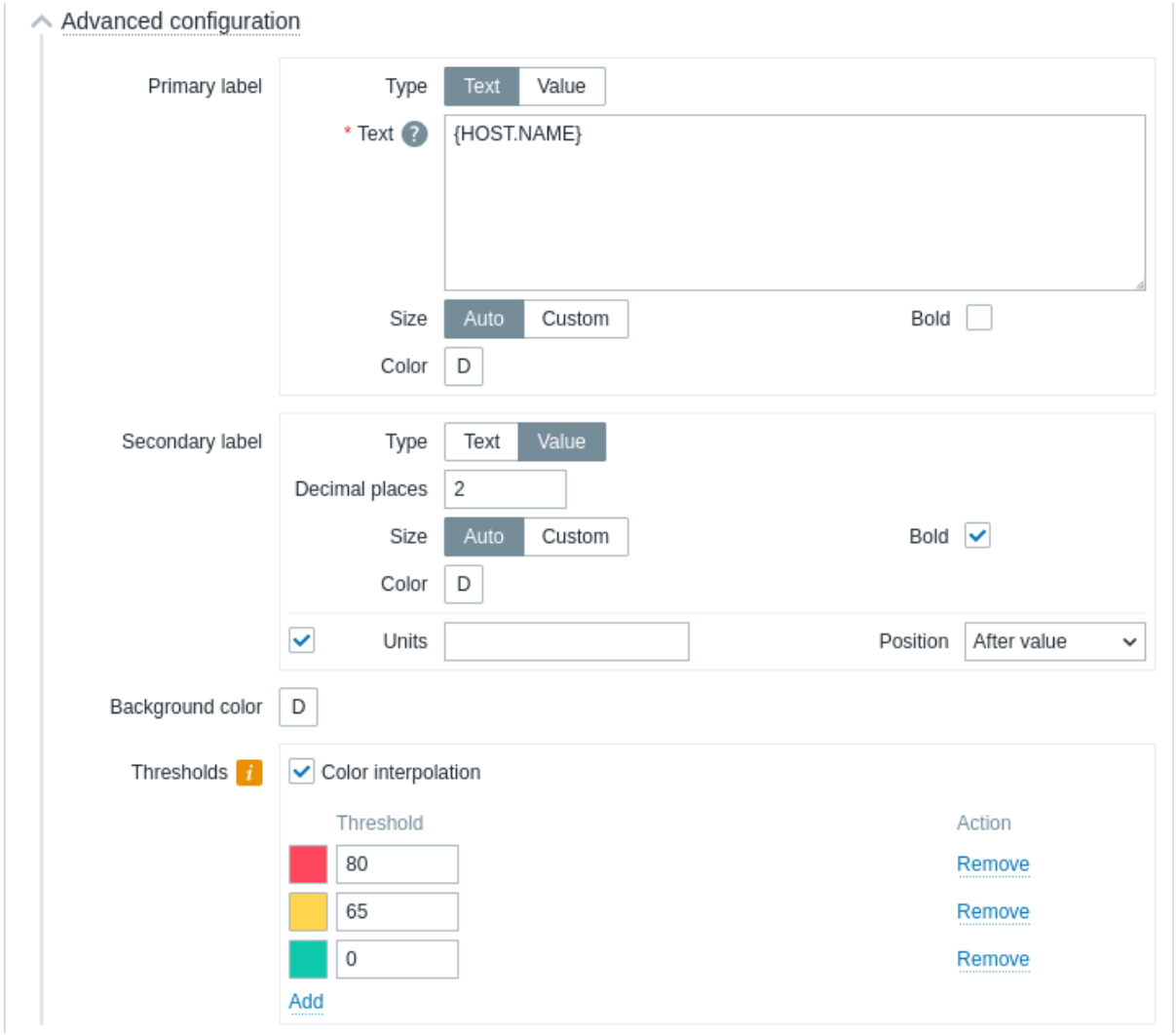
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Durch die Auswahl einer übergeordneten Host-Gruppe werden implizit alle untergeordneten 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 unterstützt Autovervollständigung; wenn Sie also 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 im 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 Beschriftung *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar und nur für Elemente, 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 im Parameter *Text* angegebenen Text an;
Wert - die Beschriftung zeigt den Wert des Datenpunkts mit der im Parameter *Dezimalstellen* angegebenen Anzahl von Nachkommastellen an.

Text — Geben Sie den Beschriftungstext ein. Dieser Text kann den Standardnamen des Datenpunkts 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 Host-Namen und innerhalb jedes Hosts nach Datenpunktnamen sortiert.
Dieser Parameter ist verfügbar, wenn *Typ* auf ”Text” gesetzt ist.

Dezimalstellen — Geben Sie die Anzahl der Nachkommastellen ein, die zusammen mit dem Wert angezeigt werden sollen.
Dieser Parameter ist verfügbar, wenn *Typ* auf ”Wert” gesetzt ist, und betrifft nur Datenpunkte, die Daten vom Typ numeric (float) zurückgeben.

Größe — Wählen Sie die Beschriftungsgröße aus:
Auto - 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 die Einheiten des Datenpunkts fett darzustellen.

Farbe — Wählen Sie die Farbe der Datenpunkteinheiten 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*.

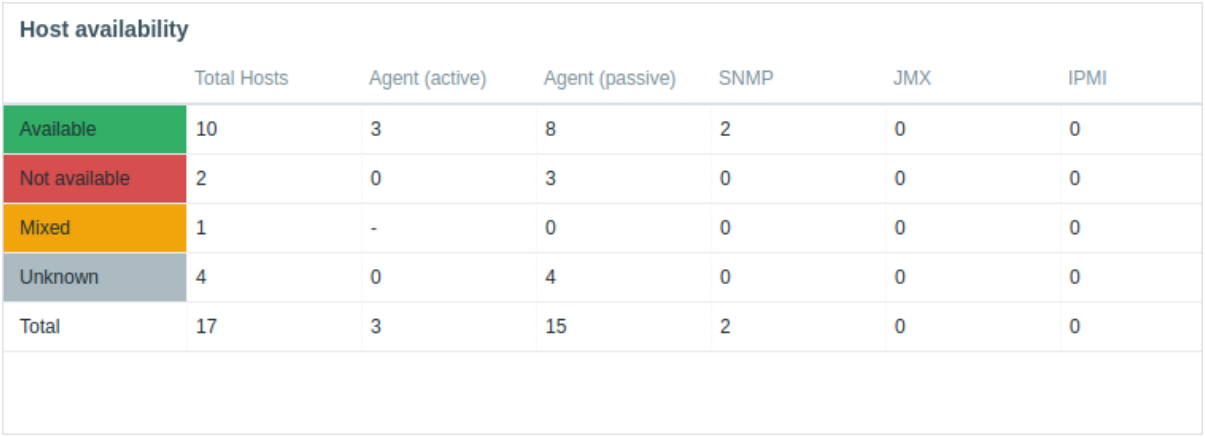
Einheiten

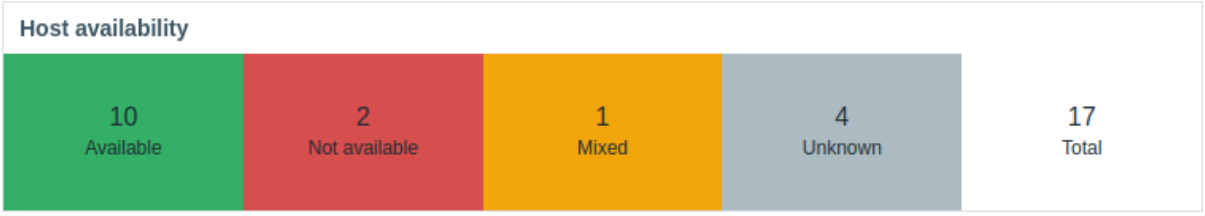
Einheiten	Aktivieren Sie das Kontrollkästchen, um Einheiten zusammen mit dem Datenpunktwert anzuzeigen. Wenn Sie einen Einheitsnamen eingeben, überschreibt dieser die in der Datenpunktkonfiguration festgelegten Einheiten. Dieser Parameter ist verfügbar, wenn *Typ* auf ”Text” gesetzt ist.
Position	Wählen 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 ”Text” gesetzt 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.

12 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.



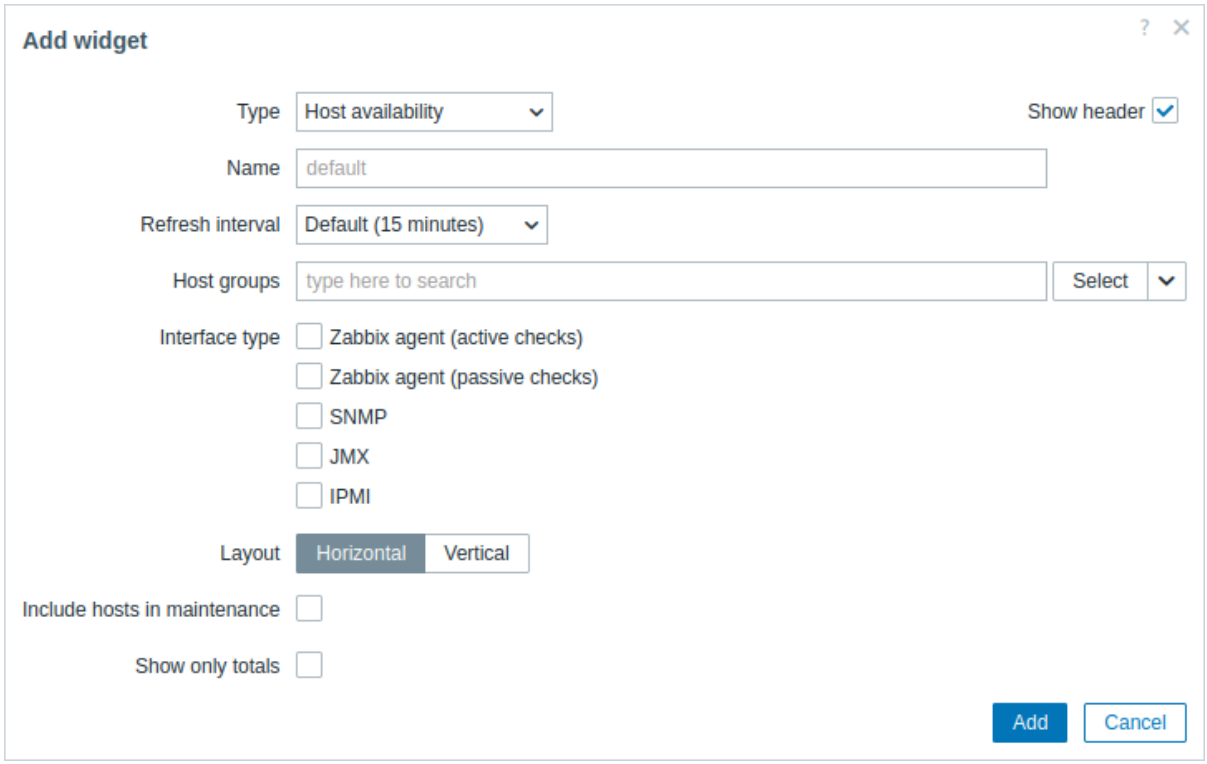


Die Host-Verfügbarkeit wird wie folgt gezählt:

- *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:



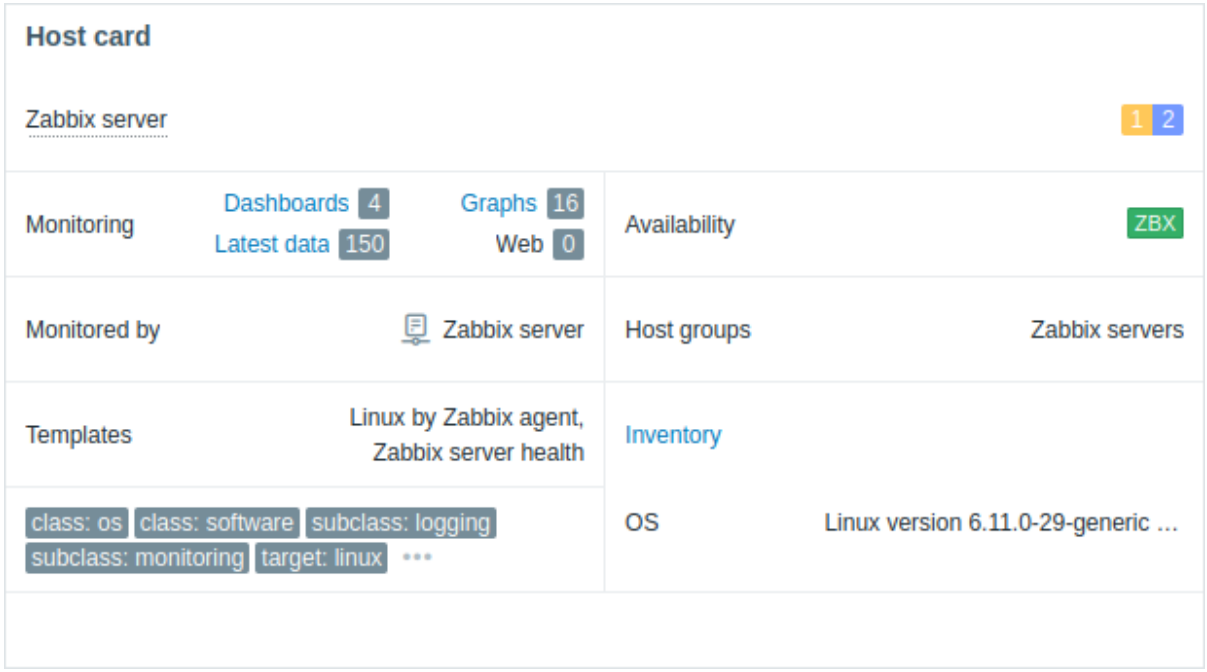
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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.

13 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.).

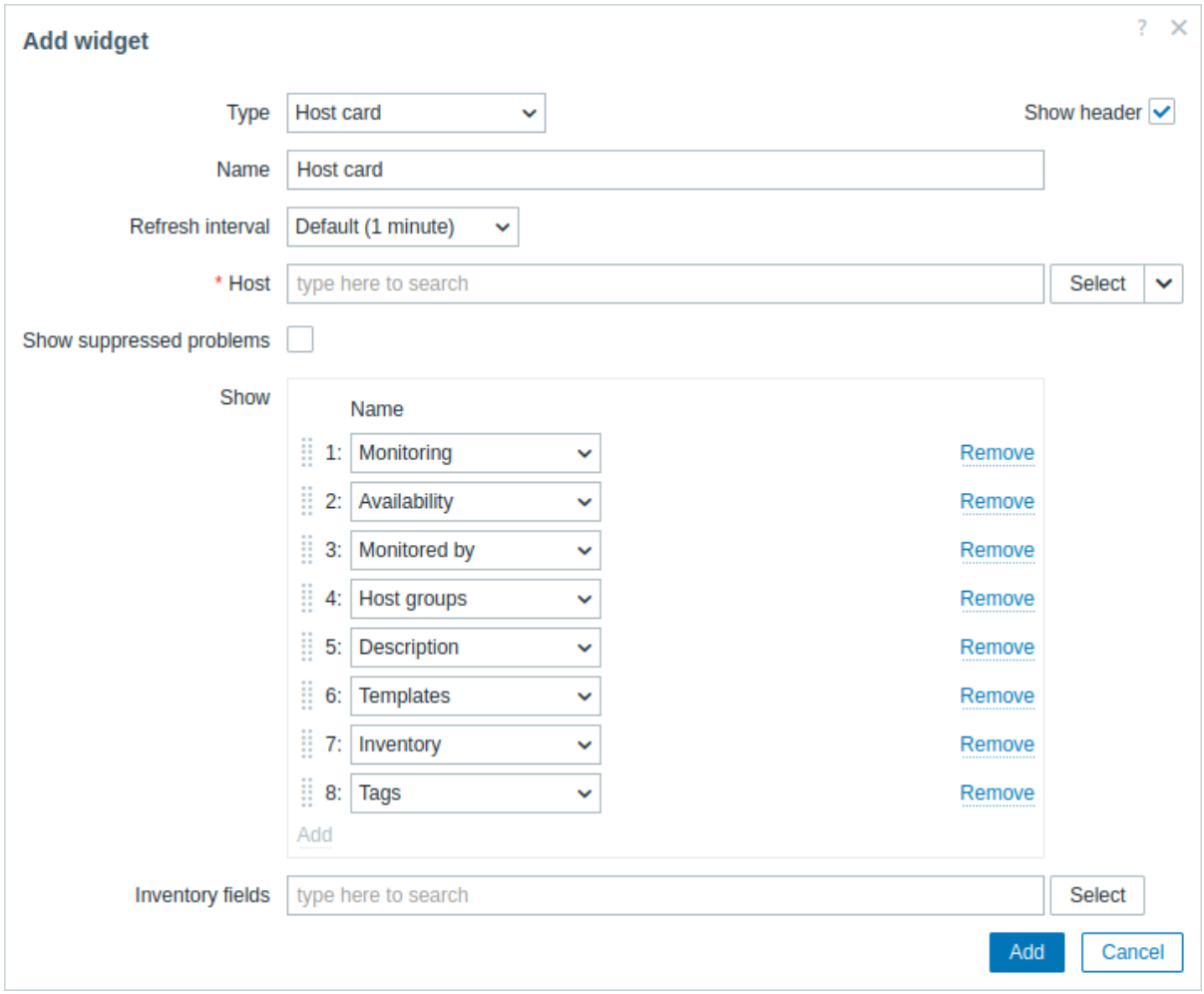


Der Host kann direkt in der Widget-Konfiguration festgelegt oder entweder aus einem kompatiblen Widget oder über den Host-Selektor im Dashboard ausgewählt werden.

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:



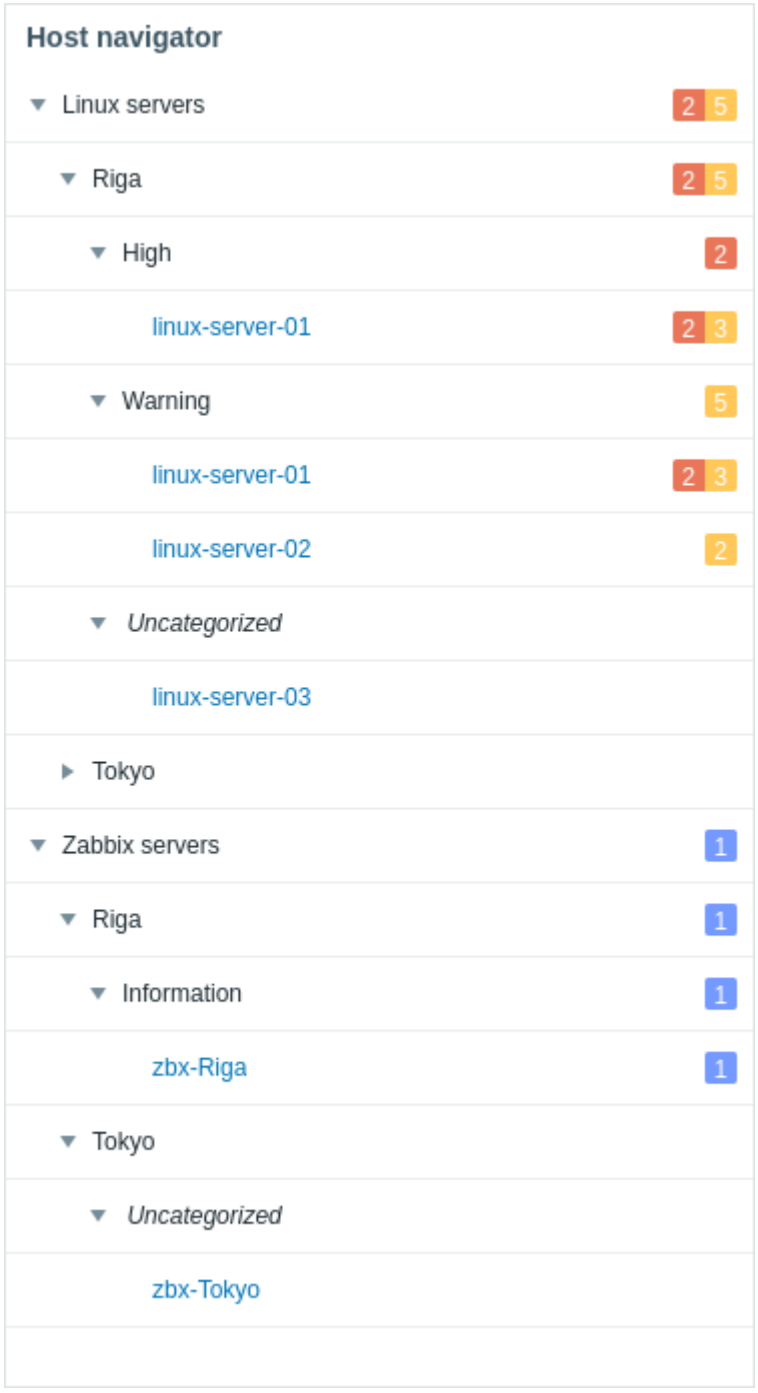
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Parameter festlegen:

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.

14 Host-Navigator

Übersicht

Das Widget *Host navigator* zeigt eine Liste von Hosts basierend auf verschiedenen Filter- und Gruppierungsoptionen an.



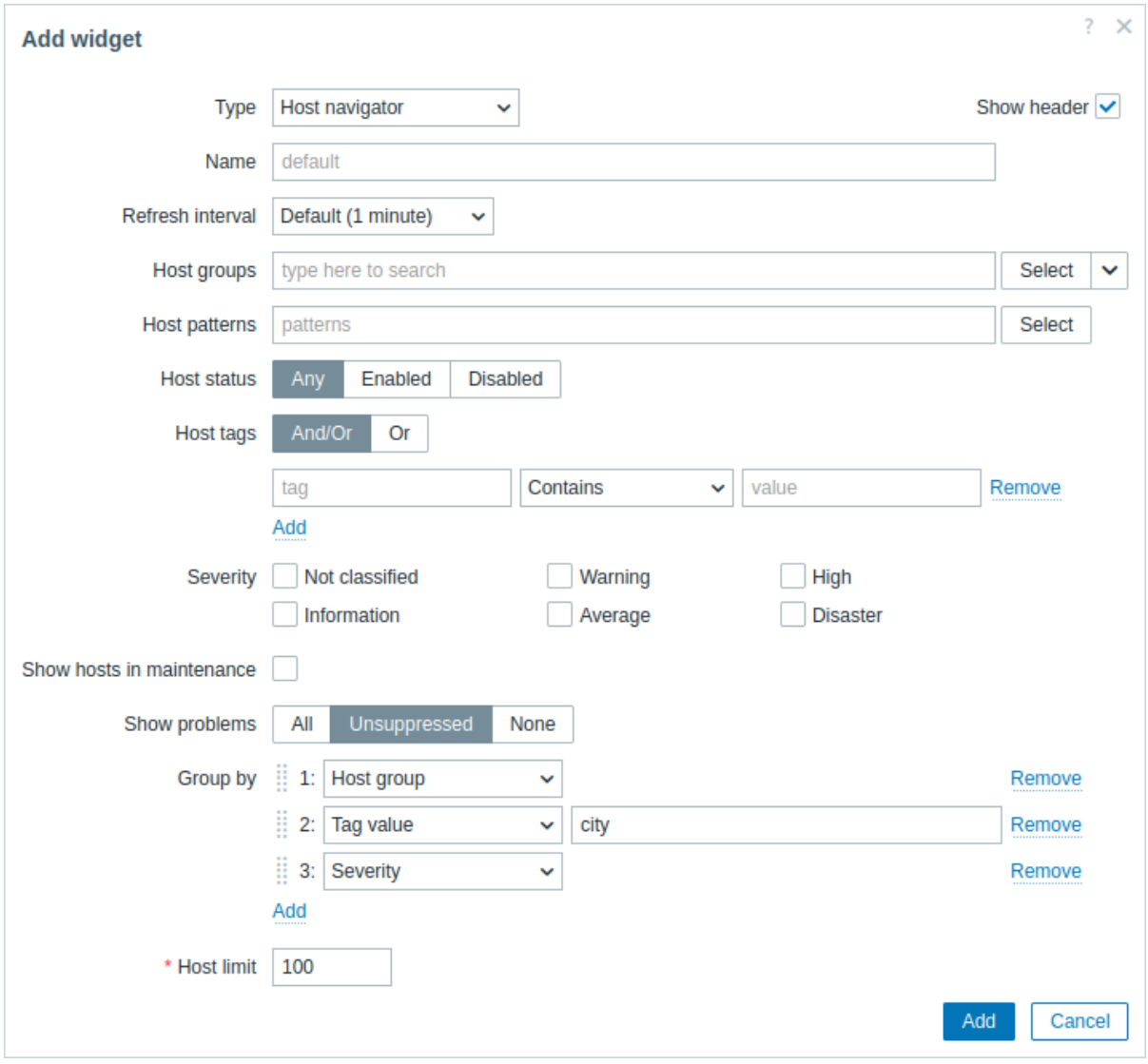
Gruppen, nach denen Hosts organisiert sind, können ein- oder ausgeklappt werden.

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 Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

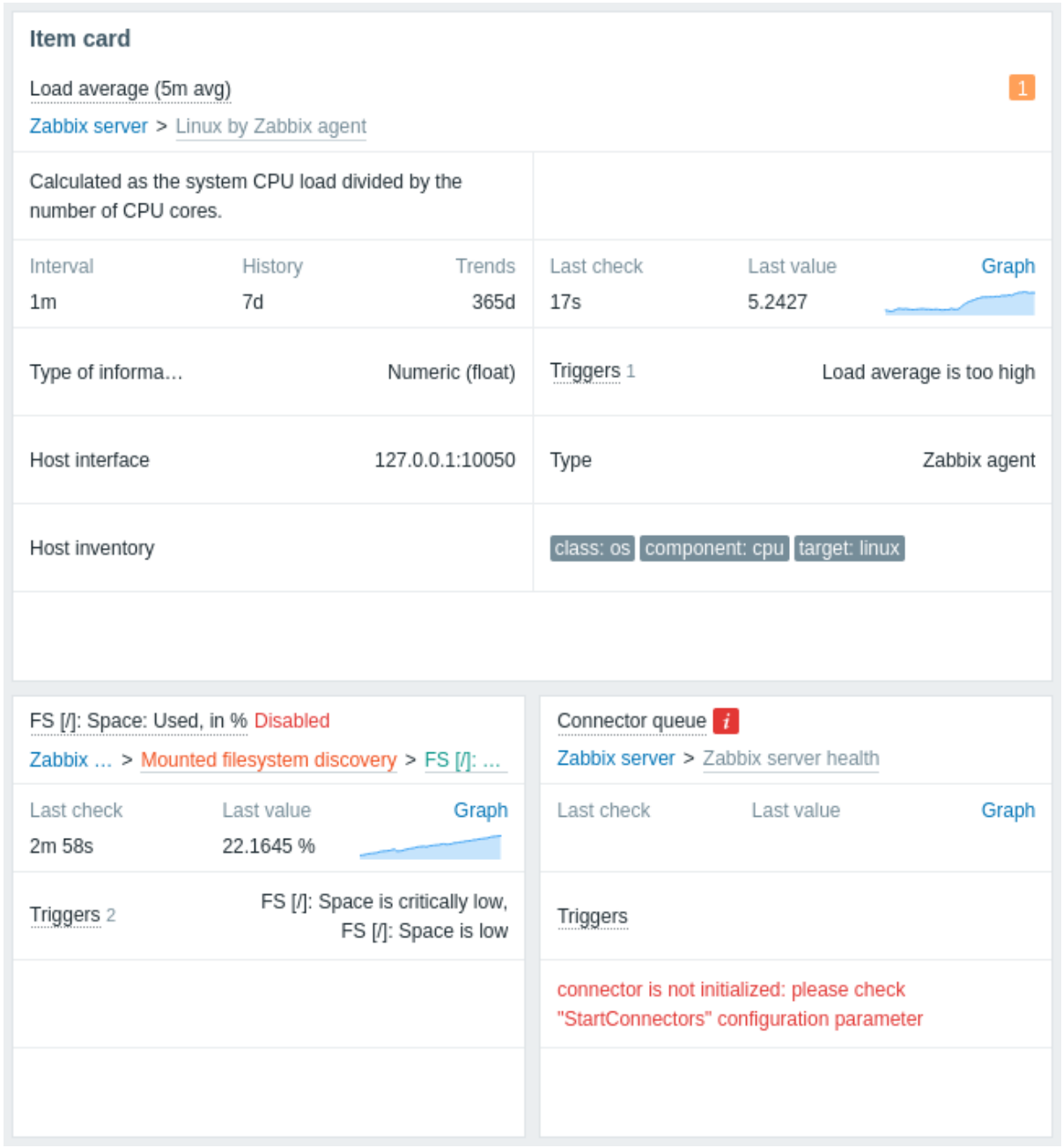
Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Die Auswahl einer übergeordneten Host-Gruppe wählt implizit auch alle darunterliegenden Host-Gruppen aus; wenn keine Host-Gruppen ausgewählt sind, zeigt das Widget alle Hosts aus allen Host-Gruppen an. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Muster	Geben Sie Host-Muster ein oder wählen Sie vorhandene Hosts als Host-Muster aus. Hosts, die den angegebenen Mustern entsprechen, werden im Host-Navigator angezeigt. Dieses Feld unterstützt 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 an. Für die Auswahl können Platzhaltermuster verwendet werden (zum Beispiel 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 *Enter*. Wenn Sie mit der Eingabe beginnen, zeigt eine Dropdown-Liste passende Hosts an, beschränkt auf Hosts innerhalb der ausgewählten *Host-Gruppen*, falls vorhanden. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Status	Filtern Sie, welche Hosts basierend auf ihrem 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. 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.
Schweregrad	Markieren Sie Problem-Schweregrade, um Hosts mit Problemen zu filtern, die im Widget angezeigt werden sollen. Wenn keine Schweregrade markiert sind, werden alle Hosts mit allen Problemen angezeigt.
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.
Probleme anzeigen	

Gruppieren nach	Fügen Sie ein Gruppierungsattribut hinzu, nach dem die ausgewählten Hosts gruppiert werden sollen: **Host-Gruppe** - Hosts nach ihrer Host-Gruppe gruppieren; **Tag-Wert** - geben Sie einen Tag-Namen ein, um Hosts nach den Werten dieses Tags zu gruppieren (geben Sie zum Beispiel „city“ ein, um Hosts nach den Werten „Riga“, „Tokyo“ usw. zu gruppieren); **Schweregrad** - Hosts nach den Schweregraden ihrer Probleme gruppieren. Wenn *Probleme anzeigen* so konfiguriert ist, dass Probleme angezeigt werden, werden diese wie folgt dargestellt: - für jede Schweregrad-Gruppe wird nur die entsprechende Problemanzahl angezeigt; - für jeden Host werden alle seine Problemanzahlen angezeigt. Beachten Sie, dass Hosts nur nach den im Parameter *Schweregrad* markierten Schweregraden gruppiert werden; wenn keine Schweregrade markiert sind, werden alle Hosts nach allen Schweregraden gruppiert. Gruppierungsattribute können durch Ziehen am Griff vor dem Gruppennamen nach oben oder unten neu angeordnet werden. Beachten Sie, dass die Reihenfolge der Gruppierungsattribute die Verschachtelungsreihenfolge der Gruppen bestimmt. Wenn Sie zum Beispiel mehrere Tag-Namen angeben (1: color, 2: city), werden Hosts zuerst nach Farbe (red, blue usw.) und dann nach Stadt (Riga, Tokyo usw.) gruppiert. Ein Host kann abhängig von den konfigurierten Gruppierungsattributen in mehreren Gruppen angezeigt werden (zum Beispiel, wenn nach Host-Gruppe gruppiert wird und der Host zu mehreren Host-Gruppen gehört). Durch Klicken auf solche Hosts werden sie in allen Gruppen ausgewählt und hervorgehoben. Hosts, die nicht mit den konfigurierten Gruppierungsattributen übereinstimmen, werden in der Gruppe *Nicht kategorisiert* 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 an Hosts ein, die angezeigt werden sollen. Mögliche Werte liegen im Bereich von 1-9999. Wenn mehr Hosts zur Anzeige verfügbar sind als das festgelegte Limit, wird unterhalb der angezeigten Hosts eine entsprechende Meldung eingeblendet (zum Beispiel „100 von 100+ Hosts werden angezeigt“). Beachten Sie, dass sich das konfigurierte Host-Limit auch auf die Anzeige der konfigurierten Gruppen auswirkt; wenn zum Beispiel das Host-Limit auf 100 gesetzt ist und Hosts nach Tag-Werten gruppiert werden (mehr als 200), werden im Widget nur die ersten 100 Tag-Werte mit den entsprechenden Hosts angezeigt. Dieser Parameter wird nicht durch den Parameter *Limit für Such- und Filterergebnisse* unter *Administration → Allgemein → GUI* beeinflusst. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

15 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.).

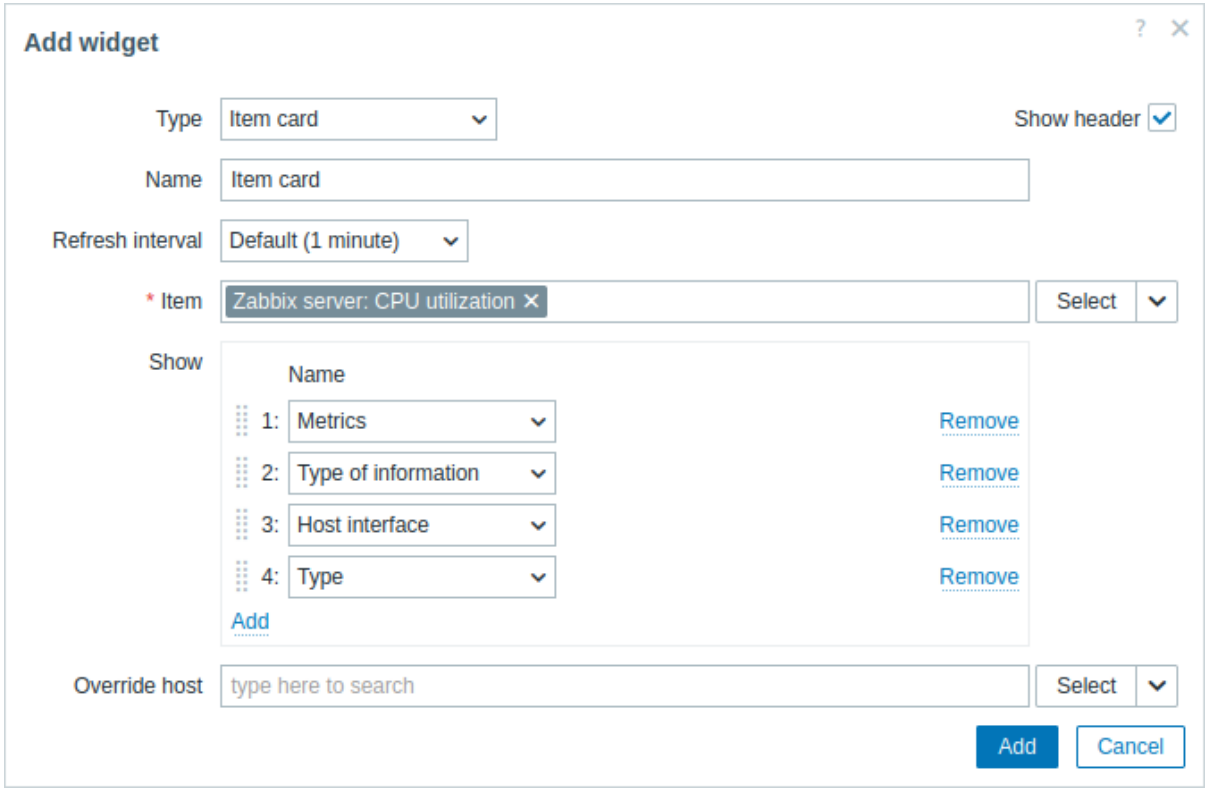


Der Datenpunkt kann direkt in der Widget-Konfiguration festgelegt oder aus einem kompatiblen Widget im Dashboard ausgewählt werden.

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:



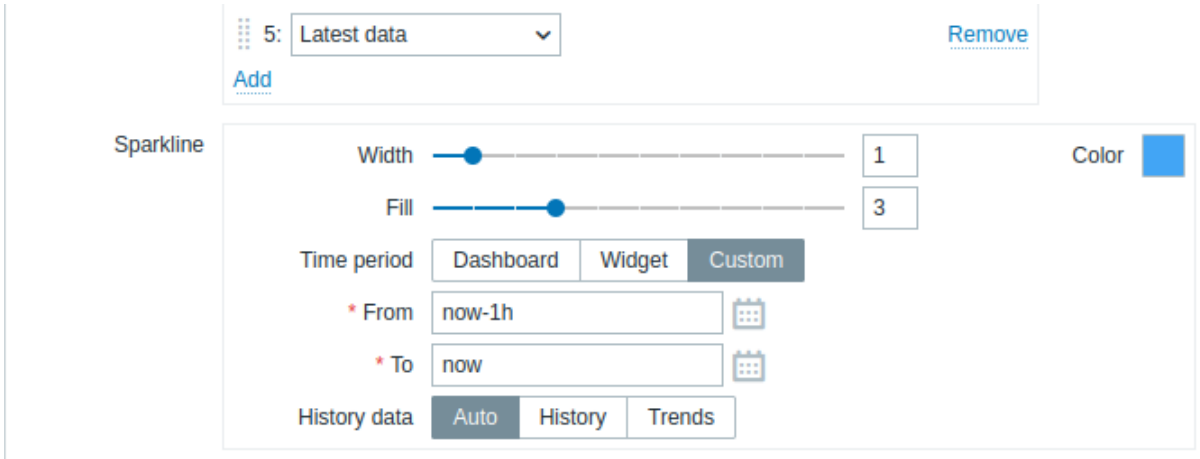
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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.



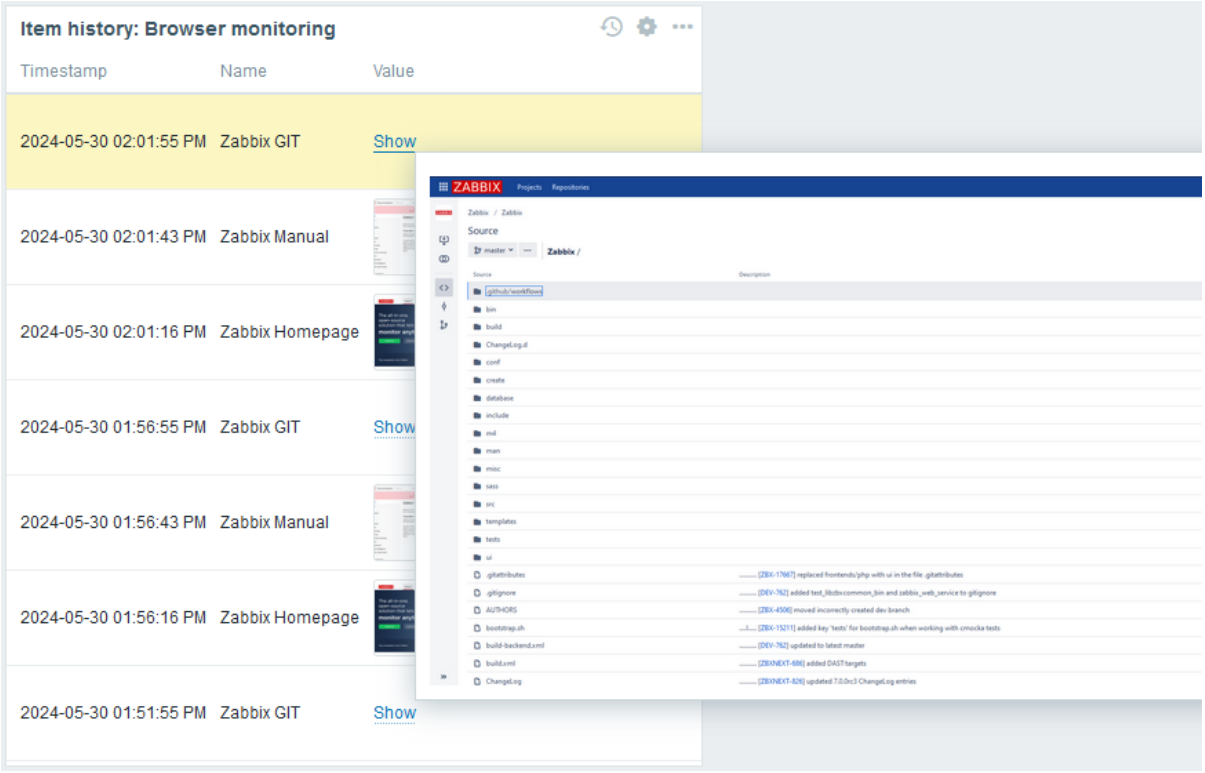
Breite	Legen Sie die Dicke der Diagrammlinie fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.
Farbe	Wählen Sie die Linien- und Füllfarbe aus.
Füllung	Legen Sie die Transparenz der Füllfarbe fest, indem Sie den Schieberegler verwenden oder manuell einen 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.

16 Datenpunkt-Verlauf

Übersicht

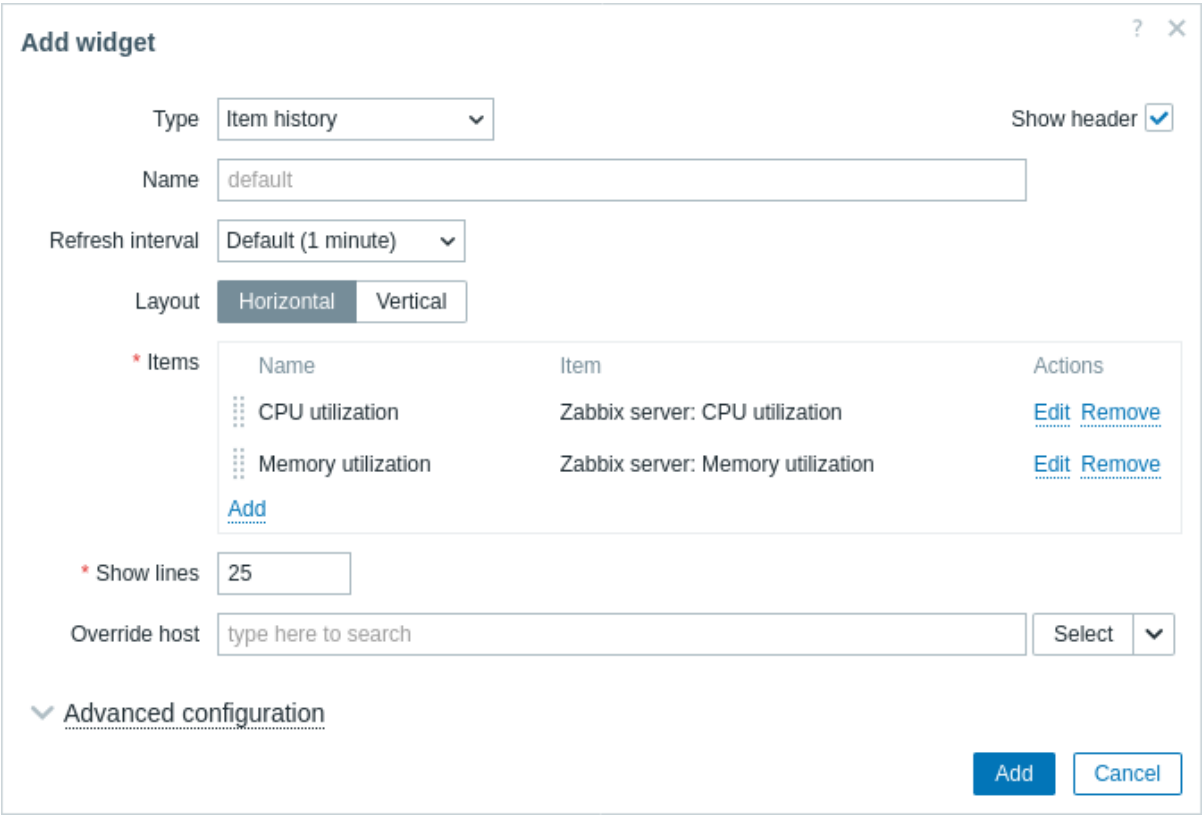
Das Widget *Datenpunkt-Verlauf* zeigt die neuesten Daten für alle Datenpunkt-Typen (numerisch, Text usw.) in Tabellenform an. Es kann Fortschrittsbalken und Bilder anzeigen (nützlich für Browser-Datenpunkte) und Werte hervorheben (nützlich für Log-dateiüberwachung).



Es können bis zu 1000 Datensätze angezeigt werden.

Konfiguration

Wählen Sie zur Konfiguration *Datenpunkthistorie* als Typ aus:



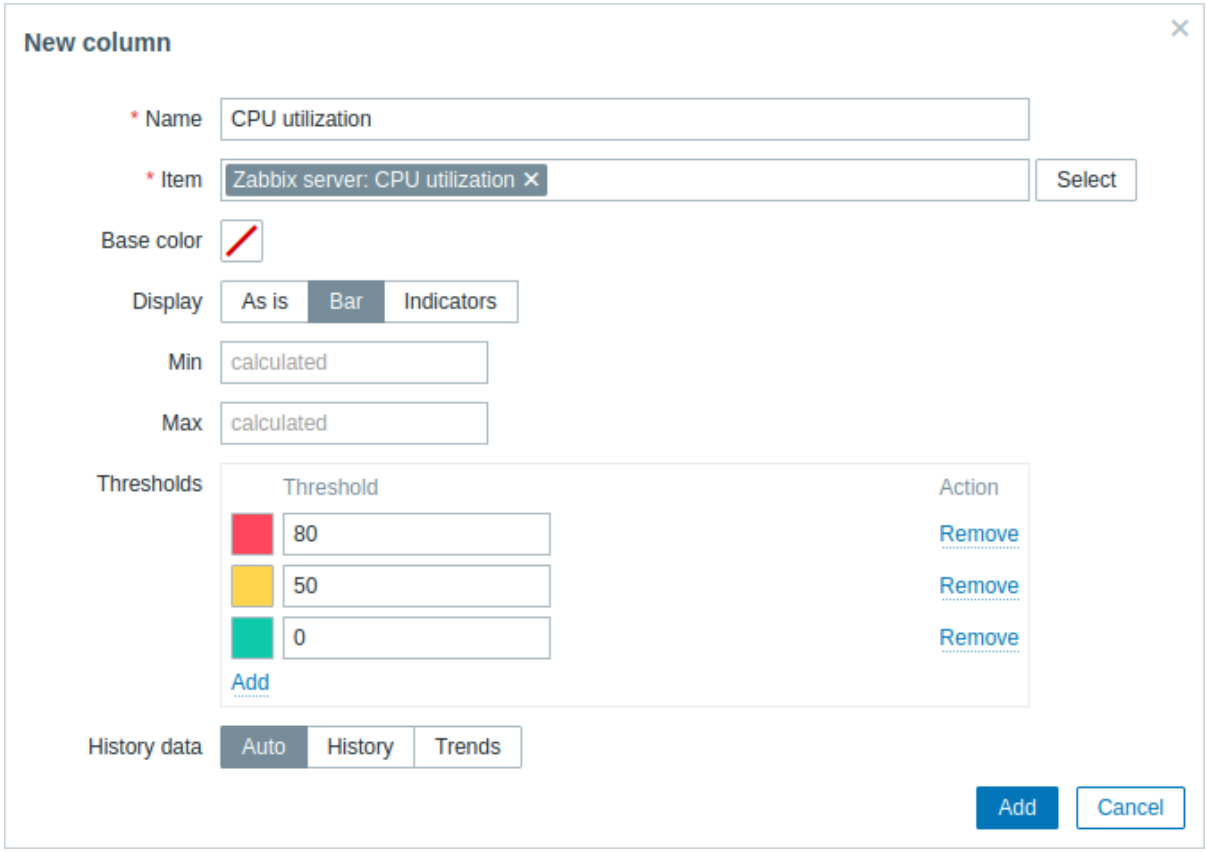
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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 speziell für Datenpunkte vom Typ Zeichen, Text, Log und JSON:

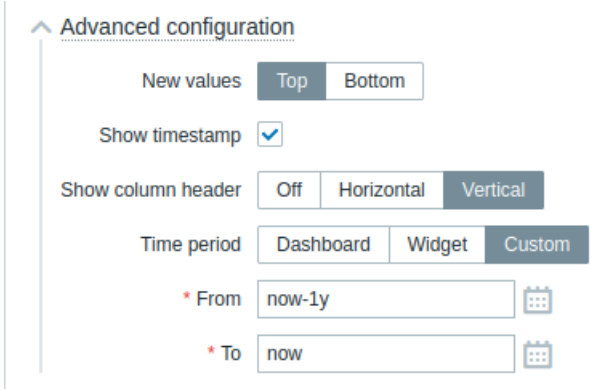
Hervorhebungen	Klicken Sie auf *Hinzufügen*, um eine Hervorhebung hinzuzufügen, wählen Sie eine Hervorhebungsfarbe aus der Farbauswahl und geben Sie einen regulären Ausdruck an. Die ausgewählte Farbe wird als Hintergrundfarbe für Datenpunktwerte verwendet, bei denen der angegebene reguläre Ausdruck auf den Text zutrifft.
Anzeige	Wählen Sie aus, wie der Datenpunktwert angezeigt werden soll: **Wie empfangen** - genau wie empfangen. Zeilenumbrüche bleiben erhalten, wenn der Wert Zeilenumbrüche enthält. Der Text wird umbrochen, wenn *Layout* auf „Vertikal“ gesetzt ist und mehr als eine Spalte konfiguriert ist; **HTML** - als HTML-formatierter Text; **Einzelne Zeile** - als einzelne Zeile, auf eine angegebene Länge gekürzt (1–500 Zeichen). Wenn Sie den gekürzten Wert mit der Maus berühren oder darauf klicken, wird ein Pop-up 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:

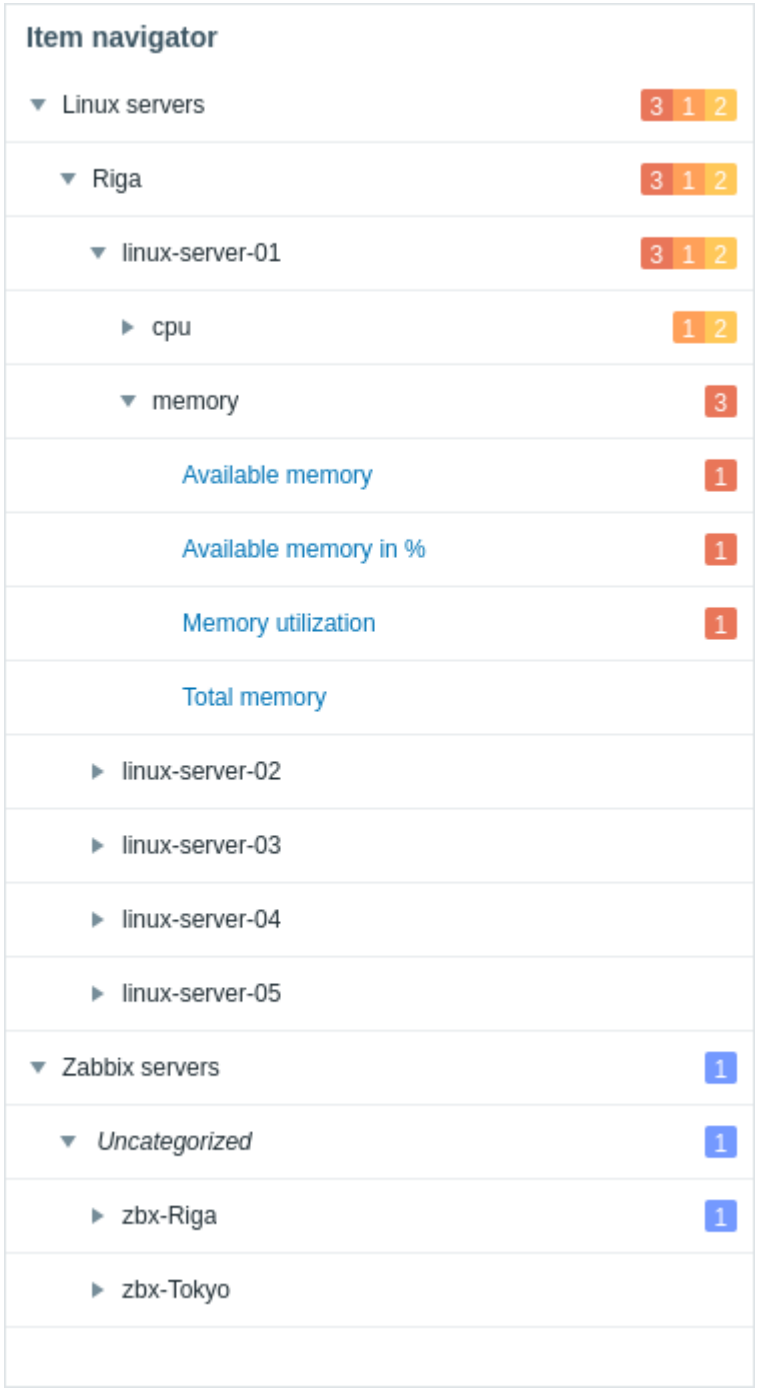


Neue Werte	Wählen Sie aus, wo neue Datenpunktwerte hinzugefügt werden sollen: **Oben** - am Anfang der Spalten; **Unten** - am Ende der Spalten.
Zeitstempel anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Zeitstempelspalte anzuzeigen (standardmäßig nicht aktiviert).
Spaltenüberschrift anzeigen	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.

17 Datenpunkt-Navigator

Übersicht

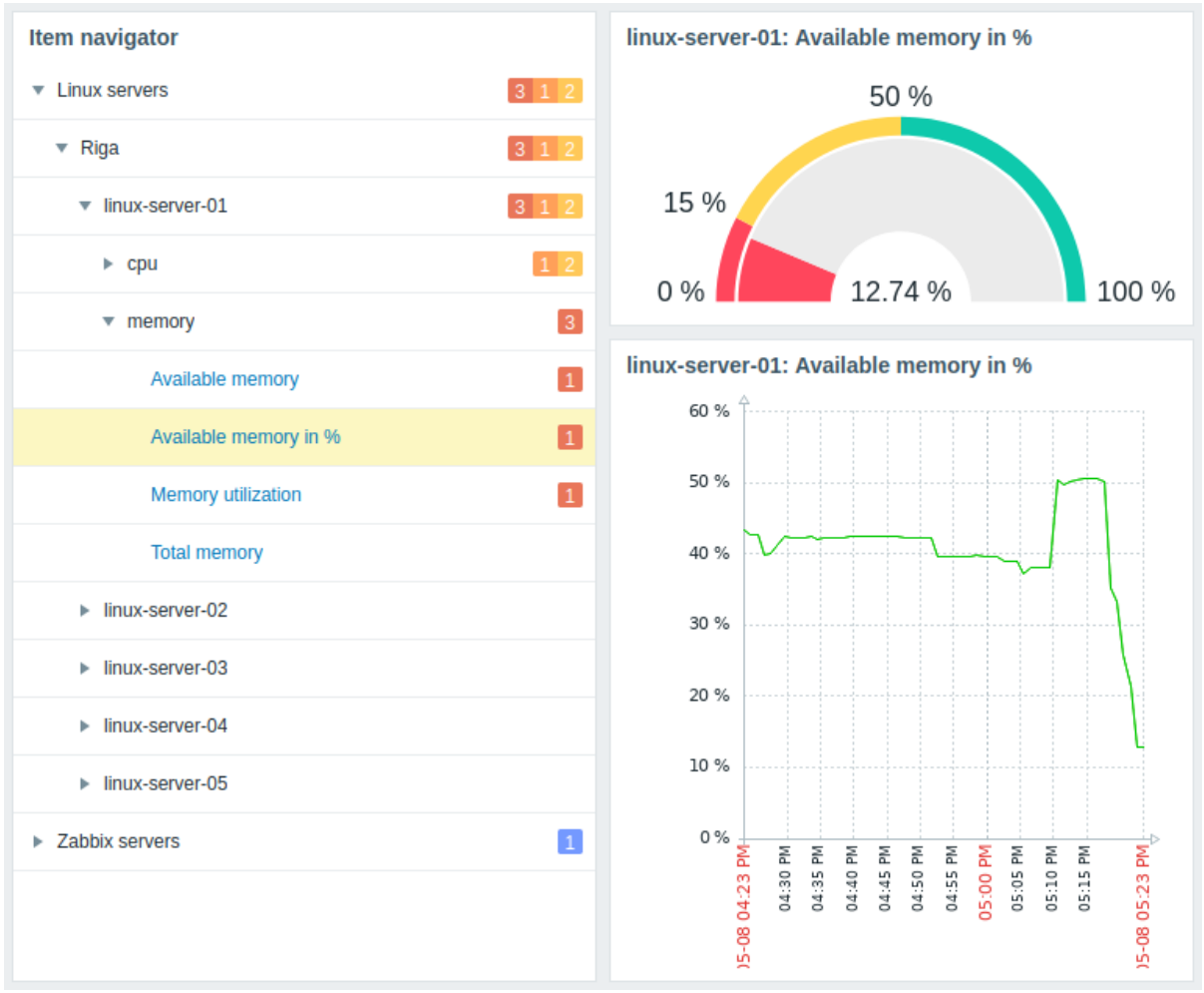
Das Widget *Datenpunkt-Navigator* zeigt eine Liste von Datenpunkten basierend auf verschiedenen Filter- und Gruppierungsoptionen an.



Gruppen, nach denen Datenpunkte organisiert sind, können ein- oder ausgeklappt werden.

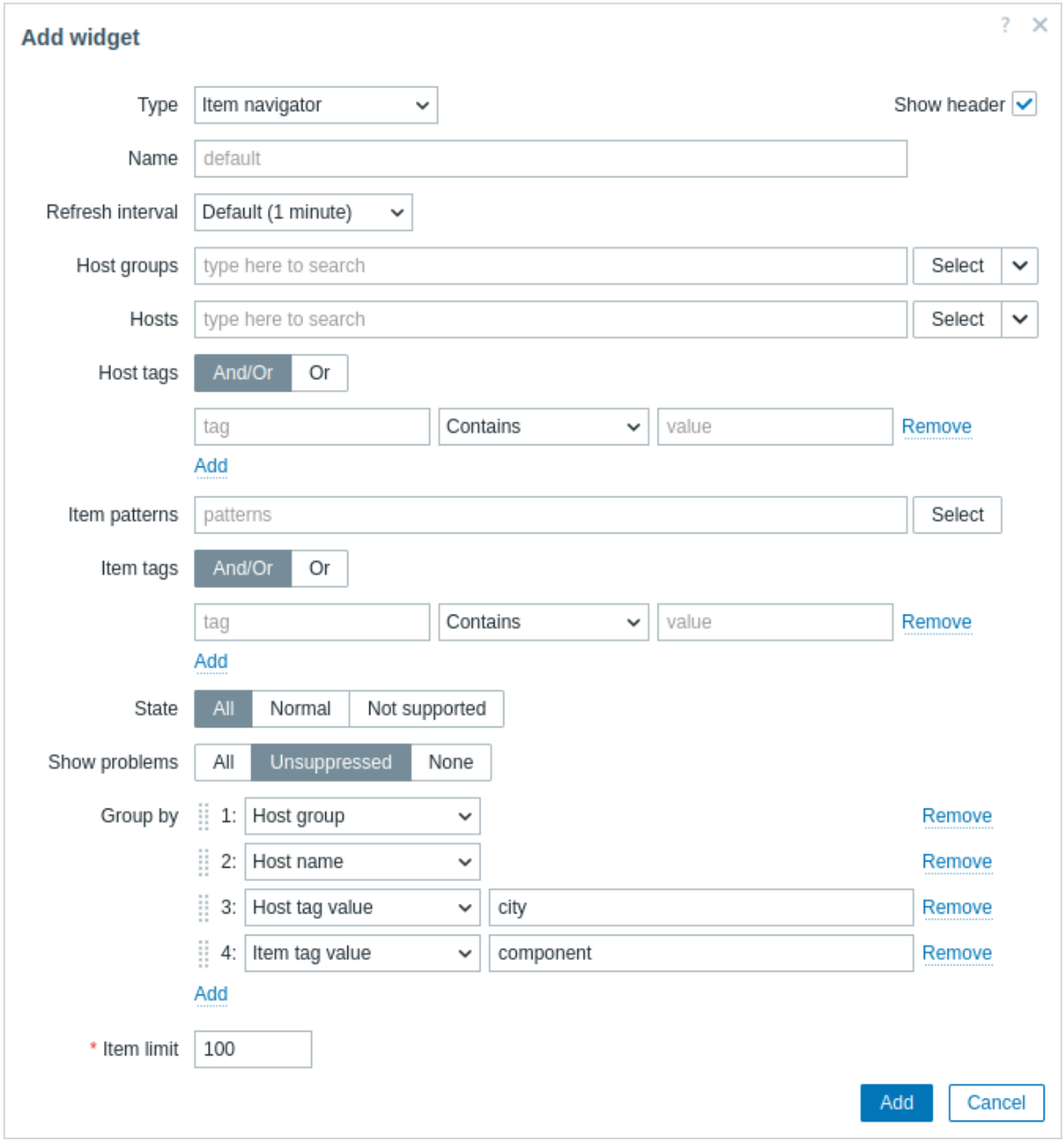
Für Gruppen und Probleme sind zusätzliche Details über Mouseover-Hinweise zugänglich.

Das Widget ist besonders nützlich, um zu steuern, was andere Widgets basierend auf dem ausgewählten Datenpunkt anzeigen.



Konfiguration

Wählen Sie zur Konfiguration *Datenpunkt-Navigator* 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 also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Die Auswahl einer übergeordneten Host-Gruppe wählt implizit auch alle untergeordneten Host-Gruppen aus; wenn keine Host-Gruppen ausgewählt sind, zeigt das Widget Datenpunkte an, die zu allen Hosts aus allen Host-Gruppen 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 verfügt über eine Autovervollständigung. Wenn Sie also 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.

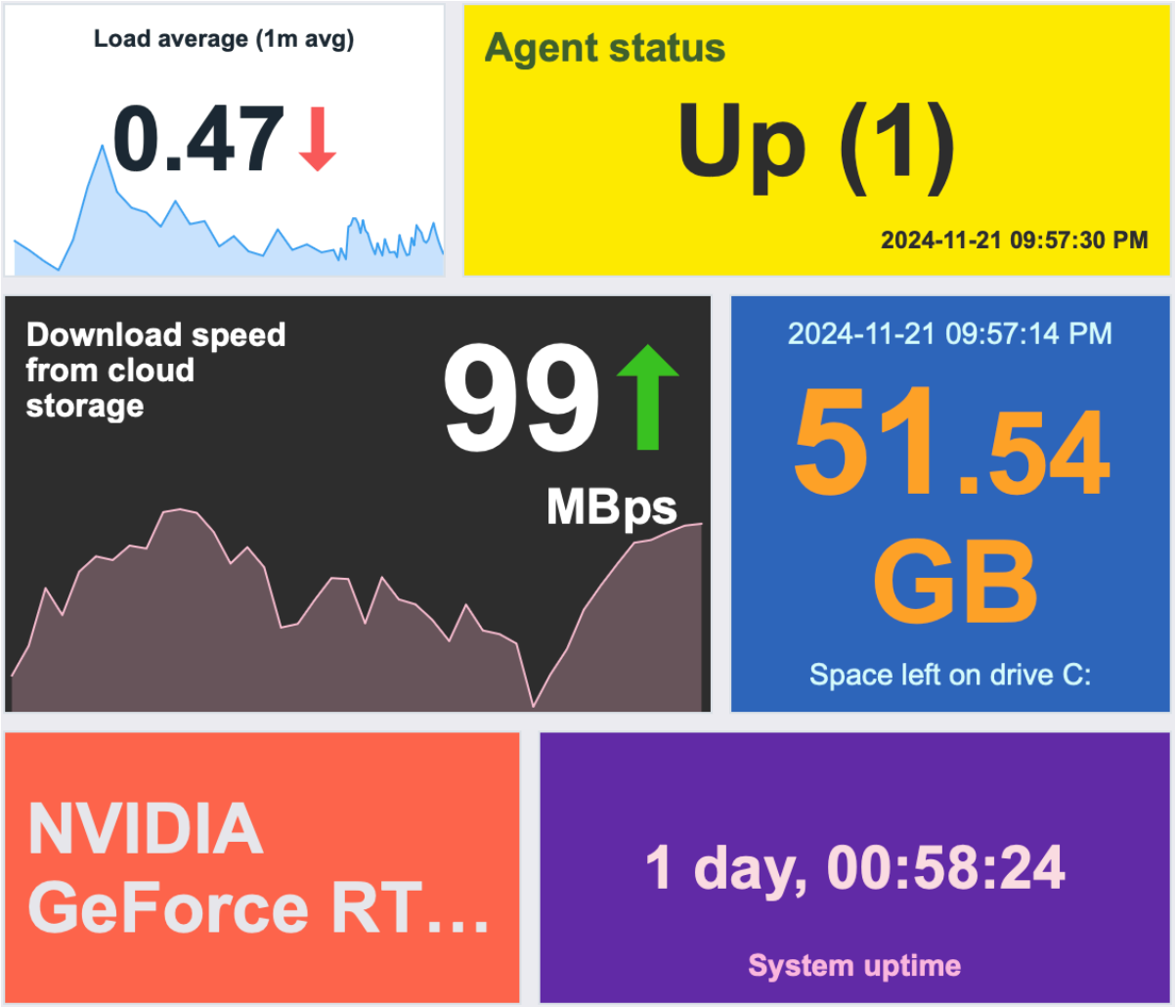
Host-Tags	Geben Sie Tags an, um die im Widget angezeigten Datenpunkte zu filtern. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede 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.
Datenpunkt-Muster	Geben Sie Datenpunkt-Muster ein oder wählen Sie vorhandene Datenpunkte als Datenpunkt-Muster aus. Datenpunkte, die den angegebenen Mustern entsprechen, werden im Datenpunkt-Navigator angezeigt. 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 im Widget angezeigten Datenpunkte 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. 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.

Status	Filtern Sie, welche Datenpunkte basierend auf ihrem Status angezeigt werden sollen (alle, normal, nicht unterstützt).
Probleme anzeigen	Filtern Sie, welche Probleme basierend auf ihrem Status angezeigt werden sollen (alle, nicht unterdrückt, keine). Probleme werden durch farbige Blöcke am Ende der Datenpunkt-Zeile angezeigt. Die Farben basieren auf dem Schweregrad des Problems, der unter *Administration* > *Allgemein* > *Optionen zur Auslöser-Anzeige* angepasst werden kann. Problem-Schweregrade können beim Aktualisieren von Problemen geändert werden.
Gruppieren nach	Fügen Sie ein Gruppierungsattribut hinzu, nach dem Datenpunkte gruppiert werden sollen: **Host-Gruppe** - Datenpunkte nach den Host-Gruppen ihrer Hosts gruppieren; **Host-Name** - Datenpunkte nach ihren Hosts gruppieren; **Host-Tag-Wert** - geben Sie einen Tag-Namen ein, um Datenpunkte nach den Werten dieses Host-Tags zu gruppieren (geben Sie zum Beispiel „city“ ein, um Datenpunkte nach Werten wie „Riga“, „Tokyo“ usw. zu gruppieren); **Datenpunkt-Tag-Wert** - geben Sie einen Tag-Namen ein, um Datenpunkte nach den Werten dieses Datenpunkt-Tags zu gruppieren (geben Sie zum Beispiel „component“ ein, um Datenpunkte nach Werten wie „cpu“, „memory“ usw. zu gruppieren). Gruppierungsattribute können neu angeordnet werden, indem sie über den Ziehpunkt vor dem Gruppennamen nach oben oder unten gezogen werden. Beachten Sie, dass die Reihenfolge der Gruppierungsattribute die Verschachtelungsreihenfolge der Gruppen bestimmt. Wenn zum Beispiel mehrere Host-Tag-Namen angegeben werden (1: color, 2: city), werden Datenpunkte zuerst nach color (red, blue usw.) und dann nach city (Riga, Tokyo usw.) gruppiert. Ein Datenpunkt kann abhängig von den konfigurierten Gruppierungsattributen in mehreren Gruppen angezeigt werden (zum Beispiel, wenn nach Host-Gruppe gruppiert wird und der Host des Datenpunkts zu mehreren Host-Gruppen gehört). Beim Anklicken solcher Datenpunkte werden sie in allen Gruppen ausgewählt und hervorgehoben. Datenpunkte, die nicht mit den konfigurierten Gruppierungsattributen übereinstimmen, werden in der Gruppe *Nicht kategorisiert* angezeigt. Wenn *Probleme anzeigen* so konfiguriert ist, dass Probleme angezeigt werden, werden sie wie folgt dargestellt: - für jede Gruppe wird die Gesamtzahl der Probleme aller untergeordneten Datenpunkte angezeigt; - für jeden Datenpunkt wird nur seine Problemanzahl angezeigt. Es können bis zu 10 Gruppierungsattribute angegeben werden, und alle müssen eindeutig sein. Wenn keine Gruppierungsattribute angegeben sind, werden Datenpunkte nicht gruppiert.
Datenpunkt-Limit	Geben Sie die maximale Anzahl anzuzeigender Datenpunkte ein. Mögliche Werte liegen im Bereich von 1 bis 9999. Wenn mehr Datenpunkte zur Anzeige verfügbar sind als das festgelegte Limit, wird unterhalb der angezeigten Datenpunkte eine entsprechende Meldung eingeblendet (zum Beispiel „100 von 100+ Datenpunkten werden angezeigt“). Beachten Sie, dass sich das konfigurierte Datenpunkt-Limit auch auf die Anzeige der konfigurierten Gruppen auswirkt; wenn zum Beispiel das Datenpunkt-Limit auf 100 gesetzt ist und Datenpunkte nach ihren Hosts gruppiert werden (jeweils mit 200 Datenpunkten), wird im Widget nur der erste Host mit seinen 100 Datenpunkten angezeigt. Dieser Parameter wird nicht durch den Parameter *Limit for search and filter results* unter *Administration → General → GUI* beeinflusst.

18 Datenpunkt-Wert

Übersicht

Das Widget *Datenpunktwert* 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.



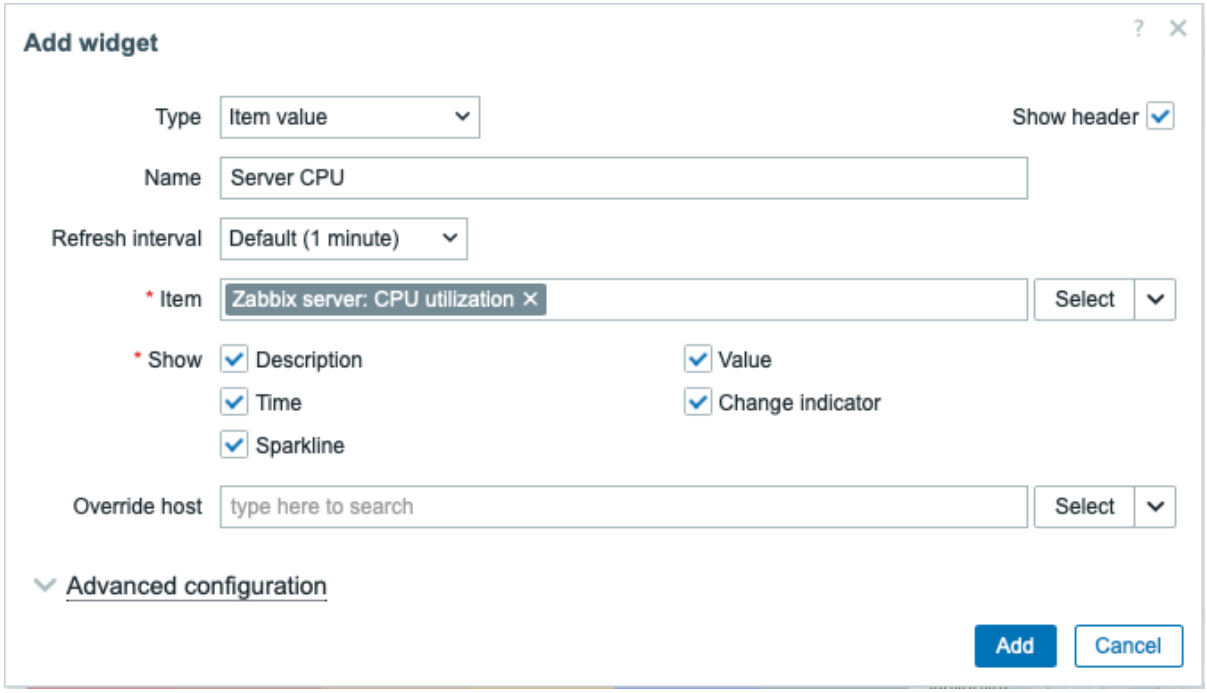
Sie können das Widget so konfigurieren, dass Folgendes angezeigt wird:

- Zeitstempel der Metrik (2024-11-21 09:57:30)
- Beschreibung des Datenpunkts (Load average (1m), Agent-Status)
- Datenpunktwert, Änderungsindikator und Einheiten (0.47 ↓, 99 ↑ Mbps)
- Hintergrundfarbe
- Sparkline-Diagramm für Werte aus dem angegebenen Zeitraum

Durch Klicken auf das Widget wird für numerische Datenpunkte ein Ad-hoc-Diagramm oder für Zeichenfolgen-Datenpunkte die letzte Datenansicht geöffnet.

Konfiguration

Um die Konfiguration vorzunehmen, 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 Datenpunkte 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 das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Erweiterte Konfiguration	Klicken Sie auf die Beschriftung *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

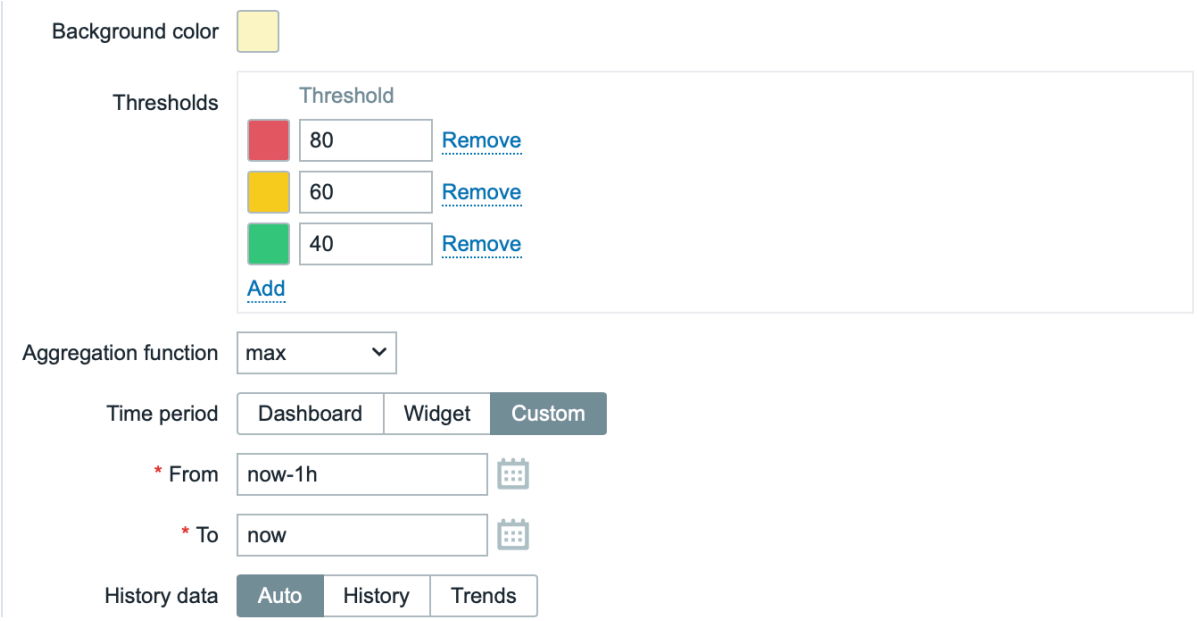
Erweiterte Konfiguration

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.

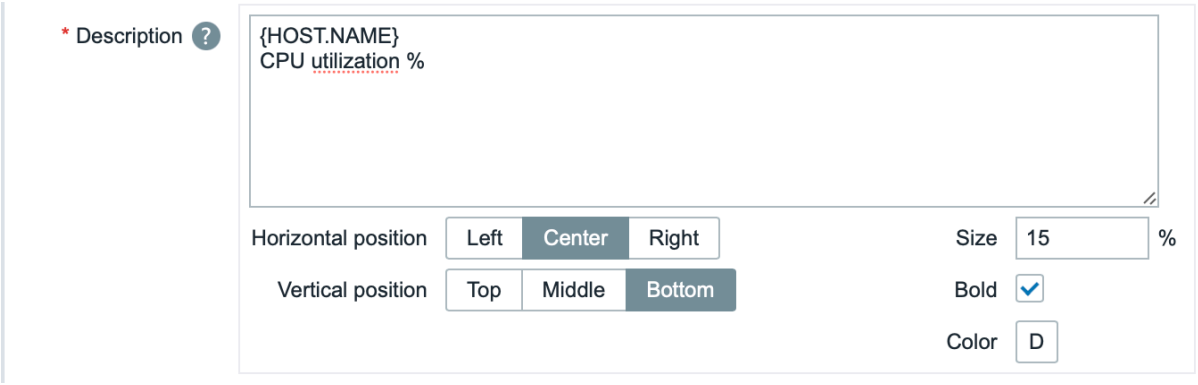


Hintergrundfarbe	Wählen Sie die Hintergrundfarbe für das gesamte Widget im Farbwähler aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie im Farbwähler auf die Schaltfläche *Standard verwenden*.
Schwellenwerte	Konfigurieren Sie die dynamische Hintergrundfarbe für das gesamte Widget. Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie die Hintergrundfarbe im Farbwähler aus und geben Sie einen numerischen Wert an. Sobald der Datenpunktwert 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 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; **not used** - den zuletzt empfangenen Wert anzeigen (keine Aggregation). Mit der Aggregation kann für das gewählte Intervall (5 Minuten, eine Stunde, ein Tag) ein aggregierter Wert anstelle des zuletzt empfangenen Werts angezeigt werden. Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Bei *count* werden nicht numerische Daten in numerische umgewandelt.
Zeitraum	Zeitraum, der für die Aggregation von Werten verwendet wird. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswahl des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Custom** - 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 die eingestellte Zeit anzeigt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration des *Zeitraums* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können. Dieser Parameter ist nicht verfügbar, wenn *Aggregationsfunktion* 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 *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 "Custom" 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 "Custom" gesetzt ist.

Verlaufsdaten	Daten aus Verlauf oder Trends verwenden: **Auto** - automatische Auswahl; **History** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden. Diese Einstellung gilt nur für numerische Daten. Nicht numerische Daten werden immer aus dem Verlauf übernommen.

Beschreibung

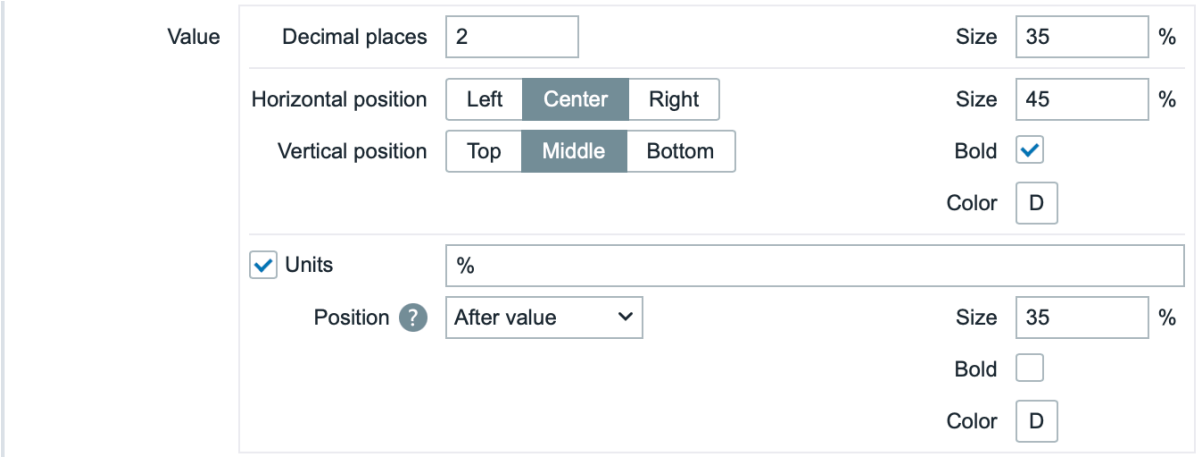
Diese Parameter bestimmen, wie die Beschreibung des Datenpunkts angezeigt werden soll.



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

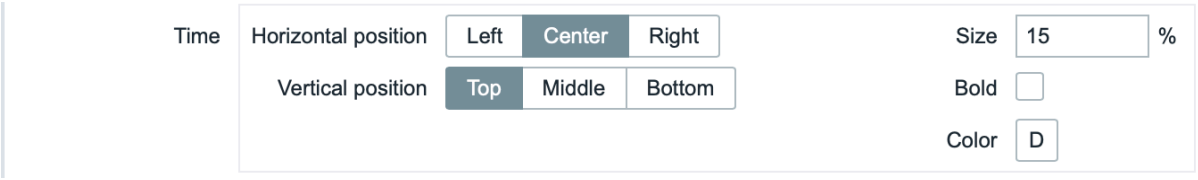
Diese Parameter bestimmen, wie der Datenpunktwert angezeigt werden soll.



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 kann die Farbe der Änderungsindikatoren über die Farbauswahl ausgewählt werden.

Die Änderungsindikatoren sind wie folgt:

- ↑ - Wert des Datenpunkts steigt (für numerische Datenpunkte)
- ↓ - Wert des Datenpunkts sinkt (für numerische Datenpunkte)
- ↕ - Wert des Datenpunkts hat sich geändert (für Zeichenfolgen-Datenpunkte und Datenpunkte mit Wertezuordnung)

Hinweis:

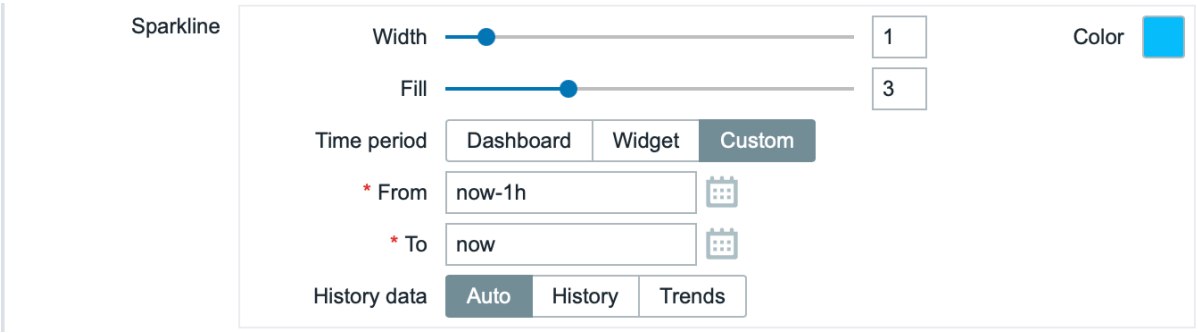
- Aufwärts- (↑) und Abwärtsindikatoren (↓) werden nicht angezeigt, wenn nur ein Wert vorhanden ist.
- Werte werden immer innerhalb desselben Bereichs verglichen; zum Beispiel:
  - Der letzte Wert wird mit dem vorherigen Wert verglichen.
  - Ein Monatswert wird mit dem vorherigen Monat verglichen.
  - Bei Aggregationen hat der vorherige Zeitraum dieselbe Dauer wie der ausgewählte Zeitraum und endet unmittelbar, bevor der ausgewählte Zeitraum beginnt.

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 (dem ganzzahligen Teil des Werts bei numerischen Datenpunkten).

Sparkline

Diese Parameter bestimmen, wie das Sparkline-Diagramm angezeigt werden soll.

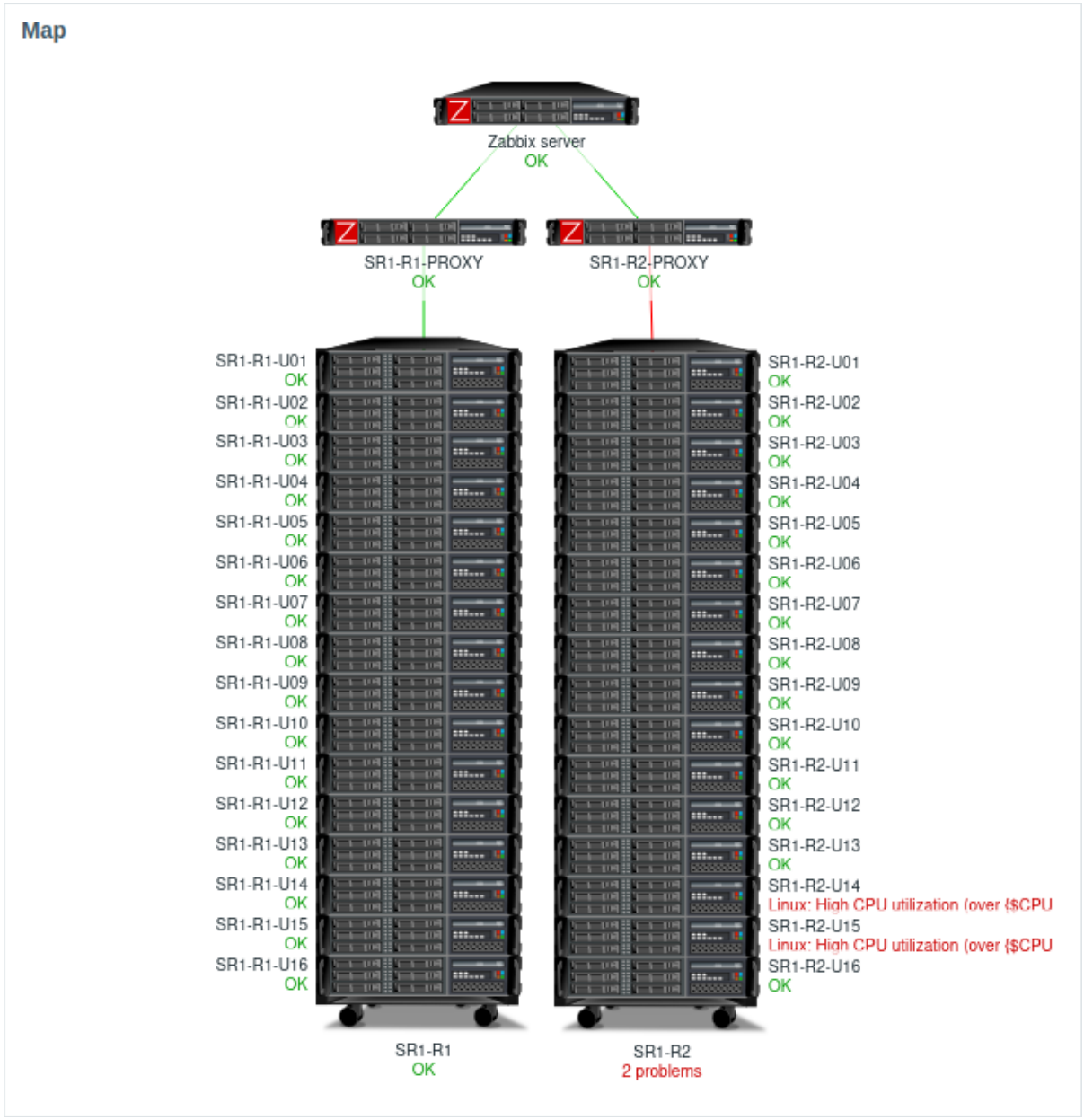


Breite	Legen Sie die Dicke der Diagrammlinie fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.
Farbe	Wählen Sie die Linien- und Füllfarbe aus.
Füllung	Legen Sie die Transparenzstufe der Füllfarbe fest, indem Sie den Schieberegler verwenden oder manuell 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.

19 Karte

Übersicht

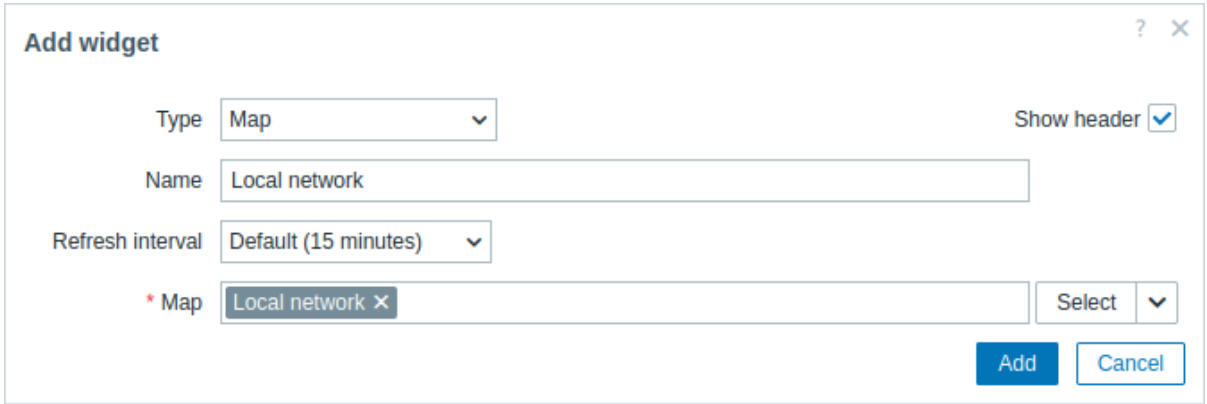
Das Widget *Karte* kann eine Netzwerkkarte oder eine ähnliche Visualisierung anzeigen und Ihnen so eine dynamische Übersicht über Ihr Netzwerk bieten.



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:



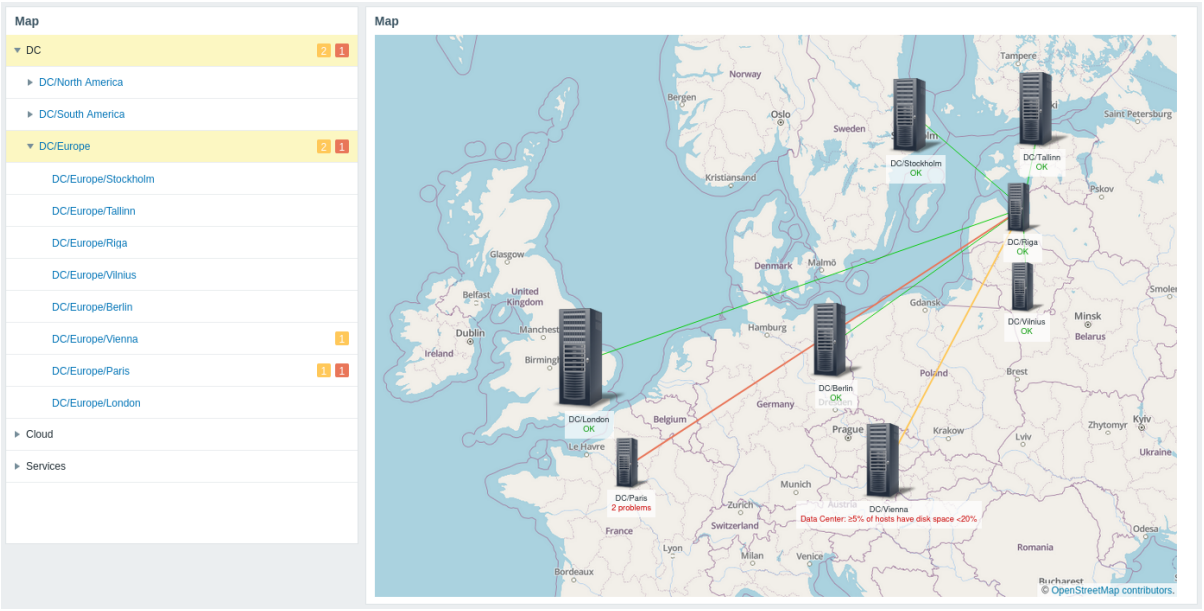
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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.

20 Navigationsbaum der Karte

Übersicht

Das Widget *Map navigation tree* zeigt eine Hierarchie vorhandener Karten sowie die Anzahl der Probleme für jede Karte und Kartengruppe an.

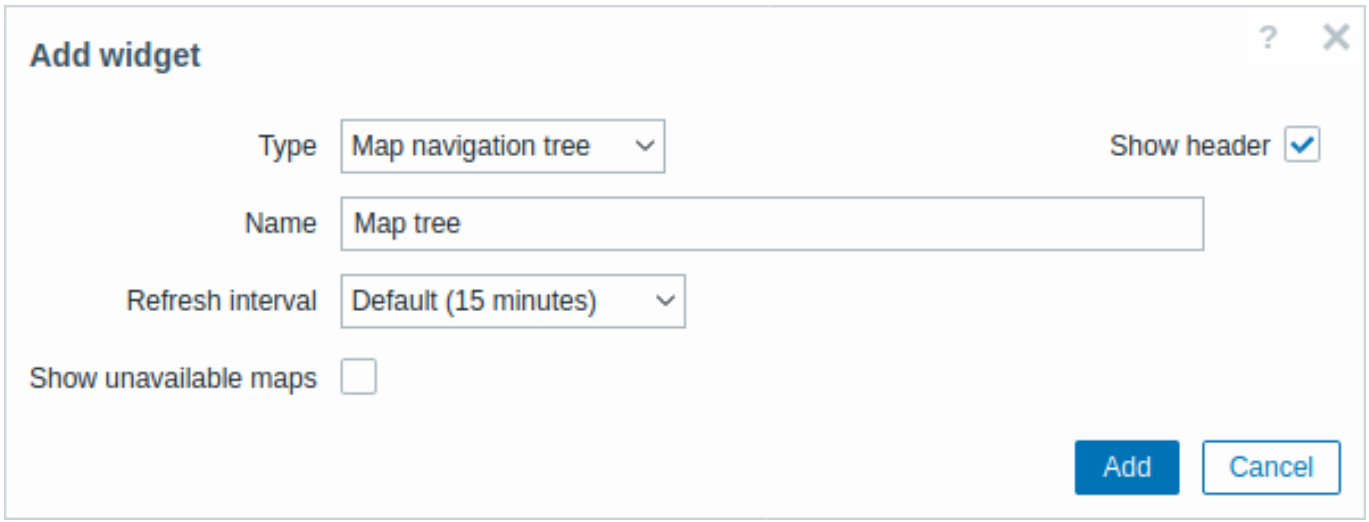


Sie können es mit dem Widget Map kombinieren, um die in der Baumstruktur ausgewählte Karte dynamisch anzuzeigen.

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:

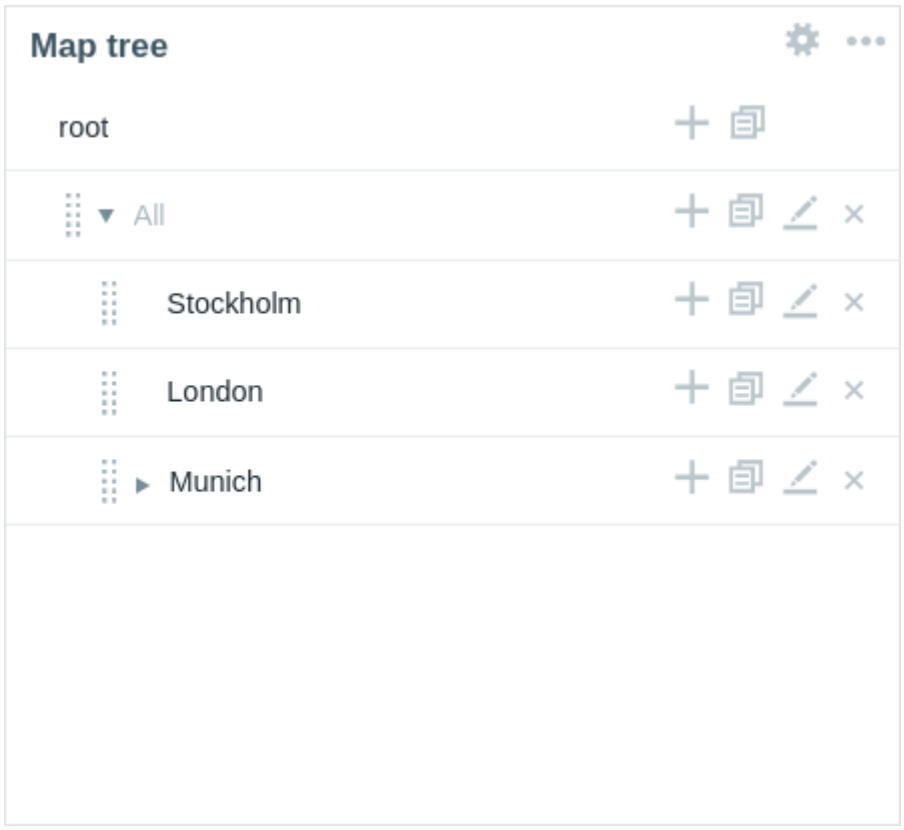


Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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.

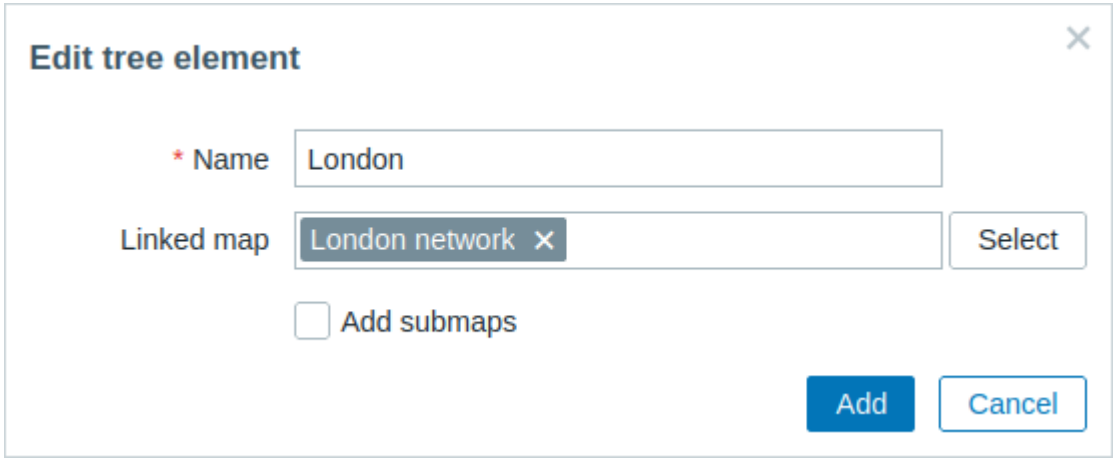
Elemente des Navigationsbaums werden in einer Liste angezeigt. Sie können:

- 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.



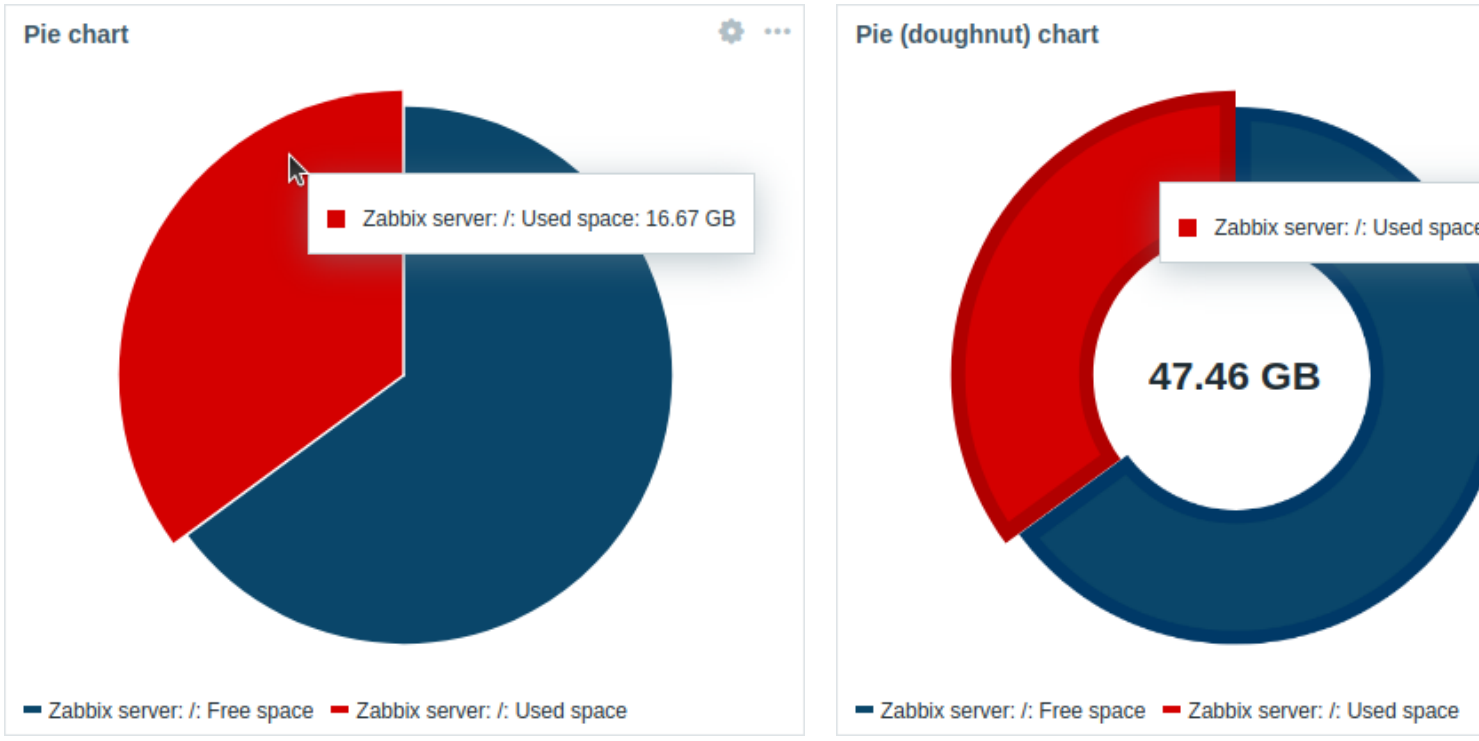
Die folgenden Konfigurationsparameter für Navigationselemente der Baumansicht sind verfügbar:

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.

21 Kreisdiagramm

Übersicht

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.

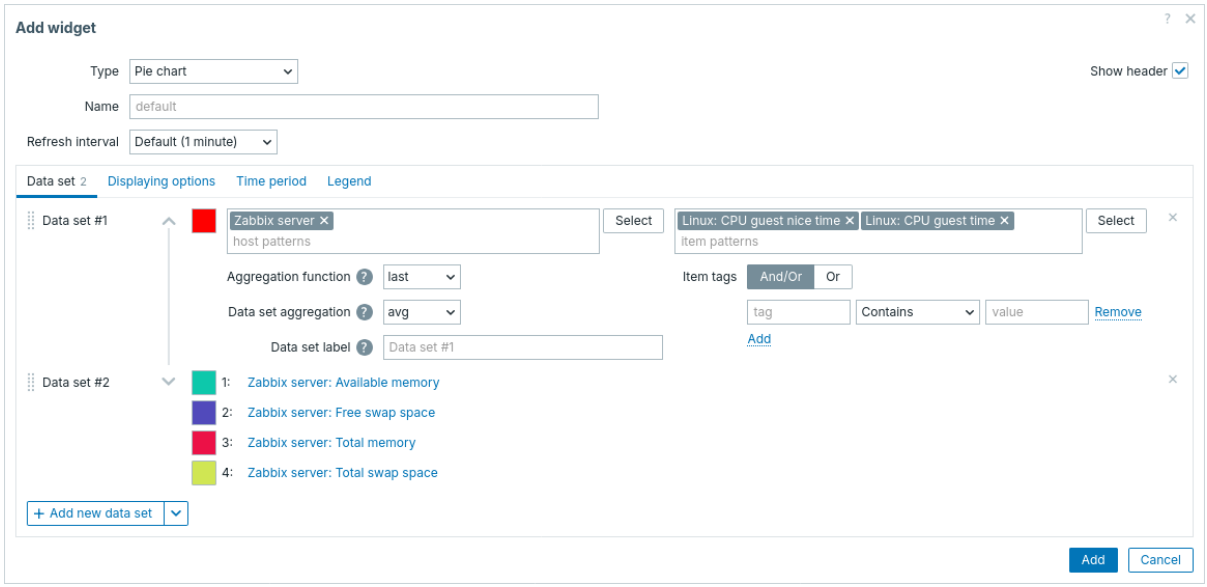


Beim Überfahren mit der Maus wird der hervorgehobene Sektor erweitert und seine Legende angezeigt; durch Klicken auf den Sektor bleibt er erweitert und seine Legende sichtbar, bis er geschlossen wird.

Die im Widget *Kreisdiagramm* angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem Sie im Widget-Menü die Option *Bild herunterladen* auswählen.

Konfiguration

Wählen Sie zur Konfiguration *Kreisdiagramm* als Typ aus:



Datensatz

Die Registerkarte **Datensatz** ermöglicht die Auswahl von Daten für das Kreisdiagramm durch Hinzufügen von Datensätzen. Es können zwei Arten von Datensätzen hinzugefügt werden:

- *Datenpunkt-Muster* - Daten aus übereinstimmenden Datenpunkten werden angezeigt. Sie können eine einzelne Grundfarbe auswählen oder eine Palettenzeile wählen, um jedem übereinstimmenden Datenpunkt unterschiedliche Farben zuzuweisen.
- *Datenpunkt-Liste* - Daten aus ausgewählten Datenpunkten werden angezeigt. Sie können die Farbe jedes Datenpunkts einzeln im Farbwähler auswählen.

Standardmäßig wird ein Datensatz vom Typ *Datenpunkt-Muster* hinzugefügt.

Datensatz	Für den Datensatz **Datenpunkt-Muster**: Wählen Sie Host- und Datenpunkt-Muster aus oder geben Sie sie ein; die Daten der Datenpunkte, die den eingegebenen Mustern entsprechen, werden im Kreisdiagramm angezeigt; es können bis zu 50 Datenpunkte angezeigt werden. Für die Auswahl können Platzhaltermuster verwendet werden (zum Beispiel liefert * Ergebnisse zurück, 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 nicht möglich, zum Beispiel einen Datenpunkt mit dem Namen *item** einzeln hinzuzufügen, wenn es andere passende Datenpunkte gibt (zum Beispiel *item2*, *item3*). Die Angabe von Host- und Datenpunkt-Mustern ist für Datensätze vom Typ ”Datenpunkt-Muster” obligatorisch. Siehe auch: Details zur Datensatzkonfiguration. Für den Datensatz **Datenpunkt-Liste**: Wählen Sie Datenpunkte für das Kreisdiagramm aus, indem Sie auf die Schaltfläche *Datenpunkt hinzufügen* klicken. Die Dropdown-Liste *Typ* nach dem Datenpunktnamen ermöglicht die Auswahl des Anzeigetyps für jeden Datenpunkt: **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 in der Legende des Kreisdiagramms an erster Stelle angezeigt 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 ”Datenpunkt-Liste” obligatorisch. Siehe auch: Details zur Datensatzkonfiguration. Beachten Sie, dass nur numerische Datenpunkttypen zulässig sind. Beim Konfigurieren des Widgets in einem Vorlagen-Dashboard ist der Parameter zur Angabe von Host-Mustern nicht verfügbar, und der Parameter zur Angabe einer Datenpunkt-Liste erlaubt nur die Auswahl der in der Vorlage konfigurierten Datenpunkte.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion für jeden Datenpunkt im Datensatz 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). Die Aggregation ermöglicht die Anzeige eines aggregierten Werts für das Intervall (5 Minuten, eine Stunde, ein Tag), das in der Registerkarte *Zeitraum* ausgewählt wurde oder für das gesamte Dashboard verwendet wird.

Datensatzaggregation	Geben Sie an, welche Aggregationsfunktion für den gesamten Datensatz verwendet werden soll: **not used** - 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. Die Aggregation ermöglicht die Anzeige eines aggregierten Werts für das Intervall (5 Minuten, eine Stunde, ein Tag), das in der Registerkarte *Zeitraum* ausgewählt wurde 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 angezeigt (bei aggregierten Datensätzen). Alle Datensätze werden nummeriert, auch diejenigen mit einer angegebenen *Datensatzbezeichnung*. Wenn keine Bezeichnung angegeben ist, wird der Datensatz automatisch entsprechend seiner Nummer bezeichnet (z. B. ”Datensatz #2”, ”Datensatz #3” usw.). Die Nummerierung der Datensätze wird nach dem Umordnen/Ziehen von Datensätzen neu berechnet. Zu lange Datensatzbezeichnungen werden gekürzt, damit sie in den verfügbaren Platz passen (z. B. ”Anzahl der Pro...”).
Datenpunkt-Tags	Geben Sie Tags an, um die im Widget angezeigten Datenpunkte 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. Für jede Bedingung sind mehrere Operatoren verfügbar: **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.

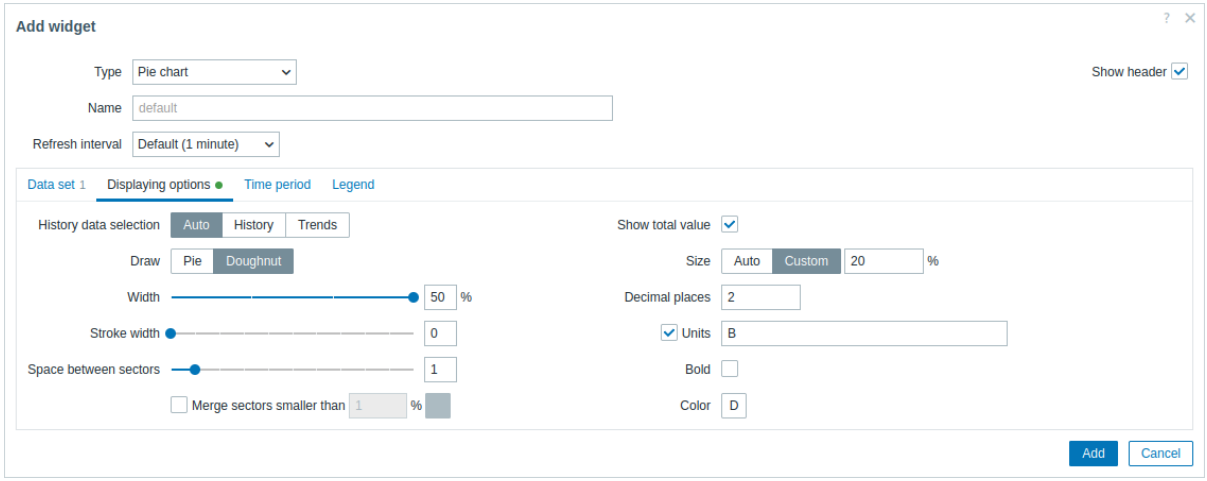
Details zur Datenpunkt-Konfiguration

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

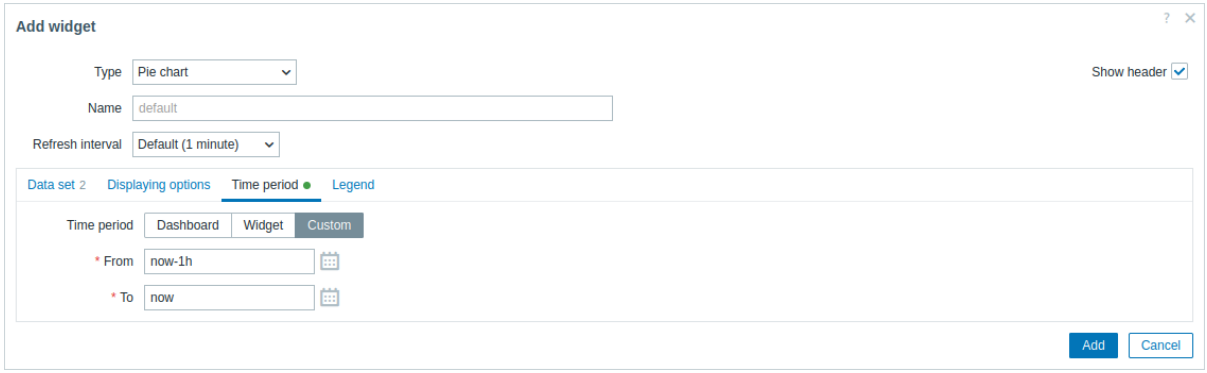
Die Registerkarte **Anzeigeoptionen** ermöglicht es, die Auswahl der Verlaufsdaten und die Visualisierungsoptionen für das Kreisdiagramm festzulegen:



Auswahl der Verlaufsdaten	Wählen Sie die Datenquelle aus: **Auto** - die Daten werden gemäß dem klassischen Algorithmus bezogen (Standard); **Verlauf** - Daten aus dem Verlauf; **Trends** - Daten aus den Trends.
Zeichnen	Wählen Sie den Visualisierungsstil des Kreisdiagramms aus: **Kreis** - ein vollständiger Kreis (Sektoren nehmen 100 % des Radius ein); **Ring** - ein Kreis mit leerem Bereich in der Mitte (Sektoren nutzen bis zu 50 % des Radius).
Abstand zwischen Sektoren	Wählen Sie die Größe des Abstands (in Einheiten von 0-10) zwischen den Sektoren aus (Standard ist „1“).
Sektoren kleiner als N % zusammenführen	Aktivieren Sie das Kontrollkästchen, um Sektoren zusammenzuführen, die kleiner als N % sind. Falls aktiviert, wählen Sie die Farbe für die zusammengeführten Sektoren und den prozentualen Schwellenwert (N) zum Zusammenführen kleiner Sektoren aus.
Zeichenstil:	
Ring	
Breite	Wählen Sie die Ringbreite aus: 20, 30, 40 oder 50 % (Standard) des Radius.
Strichbreite	Wählen Sie die Breite des Sektorrands des Rings 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 sie gut lesbar in die Mitte des Rings passt; **Benutzerdefiniert** - geben Sie die Textgröße als prozentuale Höhe der gesamten Widget-Höhe 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

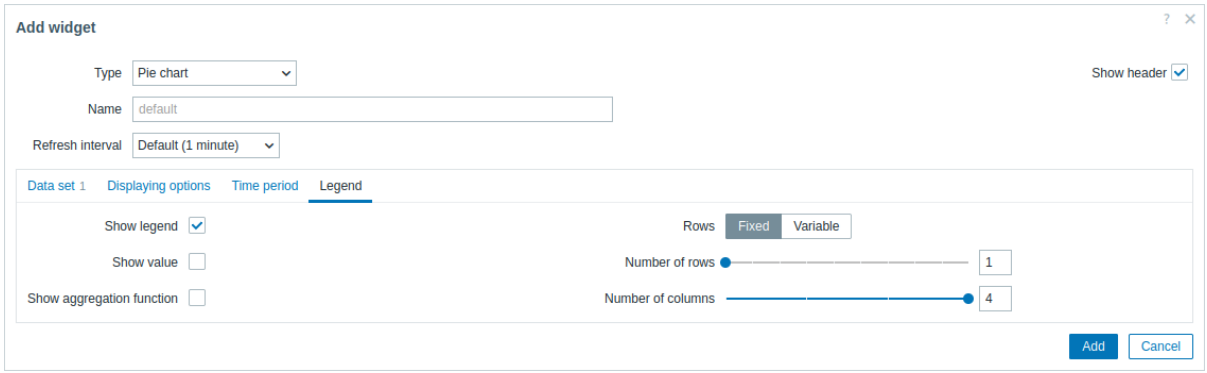
Die Registerkarte **Zeitraum** ermöglicht es, einen benutzerdefinierten Zeitraum für die Aggregationseinstellungen des Kreisdiagramms festzulegen:



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* angegeben ist; wenn dies festgelegt ist, 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 (*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

Die Registerkarte **Legende** ermöglicht die Anpassung der Legende des Kreisdiagramms:



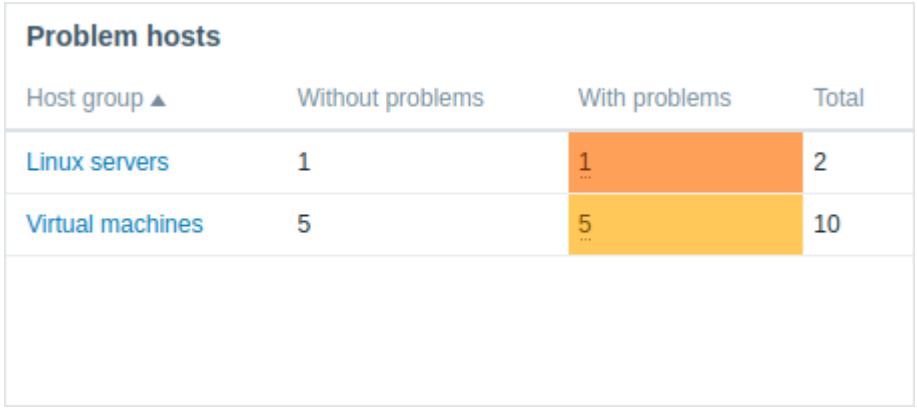
Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende im Kreisdiagramm auszublenden (standardmäßig aktiviert).
Wert anzeigen	Aktivieren Sie dieses Kontrollkästchen, um den Wert des Datenpunkts in der Legende anzuzeigen.
Aggregationsfunktion anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Aggregationsfunktion in der Legende anzuzeigen.
Zeilen	Wählen Sie den Anzeigemodus für die Legendenzeilen 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, ohne den Wert des Parameters *Maximale Anzahl der Zeilen* zu 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.

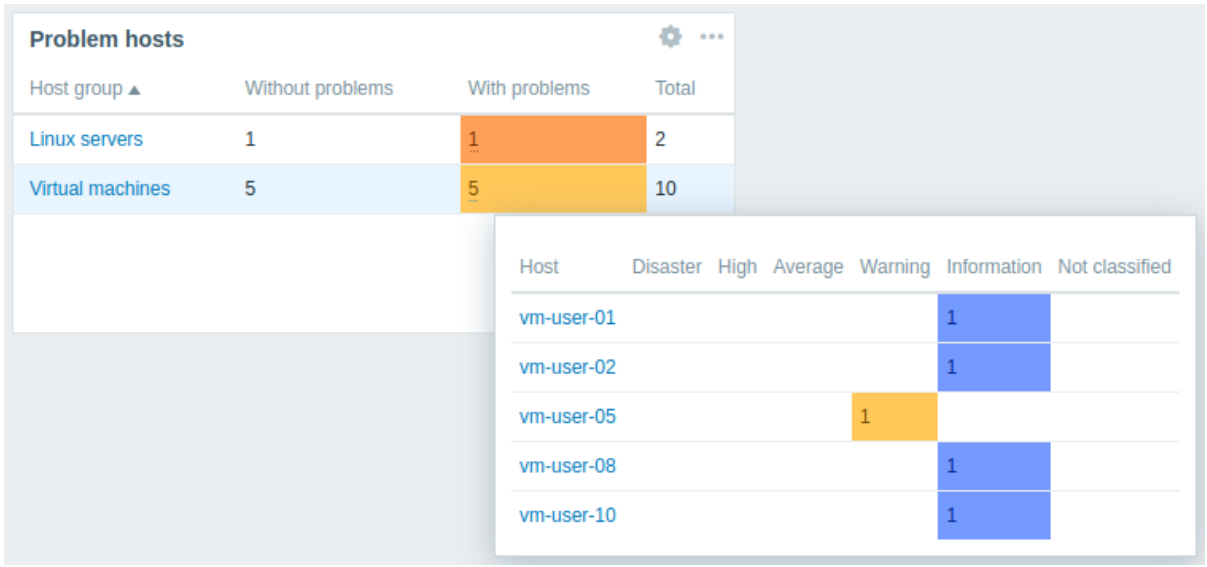
22 Problem-Hosts

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.).

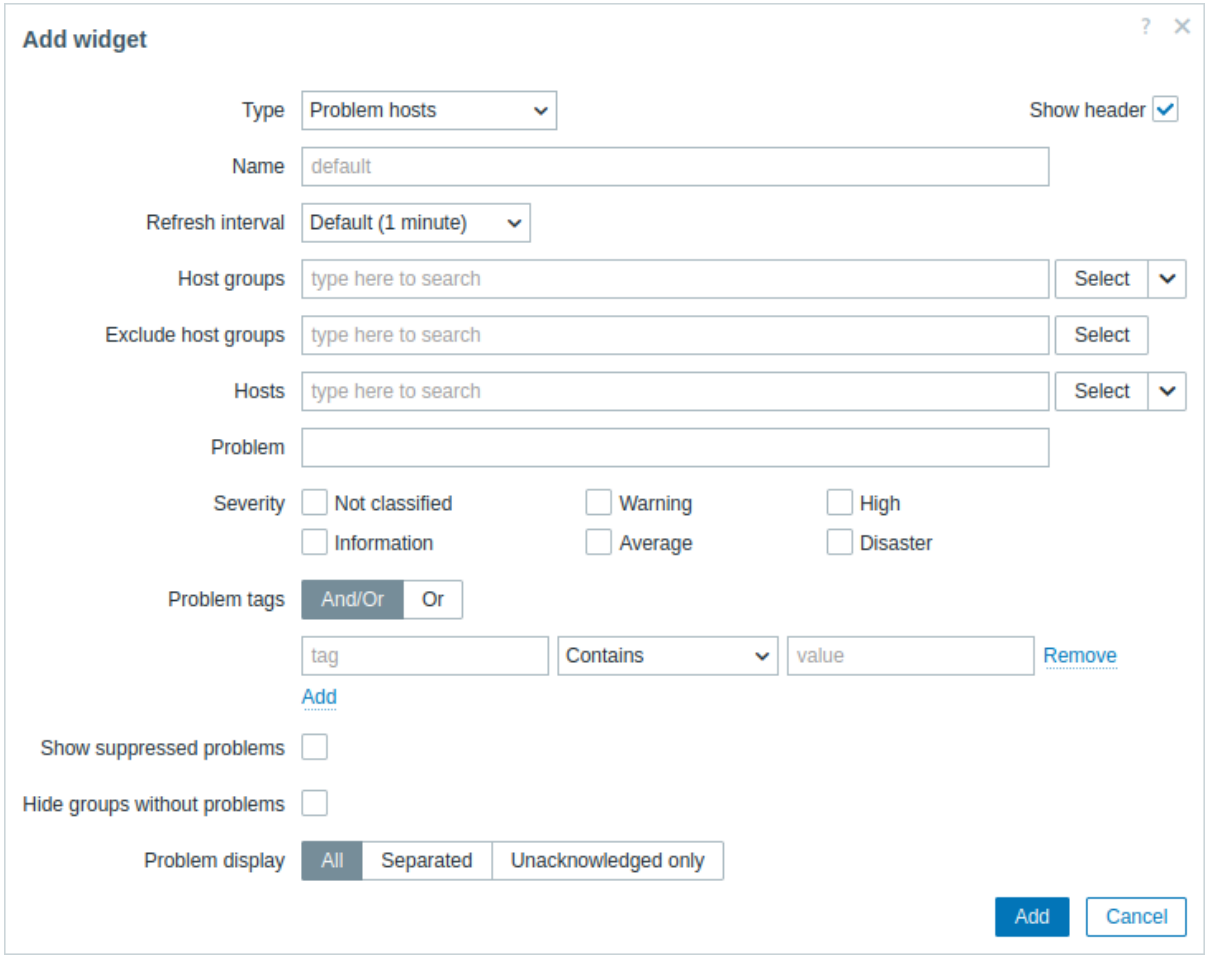


Wenn Sie mit der Maus über eine Problemanzahl fahren, wird die Liste der Hosts mit Problemen angezeigt:



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:

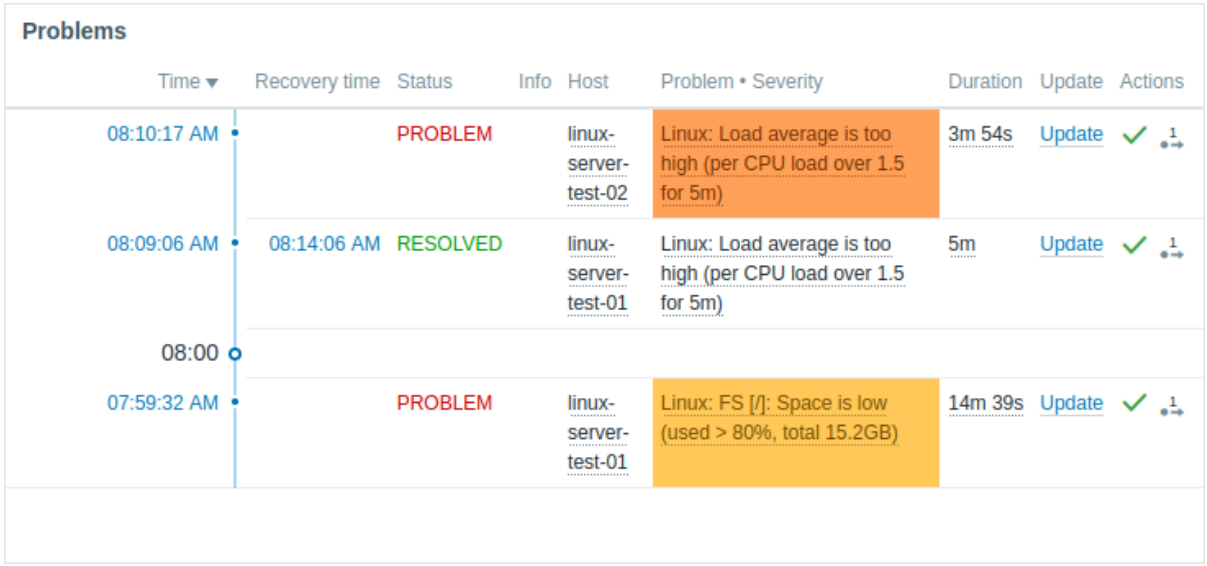
Host-Gruppen	Wählen Sie Host-Gruppen aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung. Wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste 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.

23 Probleme

Übersicht

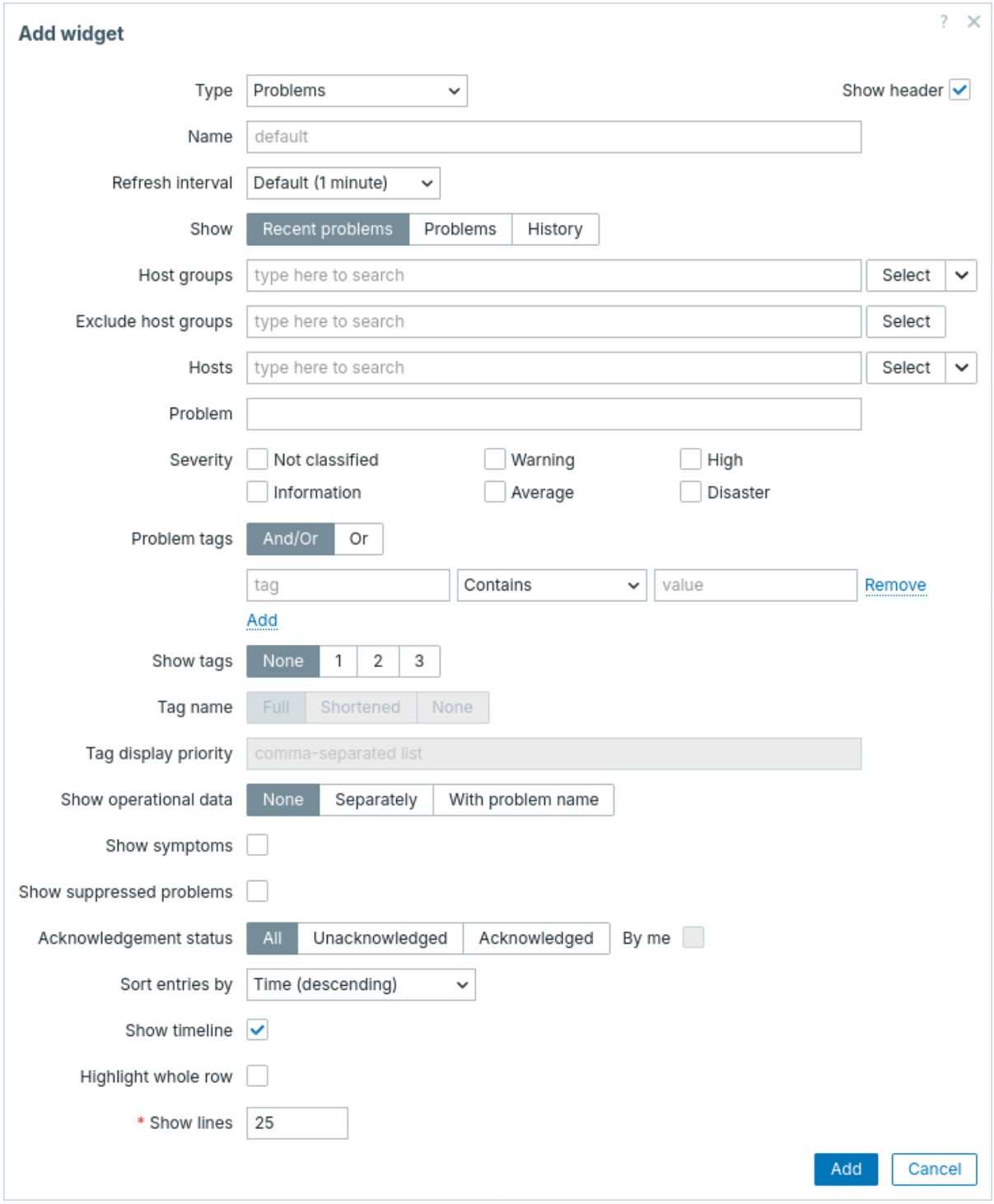
Das Widget *Probleme* zeigt aktuelle Probleme an, gefiltert nach verschiedenen Parametern (Host-Gruppen, Hosts, Problemnamen usw.), und bietet Ihnen so eine klare Übersicht darüber, was Aufmerksamkeit erfordert.



Es zeigt dieselben Daten wie *Monitoring* > *Probleme* an und kann bis zu 1000 Einträge anzeigen.

Konfiguration

Wählen Sie zur Konfiguration *Probleme* als Typ aus:



Sie können die Anzahl der im Widget angezeigten Probleme auf verschiedene Weise begrenzen – nach Problemstatus, Problemname, Schweregrad, Host-Gruppe, Host, Ereignis-Tag, Bestätigungsstatus usw.

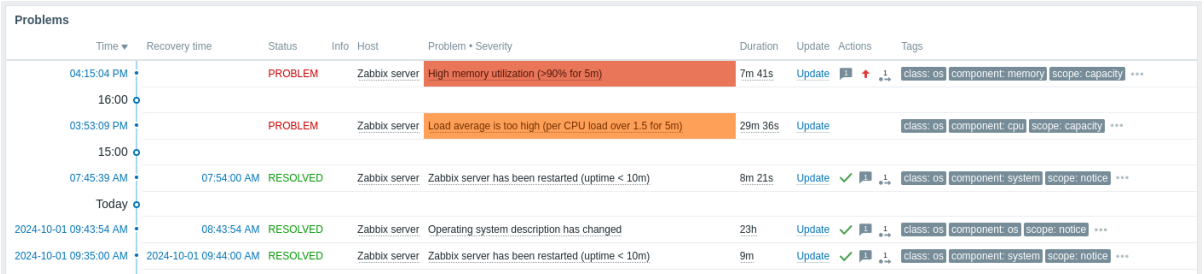
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Anzeigen	Nach Problemstatus filtern: **Aktuelle Probleme** - ungelöste und kürzlich gelöste Probleme werden angezeigt (Standard); **Probleme** - ungelöste Probleme werden angezeigt; **Verlauf** - der Verlauf aller Ereignisse wird angezeigt.

Host-Gruppen	Wählen Sie Host-Gruppen aus, deren Probleme 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. Wenn Sie eine übergeordnete Host-Gruppe angeben, werden implizit alle darunter verschachtelten Host-Gruppen ausgewählt. 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 auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen ausschließen	Wählen Sie Host-Gruppen aus, deren Probleme 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. Wenn Sie eine übergeordnete Host-Gruppe angeben, werden implizit alle darunter verschachtelten Host-Gruppen ausgewählt. Probleme 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 Widget nur Probleme von Host 001 angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-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 Autovervollständigung; wenn Sie 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 auf einem Vorlagen-Dashboard konfiguriert wird.
Problem	Sie können die Anzahl der angezeigten Probleme anhand ihres Namens begrenzen. Wenn Sie hier eine Zeichenfolge eingeben, werden nur die Probleme angezeigt, deren Name die eingegebene Zeichenfolge enthält. Makros werden nicht expandiert.
Schweregrad	Markieren Sie die Schweregrade von Problemen, 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 Bedingung *Oder* gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist. Beim Filtern werden die hier angegebenen Tags beim Problem zuerst angezeigt, sofern dies nicht durch die Liste *Priorität der Tag-Anzeige* (siehe unten) überschrieben wird.

Tags anzeigen	Wählen Sie die Anzahl der angezeigten Tags aus: **Keine** - keine Spalte *Tags*; **1** - die Spalte *Tags* enthält ein Tag; **2** - die Spalte *Tags* enthält zwei Tags; **3** - die Spalte *Tags* enthält drei Tags. Um alle Tags für das Problem zu sehen, bewegen Sie den Mauszeiger über das Symbol mit den drei Punkten.
Tag-Name	Wählen Sie den Anzeigemodus für Tag-Namen aus: **Vollständig** - Tag-Namen und -Werte werden vollständig angezeigt; **Verkürzt** - Tag-Namen werden auf 3 Zeichen gekürzt, Tag-Werte werden jedoch vollständig angezeigt; **Keine** - nur Tag-Werte werden angezeigt; keine Namen.
Priorität der Tag-Anzeige	Geben Sie die Anzeigereihenfolge der Tags für ein Problem als kommagetrennte Liste von Tags ein. Es sollten nur Tag-Namen verwendet werden, keine Werte. Beispiel: `Services,Applications,Application` Die Tags aus dieser Liste werden immer zuerst angezeigt und überschreiben die natürliche alphabetische Reihenfolge.
Betriebsdaten anzeigen	Wählen Sie den Modus für die Anzeige von Betriebsdaten: **Keine** - es werden keine Betriebsdaten angezeigt; **Getrennt** - Betriebsdaten werden in einer separaten Spalte angezeigt; **Mit Problemname** - Betriebsdaten werden an den Problemnamen angehängt, wobei für die Betriebsdaten Klammern verwendet werden.
Symptome anzeigen	Aktivieren Sie das Kontrollkästchen, um Probleme, die als Symptome klassifiziert sind, in einer eigenen Zeile anzuzeigen.
Unterdrückte Probleme anzeigen	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung oder einzelner Problemunterdrückung unterdrückt (nicht angezeigt) würden.
Bestätigungsstatus	Filtern Sie, um alle Probleme, nur unbestätigte Probleme oder nur bestätigte Probleme anzuzeigen. Aktivieren Sie das zusätzliche Kontrollkästchen, um diejenigen Probleme herauszufiltern, die jemals von Ihnen bestätigt wurden.
Einträge sortieren nach	Einträge sortieren nach: **Zeit** (absteigend oder aufsteigend); **Schweregrad** (absteigend oder aufsteigend); **Problemname** (absteigend oder aufsteigend); **Host** (absteigend oder aufsteigend). Das Sortieren von Einträgen nach **Host** (absteigend oder aufsteigend) ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Zeitachse anzeigen	Aktivieren Sie das Kontrollkästchen, um eine visuelle Zeitachse anzuzeigen.
Ganze Zeile hervorheben	Aktivieren Sie das Kontrollkästchen, um bei ungelösten Problemen die gesamte Zeile hervorzuheben. Für die Hervorhebung wird die Farbe des Problemschweregrads verwendet. *Ganze Zeile hervorheben* ist in Designs mit hohem Kontrast nicht verfügbar.
Zeilen anzeigen	Geben Sie die Anzahl der anzuzeigenden Problemzeilen an.

Verwendung des Widgets



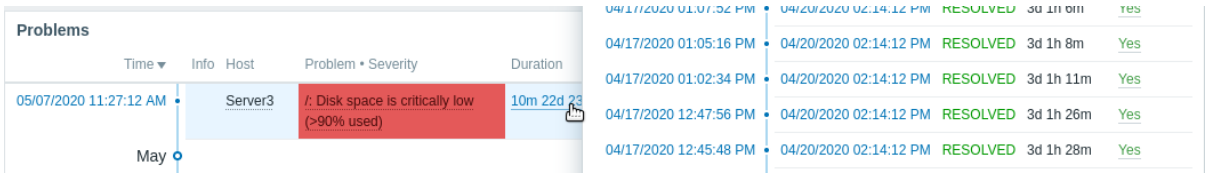
Das Problem-Widget bietet schnellen Zugriff auf zusätzliche Informationen:

- Klicken Sie auf das Problemdatum (in der Spalte Zeit) oder das Wiederherstellungsdatum (in der Spalte Wiederherstellungszeit), um die 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 Host-Namen, um das Host-Menü zu öffnen.

- Klicken Sie auf den Problemnamen, um das Ereignismenü zu öffnen.
- Bewegen Sie den Mauszeiger über die Problemdauer oder klicken Sie darauf, um das Popup-Fenster für Problemereignisse anzuzeigen.
- Durch Drücken von Aktualisieren wird ein Fenster zum Problem aktualisieren geöffnet.
- Bewegen Sie den Mauszeiger über das graue Pfeilsymbol in der Spalte Aktionen oder klicken Sie darauf, um die Liste der ausgeführten Aktionen anzuzeigen.

Popup-Fenster für Problemereignisse

Das Popup-Fenster für Problemereignisse enthält die Liste der Problemereignisse für diesen Auslöser sowie, falls definiert, die Auslöserbeschreibung und eine anklickbare URL.



So rufen Sie das Popup-Fenster für Problemereignisse auf:

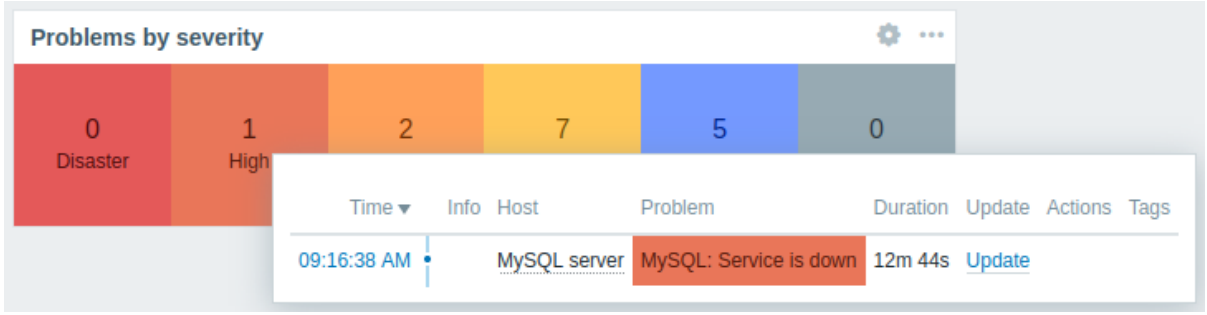
- Bewegen Sie den Mauszeiger über die Problemdauer in der Spalte *Dauer* des Widgets *Probleme*. Das Popup verschwindet, sobald Sie den Mauszeiger von der Dauer wegbewegen.
- Klicken Sie auf die Dauer in der Spalte *Dauer* des Widgets *Probleme*. Das Popup verschwindet erst, wenn Sie erneut auf die Dauer klicken.

24 Probleme nach Schweregrad

Übersicht

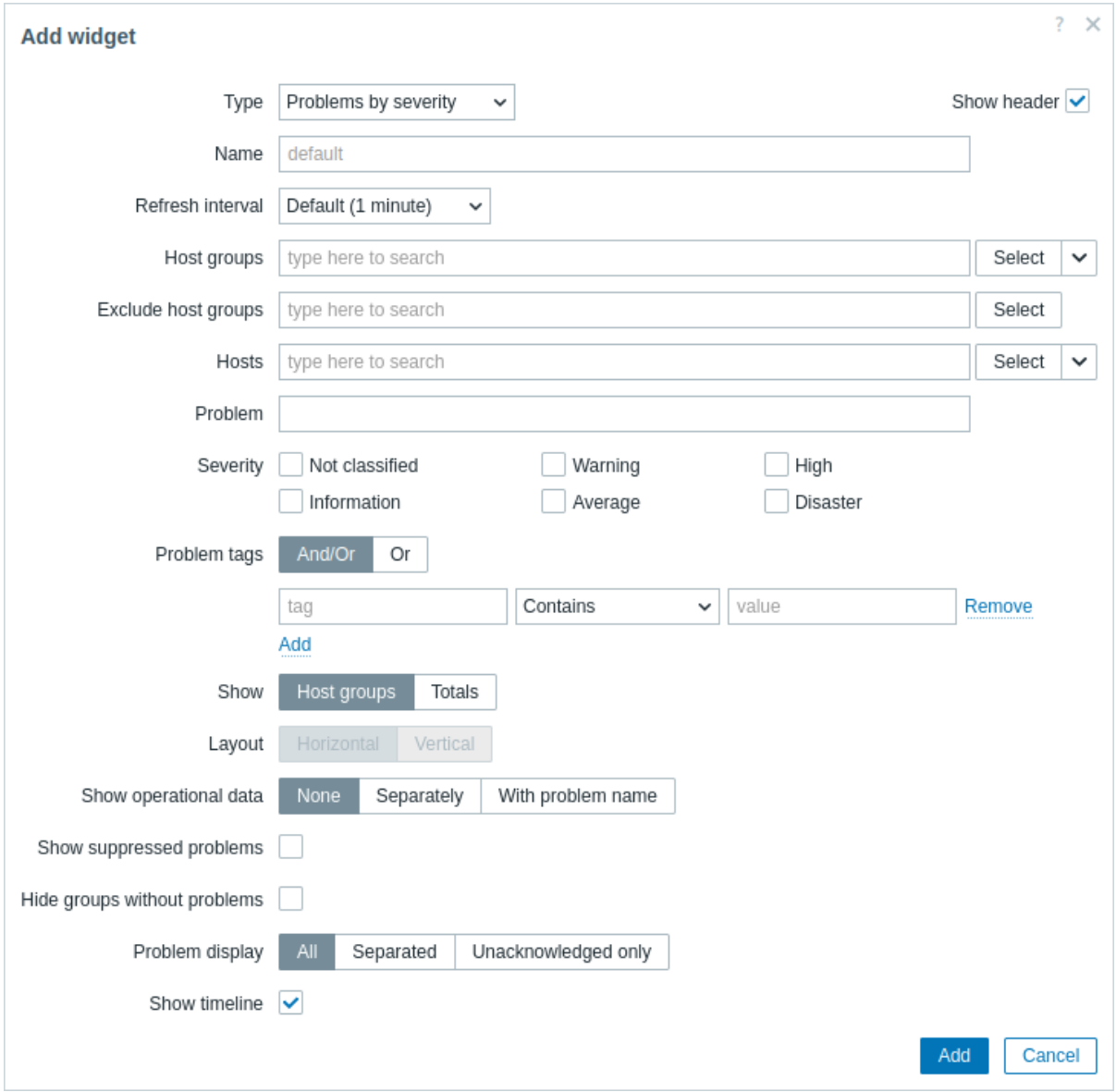
Das Widget *Probleme nach Schweregrad* zeigt die Anzahl der nach Schweregrad gruppierten Ursachenprobleme an. Es ermöglicht das Filtern von Problemen nach verschiedenen Parametern (Host-Gruppen, Hosts, Problemnamen usw.).

Wenn Sie den Mauszeiger über eine Problemanzahl bewegen, wird die Liste der zugehörigen Probleme angezeigt:



Konfiguration

Wählen Sie zur Konfiguration *Probleme nach Schweregrad* als Typ aus:



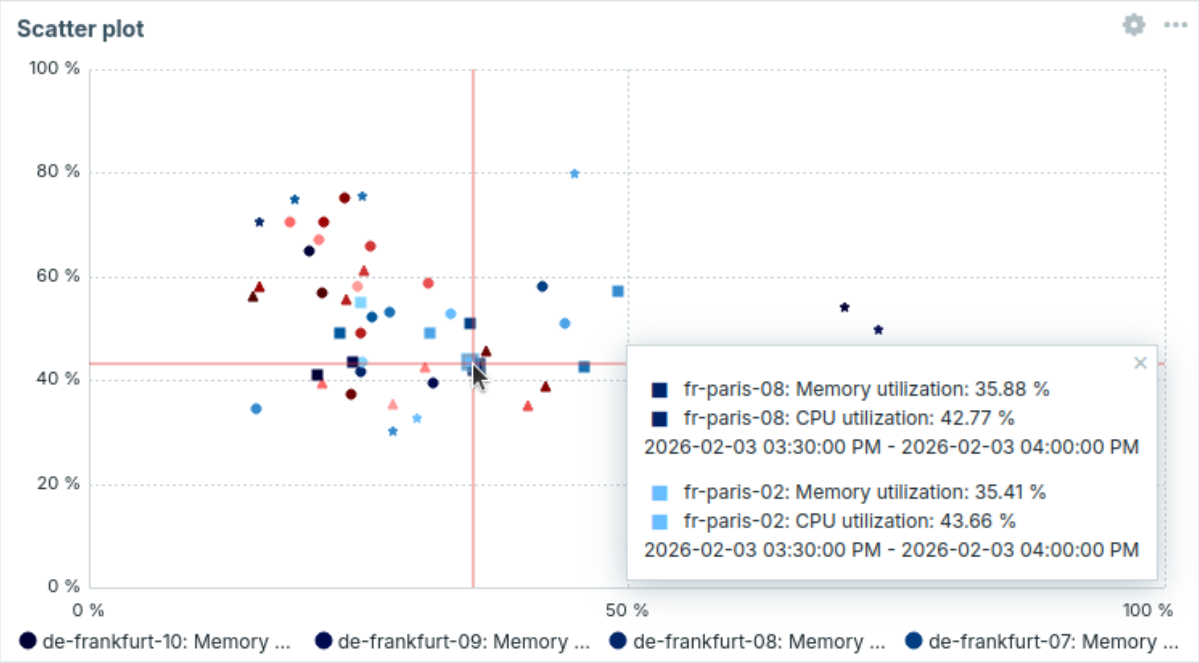
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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 verfügt über 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 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 in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen ausschließen	Wählen Sie Host-Gruppen aus, die im Widget ausgeblendet werden sollen. Dieses Feld verfügt über 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 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 in 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 verfügt über 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 in einem Vorlagen-Dashboard konfiguriert wird.
Problem	Sie können die Anzahl der angezeigten Problem-Hosts über den Problemnamen begrenzen. Wenn Sie hier eine Zeichenfolge eingeben, werden nur die Hosts angezeigt, deren Probleme einen Namen enthalten, der die eingegebene Zeichenfolge umfasst. 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.
Anzeigen	Wählen Sie die Anzeigeoption aus: **Host-Gruppen** - Probleme pro Host-Gruppe anzeigen; **Summen** - eine Problemsumme für alle ausgewählten Host-Gruppen in farbigen Blöcken anzeigen, die dem Problem-Schweregrad entsprechen. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird; es wird dann nur eine Problemsumme angezeigt.
Layout	Wählen Sie die Layout-Option aus: **Horizontal** - farbige Summenblöcke werden horizontal angezeigt; **Vertikal** - farbige Summenblöcke werden vertikal angezeigt. Dieser Parameter ist verfügbar, wenn *Anzeigen* auf ”Summen” gesetzt ist.
Betriebsdaten anzeigen	Betriebsdaten anzeigen (siehe Beschreibung von *Betriebsdaten* unter *Monitoring* > *Probleme*) als: **Keine** - Betriebsdaten werden nicht angezeigt **Getrennt** - Betriebsdaten werden als separate Zeile angezeigt **Mit Problemnamen** - Betriebsdaten werden zusammen mit dem Problemnamen angezeigt.
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 in einem Vorlagen-Dashboard konfiguriert wird.
Problemanzeige	Problemanzahl anzeigen als: **Alle** - die vollständige Problemanzahl wird angezeigt; **Getrennt** - die Anzahl nicht bestätigter Probleme wird getrennt als Zahl der gesamten Problemanzahl angezeigt; **Nur nicht bestätigte** - nur die Anzahl nicht bestätigter Probleme wird angezeigt.
Zeitachse anzeigen	Aktivieren Sie das Kontrollkästchen, um eine visuelle Zeitachse anzuzeigen.

Übersicht

Das Widget *Scatter plot* zeigt die Beziehung zwischen zwei Metriken, indem einzelne Datenpunkte entlang einer X- und Y-Achse dargestellt werden. Dies hilft dabei, Muster, Cluster, Korrelationen und Ausreißer im Datensatz sichtbar zu machen.



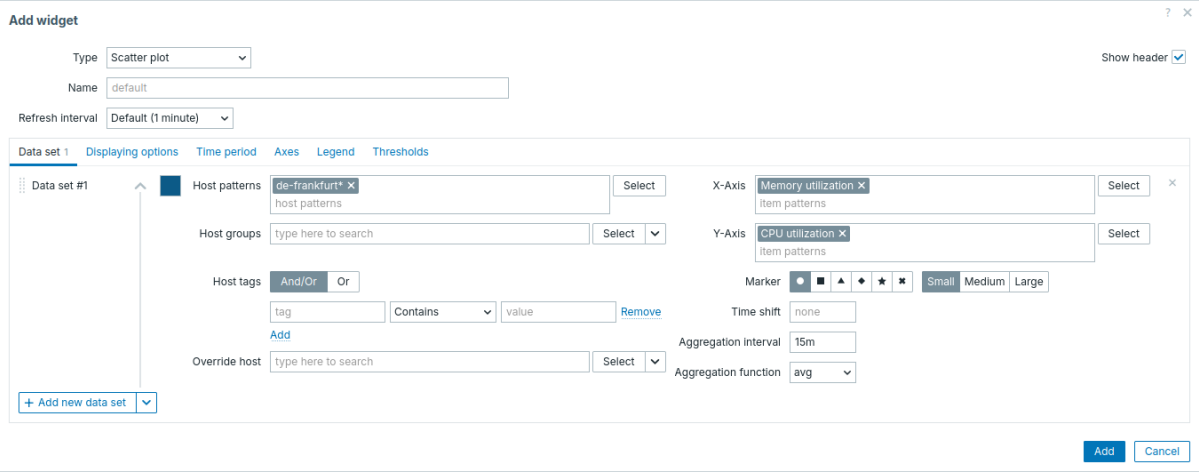
Wenn Sie mit der Maus auf einen Datenpunkt zeigen, wird eine Kurzinfo mit einer Liste von Datenpunkten dieses Punkts und anderer Punkte innerhalb eines Radius von 6 px angezeigt; sortiert zunächst nach den X-Achsenwerten des Scatter-Plots in aufsteigender Reihenfolge und anschließend nach den Y-Achsenwerten in absteigender Reihenfolge.

Wenn Sie in der Kurzinfo einen Datenpunkt auswählen, werden dessen Daten an andere Widgets übertragen.

Die im Widget *Scatter plot* angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem Sie im Widget-Menü die Option *Bild herunterladen* auswählen.

Konfiguration

Wählen Sie zum Konfigurieren *Scatter plot* als Typ aus:



Datensatz

Die Registerkarte **Datensatz** ermöglicht die Auswahl von Daten für das Streudiagramm durch Hinzufügen von Datensätzen. Es können zwei Arten von Datensätzen hinzugefügt werden:

- *Datenpunkt-Muster* - 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 unterschiedliche Farben zuzuweisen.

- *Datenpunkt-Liste* - ausgewählte Datenpunkte werden als Datenpunkte angezeigt, wobei jeder Punkt ein Paar von Datenpunkten (X und Y) darstellt. Sie können die Farbe jedes Datensatzes individuell im Farbwähler auswählen.

Datenpunkte werden nur für diejenigen Hosts/Datenpunkt-Paare erstellt, die auf beiden Achsen mindestens einen Wert haben.

Standardmäßig wird ein Datensatz vom Typ *Datenpunkt-Muster* hinzugefügt.

Datensatz	Für den Datensatz **Datenpunkt-Muster**: Wählen Sie Host- und Datenpunkt-Muster aus oder geben Sie sie ein; Daten von Datenpunkten, die diesen Mustern entsprechen, werden im Streudiagramm angezeigt; es können bis zu 100 Datenpunkte angezeigt werden (50 auf jeder Achse bei insgesamt 50 Hosts). Für die Auswahl können Platzhaltermuster verwendet werden (zum Beispiel liefert * Ergebnisse zurück, 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 nicht möglich, zum Beispiel einen Datenpunkt mit dem Namen *item** einzeln hinzuzufügen, wenn es andere passende Datenpunkte gibt (zum Beispiel *item2*, *item3*). Die Angabe von Host- und Datenpunkt-Mustern ist für Datensätze vom Typ ”Datenpunkt-Muster” erforderlich. Siehe auch: Details zur Datensatzkonfiguration. Für den Datensatz **Datenpunkt-Liste**: 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 ”Datenpunkt-Liste” erforderlich. Siehe auch: Details zur Datensatzkonfiguration. Beachten Sie, dass nur numerische Datenpunkttypen zulässig sind. Beim Konfigurieren des Widgets in einem Vorlagen-Dashboard ist der Parameter zur Angabe von Host-Mustern nicht verfügbar, und der Parameter zur Angabe einer Datenpunkt-Liste erlaubt nur die Auswahl der in der Vorlage konfigurierten Datenpunkte.
Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Auto-Vervollständigung, daher wird beim Eingeben des Gruppennamens eine Dropdown-Liste mit passenden Gruppen angeboten. Die Angabe einer übergeordneten Host-Gruppe wählt implizit 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 nur für Datensätze vom Typ *Datenpunkt-Muster* verfügbar und ist beim Konfigurieren des Widgets in einem Vorlagen-Dashboard nicht verfügbar.

Host-Tags	Geben Sie Tags an, um die im Widget angezeigten Hosts 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. 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. Dieser Parameter ist nur für Datensätze vom Typ *Datenpunkt-Muster* verfügbar und ist beim Konfigurieren des Widgets in einem Vorlagen-Dashboard nicht verfügbar.
Host überschreiben	Wählen Sie ein kompatibles Widget oder den Dashboard-Host-Selektor als Datenquelle für Hosts aus. Dieser Parameter ist beim Konfigurieren des Widgets in einem Vorlagen-Dashboard nicht verfügbar.
Markierung	Wählen Sie den Markierungstyp und die Größe für Host-Datenpunkte im Datensatz.
Zeitverschiebung	Geben Sie bei Bedarf eine Zeitverschiebung an. In diesem Feld können Zeitsuffixe verwendet werden. Negative Werte sind zulässig.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion 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; **none** - alle Werte anzeigen (keine Aggregation). Die Aggregation ermöglicht die Anzeige eines aggregierten Werts für das gewählte Intervall (5 Minuten, eine Stunde, ein Tag) anstelle aller Werte. Siehe auch: Aggregation in Diagrammen.

Aggregationsintervall	Geben Sie das Intervall für die Aggregation von Werten an. In diesem Feld können Zeitsuffixe verwendet werden. Ein numerischer Wert ohne Suffix wird als Sekunden betrachtet. Beachten Sie, dass bei einer Konfiguration des Widgets zur Anzeige historischer Daten auf Basis von Trends (*Auswahl historischer Daten* ist auf *Trends* oder *Auto* 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 kann die Verwendung eines Aggregationsintervalls, das kein Vielfaches von 1 Stunde ist (z. B. 100s, 7min, 15min, 90min usw.), zu Ergebnissen führen, die schwer zu interpretieren sind. Die Angabe eines Aggregationsintervalls ist erforderlich. Das Aggregationsintervall bestimmt das Zeitintervall und die Koordinaten für einen Datenpunkt; wenn zum Beispiel Datenpunkte der X-Achse 4 Werte und Datenpunkte der Y-Achse 1 Wert haben, spiegeln die Koordinaten die Aggregation von 4 auf X und 1 auf Y im angegebenen Intervall wider.

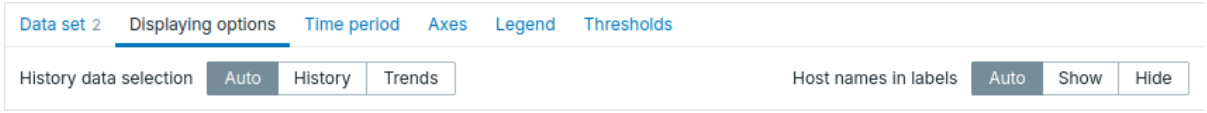
Details zur Datenmengenkonfiguration

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

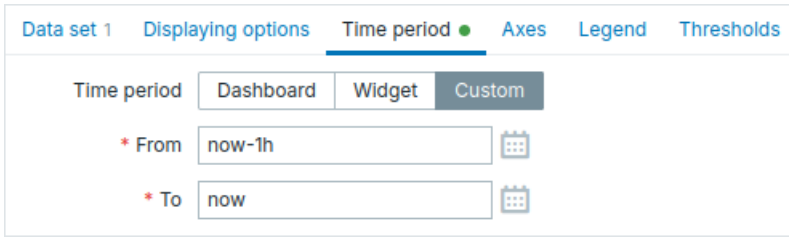
Die Registerkarte **Anzeigeoptionen** ermöglicht die Festlegung der Auswahl von Verlaufsdaten:



Auswahl der Verlaufsdaten	Legen Sie die Quelle der Streudiagrammdaten fest: **Auto** - die Daten werden gemäß dem klassischen Diagramm-Algorithmus bezogen (Standard). **Verlauf** - Daten aus dem Verlauf. **Trends** - Daten aus den Trends.
Host-Namen in Beschriftungen	Wählen Sie, ob Host-Namen in der Diagrammlegende und im Tooltip angezeigt werden sollen, der beim Überfahren der Diagrammdaten mit der Maus 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 nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Zeitraum

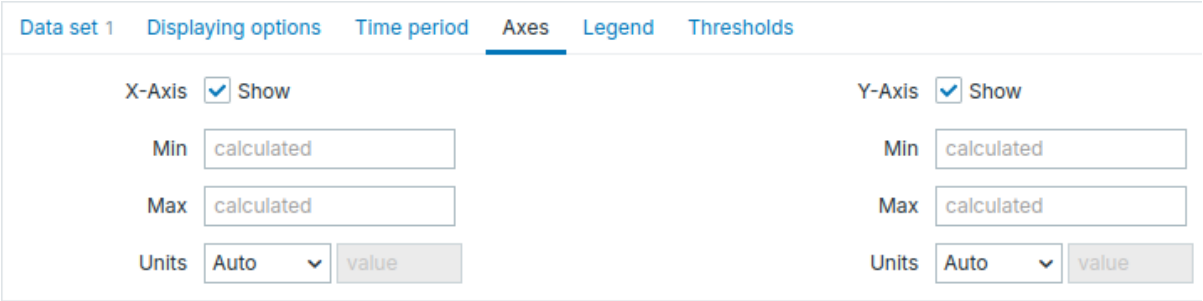
Die Registerkarte **Zeitraum** ermöglicht es, einen Zeitraum festzulegen, für den Daten im Streudiagramm angezeigt werden:



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 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

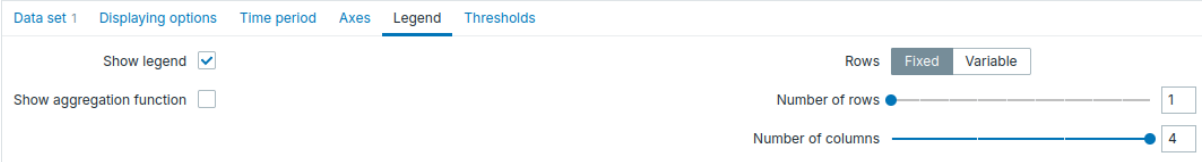
Die Registerkarte **Achsen** ermöglicht es, die Anzeige der Achsen anzupassen:



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 unter Verwendung der Einheit des ersten Datenpunkts im Datensatz angezeigt; **Statisch** - Achsenwerte werden unter Verwendung der im Eingabefeld *Wert* angegebenen Einheit angezeigt; wenn das Feld leer bleibt, werden nur numerische Werte angezeigt.

Legende

Die Registerkarte **Legende** ermöglicht die Anpassung der Scatterplot-Legende:

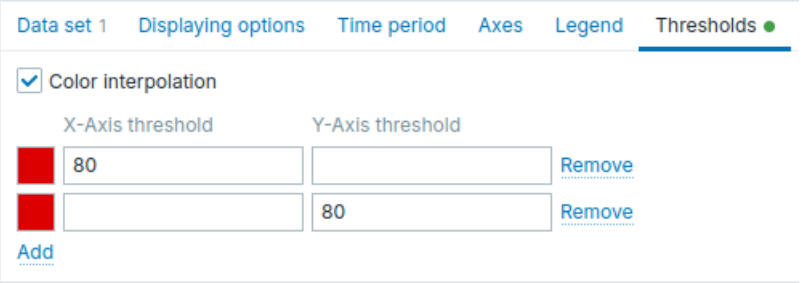


Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende im Scatterplot auszublenden (standardmäßig aktiviert).
Aggregationsfunktion anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Aggregationsfunktion in der Legende anzuzeigen.
Zeilen	Wählen Sie den Anzeigemodus für die Legendenzeilen 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, ohne den Wert des Parameters *Maximale Anzahl der Zeilen* zu 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.

Schwellenwerte

Die Registerkarte **Schwellenwerte** ermöglicht die Konfiguration von Schwellenwerten, die die Farben der Host-Markierungen bestimmen:

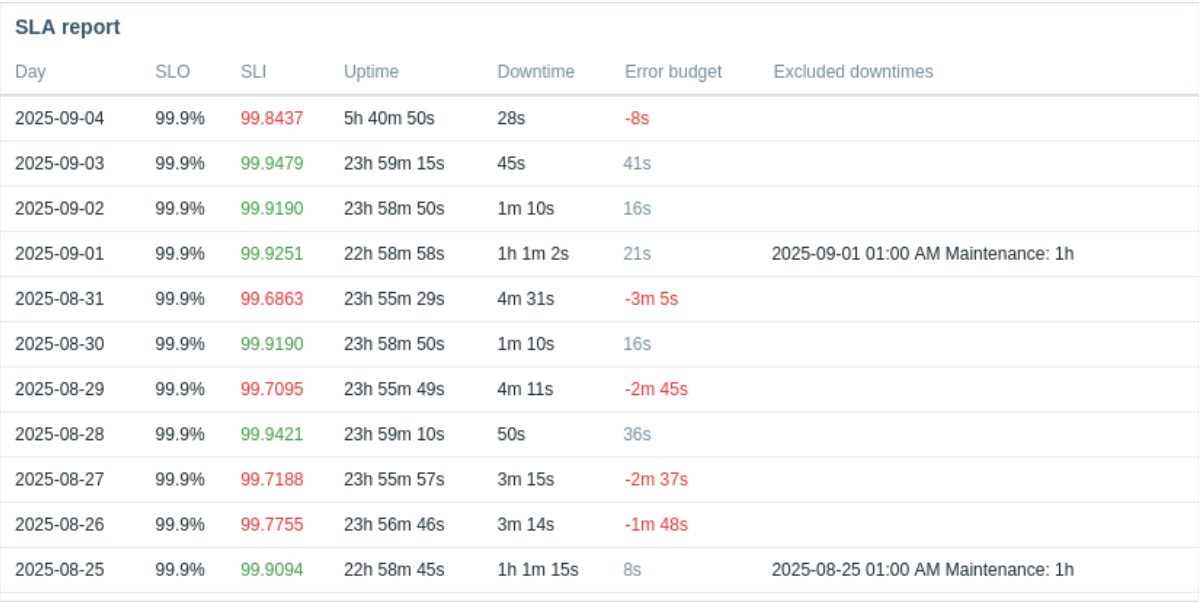


Farbinterpolation	Aktivieren Sie das Kontrollkästchen, um einen weichen Übergang zwischen den Schwellenwertfarben für Host-Markierungen zu aktivieren. Dieser Parameter gilt, wenn zwei oder mehr Schwellenwerte entweder für die X-Achse, die Y-Achse oder gleichzeitig für beide Achsen festgelegt sind.
Schwellenwert für X-Achse/Y-Achse	Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie eine Schwellenwertfarbe aus der Farbauswahl aus und geben Sie einen numerischen Wert an. Schwellenwerte können auf eine einzelne Achse, auf beide Achsen getrennt 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 (zum Beispiel „1d“, „2w“, „4K“, „8G“) werden unterstützt. Wertzuordnungen werden unterstützt.

26 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.

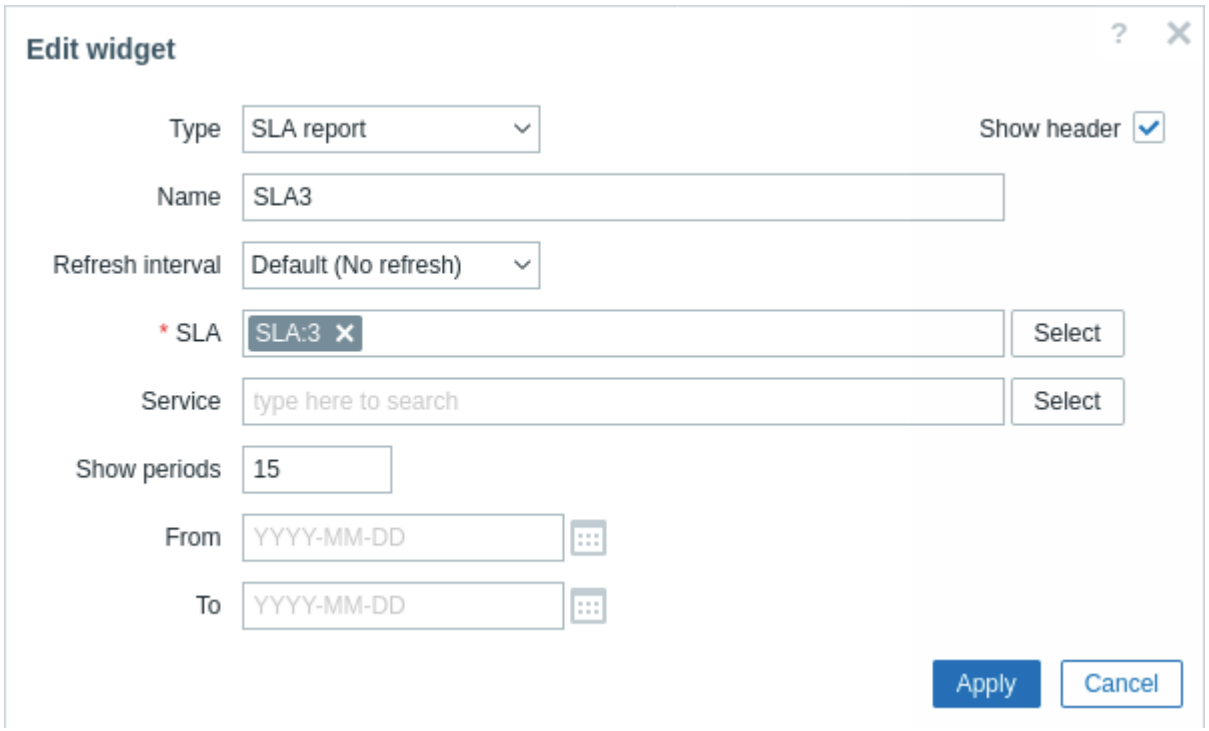

SLA report

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:



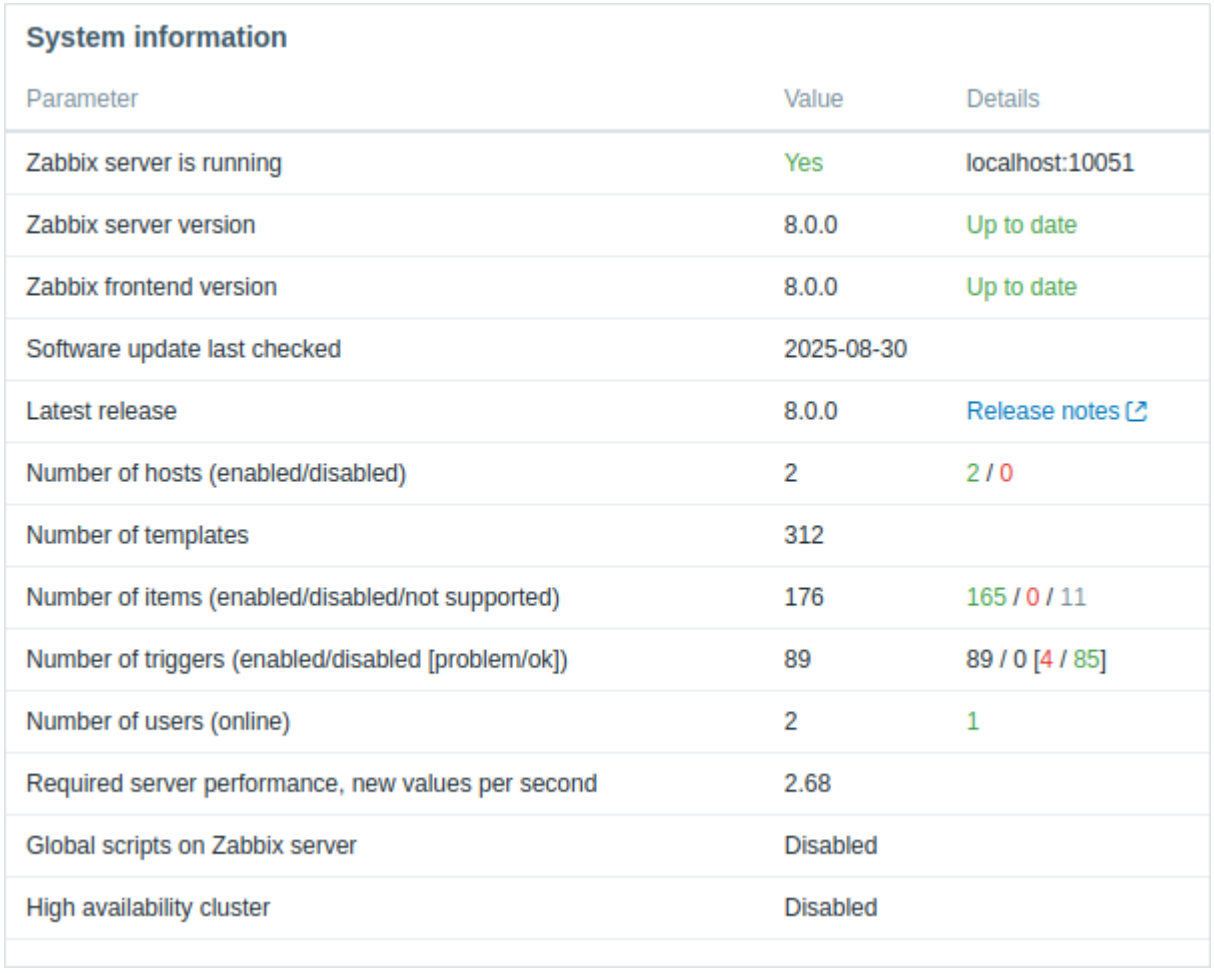
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

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.

27 Systeminformationen

Übersicht

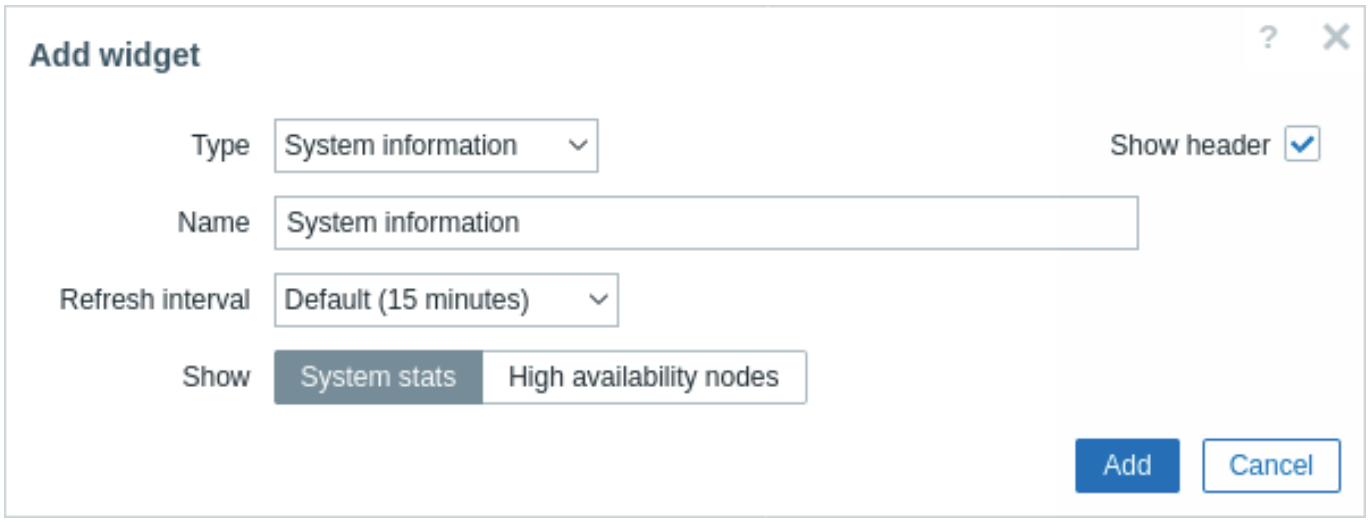
Das Widget *Systeminformationen* zeigt eine Zusammenfassung wichtiger Zabbix-Server- und Systemdaten oder Details zu Hochverfügbarkeitsknoten an.



Es zeigt dieselben Daten an wie *Berichte* > *Systeminformationen*.

Konfiguration

Um zu konfigurieren, wählen Sie *System information* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

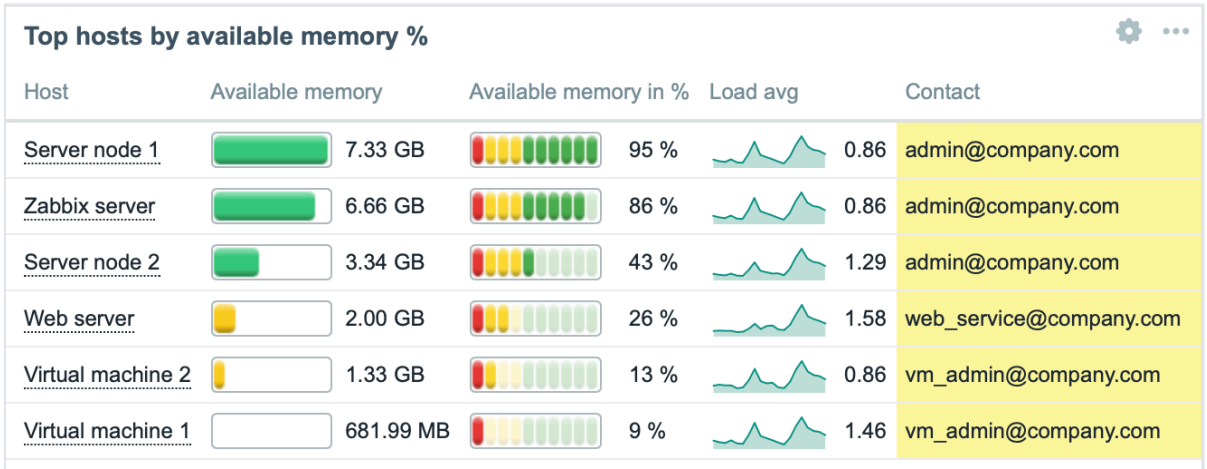
Show	Wählen Sie aus, was angezeigt werden soll: **System stats** - zeigt eine Zusammenfassung der wichtigsten Zabbix-Server- und Systemdaten an; **High availability nodes** - zeigt den Status der Hochverfügbarkeitsknoten an (wenn der Hochverfügbarkeitscluster aktiviert ist).

Show software update check details	Aktivieren Sie das Kontrollkästchen, um Details zur Prüfung auf Zabbix-Softwareaktualisierungen anzuzeigen. Diese Option ist nur verfügbar, wenn die Prüfung auf Softwareaktualisierungen in der Zabbix-Server-Konfiguration aktiviert ist und im Feld *Show* „System stats“ ausgewählt wurde.

28 Top-Hosts

Übersicht

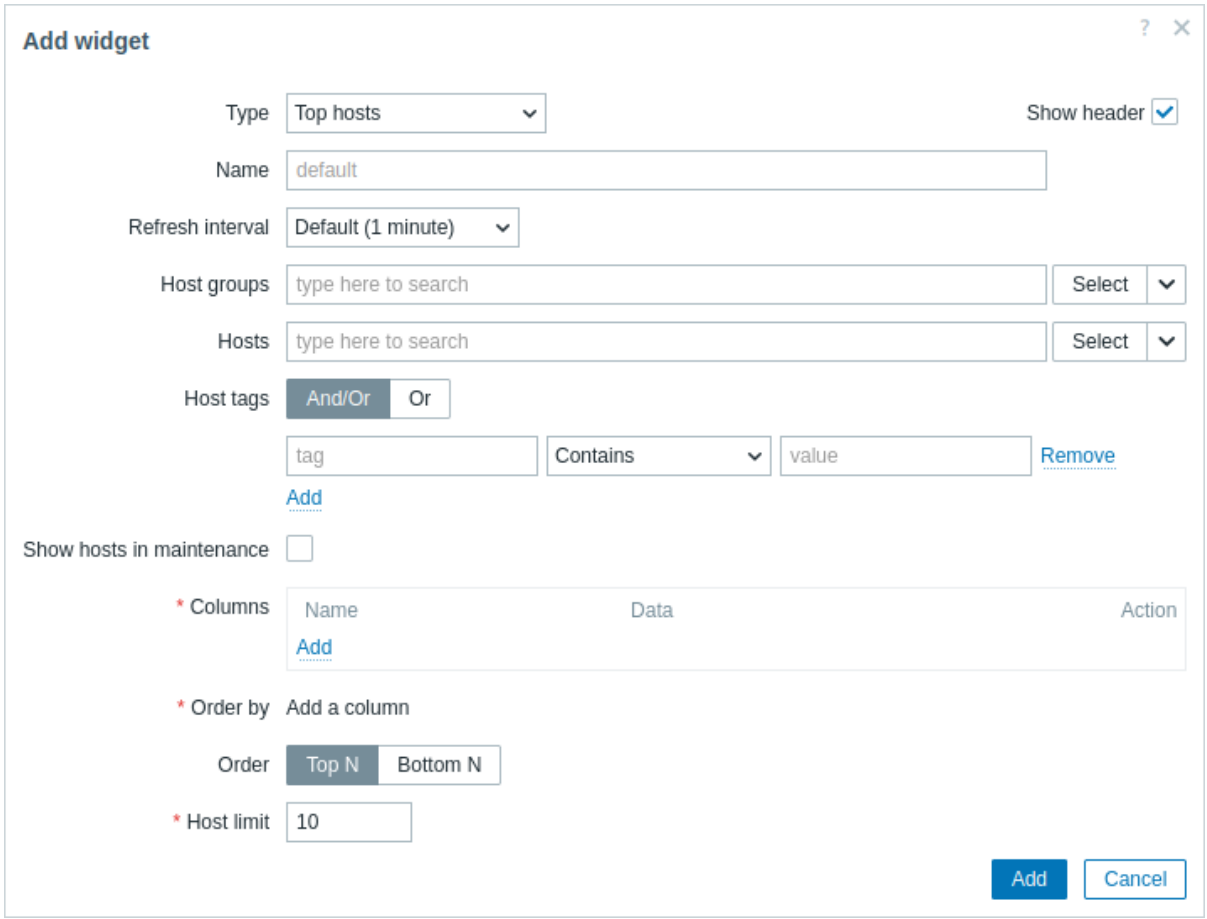
Das Widget *Top hosts* zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte von mehreren Hosts an.



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:

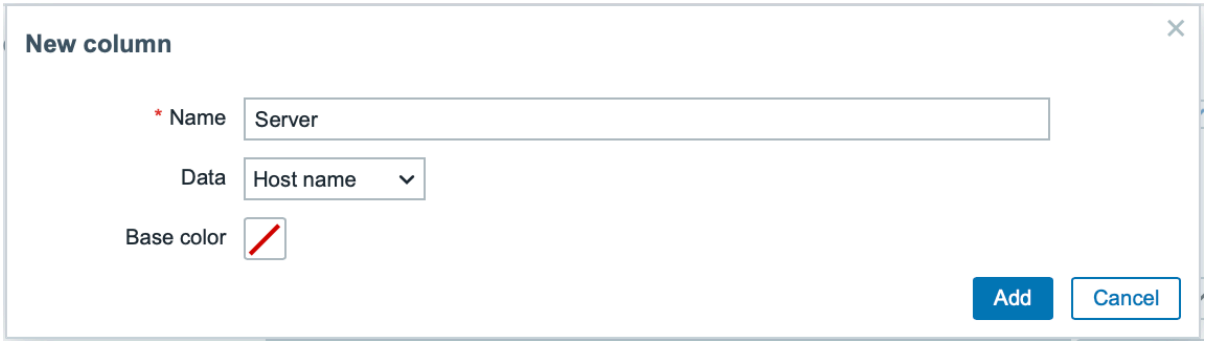
Host-Gruppen	Wählen Sie die Host-Gruppen aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit 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.

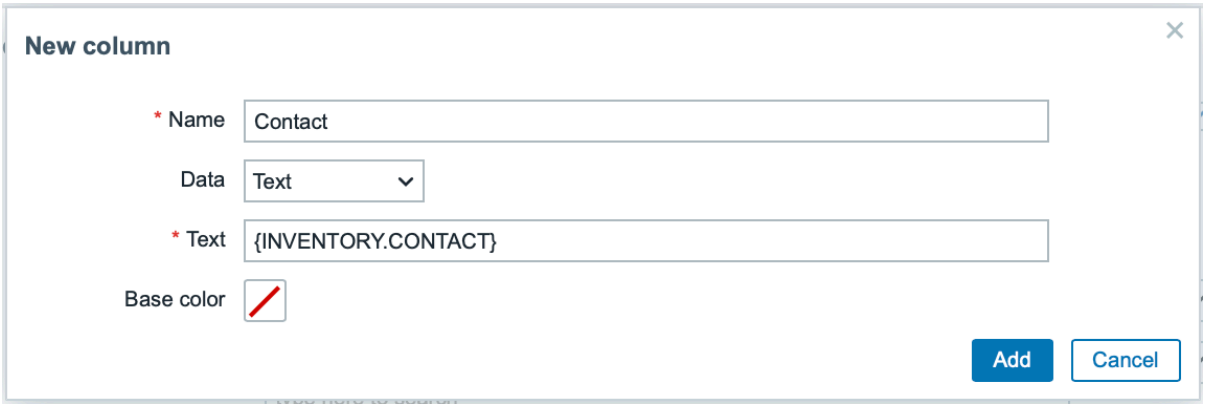


Unterstützte Parameter:

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.

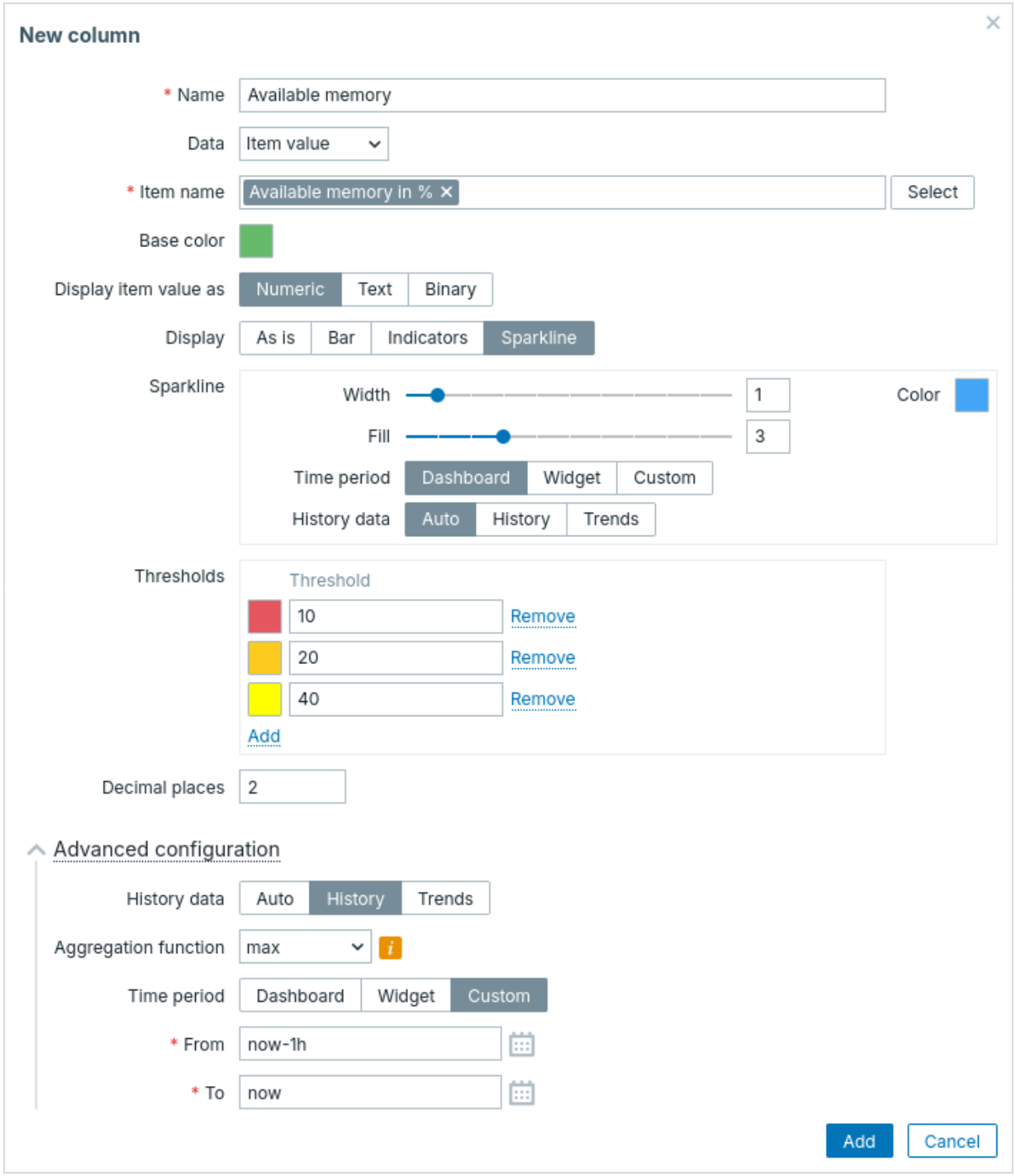


Unterstützte Parameter:

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.

Datenwertspalte

Die Datenwertspalte wird verwendet, um den Wert des angegebenen Datenpunkts anzuzeigen; sie unterstützt mehrere Anzeigeformate und Optionen für Werte.



Unterstützte Parameter:

Name	Name der Spalte.
Data	Datentyp, der in der Spalte angezeigt werden soll; wählen Sie **Datenwert** aus.
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 hat, 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 in einem Vorlagen-Dashboard können nur in der Vorlage konfigurierte Datenpunkte ausgewählt werden.
Base color	Hintergrundfarbe der Spalte; Füllfarbe, wenn *Datenwert*-Daten als Balken/Indikatoren angezeigt werden. Die Standardfarbe kann durch eine benutzerdefinierte Farbe überschrieben werden, wenn der Datenwert einen der angegebenen Schwellenwerte überschreitet.
Display item value as	Format für die Anzeige des Datenwerts: **Numerisch**, **Text** oder **Binär**. Die ausgewählte Option bestimmt, welche zusätzlichen Parameter verfügbar sind. Siehe die jeweilige Parameterliste für jedes Format unten.
Advanced configuration	

History data	Daten aus Verlauf oder Trends verwenden: **Auto** - automatische Auswahl; **History** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden. Dieser Parameter ist nur für numerische Datenwerte verfügbar. Für Text- und Binärwerte werden Daten immer aus dem Verlauf verwendet.
Aggregation function	Geben Sie an, welche Aggregationsfunktion 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; **not used** - den aktuellsten Wert anzeigen (keine Aggregation). Mit der Aggregation kann für das gewählte 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.
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 Zeitraumauswahl 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 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 *Time period*-Konfiguration 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 numerische Datenwerte:

Display	Definieren Sie, wie der Wert angezeigt werden soll: **As is** - als normaler Text; **Bar** - als durchgehender, farbig gefüllter Balken; **Indicators** - als segmentierter, farbig gefüllter Balken; **Sparkline** - Mini-Liniendiagramm.
Thresholds	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 Datenwert 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 Linienstärke des Diagramms 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 einbezogen 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); **Custom** - einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *From* und *To* 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 *Time period*-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 *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 Verlauf oder Trends verwenden: **Auto** - automatische Auswahl; **History** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden.

Parameter für Text-Datenwerte:

Highlights	Geben Sie die regulären Ausdrücke an, bei deren Übereinstimmung sich die Hintergrund-/Füllfarbe ändern soll.

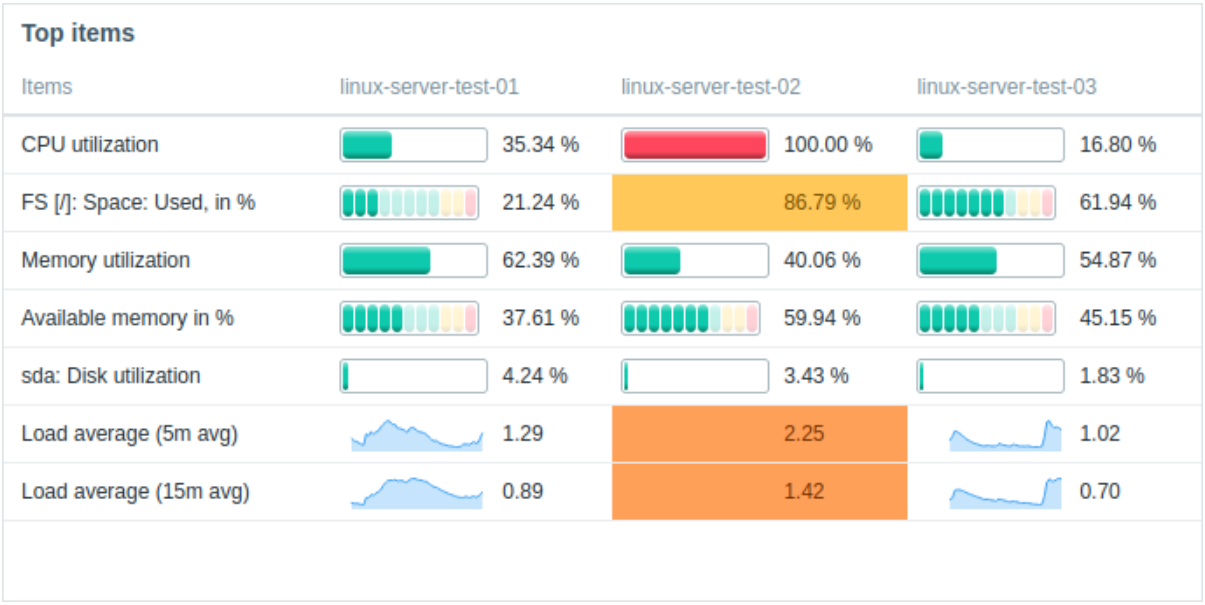
Parameter für binäre Datenwerte:

Show thumbnail	Geben Sie an, ob für das Bild mit Binärdaten eine Miniaturansicht erstellt und angezeigt werden soll oder ob in der Wertspalte ein Hyperlink *Show* angezeigt werden soll, der zum Bild in voller Größe führt.

29 Top-Datenpunkte

Übersicht

Das Widget *Top items* zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte an und bietet einen schnellen Überblick über deren Leistung.

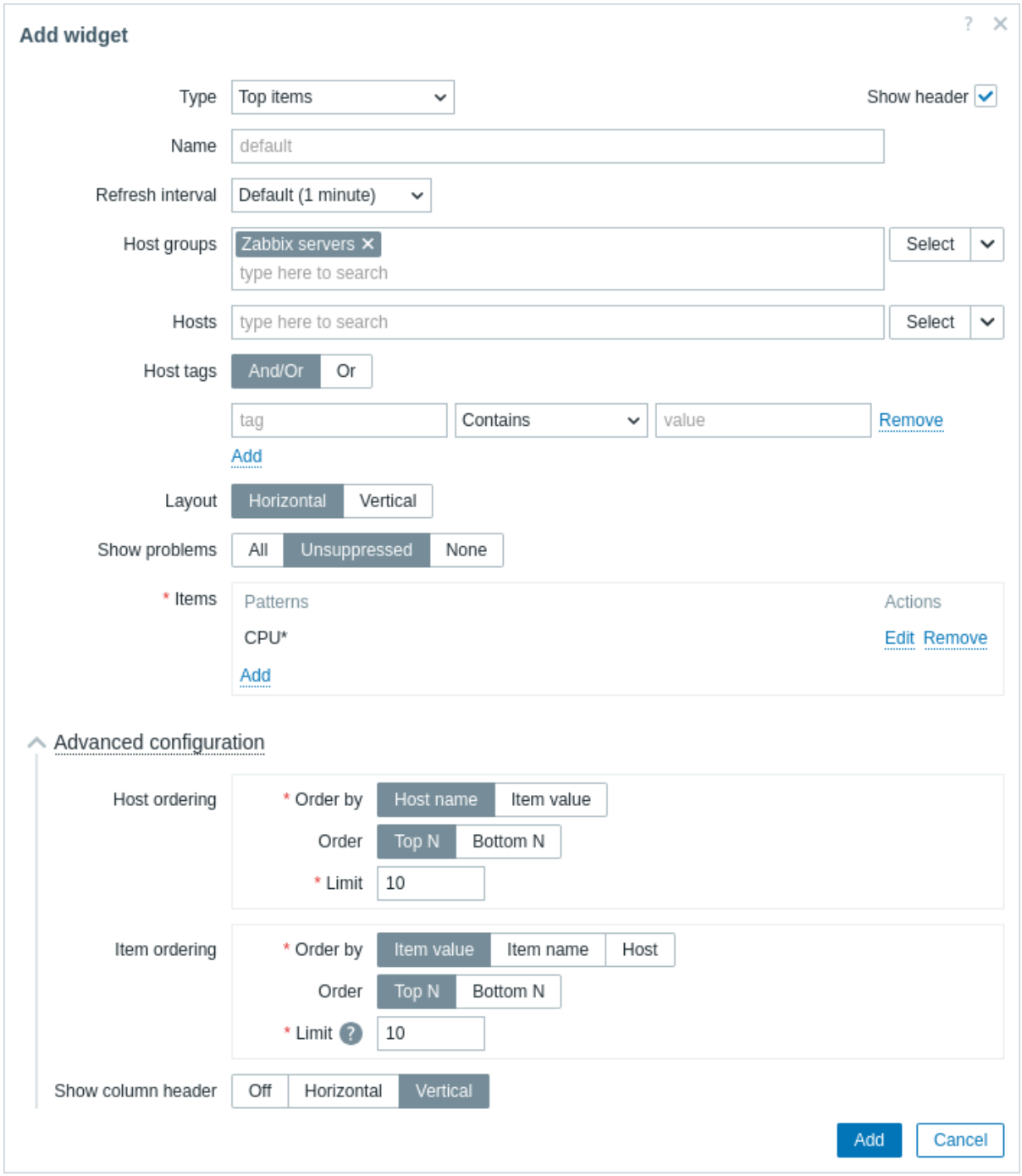


Standardmäßig zeigt das Widget Werte aus den letzten 24 Stunden an. Dieser Zeitraum kann unter *Administration* > *General* > *GUI* (*Maximale Anzeigeperiode des Verlaufs*) geändert werden.

Durch Klicken auf den Datenpunktwert wird das Datenpunktmenü geöffnet.

Konfiguration

Wählen Sie zur Konfiguration *Top items* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

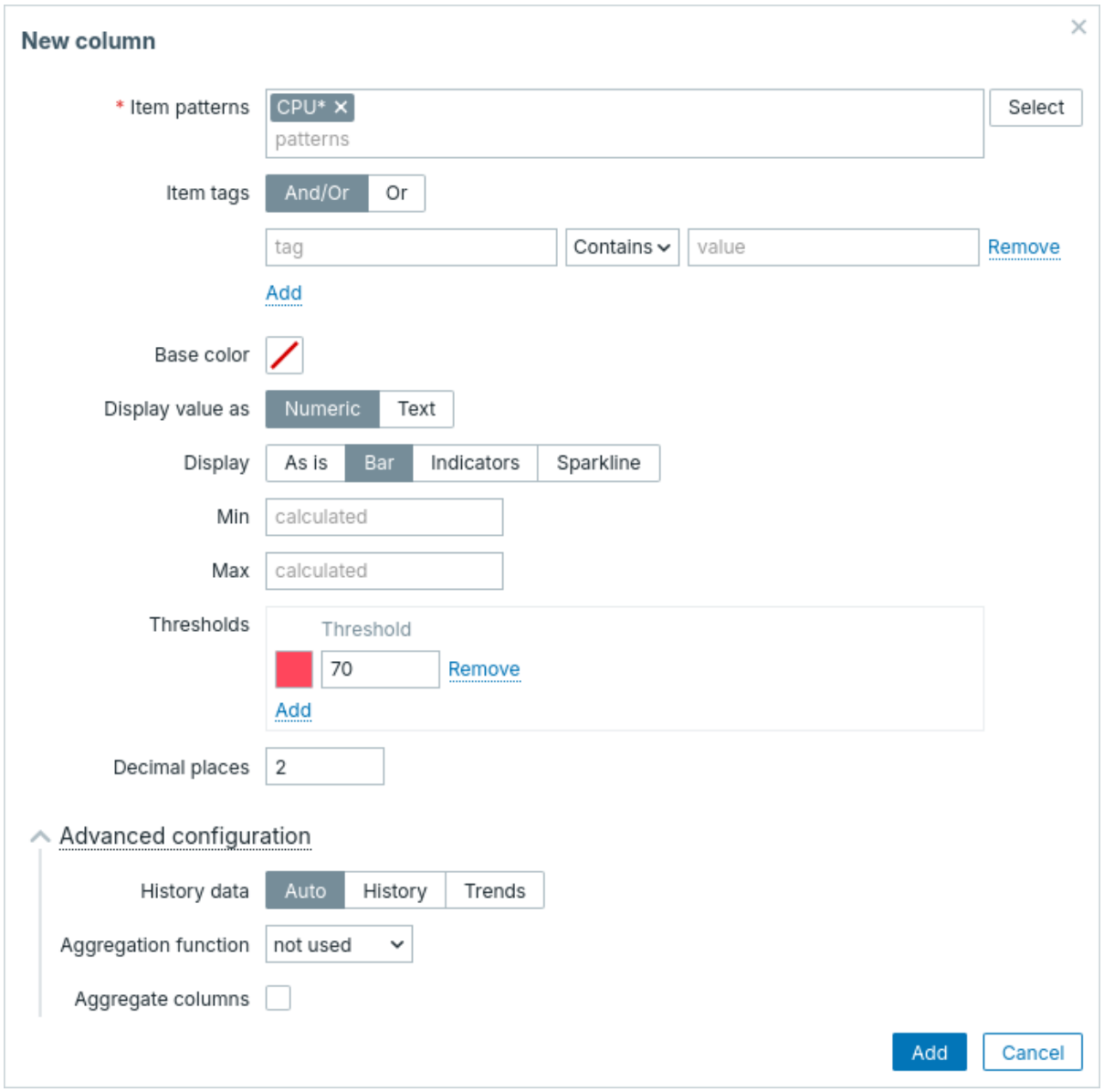
Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Dieser Parameter ist nicht verfügbar, wenn das Widget in 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 in einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Host-Daten 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.
Layout	Wählen Sie die Layout-Option aus: **Horizontal** - Host-Namen werden links angezeigt; **Vertikal** - Host-Namen werden oben angezeigt.
Probleme anzeigen	Filtern Sie, welche Probleme basierend auf ihrem Status angezeigt werden sollen (alle, nicht unterdrückte, keine). Probleme werden durch farbige Datenpunkt-Werte angezeigt, wodurch die Anzeigeeinstellungen der Datenpunkt-Spalte überschrieben werden. 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.
Datenpunkte	Fügen Sie Datenpunkt-Muster oder bestimmte Datenpunkte zur Anzeige hinzu (siehe Spaltenkonfiguration).
Erweiterte Konfiguration	
Host-Sortierung	Wählen Sie Sortieroptionen für die Host-Spalte/-Zeile aus.
Sortieren nach	Hosts sortieren nach: **Host-Name** - Hosts werden nach Host-Namen sortiert; **Datenpunkt-Wert** - Hosts werden nach dem Wert der ausgewählten Datenpunkte sortiert.
Reihenfolge	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* ausgewählt.
Limit	Geben Sie das Limit für anzeigbare Hosts ein (Bereich 1-1000; Standard 10). Dieser Wert wird zum Wert von N im Feld *Reihenfolge*.
Datenpunkt-Sortierung	Wählen Sie Sortieroptionen für die Datenpunkt-Spalte/-Zeile aus.

Sortieren nach	Datenpunkte sortieren nach: **Datenpunkt-Wert** - Datenpunkte werden nach Datenpunkt-Wert sortiert; **Datenpunkt-Name** - Datenpunkte werden nach Datenpunkt-Namen sortiert; **Host** - Datenpunkte werden nach dem ausgewählten Host-Muster sortiert.
Reihenfolge	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* ausgewählt.
Limit	Geben Sie das Limit für anzeigbare Datenpunkte ein (Bereich 1-1000; Standard 10). Dieser Wert wird zum Wert von N im Feld *Reihenfolge*.
Spaltenüberschrift anzeigen	Wählen Sie Anzeigeoptionen für die Spaltenüberschrift aus: **Aus** - Spaltenüberschrift nicht anzeigen; **Horizontal** - Text horizontal in der Überschrift anzeigen; **Vertikal** - Text vertikal in der Überschrift 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	Definieren Sie, wie der Wert angezeigt werden soll: **Wie er ist** - als normaler Text; **Balken** - als durchgehender, farbig gefüllter Balken; **Indikatoren** - als segmentierter, farbig gefüllter Balken; **Sparkline** - Mini-Liniendiagramm.
Min	Minimalwert für die Anzeige als Balken/Indikatoren.
Max	Maximalwert für die Anzeige als Balken/Indikatoren.
Breite	Legen Sie die Dicke der Diagrammlinie fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben. Dieser Parameter gilt nur für die Sparkline-Anzeige.
Füllung	Legen Sie die Transparenzstufe der Füllfarbe fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben. Dieser Parameter gilt nur für die Sparkline-Anzeige.
Farbe	Wählen Sie die Linien- und Füllfarbe aus. Dieser Parameter gilt nur für die Sparkline-Anzeige.
Siehe *Erweiterte Konfiguration* für die Beschreibung der Felder zur Auswahl des Zeitraums und der Verlaufsdaten.	
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 Datenpunktwert angewendet.
Dezimalstellen	Geben Sie an, wie viele Dezimalstellen mit dem Wert angezeigt werden sollen.

Erweiterte Konfiguration	
Verlaufsdaten	Daten aus Verlauf oder Trends verwenden: **Auto** - automatische Auswahl; **Verlauf** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden. Dieser Parameter ist nur für numerische Datenpunktwerte verfügbar. Bei Textwerten werden Daten immer aus dem Verlauf verwendet.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion 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; **not used** - den neuesten Wert anzeigen (keine Aggregation). Die Aggregation ermöglicht die Anzeige eines aggregierten Werts für das gewählte Intervall (5 Minuten, eine Stunde, ein Tag) anstelle des neuesten Werts. Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Bei *count* werden nicht numerische Daten in numerische 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 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 die eingestellte Zeit anzeigt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration des *Zeitraums* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können. Dieser Parameter ist nicht verfügbar, wenn *Aggregationsfunktion* 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 *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.
Spalten aggregieren	Wenn diese Option aktiviert ist, werden alle Datenpunkte, die den Datenpunktmustern der Spalte entsprechen, zu einer einzigen kombinierten Spalte/Zeile gruppiert. Werte aus übereinstimmenden Datenpunkten werden mit der ausgewählten *Spaltenaggregationsfunktion* aggregiert und unter dem *Namen der kombinierten Spalte* angezeigt. Standard: aus (nicht aktiviert). 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.

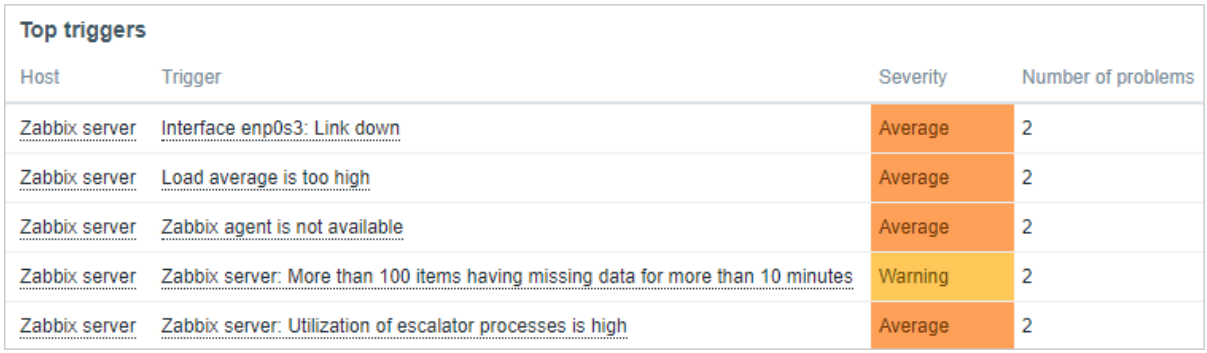
Spaltenparameter für Textwerte:

Hervorhebungen	Geben Sie die regulären Ausdrücke an, bei deren Übereinstimmung sich die Hintergrund-/Füllfarbe ändern soll.

30 Top-Auslöser

Übersicht

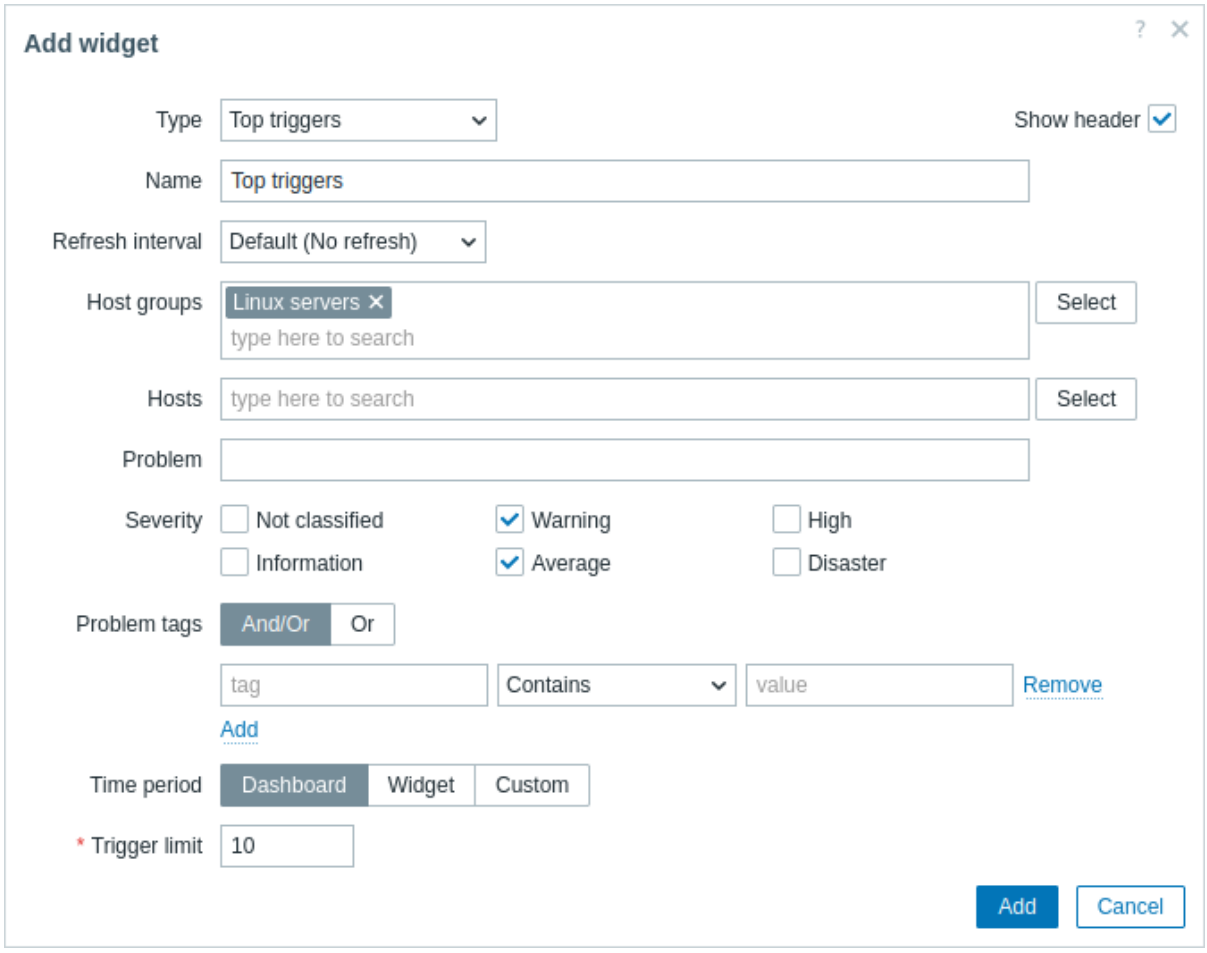
Das Widget *Top-Auslöser* zeigt Auslöser mit der höchsten Anzahl an Problemen an.



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.

31 Übersicht über Auslöser

Übersicht

Das Widget *Auslöserübersicht* 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.

Das Widget verwendet mehrere visuelle Elemente, um Details zu Auslösern zu vermitteln:

- Die Blockfarben für PROBLEM-Auslöser entsprechen den Schweregraden der Probleme.
- Blinkende Blöcke weisen auf kürzliche Änderungen des Auslöserzustands hin (innerhalb der letzten 2 Minuten).
- Graue Pfeile kennzeichnen Auslöser mit Abhängigkeiten (fahren Sie mit der Maus darüber, um Details anzuzeigen).
- Kontrollkästchen-Symbole kennzeichnen bestätigte Probleme; damit dieses Symbol erscheint, müssen alle aktiven oder behobenen Probleme für den Auslöser bestätigt sein.

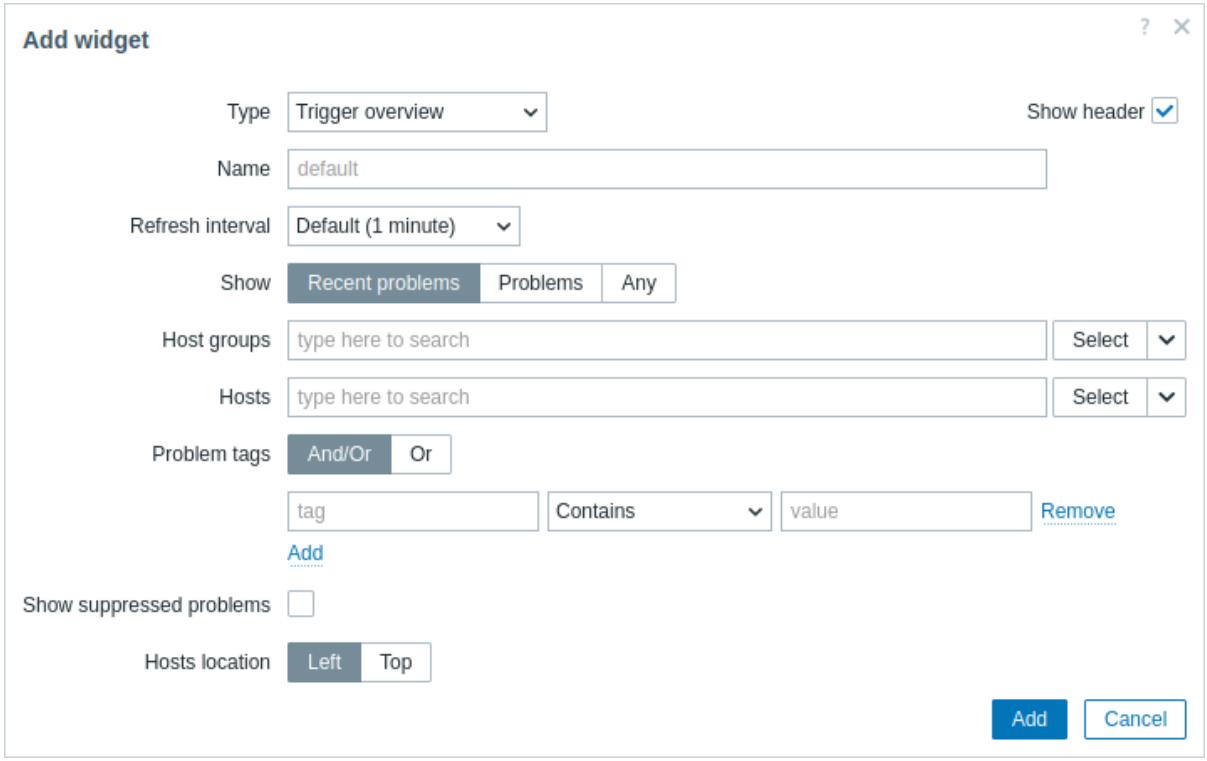
Note:
Die Farben der Auslöserschweregrade können unter *Administration* > *General* angepasst werden. Auslöserschweregrade können beim Aktualisieren von Problemen geändert werden.

Durch Klicken auf einen Block erhalten Sie Links zu auslöserbezogenen Informationen (Probleme, Auslöserkonfiguration usw.).

Standardmäßig werden bis zu 50 Einträge angezeigt (konfigurierbar unter *Administration* > *General* > *GUI* im Feld *Maximale Anzahl von Spalten und Zeilen in Übersichtstabellen*). Wenn mehr Auslöser vorhanden sind, werden Sie durch eine Meldung aufgefordert, die Filterkriterien zu verfeinern. Dieses Limit 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:

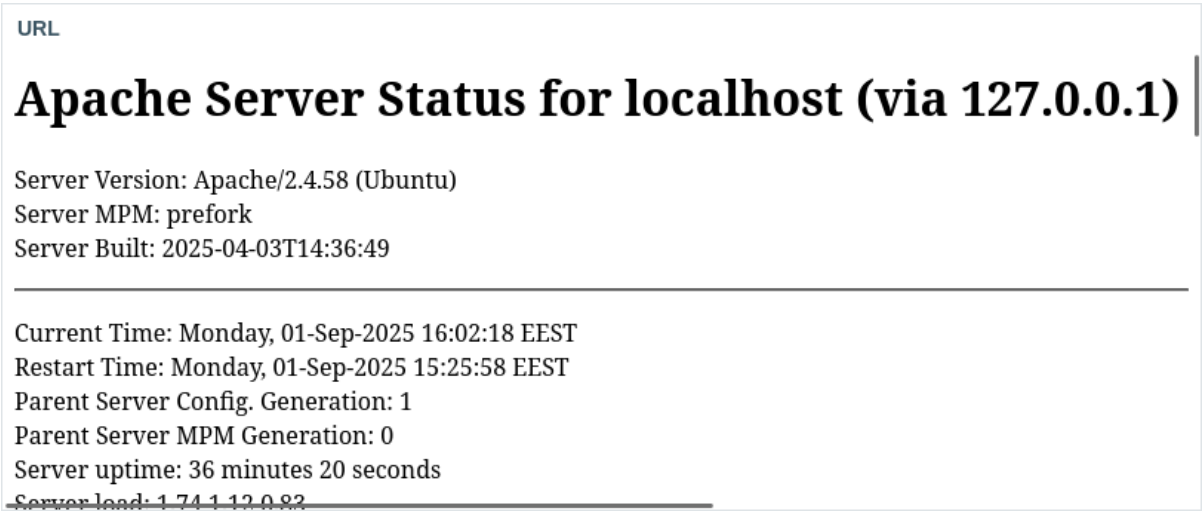
Anzeigen	Filtern Sie Auslöser nach Auslöserstatus: **Aktuelle Probleme** - *(Standard)* zeigt Auslöser an, die vor Kurzem in einem PROBLEM-Status waren oder sich noch darin befinden (gelöst und ungelöst); **Probleme** - zeigt Auslöser an, die sich in einem PROBLEM-Status befinden (ungelöst); **Beliebig** - 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.

32 URL

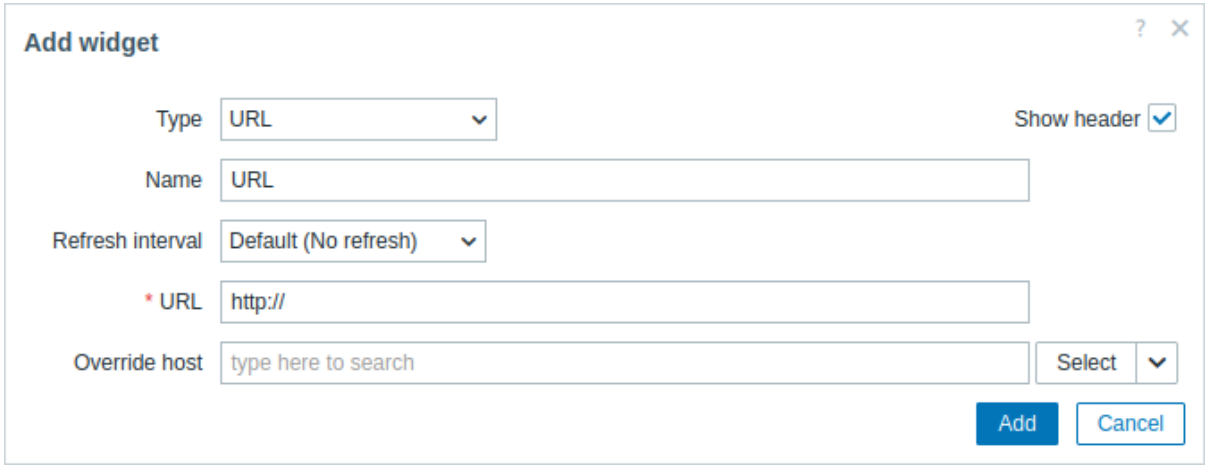
Übersicht

Das *URL*-Widget zeigt den von einer URL abgerufenen Inhalt an.



Konfiguration

Wählen Sie zur Konfiguration *URL* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

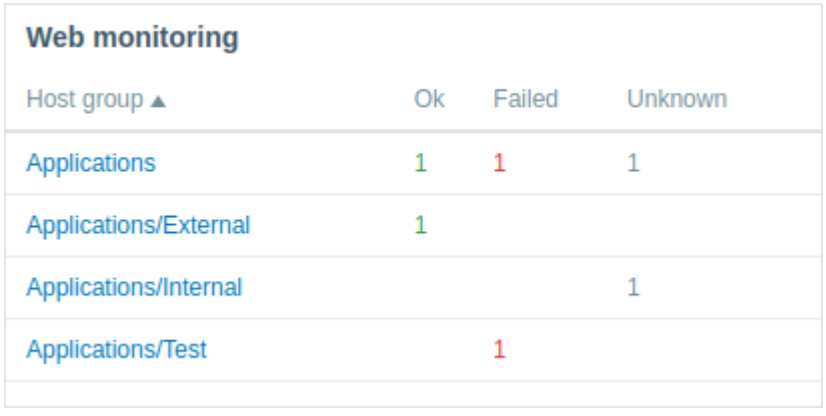
URL	Geben Sie die anzuzeigende URL ein (bis zu 2048 Zeichen). Externe URLs müssen mit `http://` oder `https://` beginnen. Interne URLs unterstützen relative Pfade (zum Beispiel `zabbix.php?action=report.status`). {HOST.*}-Makros werden unterstützt.
Override host	Wählen Sie ein kompatibles Widget oder den Dashboard-Host-Selektor als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Attention:
Browser laden möglicherweise keine im Widget konfigurierte HTTP-Seite, wenn auf das Zabbix Frontend über HTTPS zugegriffen wird. Möglicherweise müssen Sie außerdem den X-Frame-Options HTTP response header ändern, um Inhalte außerhalb des Zabbix Frontend anzuzeigen.

33 Web-Überwachung

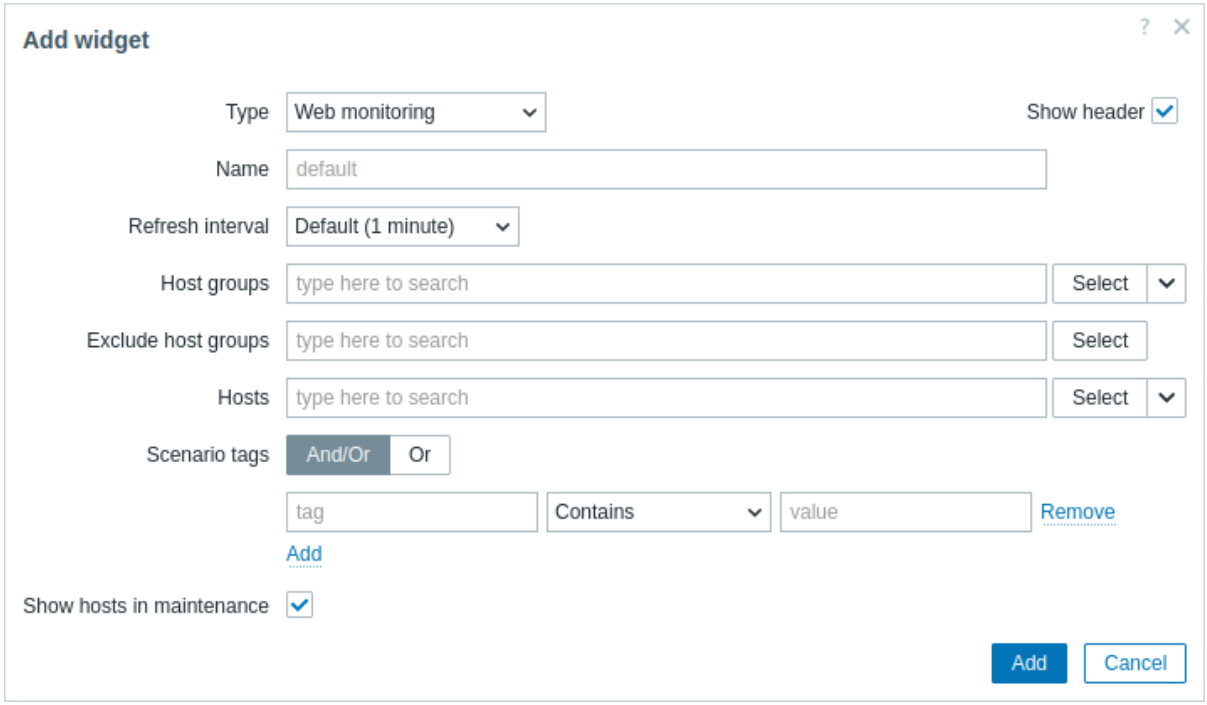
Übersicht

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.

2 Überwachung

Übersicht

Im Menü „Monitoring“ dreht sich alles um die Anzeige von Daten. Unabhängig davon, welche Informationen Zabbix zum Erfassen, Visualisieren und Verarbeiten konfiguriert ist, werden sie in den verschiedenen Bereichen des Menüs „Monitoring“ angezeigt.

Schaltflächen für den Ansichtsmodus

Die folgenden Schaltflächen in der oberen rechten Ecke sind in jedem Abschnitt gleich:

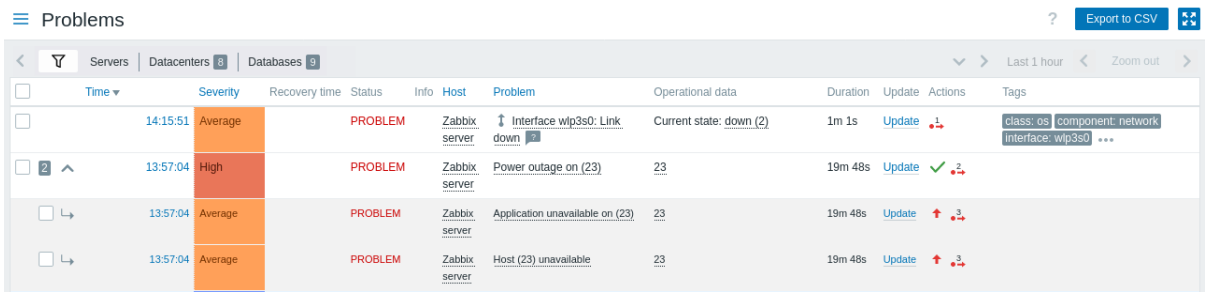
	Seite im Kioskmodus anzeigen. In diesem Modus wird nur der Seiteninhalt angezeigt. Um den Kioskmodus zu verlassen, bewegen Sie den Mauszeiger, bis die Schaltfläche  zum Beenden erscheint, und klicken Sie darauf. Sie kehren dann zum normalen Modus zurück.

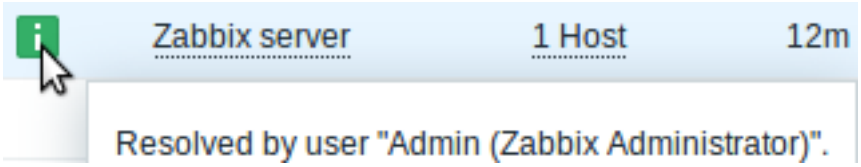
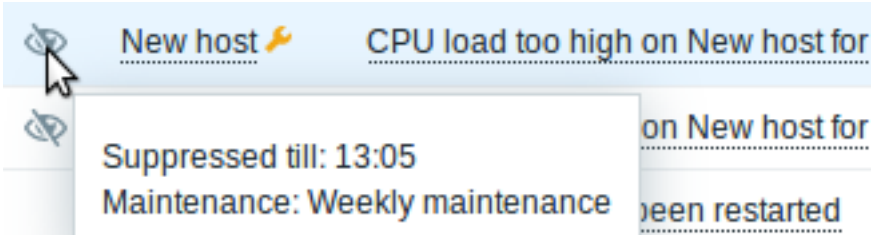
1 Probleme

Übersicht

Unter *Monitoring* > *Probleme* können Sie sehen, welche Probleme derzeit vorliegen. Probleme sind diejenigen Auslöser, die sich im Status „Problem“ befinden.

Standardmäßig werden alle neuen Probleme als Ursacheprobleme klassifiziert. Bestimmte Probleme können manuell als Symptomproblem eines Ursacheproblems umklassifiziert werden. Weitere Details finden Sie unter Ursachen- und Symptomereignisse.



Spalte	Beschreibung
Kontrollkästchen	Kontrollkästchen zur Problemauswahl werden angezeigt. Die Symbole neben den Kontrollkästchen haben folgende Bedeutung:  - die Anzahl der Symptomereignisse für das Ursacheproblem;  - erweitern, um Symptomereignisse anzuzeigen;  - einklappen, um Symptomereignisse auszublenden;  - dies ist ein Symptomereignis.
Zeit	Die Startzeit des Problems wird angezeigt.
Schweregrad	Der Schweregrad des Problems wird angezeigt. Der Schweregrad des Problems basiert ursprünglich auf dem Schweregrad des zugrunde liegenden Problem-Auslösers, kann jedoch nach Eintritt des Ereignisses über den Bildschirm *Problem aktualisieren* geändert werden. Die Farbe des Problemschweregrads wird während der Problemzeit als Zellhintergrund verwendet.
Wiederherstellungszeit	Die Zeit der Problembehebung wird angezeigt.
Status	Der Problemstatus wird angezeigt: **Problem** - ungelöstes Problem **Behoben** - kürzlich behobenes Problem. Kürzlich behobene Probleme können über den Filter ausgeblendet werden. Neue und kürzlich behobene Probleme blinken 2 Minuten lang. Behobene Probleme werden insgesamt 5 Minuten lang angezeigt. Beide Werte sind unter *Administration > General > Optionen zur Auslöseranzeige* konfigurierbar.
Info	Ein grünes Informationssymbol wird angezeigt, wenn ein Problem durch globale Korrelation oder manuell beim Aktualisieren des Problems geschlossen wurde. Wenn Sie mit der Maus über das Symbol fahren, werden weitere Details angezeigt:  Das folgende Symbol wird angezeigt, wenn ein unterdrücktes Problem dargestellt wird (siehe Option *Unterdrückte Probleme anzeigen* im Filter). Wenn Sie mit der Maus über das Symbol fahren, werden weitere Details angezeigt: 
Host	Der Host des Problems wird angezeigt. Ein Klick auf den Hostnamen öffnet das Host-Menü.
Problem	Der Problemname wird angezeigt. Der Problemname basiert auf dem Namen des zugrunde liegenden Problem-Auslösers. Makros im Auslösernamen werden zum Zeitpunkt des Problemauftretens aufgelöst; die aufgelösten Werte werden danach nicht mehr aktualisiert. *Beachten Sie*, dass es möglich ist, den Problemnamen um Betriebsdaten zu erweitern, die einige aktuelle Datenpunktwerte anzeigen. Ein Klick auf den Problemnamen öffnet das Ereignismenü. Wenn Sie nach dem Problemnamen auf das Symbol  zeigen, wird die Auslöserbeschreibung angezeigt (für Probleme, bei denen eine vorhanden ist).
Betriebsdaten	Betriebsdaten werden angezeigt und enthalten aktuelle Datenpunktwerte. Betriebsdaten können eine Kombination aus Text und Datenpunktwert-Makros sein, wenn dies auf Auslöserebene konfiguriert ist. Wenn auf Auslöserebene keine Betriebsdaten konfiguriert sind, werden die aktuellen Werte aller Datenpunkte aus dem Ausdruck angezeigt. Diese Spalte wird nur angezeigt, wenn im Filter für *Betriebsdaten anzeigen* die Option *Separat* ausgewählt ist.
Dauer	Die Problemdauer wird angezeigt. Siehe auch: Negative Problemdauer

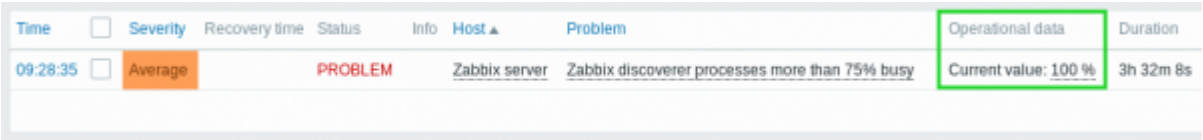
Spalte	Beschreibung
Aktualisieren	Klicken Sie auf den Link *Aktualisieren*, um zum Bildschirm Problem aktualisieren zu gelangen, auf dem verschiedene Aktionen für das Problem durchgeführt werden können, einschließlich Kommentieren und Bestätigen des Problems.
Aktionen	Der Verlauf der Aktivitäten zum 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 Problemschweregrad wurde erhöht (z. B. Information > Warnung).  - der Problemschweregrad wurde verringert (z. B. Warnung > Information).  - der Problemschweregrad wurde geändert, aber auf die ursprüngliche Stufe zurückgesetzt (z. B. Warnung > Information > Warnung).  - das Problem wurde unterdrückt.  - die Unterdrückung des Problems wurde aufgehoben.  - Aktionen wurden ausgeführt. Die Anzahl der Aktionen wird ebenfalls angezeigt.  - Aktionen wurden ausgeführt, mindestens eine davon ist noch in Bearbeitung. Die Anzahl der Aktionen wird ebenfalls angezeigt.  - Aktionen wurden ausgeführt, mindestens eine davon ist fehlgeschlagen. Die Anzahl der Aktionen wird ebenfalls angezeigt. Wenn Sie mit der Maus über die Symbole fahren, werden Popups mit Details zur Aktivität angezeigt. Unter Details anzeigen erfahren Sie mehr über die im Popup für ausgeführte Aktionen verwendeten Symbole.
Tags	Tags werden angezeigt (falls vorhanden). Zusätzlich können auch Tags aus einem externen Ticketsystem angezeigt werden (siehe die Option *Tags verarbeiten* bei der Konfiguration von webhooks).

Betriebsdaten von Problemen

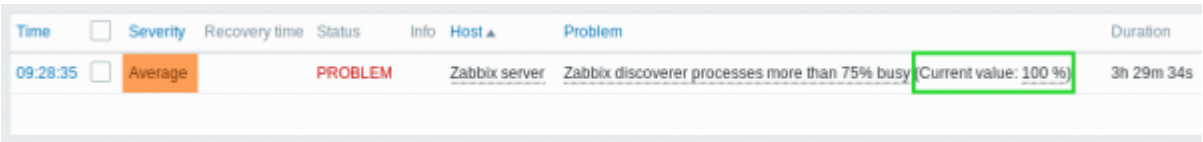
Es ist möglich, Betriebsdaten für aktuelle Probleme anzuzeigen, d. h. die neuesten Datenpunkt-Werte im Gegensatz zu den Datenpunkt-Werten zum Zeitpunkt des Problems.

Die Anzeige von Betriebsdaten kann im Filter unter *Monitoring* > *Probleme* oder in der Konfiguration des jeweiligen Dashboard-Widgets konfiguriert werden, indem eine von drei Optionen ausgewählt wird:

- *Keine* - es werden keine Betriebsdaten angezeigt
- *Getrennt* - Betriebsdaten werden in einer separaten Spalte angezeigt



- *Mit Problemnamen* - Betriebsdaten werden an den Problemnamen angehängt, und zwar in Klammern. Betriebsdaten werden nur dann an den Problemnamen angehängt, wenn das Feld *Betriebsdaten* in der Auslöser-Konfiguration nicht leer ist.



Der Inhalt der Betriebsdaten kann für jeden Auslöser im Feld *Betriebsdaten* konfiguriert werden. Dieses Feld akzeptiert eine beliebige Zeichenfolge mit Makros, insbesondere das Makro `{ITEM.LASTVALUE<1-9>}`.

`{ITEM.LASTVALUE<1-9>}` in diesem Feld wird immer zu den neuesten Werten der Datenpunkte im Auslöser-Ausdruck aufgelöst.
`{ITEM.VALUE<1-9>}` in diesem Feld wird zu den Datenpunkt-Werten zum Zeitpunkt der Änderung des Auslöser-Status aufgelöst

(d. h. beim Wechsel zu **Problem**, beim Wechsel zu **OK**, beim manuellen Schließen durch den Benutzer oder beim Schließen durch Korrelation).

Beachten Sie, dass das manuelle Schließen des Problems keinen neuen Wert erzeugt, sodass der aufgelöste Wert von {ITEM.LASTVALUE<1-9>} oder {ITEM.VALUE<1-9>} weiterhin den Wert aus dem Problemzeitpunkt anzeigt.

{ITEM.LASTVALUE<1-9>} oder {ITEM.VALUE<1-9>} wird zu *UNKNOWN* aufgelöst, wenn der neueste Verlaufswert vor mehr als der Zeitspanne *Max history display period* erfasst wurde (siehe *Administration > General*).

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 zur Massenbearbeitung

Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Massenaktualisierung* - die ausgewählten Probleme aktualisieren, indem Sie zum Bildschirm Problemaktualisierung navigieren

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:

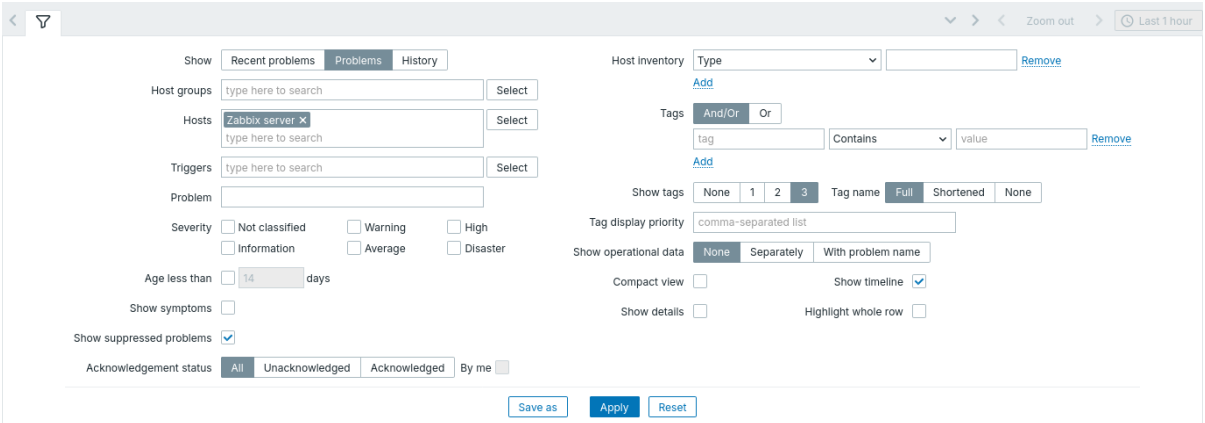
Export to CSV	Inhalt von allen Seiten in eine CSV-Datei exportieren.

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. Bevorzugte Filtereinstellungen können als Registerkarten gespeichert und dann durch Klicken auf die Registerkarten oberhalb des Filters schnell aufgerufen werden.



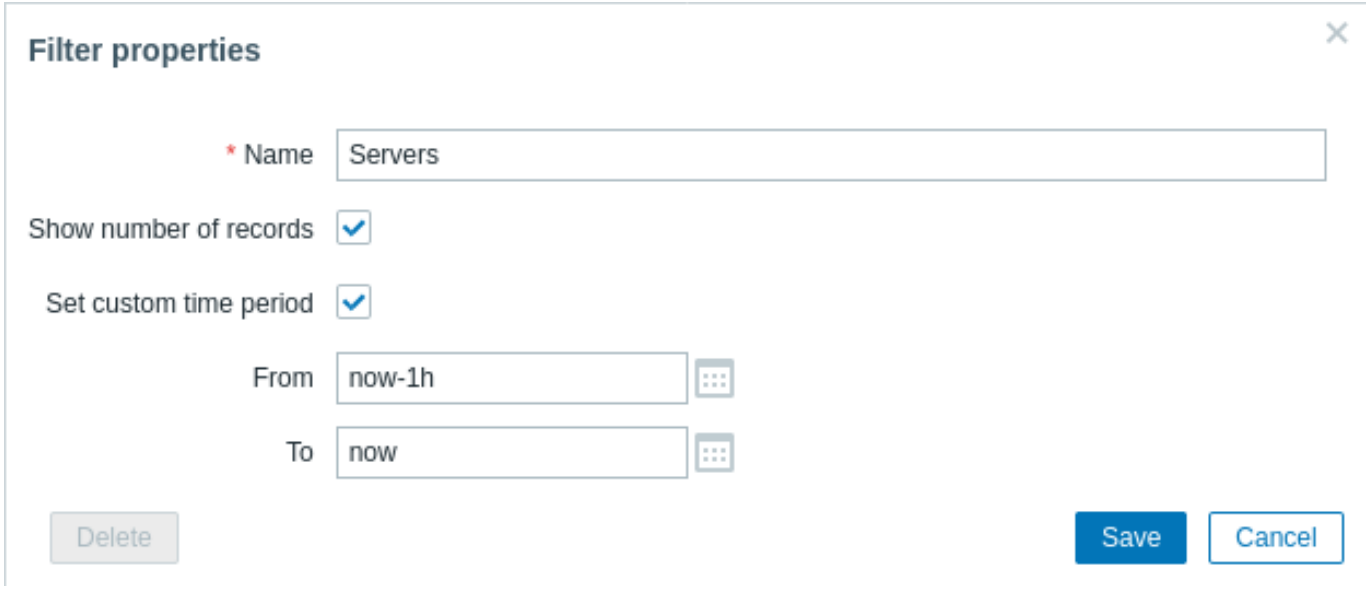
Parameter	Beschreibung
Show	Nach Problemstatus filtern: **Recent problems** - nicht gelöste und kürzlich gelöste Probleme werden angezeigt (Standard) **Problems** - nicht gelöste Probleme werden angezeigt **History** - der Verlauf aller Ereignisse wird angezeigt
Host groups	Nach einer oder mehreren Host-Gruppen filtern. Die Angabe einer übergeordneten Host-Gruppe wählt implizit auch alle darunterliegenden Host-Gruppen aus.
Hosts	Nach einem oder mehreren Hosts filtern.
Triggers	Nach einem oder mehreren Auslösern filtern.
Problem	Nach Problemnamen filtern.
Severity	Nach Auslöser-(Problem-)Schweregrad filtern.
Age less than	Danach filtern, wie alt das Problem ist.
Show symptoms	Aktivieren Sie das Kontrollkästchen, um Probleme, die als Symptome klassifiziert sind, in einer eigenen Zeile anzuzeigen.
Show suppressed problems	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung oder einzelner Problemunterdrückung unterdrückt (nicht angezeigt) würden.
Acknowledgement status	Filtern, um alle Probleme, nur unbestätigte Probleme oder nur bestätigte Probleme anzuzeigen. Aktivieren Sie das zusätzliche Kontrollkästchen, um Probleme herauszufiltern, die jemals von Ihnen bestätigt wurden.
Host inventory	Nach Inventartyp und Wert filtern.
Tags	Nach Name und Wert des Ereignis-Tags 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. 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 Beim Filtern werden die hier angegebenen Tags beim Problem zuerst angezeigt, sofern dies nicht durch die Liste *Tag display priority* (siehe unten) überschrieben wird.
Show tags	Wählen Sie die Anzahl der angezeigten Tags: **None** - keine Spalte *Tags* in *Monitoring* > *Problems* **1** - die Spalte *Tags* enthält ein Tag **2** - die Spalte *Tags* enthält zwei Tags **3** - die Spalte *Tags* enthält drei Tags Um alle Tags für das Problem zu sehen, 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; Tag-Werte werden vollständig angezeigt **None** - nur Tag-Werte werden angezeigt; keine Namen
Tag display priority	Geben Sie die Anzeigereihenfolge der Tags für ein Problem als kommagetrennte Liste von Tags ein (zum Beispiel: `customer,scope,component`). Es sollten nur Tag-Namen verwendet werden, keine Werte. Die Tags aus 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 in Klammern an den Problemnamen angehängt
Compact view	Aktivieren Sie das Kontrollkästchen, um die Kompaktansicht zu aktivieren.
Show details	Aktivieren Sie das Kontrollkästchen, um die zugrunde liegenden Auslöser-Ausdrücke der Probleme anzuzeigen. Deaktiviert, wenn das Kontrollkästchen *Compact view* aktiviert ist.

Parameter	Beschreibung
Show timeline	Aktivieren Sie das Kontrollkästchen, um die visuelle Zeitleiste und Gruppierung anzuzeigen. Deaktiviert, wenn das Kontrollkästchen *Compact view* aktiviert ist.
Highlight whole row	Aktivieren Sie das Kontrollkästchen, um bei nicht gelösten Problemen die gesamte Zeile hervorzuheben. Für die Hervorhebung wird die Problem-Schweregradfarbe verwendet. *Highlight whole row* ist in den kontrastreichen Themes nicht verfügbar.

Registerkarten für bevorzugte Filter

Häufig verwendete Sätze von Filterparametern können in Registerkarten gespeichert werden.

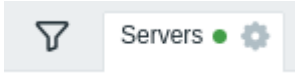
Um einen neuen Satz von Filterparametern zu speichern, öffnen Sie die Hauptregisterkarte und konfigurieren Sie die Filtereinstellungen. Klicken Sie dann auf die Schaltfläche *Save as*. Definieren Sie im neuen Popup-Fenster die *Filter properties*.



Parameter	Beschreibung
Name	Der Name des Filters, der in der Registerkartenliste angezeigt wird.
Show number of records	Aktivieren Sie diese Option, wenn die Anzahl der Probleme neben dem Registerkartennamen angezeigt werden soll.
Override time period selector	Aktivieren Sie diese Option, um für diesen Filtersatz einen bestimmten Standardzeitraum festzulegen. Wenn diese Option gesetzt ist, können Sie den Zeitraum für diese Registerkarte nur durch Aktualisieren der Filtereinstellungen ändern. Bei Registerkarten ohne benutzerdefinierten Zeitraum kann der Zeitraum durch Klicken auf die Zeitauswahl-Schaltfläche in der oberen rechten Ecke geändert werden (der Name der Schaltfläche hängt vom ausgewählten Zeitintervall ab: Diese Woche, Letzte 30 Minuten, Gestern usw.). Diese Option ist nur für Filter unter *Monitoring* > *Problems* verfügbar.
From/To	Beginn und Ende des Zeitraums 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 benannte Filterregisterkarte erstellt und sofort aktiviert.

Um die Filtereigenschaften eines vorhandenen Filters zu bearbeiten, klicken Sie auf das Zahnradsymbol neben dem Namen der aktiven Registerkarte.



Hinweise:

- Um den Filterbereich auszublenden, klicken Sie auf den Namen der aktuellen Registerkarte. Klicken Sie erneut auf den Namen der aktiven Registerkarte, um den Filterbereich wieder zu öffnen.
- Die Tastaturnavigation wird unterstützt: Verwenden Sie die Pfeiltasten, um zwischen den Registerkarten zu wechseln, und drücken Sie *Enter*, um sie zu öffnen.

- Mit den Links-/Rechts-Schaltflächen über dem Filter können Sie zwischen gespeicherten Filtern wechseln. Alternativ öffnet die nach unten zeigende Schaltfläche ein Dropdown-Menü mit allen gespeicherten Filtern, aus dem Sie den benötigten auswählen können.
- Filterregisterkarten können per Drag-and-drop neu angeordnet werden.
- 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.
- Die aktuellen Filtereinstellungen werden im Benutzerprofil gespeichert. Wenn der Benutzer die Seite erneut öffnet, bleiben die Filtereinstellungen unverändert.

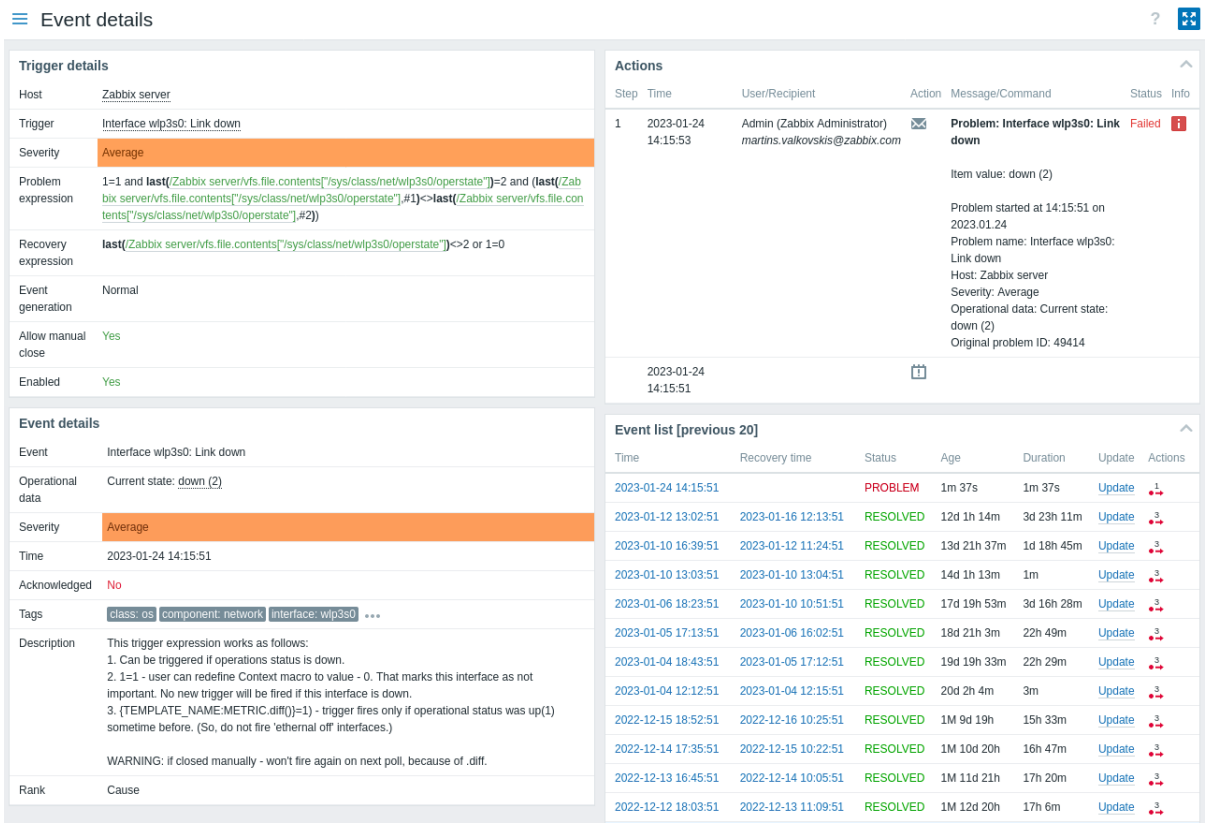
Note:
Um Filter zu teilen, kopieren Sie die URL eines aktiven Filters und senden Sie sie an andere. Nach dem Öffnen dieser URL können andere Benutzer diesen Parametersatz als permanenten Filter in ihrem Zabbix-Konto speichern.
Siehe auch: Seitenparameter.

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 Zeitangaben für Problemstart und Wiederherstellung in *Monitoring* > *Probleme* sind Links. Ein Klick darauf öffnet weitere Details zum Ereignis.



Beachten Sie, dass sich der Schweregrad des Problems für den Auslöser und das Problemereignis unterscheiden kann – wenn er für das Problemereignis über den *Problem aktualisieren*-Bildschirm aktualisiert wurde.

In der Aktionsliste werden die folgenden Symbole verwendet, um den Aktivitätstyp anzuzeigen:

-  - Problemereignis erzeugt.
-  - Nachricht wurde gesendet.
-  - Problemereignis bestätigt.
-  - Bestätigung des Problemereignisses aufgehoben.
-  - ein Kommentar wurde hinzugefügt.
-  - Problemschweregrad wurde erhöht (z. B. Information > Warnung).
-  - Problemschweregrad wurde verringert (z. B. Warnung > Information).
-  - Problemschweregrad wurde geändert, aber auf die ursprüngliche Stufe zurückgesetzt (z. B. Warnung > Information > Warnung).
-  - 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 in ein Symptomproblem umgewandelt.
-  - das Problem wurde in ein Ursachproblem umgewandelt.

1 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.

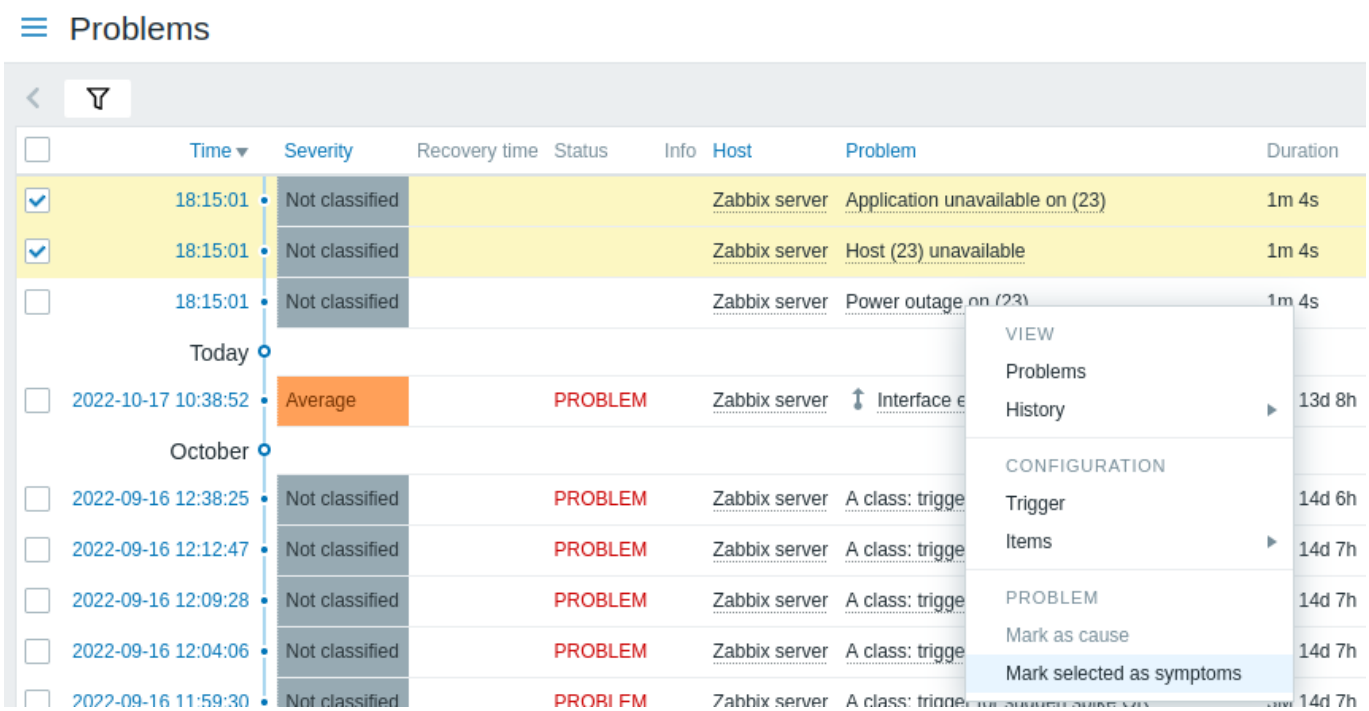
Nur Ursacheprobleme werden in den Problemgesamtzahlen auf Karten, in Dashboard-Widgets wie *Probleme nach Schweregrad* oder *Problem-Hosts* usw. gezählt. Die Problemrangfolge wirkt sich jedoch nicht auf Dienste aus.

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 *Ausgewählte als Symptome markieren*.



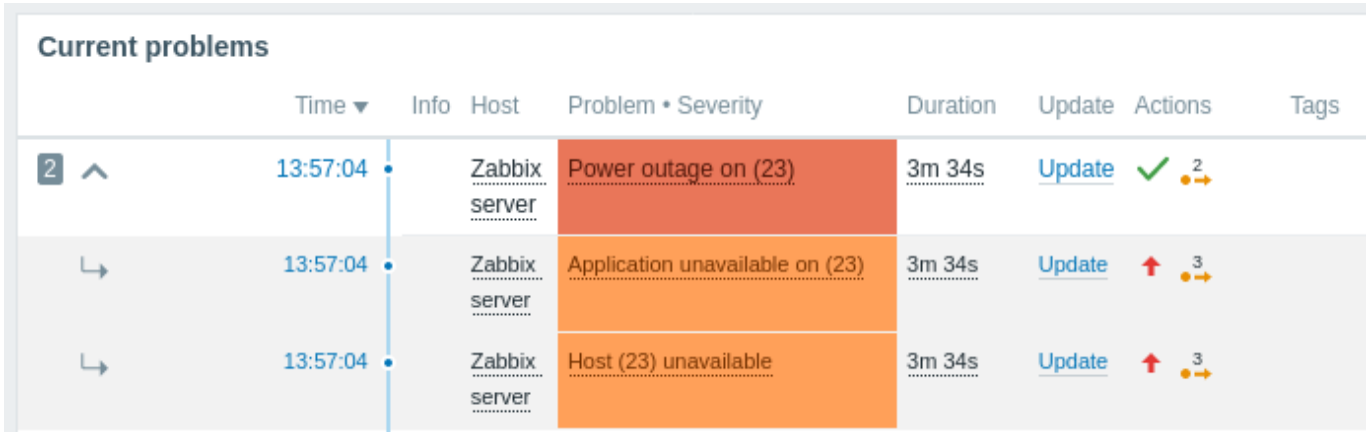
Danach werden die ausgewählten Probleme vom Server zu Symptomproblemen des Ursacheproblems aktualisiert.

Während der Status des Problems aktualisiert wird, wird es auf eine von zwei Arten angezeigt:

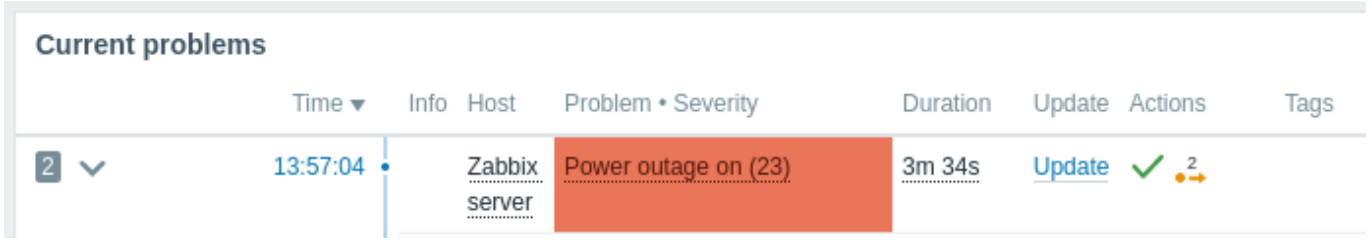
- In der Spalte Status wird ein blinkender Status ”UPDATING” angezeigt;
- Ein blinkendes Symbol oder in der Spalte Info (dies gilt, wenn im Filter nur *Probleme* ausgewählt sind und daher die Spalte Status nicht angezeigt wird).

Anzeige

Symptomprobleme werden unterhalb des Ursacheproblems angezeigt und entsprechend markiert in *Monitoring* -> *Probleme* (sowie im Dashboard-Widget *Probleme*) – mit einem Symbol, kleinerer Schrift und einem anderen Hintergrund.



In der eingeklappten Ansicht ist nur das Ursacheproblem sichtbar; das Vorhandensein von Symptomproblemen wird durch die Zahl am Anfang der Zeile und das Symbol zum Erweitern der Ansicht angezeigt.



Es ist auch möglich, Symptomprobleme zusätzlich in normaler Schrift und in einer eigenen Zeile anzuzeigen. Wählen Sie dazu *Symptome anzeigen* in den Filtereinstellungen oder in der Widget-Konfiguration aus.

Zurücksetzen auf Ursacheproblem

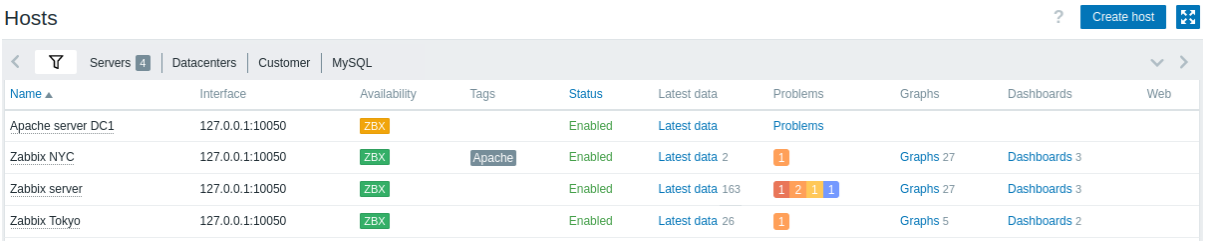
Ein Symptomproblem kann wieder in ein Ursacheproblem zurückgesetzt werden. Gehen Sie dazu wie folgt vor:

- Klicken Sie im Kontextmenü des Symptomproblems auf die Option *Als Ursache markieren*.
- Aktivieren Sie im Bildschirm zur Problemaktualisierung die Option *In Ursache umwandeln* und klicken Sie auf *Aktualisieren* (diese Option funktioniert auch, wenn mehrere Probleme ausgewählt sind).

2 Hosts

Übersicht

Der Abschnitt *Überwachung* → *Hosts* zeigt eine vollständige Liste der überwachten Hosts mit detaillierten Informationen über Host-Schnittstellen, Verfügbarkeit, Tags, aktuelle Probleme, Status (aktiviert/deaktiviert) sowie Links, um einfach zu den neuesten Daten des Hosts, zur Problemhistorie, zu Graphen, Dashboards und Webszenarien zu navigieren.



Spalte	Beschreibung
Name	Der sichtbare Host-Name. Ein Klick auf den Namen öffnet das Host-Menü. Ein oranges Schraubenschlüssel-Symbol nach dem Namen zeigt an, dass sich dieser Host in Wartung befindet. Klicken Sie auf die Spaltenüberschrift, um Hosts nach Namen in auf- oder absteigender Reihenfolge zu sortieren.
Schnittstelle	Die Hauptschnittstelle des Hosts wird angezeigt.

Spalte	Beschreibung
Verfügbarkeit	Die Host-Verfügbarkeit pro konfigurierter Schnittstelle wird angezeigt. Die Verfügbarkeitssymbole stellen den aktuellen Status der Host-Schnittstelle auf dem Zabbix-Server dar. Wenn Sie daher einen Host im Frontend deaktivieren, wird seine Verfügbarkeit aktualisiert, nachdem der Zabbix-Server die Konfigurationsänderungen synchronisiert hat. Ebenso wird bei Aktivierung eines Hosts seine Verfügbarkeit aktualisiert, nachdem der Zabbix-Server die Konfigurationsänderungen synchronisiert und den Host abgefragt hat. Die Verfügbarkeitssymbole stellen nur die konfigurierten Schnittstellentypen dar (Agent, SNMP, IPMI, JMX). 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 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 bei 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 (Zeitüberschreitung, Firewall-Probleme usw.); **Unbekannt** - die Verbindung zum Host wurde nicht versucht 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 einzelnen Typs (Agent, SNMP, IPMI, JMX) wird durch diejenigen 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 auch die gesamte Verfügbarkeit der Agent-Schnittstelle wie oben beschrieben. Zur Bestimmung der Verfügbarkeit aktiver Prüfungen werden Heartbeat-Nachrichten im Thread für aktive Agent-Prüfungen 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 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.
Tags	Tags des Hosts und aller verknüpften Vorlagen, wobei Makros nicht aufgelöst werden.
Status	Host-Status - *Aktiviert* oder *Deaktiviert*. Klicken Sie auf die Spaltenüberschrift, um Hosts nach Status in auf- oder absteigender Reihenfolge zu sortieren.
Neueste Daten	Ein Klick auf den Link öffnet die Seite *Überwachung* **-** *Neueste Daten* mit allen zuletzt vom Host erfassten Daten. Die Anzahl der Datenpunkte mit neuesten Daten wird grau angezeigt.
Probleme	Die Anzahl offener Host-Probleme, nach Schweregrad sortiert. Die Farbe des Quadrats zeigt den Schweregrad des Problems an. Die Zahl im Quadrat gibt die Anzahl der Probleme für den jeweiligen Schweregrad an. Ein Klick auf das Symbol öffnet die Seite *Überwachung* **-** *Probleme* für den aktuellen Host. Wenn ein Host keine Probleme hat, wird ein Link zum Abschnitt *Probleme* für diesen Host als Text angezeigt. Verwenden Sie den Filter, um auszuwählen, ob unterdrückte Probleme einbezogen werden sollen (standardmäßig nicht einbezogen).

Spalte	Beschreibung
Graphen	Ein Klick auf den Link zeigt die für den Host konfigurierten Graphen an. Die Anzahl der Graphen wird grau angezeigt. Wenn ein Host keine Graphen hat, wird kein Link angezeigt.
Dashboards	Ein Klick auf den Link zeigt die für den Host konfigurierten Dashboards an. Die Anzahl der Dashboards wird grau angezeigt. Wenn ein Host keine Dashboards hat, wird kein Link angezeigt.
Web	Ein Klick auf den Link zeigt die für den Host konfigurierten Webszenarien an. Die Anzahl der Webszenarien wird grau angezeigt. Wenn ein Host keine Webszenarien hat, wird kein Link angezeigt.

Schaltflächen

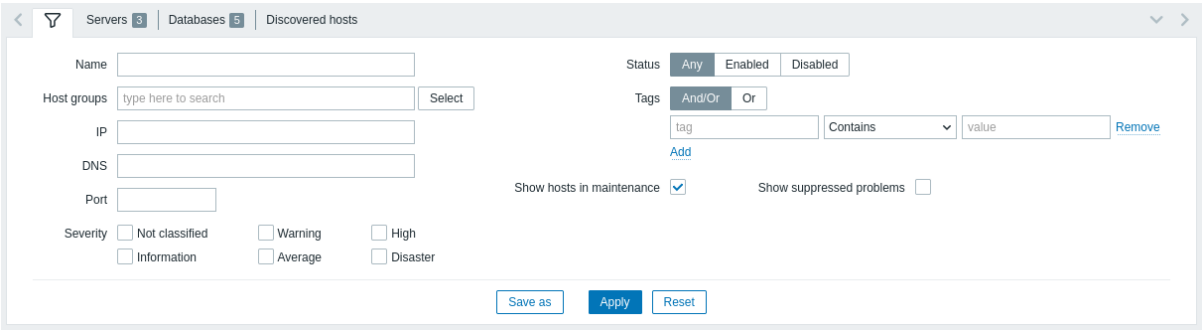
Host erstellen ermöglicht das Erstellen eines neuen Hosts. Diese Schaltfläche ist nur für Benutzer mit den Rollen Admin und Super Admin verfügbar.

Die Schaltflächen des Ansichtsmodus, die in allen Bereichen gleich sind, werden auf der Seite Monitoring beschrieben.

Filter verwenden

Sie können den Filter verwenden, um nur die Hosts anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Filter befindet sich oberhalb der Tabelle. Es ist möglich, Hosts nach Name, Host-Gruppe, IP oder DNS, Interface-Port, Tags, Problem-Schweregrad, Status (aktiviert/deaktiviert/beliebig) zu filtern; außerdem können Sie auswählen, ob unterdrückte Probleme und Hosts angezeigt werden sollen, die sich derzeit in Wartung befinden.



Parameter	Beschreibung
Name	Nach sichtbarem Host-Namen filtern.
Host groups	Nach einer oder mehreren Host-Gruppen filtern. Die Angabe einer übergeordneten Host-Gruppe wählt implizit auch alle darunterliegenden Host-Gruppen aus.
IP	Nach IP-Adresse filtern.
DNS	Nach DNS-Namen filtern.
Port	Nach Portnummer filtern.
Severity	Nach Problem-Schweregrad filtern. Standardmäßig werden Probleme aller Schweregrade angezeigt. Probleme werden angezeigt, wenn sie nicht unterdrückt sind.
Status	Nach Host-Status filtern.

Parameter	Beschreibung
Tags	Nach Host-Tag-Name und -Wert filtern. Hosts können sowohl nach Tags auf Host-Ebene als auch nach Tags aus allen verknüpften Vorlagen, einschließlich verschachtelter Vorlagen, gefiltert werden. 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
Show hosts in maintenance	Aktivieren Sie das Kontrollkästchen, um Hosts anzuzeigen, die sich in Wartung befinden (standardmäßig angezeigt).
Show suppressed problems	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung oder einzelner problem suppression unterdrückt würden (nicht angezeigt).

Filter speichern

Bevorzugte Filtereinstellungen können als Tabs gespeichert und dann schnell aufgerufen werden, indem Sie auf den entsprechenden Tab oberhalb des Filters klicken.

Weitere Details zum Speichern von Filtern.

1 Diagramme

Übersicht

Auf Host-Diagramme kann über *Monitoring* → *Hosts* zugegriffen werden, indem Sie für den jeweiligen Host auf Diagramme klicken.

Alle benutzerdefinierten Diagramme, die für den Host konfiguriert wurden, können angezeigt werden, ebenso wie alle einfachen Diagramme.

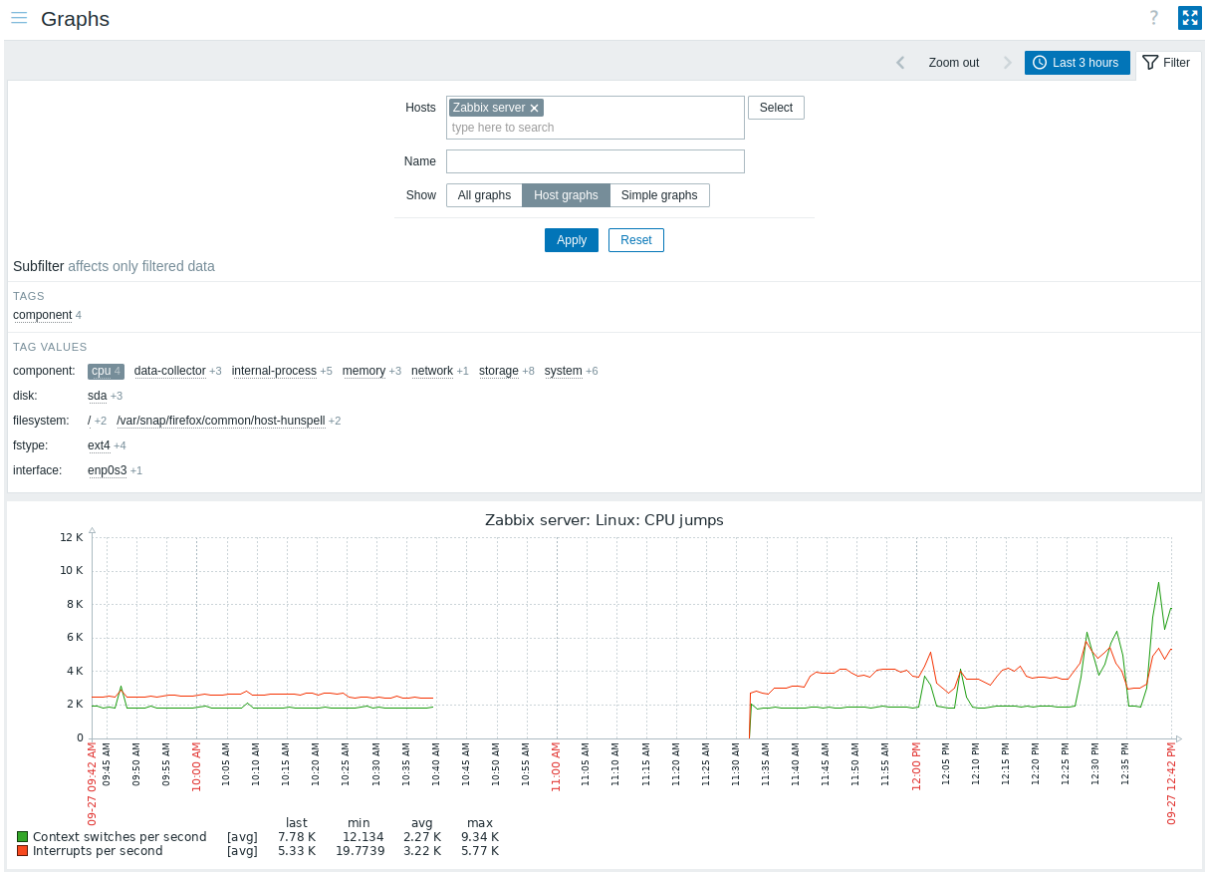


Diagramme werden sortiert nach:

- Diagrammname (benutzerdefinierte Diagramme)
- Datenpunktname (einfache Diagramme)

Diagramme für deaktivierte Hosts sind ebenfalls zugänglich.

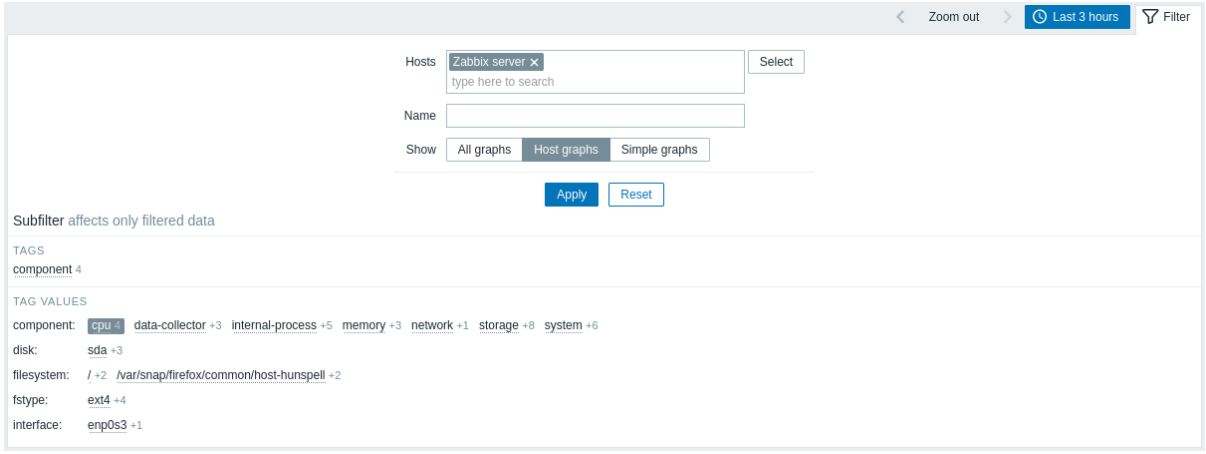
Auswahl des Zeitraums

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 Diagrammnamen und die Option *Anzeigen* (alle/Host-Diagramme/einfache Diagramme) festlegen.



Wenn im Filter kein Host ausgewählt ist, werden keine Diagramme angezeigt.

Verwendung des Unterfilters

Der Unterfilter ermöglicht es, die Filterung aus dem Hauptfilter weiter anzupassen.

Er enthält anklickbare Links für einen schnellen Zugriff auf zugehörige Diagramme. Diagramme sind über gemeinsame Entitäten verknüpft – Tag-Name oder Wert. Wenn auf einen Tag-Namen/-Wert geklickt wird, wird dieser mit einem grauen Hintergrund hervorgehoben, und die Diagramme werden sofort gefiltert (es ist nicht nötig, im Hauptfilter auf *Anwenden* zu klicken). Durch Klicken auf einen weiteren Tag-Namen/-Wert wird dieser zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Klicken auf den Tag-Namen/-Wert wird die Filterung entfernt.

Unterfilter werden auf Grundlage der gefilterten Daten erzeugt, die auf 1000 Datensätze begrenzt sind. Wenn Sie im Unterfilter mehr Datensätze sehen möchten, müssen Sie den Wert des Parameters *Limit for search and filter results* erhöhen (unter *Administration -> General -> GUI*).

Im Gegensatz zum Hauptfilter wird der Unterfilter zusammen mit jeder Aktualisierungsanfrage der Tabelle aktualisiert, damit stets aktuelle Informationen über verfügbare Filteroptionen und deren Zählerwerte abgerufen werden.

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. Vertikale Listen (wie Tags mit ihren Werten) sind auf 20 Einträge begrenzt. Wenn es mehr gibt, wird ein Symbol mit drei Punkten angezeigt; es ist nicht anklickbar.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der Diagramme an, die sie in den Ergebnissen des Hauptfilters hat.

Sobald eine Entität ausgewählt ist, werden die Zahlen bei anderen verfügbaren Entitäten mit einem Pluszeichen angezeigt, das angibt, wie viele Diagramme zur aktuellen Auswahl hinzugefügt werden können.

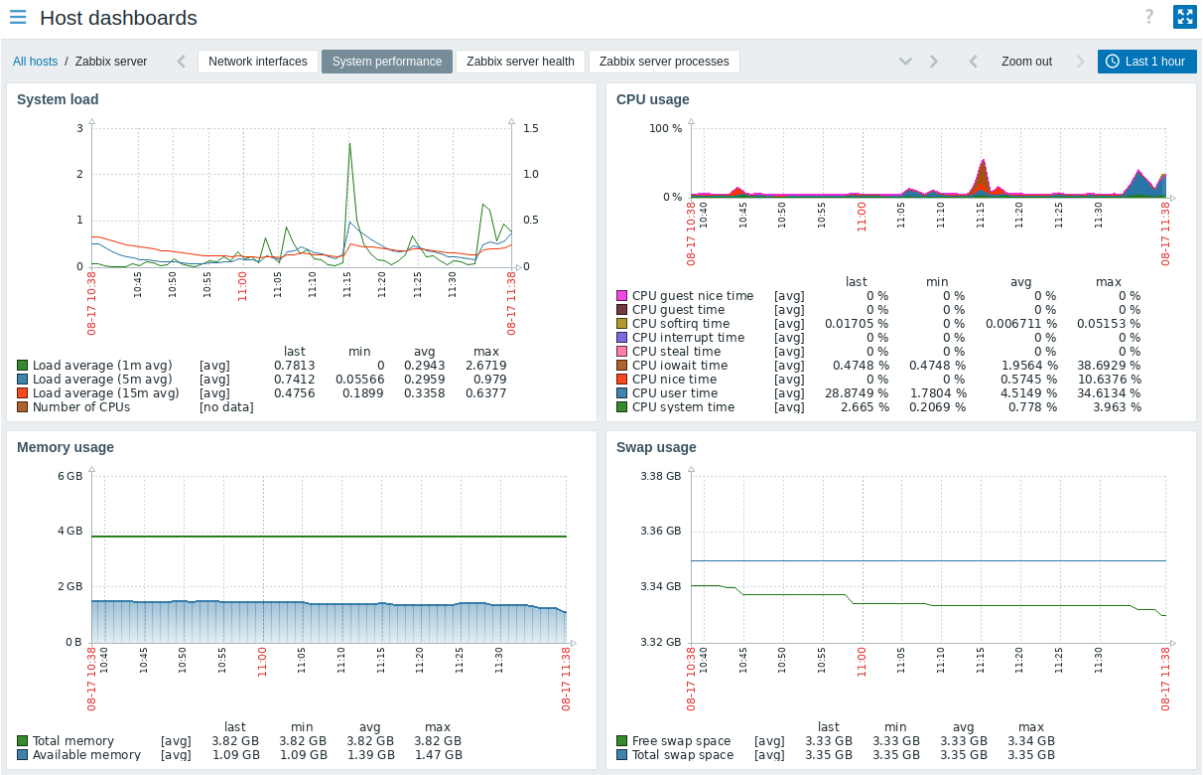
Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Bereichen gemeinsam verwendet werden, sind auf der Seite Monitoring beschrieben.

2 Host-Dashboards

Übersicht

Host-Dashboards sehen ähnlich aus wie globale Dashboards; allerdings haben Host-Dashboards 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, die die vollständige Liste der verfügbaren Host-Dashboards anzeigt.

Um zum Abschnitt *Monitoring* → *Hosts* zu wechseln, klicken Sie auf den Navigationslink *All 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.

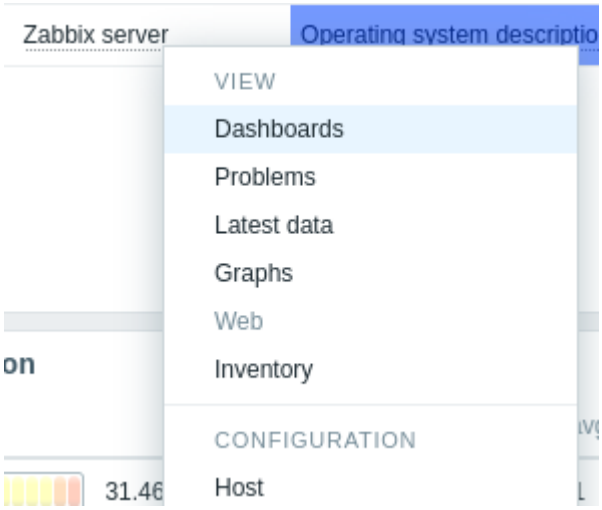
Widgets von Host-Dashboards können ebenfalls nur auf der Ebene der Vorlage konfiguriert werden, mit Ausnahme der Änderung des Aktualisierungsintervalls. Außerdem können Widgets von Host-Dashboards nur in andere Host-Dashboards innerhalb derselben Vorlage kopiert werden. Beachten Sie, dass Widgets aus 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

Auf Host-Dashboards kann zugegriffen werden:

- nach der Suche nach einem Host-Namen in der globalen Suche (klicken Sie auf den Link *Dashboards* in den Suchergebnissen);
- nach einem Klick auf einen Host-Namen unter *Inventar* → *Hosts* (klicken Sie auf den Link *Dashboards* in der Host-Übersicht);
- über das Host-Menü durch Klicken auf *Dashboards*.



Beachten Sie, dass auf Host-Dashboards im Abschnitt *Dashboards* nicht direkt zugegriffen werden kann, da dieser für globale Dashboards vorgesehen ist.

3 Web-Szenarien

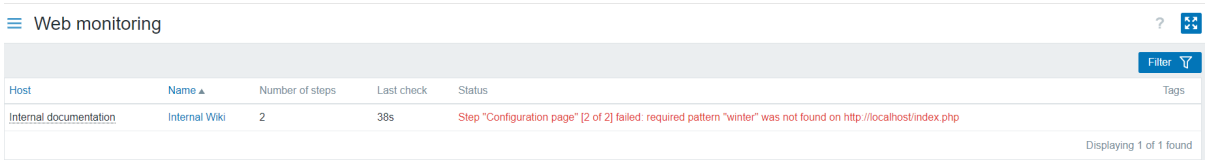
Übersicht

Informationen zu Webszenarien eines Hosts können unter *Monitoring* → *Hosts* aufgerufen werden, indem Sie beim jeweiligen Host auf Web klicken.

Ein Klick auf den Hostnamen öffnet das Host-Menü. Daten deaktivierter Hosts sind ebenfalls zugänglich. Der Name eines deaktivierten Hosts wird rot angezeigt.

Der Status eines Webszenarios 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 für Hinweise darauf, welche Parameter möglicherweise korrigiert werden müssen).
- Unbekannt (grau) - es ist noch kein Schrittstatus verfügbar oder es wurden keine Daten erfasst.



Die maximale Anzahl der pro Seite angezeigten Szenarien hängt von der Benutzereinstellung *Zeilen pro Seite* im Profil ab.

Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Diese Begrenzung wurde eingeführt, um die anfänglichen Ladezeiten bei großen Seiten mit aktuellen Daten zu verbessern. Sie können diesen Zeitraum verlängern, indem Sie den Wert des Parameters *Max history display period* im Menübereich *Administration → General → GUI* ändern.

Der Name des Szenarios ist ein Link zu detaillierteren Statistiken darüber:

Filter verwenden

Die Seite zeigt eine Liste aller Webszenarien des ausgewählten Hosts an. Um Webszenarien für einen anderen Host oder eine andere Host-Gruppe 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.

3 Letzte Daten

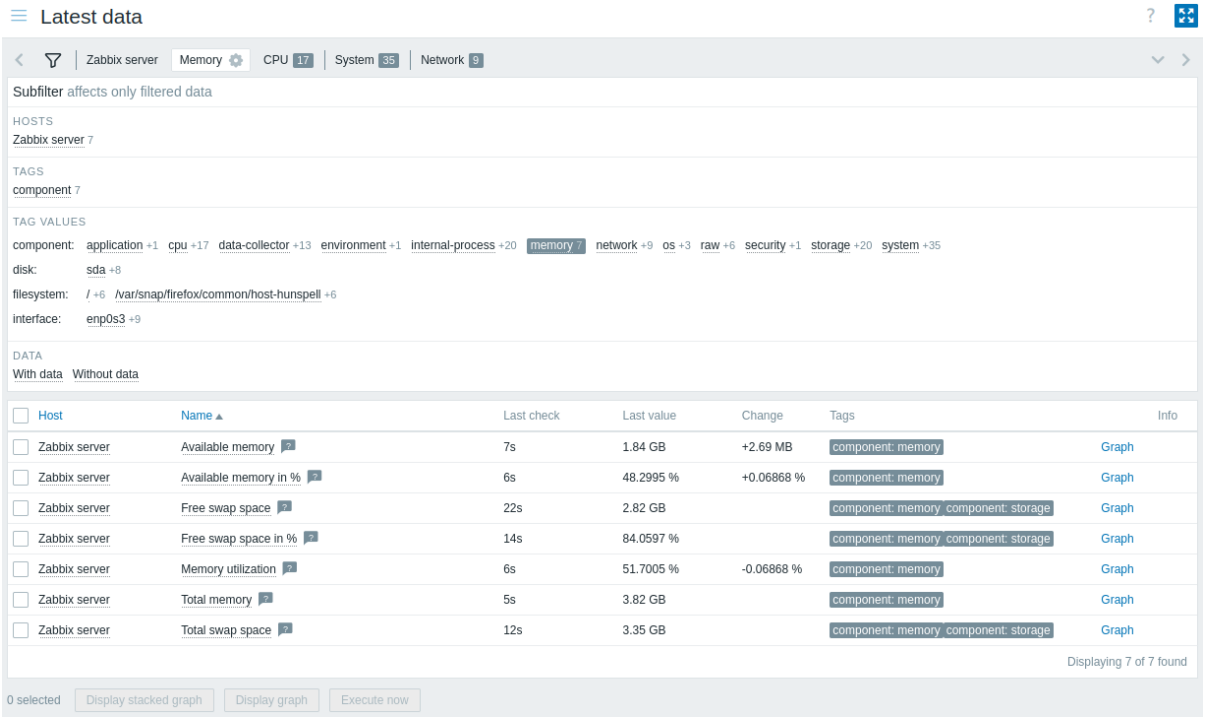
Übersicht

Der Abschnitt *Monitoring* → *Letzte Daten* zeigt die zuletzt von Datenpunkten erfassten Werte an.

Dieser Abschnitt enthält die folgenden Elemente:

- Filter
- Unterfilter
- Datenpunktliste

Note:
Der Unterfilter und die Datenpunktliste werden nur angezeigt, wenn der Filter gesetzt ist und Ergebnisse zur Anzeige vorhanden sind.



Spalte	Beschreibung
Host	Name des Hosts, zu dem der Datenpunkt gehört. Durch Klicken auf den Namen wird das Host-Kontextmenü geöffnet. Befindet sich ein Host in Wartung, wird nach dem Hostnamen ein orangefarbenes Schraubenschlüssel-Symbol angezeigt. Ist ein Host deaktiviert, wird der Name des Hosts rot dargestellt. Beachten Sie, dass Daten deaktivierter Hosts (einschließlich Graphen und Datenpunkt-Wertelisten) im Abschnitt *Letzte Daten* zugänglich sind.
Name	Name des Datenpunkts. Durch Klicken auf den Namen wird das Datenpunktmenü geöffnet. Neben dem Namen des Datenpunkts wird für alle Datenpunkte mit einer Beschreibung ein Fragezeichen-Symbol angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Beschreibung des Datenpunkts anzuzeigen.
Letzte Prüfung	Zeit seit der letzten Prüfung des Datenpunkts.
Letzter Wert	Der zuletzt erfasste Wert des Datenpunkts. Werte werden unter Anwendung von Einheitumrechnung und Wertezuordnung angezeigt. Bewegen Sie den Mauszeiger über den Wert, um Rohdaten anzuzeigen (auf 8192 Zeichen gekürzt). Binärwerte werden als Platzhalter (*Binärwert*) statt als tatsächlicher Wert angezeigt. Standardmäßig werden nur Werte angezeigt, die in den letzten 24 Stunden empfangen wurden. Diese Begrenzung verbessert die anfänglichen Ladezeiten bei großen Seiten mit letzten Daten; um sie zu erweitern, aktualisieren Sie den Wert des Parameters *Max history display period* unter *Administration* → *General* → *GUI*.

Spalte	Beschreibung
Änderung	Differenz zwischen dem vorherigen Wert und dem zuletzt erfassten Wert. Bei Datenpunkten mit einer Aktualisierungsfrequenz von 1 Tag oder mehr wird der Änderungsbetrag niemals angezeigt (bei der Standardeinstellung). 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. Tags in der Datenpunktliste sind anklickbar. Durch Klicken auf ein Tag wird es im Unterfilter aktiviert, sodass die Datenpunktliste nur Datenpunkte anzeigt, die dieses Tag enthalten (sowie alle anderen zuvor im Unterfilter ausgewählten Tags). Beachten Sie, dass Tags in der Datenpunktliste nicht mehr anklickbar sind, sobald Datenpunkte auf diese Weise gefiltert wurden. Weitere Änderungen auf Basis von Tags (zum Beispiel zum Entfernen von Tags oder zum Festlegen anderer Filter) müssen im Unterfilter vorgenommen werden.
Graph/Verlauf	Link zum einfachen Graphen/Verlauf der Datenpunktwerte.
Info	Zusätzliche Informationen zum Datenpunkt. Wenn ein Datenpunkt Fehler aufweist (zum Beispiel nicht mehr unterstützt wird), wird ein Informationssymbol angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um Details anzuzeigen.

Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Abschnitten gemeinsam verwendet werden, sind auf der Seite Monitoring beschrieben.

Massenaktionen

Schaltflächen unterhalb der Liste bieten Massenaktionen für einen oder mehrere ausgewählte Datenpunkte:

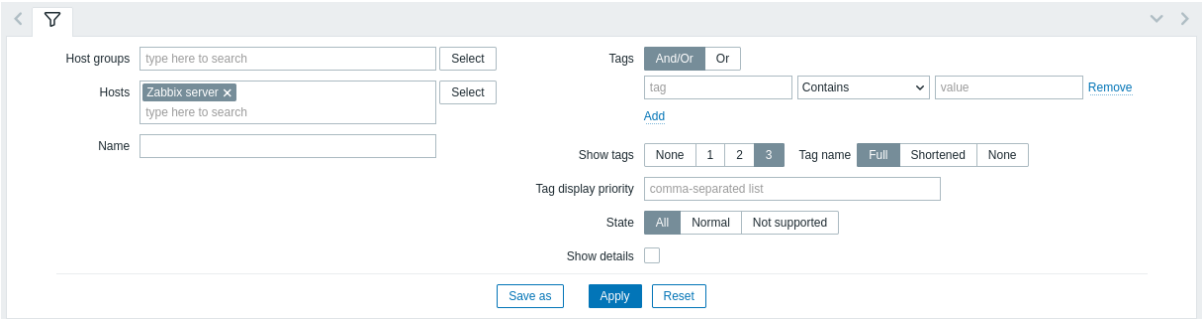
- *Gestapeltes Diagramm anzeigen* - ein gestapeltes Ad-hoc-Diagramm anzeigen.
- *Diagramm anzeigen* - ein einfaches Ad-hoc-Diagramm anzeigen.
- *Jetzt ausführen* - eine Prüfung auf neue Datenpunktwerte sofort ausführen. Nur für **passive** Prüfungen unterstützt (siehe weitere Details). Diese Option ist nur für Hosts mit Lese-/Schreibzugriff verfügbar. Der Zugriff auf diese Option für Hosts mit schreibgeschützten Berechtigungen hängt von der Option der Benutzerrolle namens *„Jetzt ausführen“ auf schreibgeschützten Hosts 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 für den Filter befindet sich oberhalb der Datenpunktliste und des Unterfilters. Klicken Sie darauf, um den Filter zu erweitern.



Mit dem Filter kann die Datenpunktliste nach Host-Gruppe, Host, Datenpunktname, Tag, Status und anderen Einstellungen eingegrenzt werden. Wenn im Filter eine übergeordnete Host-Gruppe angegeben wird, werden implizit auch alle verschachtelten Host-Gruppen ausgewählt. Weitere Informationen zum Filtern nach Tags finden Sie unter *Monitoring* → *Probleme*.

Mit der Filteroption *Details anzeigen* können die für die Datenpunkte angezeigten Informationen erweitert werden. Aktivieren Sie diese Option, um Details wie das Aktualisierungsintervall des Datenpunkts, Verlaufs- und Trend-Einstellungen, den Datenpunkttyp sowie Datenpunktfehler (in Ordnung/nicht unterstützt) anzuzeigen.

Filter speichern

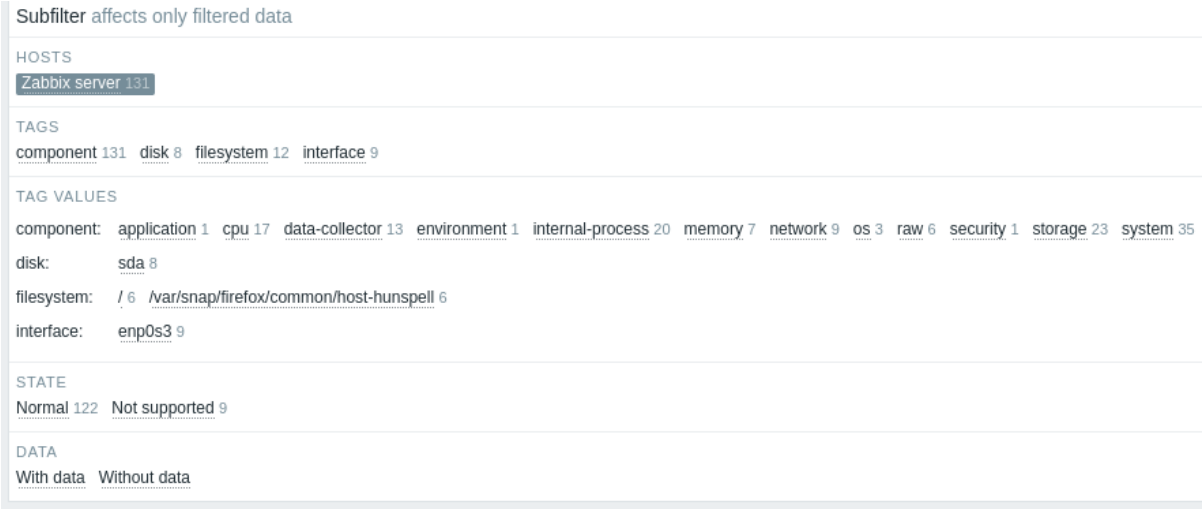
Bevorzugte Filtereinstellungen können als Registerkarten gespeichert und dann durch Klicken auf die jeweilige Registerkarte oberhalb des Filters schnell aufgerufen werden.

Weitere Details zum Speichern von Filtern.

Verwendung des Subfilters

Der Subfilter ermöglicht es, die Filterung aus dem Hauptfilter weiter anzupassen.

Er enthält anklickbare Links für einen schnellen Zugriff auf zugehörige Datenpunkte. Datenpunkte sind über eine gemeinsame Entität miteinander verknüpft – Host, Tag-Name oder -Wert, Datenpunktstatus oder Datenstatus. Wenn auf eine Entität geklickt wird, wird diese mit einem grauen Hintergrund hervorgehoben, und die Datenpunkte werden sofort gefiltert (es ist nicht nötig, im Hauptfilter auf *Anwenden* zu klicken). Durch Klicken auf eine weitere Entität wird diese zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Klicken auf die Entität wird die Filterung entfernt.



Subfilter werden auf Grundlage der gefilterten Daten erzeugt, die auf 1000 Datensätze begrenzt sind. Wenn es 20 Hosts mit jeweils 100 Datenpunkten gibt (also insgesamt 2000 Datensätze), ist nur die Hälfte der Hosts im Subfilter sichtbar. Wenn Sie mehr Datensätze im Subfilter sehen möchten, müssen Sie den Wert des Parameters *Limit for search and filter results* erhöhen (unter *Administration -> General -> GUI*).

Im Gegensatz zum Hauptfilter wird der Subfilter bei jeder Anforderung zur Tabellenaktualisierung aktualisiert, damit stets aktuelle Informationen über verfügbare Filteroptionen und deren Zählerwerte vorliegen.

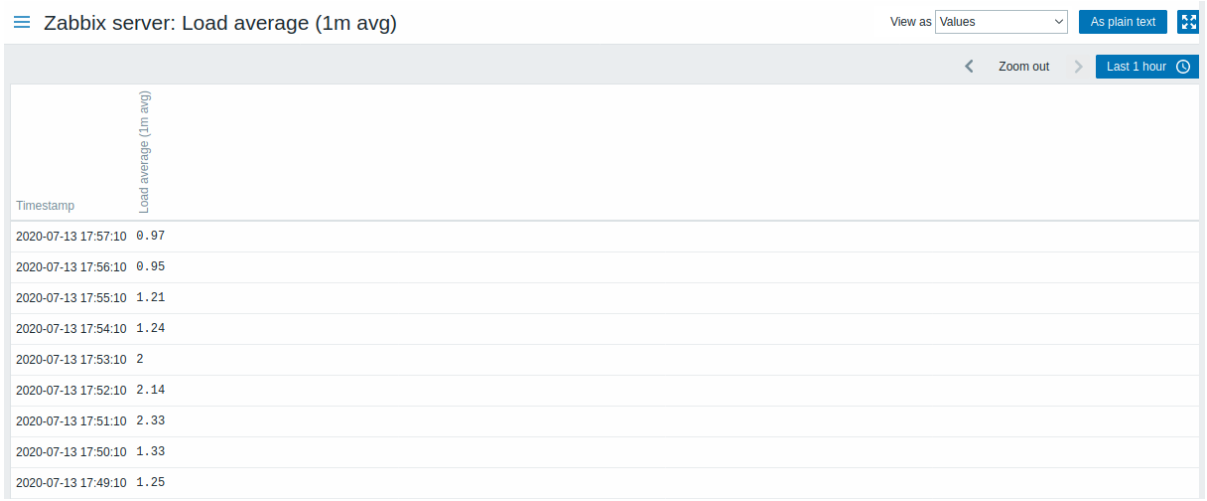
Für jede Entitätsgruppe (Hosts, Tags, Tag-Werte usw.) werden bis zu 10 Zeilen mit Entitäten angezeigt. Wenn es mehr Entitäten gibt, kann diese Liste durch Klicken auf das Dreipunkt-Symbol *** am Ende der Liste erweitert werden, um maximal 1000 Einträge anzuzeigen (der Wert von SUBFILTER_VALUES_PER_GROUP 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 eingeklappt werden kann.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der darin gruppierten Datenpunkte an (basierend auf den Ergebnissen des Hauptfilters). Wenn auf eine Entität geklickt wird, werden die Zahlen bei anderen verfügbaren Entitäten mit einem Pluszeichen angezeigt, das angibt, wie viele Datenpunkte zur aktuellen Auswahl hinzugefügt werden können. Entitäten ohne Datenpunkte werden nicht angezeigt, es sei denn, sie wurden zuvor im Subfilter ausgewählt.

Graphen und Verlauf

Die Spalte *Graph/History* in der Datenpunktliste bietet die folgenden Links:

- **History** - für alle textuellen Datenpunkte; führt zu Listen (*Values/500 latest values*), die den Verlauf vorheriger Datenpunktwerte anzeigen.
- **Graph** - für alle numerischen Datenpunkte; führt zu einem einfachen Graphen. Beachten Sie, dass beim Anzeigen des Graphen ein Dropdown-Menü oben rechts die Möglichkeit bietet, ebenfalls zu *Values/500 latest values* zu wechseln.



Die in dieser Liste angezeigten Werte sind Rohwerte, das heißt, es wird keine Nachbearbeitung angewendet.

Note:
Die Gesamtzahl der angezeigten Werte wird durch den Wert des Parameters *Limit for search and filter results* bestimmt, der unter *Administration* → *General* → *GUI* festgelegt wird.

4 Karten

Übersicht

Im Abschnitt *Monitoring* → *Karten* können Sie Netzwerkkarten konfigurieren, verwalten und anzeigen.

Wenn Sie diesen Abschnitt öffnen, sehen Sie entweder die zuletzt aufgerufene Karte oder eine Liste aller Karten, auf die Sie Zugriff haben.

Alle Karten können entweder öffentlich oder privat sein. Öffentliche Karten stehen allen Benutzern zur Verfügung, während private Karten nur für ihren Eigentümer und die Benutzer zugänglich sind, mit denen die Karte geteilt wurde.

Kartenübersicht

Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Karte. Klicken Sie auf den Namen, um die Karte anzuzeigen.
Width	Die Breite der Karte wird angezeigt.
Height	Die Höhe der Karte wird angezeigt.
Actions	Zwei Aktionen sind verfügbar: **Properties** - allgemeine Eigenschaften der Karte festlegen **Edit** - Zugriff auf das Raster zum Hinzufügen von Kartenelementen

Um eine neue Karte zu konfigurieren, 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 ihr Besitzer 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

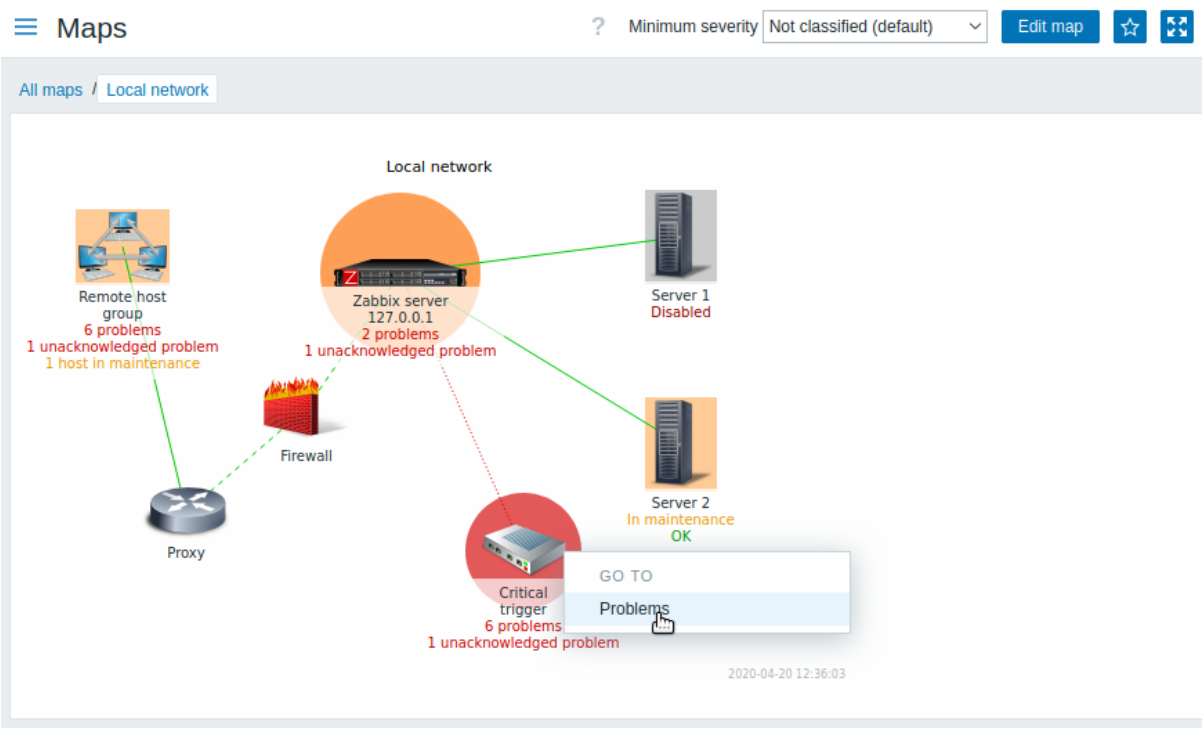
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den entsprechenden Karten und klicken Sie dann auf die gewünschte Schaltfläche.

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.

Karten anzeigen

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 die niedrigste Schweregradstufe der anzuzeigenden Problem-Auslöser auszuwählen. Der als *Standard* markierte Schweregrad ist die in der Kartenkonfiguration festgelegte Stufe. Wenn die Karte eine Unterkarte enthält, wird beim Navigieren zur Unterkarte der Schweregrad der übergeordneten Karte beibehalten (außer wenn er *Nicht klassifiziert* ist; in diesem Fall wird er nicht an die Unterkarte übergeben).

Hervorhebung von Symbolen

Wenn sich ein Kartenelement im Problemzustand befindet, wird es mit einem runden Kreis hervorgehoben. Die Füllfarbe des Kreises entspricht der Schweregradfarbe des Problems. Nur Probleme auf oder über dem ausgewählten Schweregrad werden zusammen mit dem Element angezeigt. Wenn alle Probleme bestätigt sind, wird ein dicker grüner Rand um den Kreis angezeigt.

Zusätzlich:

- Ein Host in Wartung wird mit einem orangefarbenen, ausgefüllten Quadrat hervorgehoben. Beachten Sie, dass die Wartung-shervorhebung Vorrang vor der Hervorhebung des Problemschweregrads hat, wenn das Kartenelement ein Host ist.
- Ein deaktivierter (nicht überwachter) Host wird mit einem grauen, ausgefüllten Quadrat hervorgehoben.

Die Hervorhebung wird angezeigt, wenn das Kontrollkästchen *Icon highlighting* in der Karten-Konfiguration aktiviert ist.

Markierungen für aktuelle Änderungen

Nach innen zeigende rote Dreiecke um ein Element zeigen eine aktuelle Änderung des Auslöserstatus an – eine, die innerhalb der letzten 30 Minuten stattgefunden hat. Diese Dreiecke werden angezeigt, wenn das Kontrollkästchen *Elemente bei Änderung des Auslöserstatus markieren* in der Konfiguration der Karte 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 die folgenden Optionen:

Edit map	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

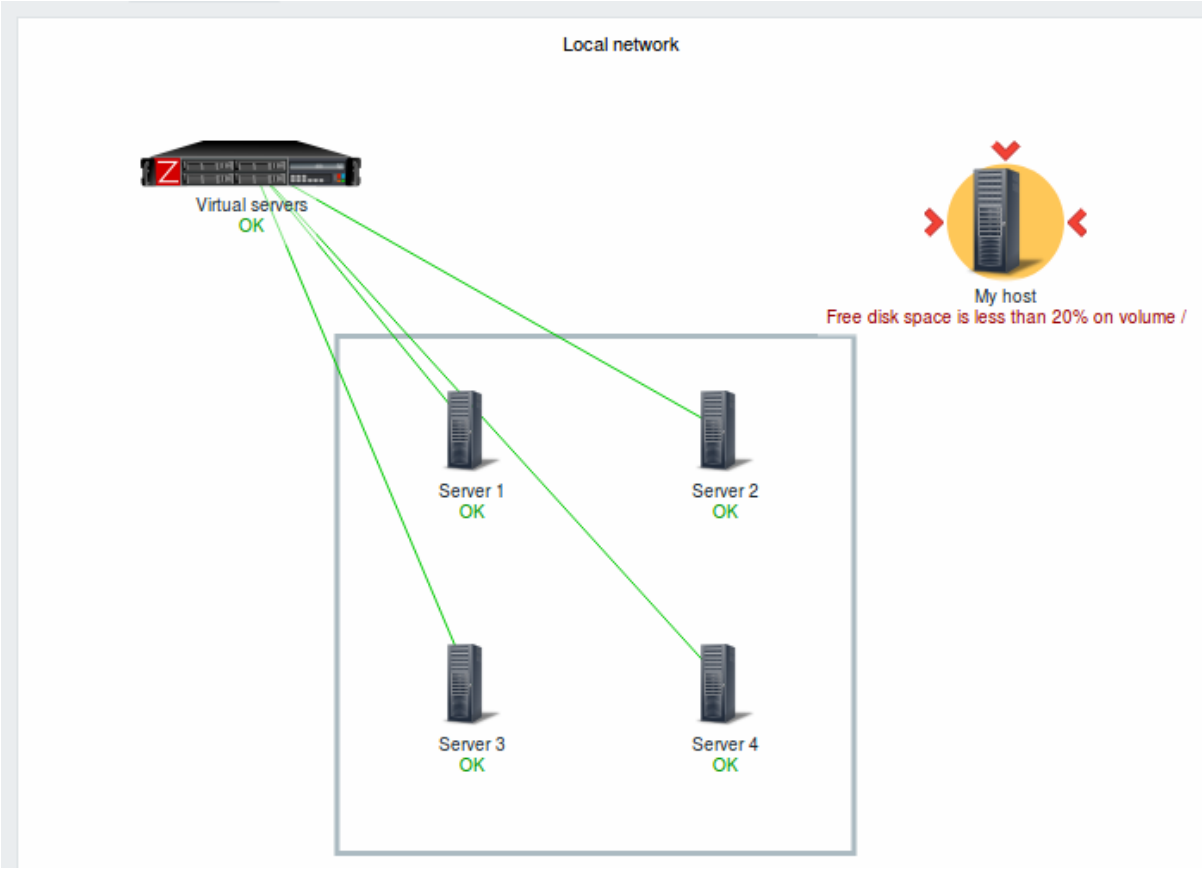
Eine verborgene Eigenschaft „aria-label“ ist verfügbar, die es ermöglicht, Karteninformationen mit einem Screenreader vorzulesen. Sowohl eine allgemeine Kartenbeschreibung als auch die Beschreibung einzelner Elemente sind im folgenden Format verfügbar:

- für die Kartenbeschreibung: `<Kartenname>, <* von * Elementen im Problemstatus>, <* Probleme insgesamt>.`
- zur Beschreibung eines Elements mit einem Problem: `<Elementtyp>, Status <Elementstatus>, <Elementname>, <Problembeschreibung>.`
- zur Beschreibung eines Elements mit mehreren Problemen: `<Elementtyp>, Status <Elementstatus>, <Elementname>, <* Probleme>.`
- zur Beschreibung eines Elements ohne Probleme: `<Elementtyp>, Status <Elementstatus>, <Elementname>.`

Zum Beispiel ist folgende Beschreibung verfügbar:

```
'Lokales Netzwerk, 1 von 6 Elementen im Problemstatus, 1 Problem insgesamt. Host, Status Problem, Mein Hos
```

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
```

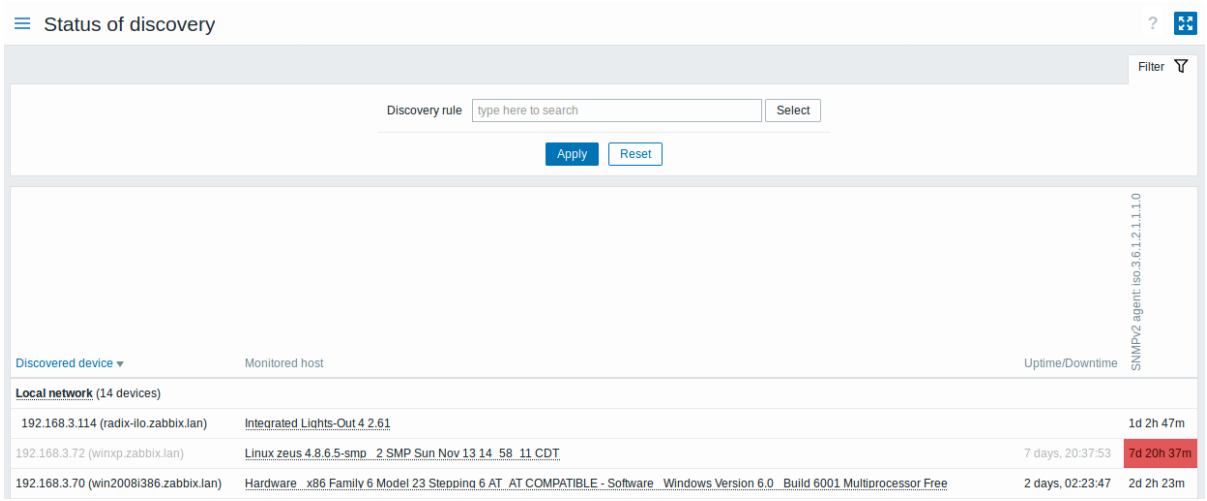
öffnet die Karte mit diesem Namen (Local network).

Wenn sowohl `sysmapid` (Karten-ID) als auch `mapname` (Kartenname) angegeben sind, hat `mapname` eine höhere Priorität.

5 Discovery

Übersicht

Im Abschnitt *Monitoring* → *Discovery* werden die Ergebnisse der Netzwerkerkennung angezeigt. Erkannte Geräte werden nach der Erkennungsregel sortiert.



Angezeigte Daten:

Spalte	Beschreibung
Erkanntes Gerät	Erkannte Geräte werden aufgelistet und nach der Erkennungsregel gruppiert. Durch Klicken auf die Erkennungsregel wird das Regelmenü mit dem Link zum Konfigurationsformular der Erkennungsregel geöffnet.
Überwachter Host	Wenn ein Gerät bereits überwacht wird, wird der Host-Name in dieser Spalte aufgeführt. Durch Klicken auf den Host-Namen wird das Host-Menü geöffnet.
Uptime/Downtime	Die Dauer, seit der ein Gerät erkannt wurde oder nach einer vorherigen Erkennung als verloren gilt, wird in dieser Spalte angezeigt.
Erkennungsprüfung	Der Status des einzelnen Dienstes (Erkennungsprüfung) für jedes erkannte Gerät wird angezeigt. Eine rote Zelle zeigt an, dass der Dienst nicht verfügbar ist. Die Uptime oder Downtime des Dienstes wird innerhalb der Zelle angezeigt. Diese Spalte wird nur angezeigt, wenn der Dienst auf mindestens einem erkannten Gerät gefunden wurde.

Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Abschnitten gemeinsam verwendet werden, sind 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 damit, ihren Namen in den Filter einzugeben. Alle übereinstimmenden aktivierten Discovery-Regeln werden zur Auswahl aufgelistet. Es kann mehr als eine Discovery-Regel ausgewählt werden.

3 Services

Übersicht

Das Menü **Services** ist für die Funktionen der Service-Überwachung von Zabbix vorgesehen.

1 Services

Übersicht

In diesem Abschnitt sehen Sie einen allgemeinen Status aller Services, die in Zabbix auf Grundlage Ihrer Infrastruktur konfiguriert wurden.

Ein Service kann eine Hierarchie sein, die aus mehreren Ebenen anderer Services besteht, den sogenannten „untergeordneten“ Services, die zum Gesamtstatus des Service beitragen (siehe auch die Übersicht zur Funktion Serviceüberwachung.)

Die Hauptkategorien des Servicezustands sind *OK* oder *Problem*, wobei der Status *Problem* durch den entsprechenden Namen und die Farbe des Problemschweregrads ausgedrückt wird.

Während Sie im Ansichtsmodus Services mit ihrem Status und weiteren Details überwachen können, können Sie in diesem Abschnitt durch Wechsel in den Bearbeitungsmodus auch die Servicehierarchie konfigurieren (Services und untergeordnete Services hinzufügen/bearbeiten).

Um vom Ansichtsmodus in den Bearbeitungsmodus zu wechseln (und zurück), klicken Sie auf die entsprechende Schaltfläche in der oberen rechten Ecke:

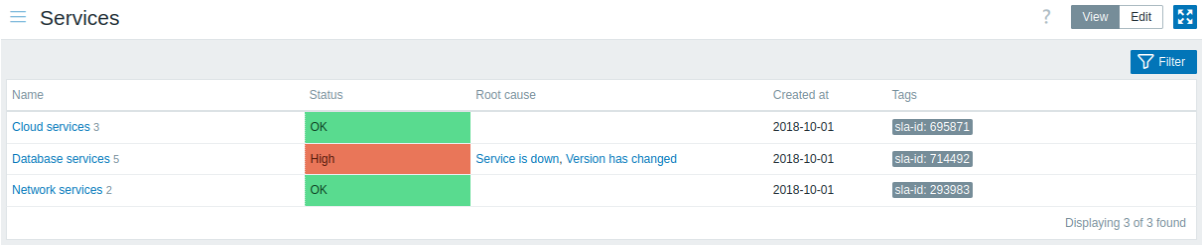
- 

- Services anzeigen
- 

- Services und untergeordnete Services hinzufügen/bearbeiten

Beachten Sie, dass der Zugriff auf die Bearbeitung von den Einstellungen der Benutzerrolle abhängt.

Dienste anzeigen



Eine Liste der vorhandenen Dienste wird angezeigt.

Angezeigte Daten:

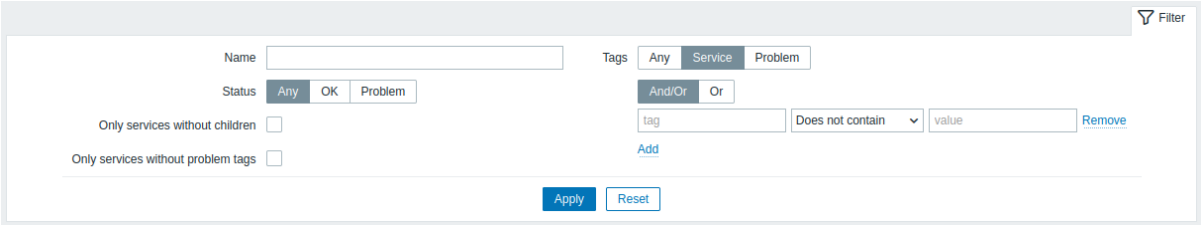
Parameter	Beschreibung
Name	Name des Dienstes. Der Dienstname ist ein Link zu den Dienstdetails. Die Zahl nach dem Namen gibt an, wie viele untergeordnete Dienste der Dienst hat.
Status	Dienststatus: **OK** - keine Probleme **<Problemfarbe und Schweregrad>** - weist auf ein Problem und dessen Schweregrad hin. Bei mehreren Problemen werden die Farbe und der Schweregrad des kritischsten Problems angezeigt.
Root cause	Zugrunde liegende Probleme, die den Dienststatus direkt oder indirekt beeinflussen, werden aufgelistet. Dieselben Probleme werden auch von dem Makro {SERVICE.ROOTCAUSE} macro zurückgegeben. Klicken Sie auf den Problemnamen, um weitere Details dazu unter *Monitoring* → *Problems* anzuzeigen. Probleme, die den Dienststatus nicht beeinflussen, werden nicht in der Liste aufgeführt.
Created at	Die Uhrzeit, 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 Bereichen gemeinsam verwendet werden, werden auf der Seite Monitoring beschrieben.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Services anzuzeigen, die Sie interessieren.

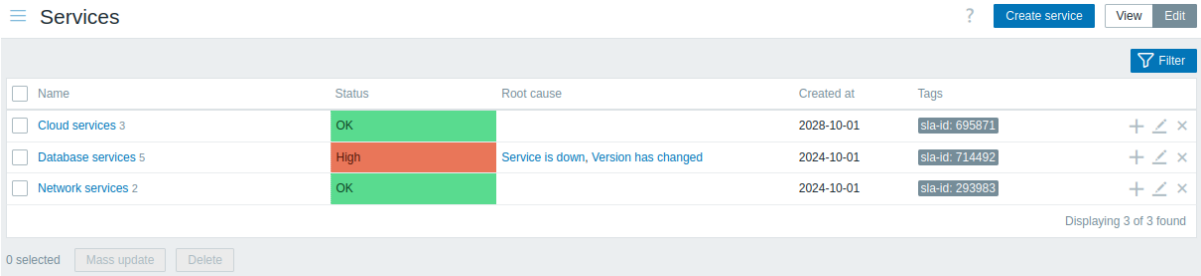


Parameter	Beschreibung
Name	Nach Servicenamen filtern.
Status	Nach Servicestatus filtern.
Only services without children	Aktivieren Sie das Kontrollkästchen, um nur Services ohne untergeordnete Services anzuzeigen. Dieser Parameter ist nur im Service-Bearbeitungsmodus verfügbar.
Only services without 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.

Services bearbeiten

Klicken Sie auf die Schaltfläche *Edit*, um den Bearbeitungsmodus zu öffnen. Im Bearbeitungsmodus wird die Liste durch Kontrollkästchen vor den Einträgen sowie durch diese zusätzlichen Optionen ergänzt:

- - diesem Service einen untergeordneten Service hinzufügen
- - diesen Service bearbeiten
- - diesen Service löschen

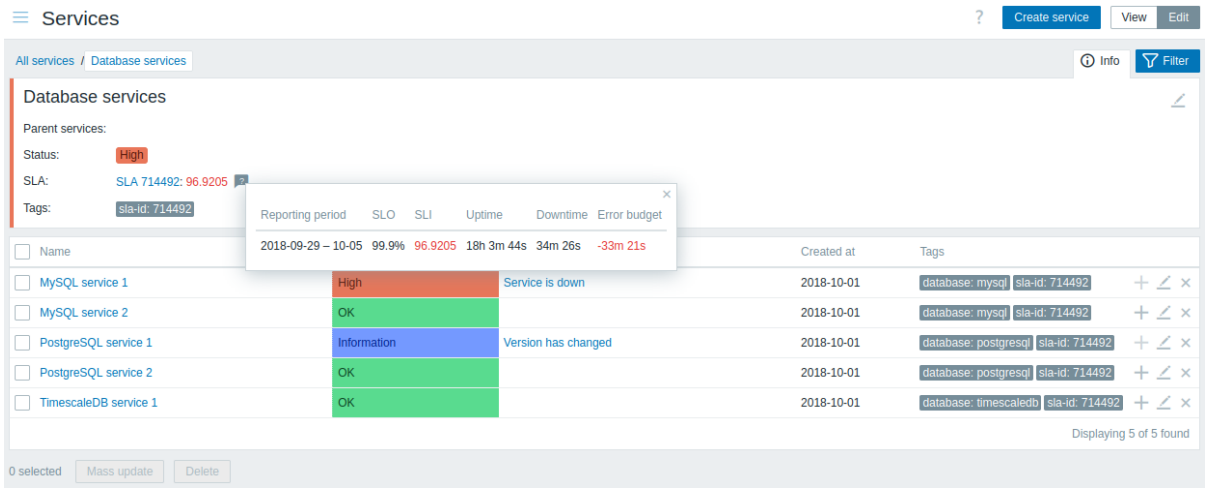


Um einen neuen Service zu konfigurieren, klicken Sie auf die Schaltfläche *Create service* in der oberen rechten Ecke.

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 den Reiter *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-Indikator`. 'SLA-Name' ist außerdem ein Link zum SLA-Bericht für diesen Service. Wenn Sie den Mauszeiger in der Infobox neben dem Service-Level-Indikator (SLI) positionieren, wird eine Pop-up-Informationsliste mit SLI-Details angezeigt. Der Service-Level-Indikator zeigt das aktuelle 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 den Reiter *Filter*.

Im Bearbeitungsmodus wird die Liste der untergeordneten Services um zusätzliche Bearbeitungsoptionen ergänzt:

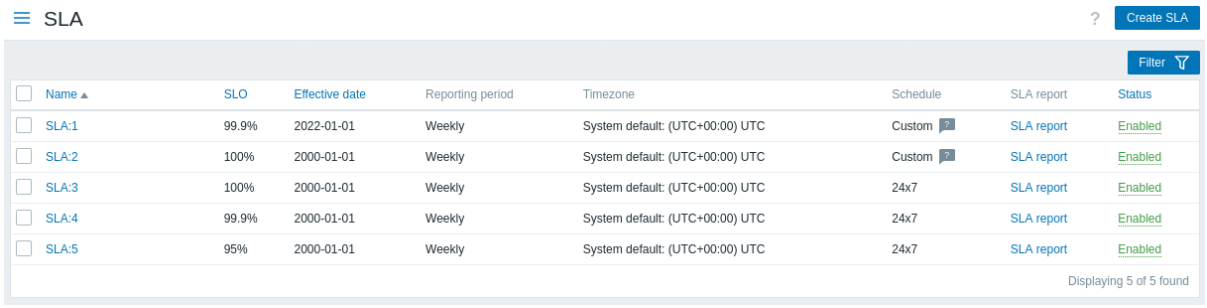
-  - einen untergeordneten Service zu diesem Service hinzufügen
-  - diesen Service bearbeiten
-  - diesen Service löschen

2 SLA

Übersicht

In diesem Abschnitt können SLAs angezeigt und konfiguriert werden.

SLAs



Eine Liste der konfigurierten SLAs wird angezeigt. *Beachten Sie*, dass nur die SLAs angezeigt werden, die sich auf Services beziehen, auf die der Benutzer zugreifen kann (schreibgeschützt, sofern *SLA verwalten* nicht für die Benutzerrolle aktiviert ist).

Angezeigte Daten:

Parameter	Beschreibung
Name	Der SLA-Name wird angezeigt. Der Name ist ein Link zur SLA-Konfiguration.
SLO	Das Service Level Objective (SLO) wird angezeigt.

Parameter	Beschreibung
Gültig ab	Das Startdatum der SLA-Berechnung wird angezeigt.
Berichtszeitraum	Der im SLA-Bericht verwendete Zeitraum wird angezeigt – *täglich*, *wöchentlich*, *monatlich*, *vierteljährlich* oder *jährlich*.
Zeitzone	Die SLA-Zeitzone wird angezeigt.
Zeitplan	Der SLA-Zeitplan wird angezeigt – 24x7 oder benutzerdefiniert.
SLA-Bericht	Klicken Sie auf den Link, um den SLA-Bericht für dieses SLA anzuzeigen.
Status	Der SLA-Status wird angezeigt – aktiviert oder deaktiviert.

3 SLA-Bericht

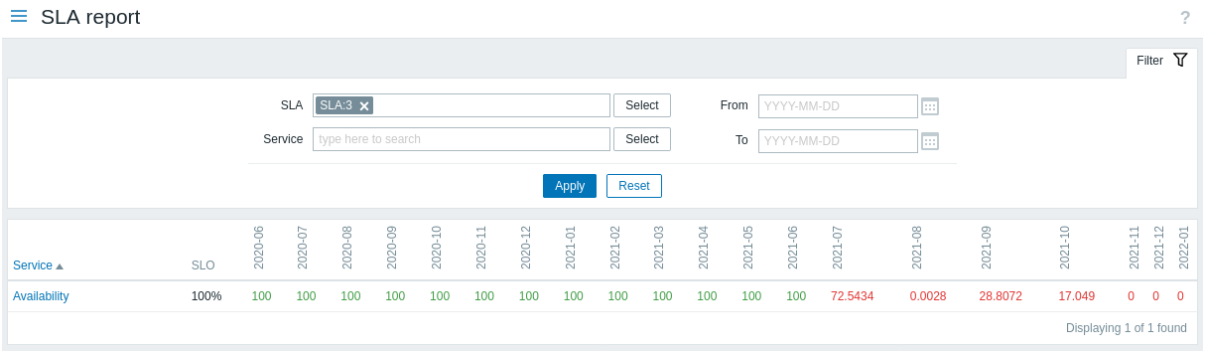
Übersicht

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 kann der Bericht anhand des SLA-Namens sowie des Servicenamens ausgewählt werden. Es ist auch möglich, den angezeigten Zeitraum zu begrenzen.

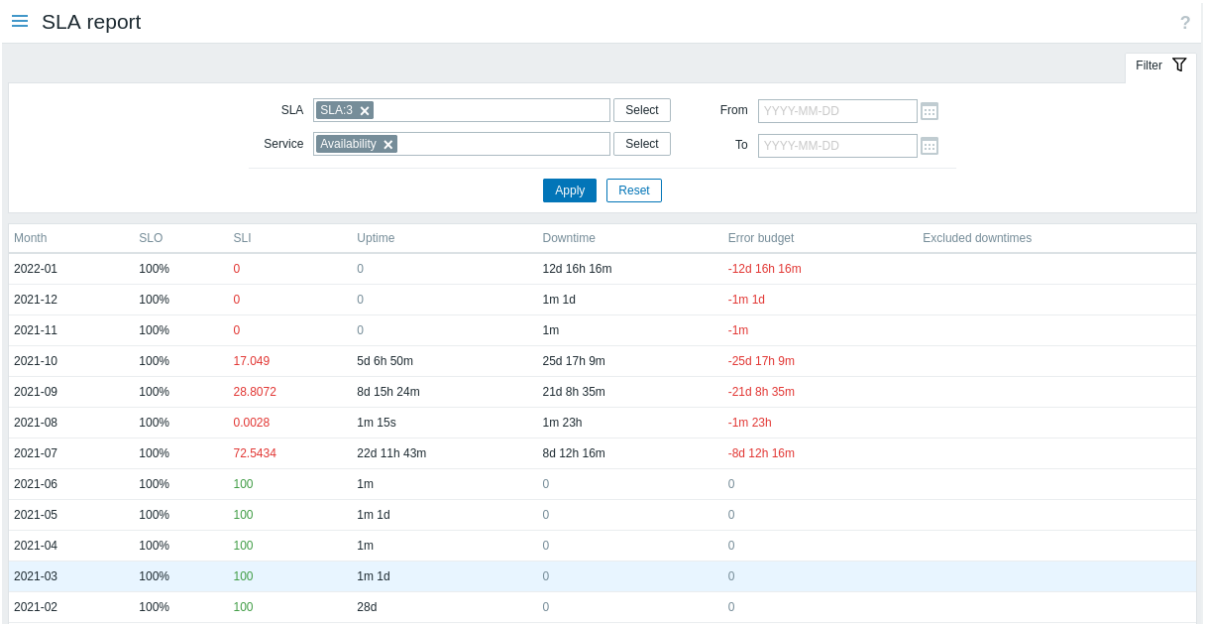


Jede Spalte (Zeitraum) zeigt den SLI für diesen Zeitraum an. SLIs, die den festgelegten SLO verletzen, werden rot hervorgehoben.

Im Bericht werden 20 Zeiträume angezeigt. Maximal 100 Zeiträume können angezeigt werden, wenn sowohl das *Von*-Datum als auch das *Bis*-Datum angegeben sind.

Berichtdetails

Wenn Sie im Bericht auf den Servicenamen klicken, können Sie auf einen weiteren Bericht zugreifen, der eine detailliertere Ansicht anzeigt.



Beachten Sie, dass sich die negative Problemdauer nicht auf die SLA-Berechnung oder Berichterstattung auswirkt.

4 Inventar

Übersicht

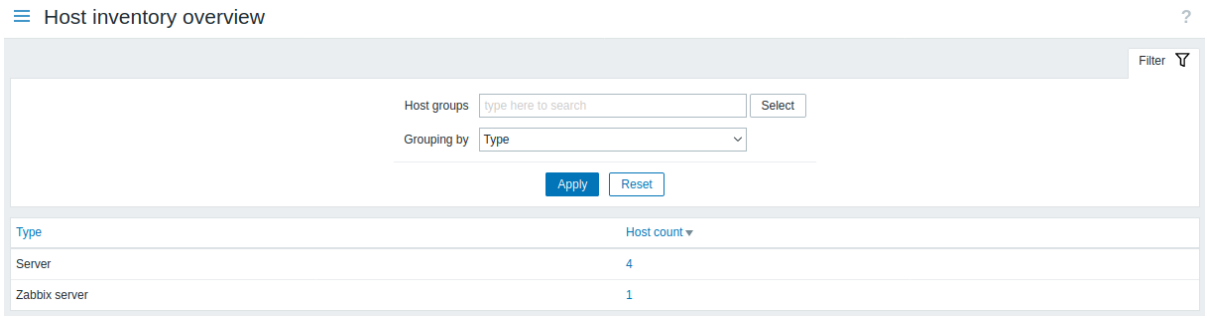
Das Menü „Inventar“ enthält Abschnitte, die eine Übersicht über die Inventardaten von Hosts nach einem ausgewählten Parameter bieten, sowie die Möglichkeit, Details zum Host-Inventar anzuzeigen.

1 Übersicht

Übersicht

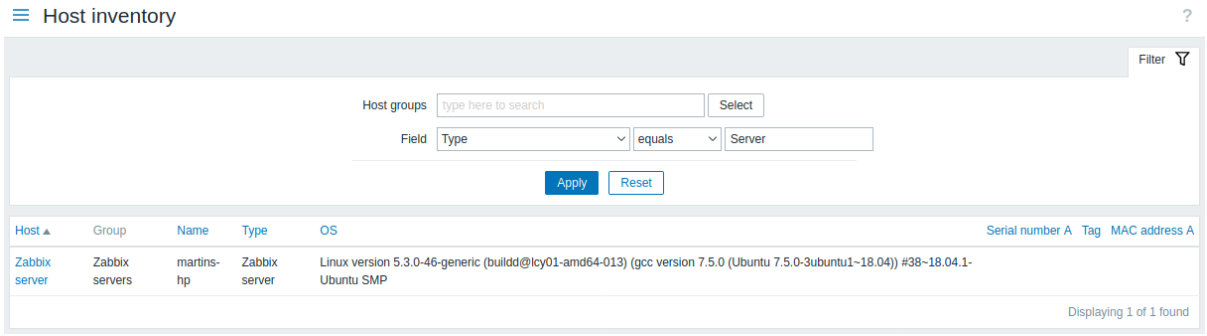
Der Abschnitt *Inventar → Übersicht* bietet Möglichkeiten, sich einen Überblick über die Daten des Host-Inventars zu verschaffen.

Damit eine Übersicht angezeigt wird, wählen Sie Host-Gruppen (oder keine) sowie das Inventarfeld aus, nach dem die Daten angezeigt werden sollen. Die Anzahl der Hosts, die jedem Eintrag des ausgewählten Feldes entsprechen, wird angezeigt.



Wie vollständig eine Übersicht ist, hängt davon ab, wie viele Inventarinformationen bei den Hosts gepflegt werden.

Die Zahlen in der Spalte *Host-Anzahl* sind Links; sie führen dazu, dass diese Hosts in der Tabelle *Host-Inventare* herausgefiltert werden.

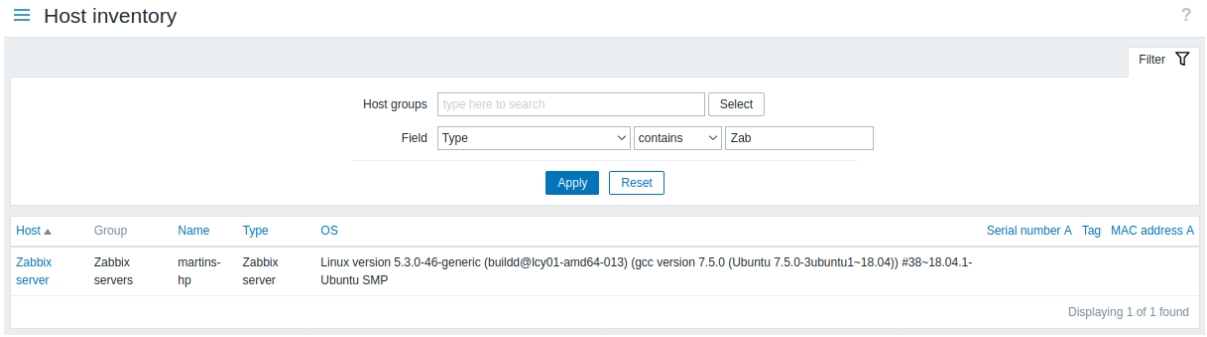


2 Hosts

Übersicht

Im Abschnitt *Inventar → Hosts* werden Inventardaten von Hosts angezeigt.

Sie können die Hosts nach Hostgruppe(n) und nach jedem beliebigen Inventarfeld filtern, um nur die Hosts anzuzeigen, die Sie interessieren.

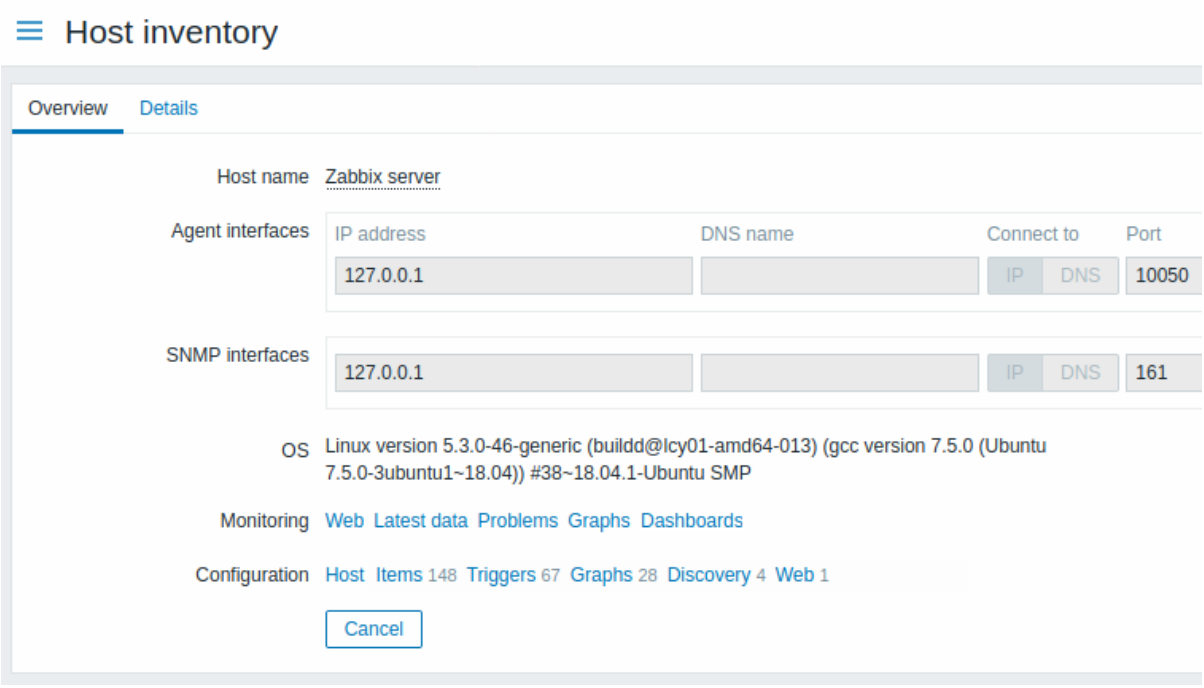


Um alle Host-Inventare anzuzeigen, klicken Sie auf die Schaltfläche „Zurücksetzen“.

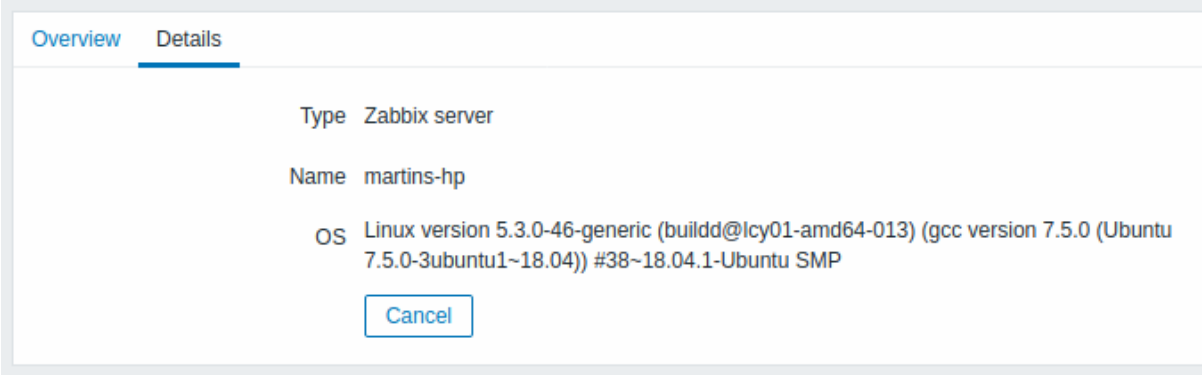
Während in der Tabelle nur einige wichtige Inventarfelder angezeigt werden, können Sie auch alle verfügbaren Inventarinformationen für diesen Host anzeigen. Klicken Sie dazu in der ersten Spalte auf den Hostnamen.

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.

5 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.

1 Systeminformationen

Übersicht

Unter *Berichte* → *Systeminformationen* wird eine Zusammenfassung der wichtigsten Zabbix-Server- und Systemdaten angezeigt. Systemdaten werden mithilfe von internen Datenpunkten erfasst.

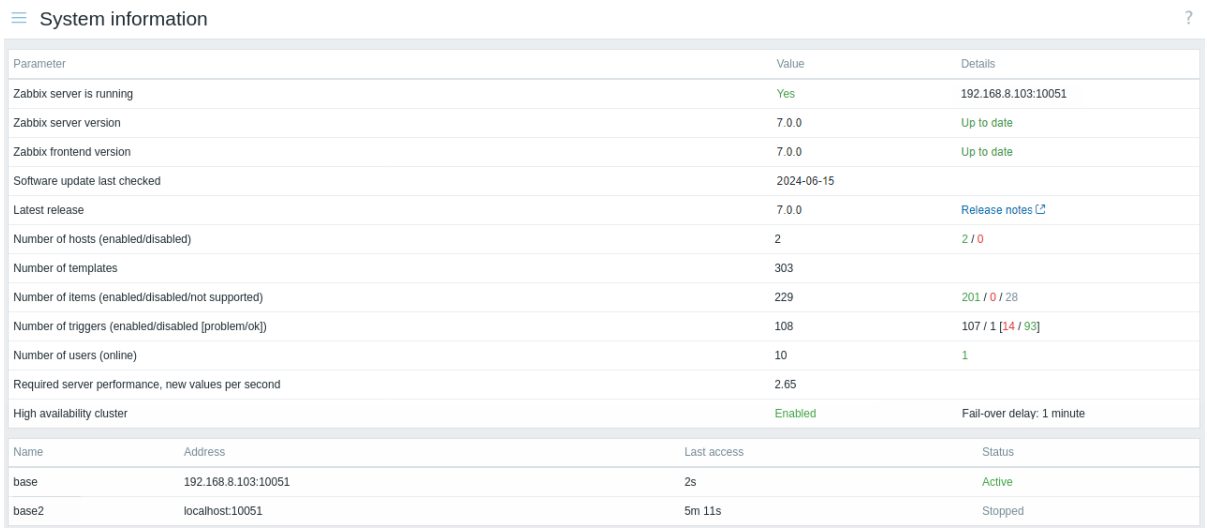
Beachten Sie, dass es in einer Hochverfügbarkeitskonfiguration möglich ist, die Quelle der Systeminformationen (Server-Instanz) umzuleiten. Bearbeiten Sie dazu die Datei *zabbix.conf.php* – entfernen Sie die Auskommentierung von `$ZBX_SERVER` oder von sowohl `$ZBX_SERVER` als auch `$ZBX_SERVER_PORT` und setzen Sie diese auf einen anderen Server als den aktuell als aktiv

angezeigten. Beachten Sie, dass bei der Einstellung nur von $ZBX_SERVER ein Standardwert (10051) für $ZBX_SERVER_PORT verwendet wird.

Wenn die Hochverfügbarkeitskonfiguration aktiviert ist, wird unterhalb der Systemstatistiken ein separater Block mit Details zu den Hochverfügbarkeitsknoten angezeigt. Dieser Block ist nur für Zabbix-Benutzer des Typs *Super Admin* sichtbar.

Systeminformationen sind auch als Dashboard-Widget verfügbar.

Systemstatistiken



Angezeigte Daten:

Parameter	Wert	Details
Zabbix-Server läuft	Status des Zabbix-Servers: **Ja** - Server läuft **Nein** - Server läuft nicht *Hinweis:* Um die restlichen Informationen anzuzeigen, benötigt das Frontend einen laufenden Server, und auf dem Server muss mindestens ein Trapper-Prozess gestartet sein (Parameter StartTrappers in der Datei zabbix_server.conf > 0).	Standort und Port des Zabbix-Servers.
Zabbix-Server-Version	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 neuere Version ist verfügbar; **Veraltet** - der vollständige Supportzeitraum für diese Version ist abgelaufen. Diese Information ist nur verfügbar, wenn die Prüfung auf Software-Updates in der Zabbix-Server-Konfiguration aktiviert ist. Es wird nichts angezeigt, wenn die letzte Prüfung auf Software-Updates vor mehr als einer Woche durchgeführt wurde oder keine Daten zur aktuellen Version vorhanden sind.

Parameter	Wert	Details
Zabbix-Frontend-Version	Die Versionsnummer des Zabbix-Frontends wird angezeigt.	Der Status der Frontend-Version wird angezeigt: **Aktuell** - die neueste Version wird verwendet; **Neues Update verfügbar** - eine neuere Version ist verfügbar; **Veraltet** - der vollständige Supportzeitraum für diese Version ist abgelaufen. Diese Information ist nur verfügbar, wenn die Prüfung auf Software-Updates in der Zabbix-Server-Konfiguration aktiviert ist. Es wird nichts angezeigt, wenn die letzte Prüfung auf Software-Updates vor mehr als einer Woche durchgeführt wurde oder keine Daten zur aktuellen Version vorhanden sind.
Letzte Prüfung auf Software-Updates	Das Datum der letzten Prüfung auf Zabbix-Software-Updates wird angezeigt. Diese Information ist nur verfügbar, wenn die Prüfung auf Software-Updates in der Zabbix-Server-Konfiguration aktiviert ist.	
Neueste Version	Die Nummer einer neueren Version (falls verfügbar) für die aktuelle Zabbix-Version wird angezeigt. Diese Information ist nur verfügbar, wenn die Prüfung auf Software-Updates in der Zabbix-Server-Konfiguration aktiviert ist. Es wird nichts angezeigt, wenn die letzte Prüfung auf Software-Updates vor mehr als einer Woche durchgeführt wurde oder keine Daten zur aktuellen Version vorhanden sind.	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 auch Auslöser mit dem Status „Unbekannt“. Auslöser, die von deaktivierten Datenpunkten abhängen oder deaktivierten Hosts zugewiesen sind, werden als deaktiviert gezählt.
Anzahl der Benutzer	Die Gesamtzahl der konfigurierten Benutzer wird angezeigt.	Anzahl der online befindlichen Benutzer.

Parameter	Wert	Details
Erforderliche Serverleistung, neue Werte pro Sekunde	Die erwartete Anzahl neuer Werte, die pro Sekunde vom Zabbix-Server verarbeitet werden, wird angezeigt.	*Erforderliche Serverleistung* ist ein Schätzwert und kann als Richtwert nützlich sein. Für genaue Zahlen der verarbeiteten Werte verwenden Sie den internen Datenpunkt `zabbix[wcache,values,all]`. Aktivierte Datenpunkte von überwachten Hosts werden in die Berechnung einbezogen. Log-Datenpunkte werden als ein Wert pro Aktualisierungsintervall des Datenpunkts gezählt. Werte aus regulären Intervallen werden gezählt; Werte aus flexiblen Intervallen und Planungsintervallen nicht. Die Berechnung wird während eines „nodata“-Wartungszeitraums 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 in der Serverkonfiguration EnableGlobalScripts=0 gesetzt wird.	
Hochverfügbarkeitscluster	Status des Hochverfügbarkeitsclusters für den Zabbix-Server: **Deaktiviert** - eigenständiger Server **Aktiviert** - mindestens ein Hochverfügbarkeitsknoten existiert	Falls aktiviert, wird die Failover-Verzögerung angezeigt.

Systeminformationen zeigen außerdem unter den folgenden Bedingungen eine Fehlermeldung an:

- Die verwendete Datenbank hat nicht den erforderlichen Zeichensatz oder die erforderliche Sortierung (UTF-8).
- Die Version der Datenbank liegt unterhalb oder oberhalb des unterstützten Bereichs (nur verfügbar für Benutzer mit dem Rollentyp *Super admin role*).
- Housekeeping für TimescaleDB ist falsch konfiguriert (History- oder Trend-Tabellen enthalten komprimierte Chunks, aber die Optionen *Override item history period* oder *Override item trend period* sind deaktiviert).

Hochverfügbarkeitsknoten

Wenn der Hochverfügbarkeitscluster aktiviert ist, wird ein weiterer Datenblock mit dem Status jedes Hochverfügbarkeitsknotens angezeigt.

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:

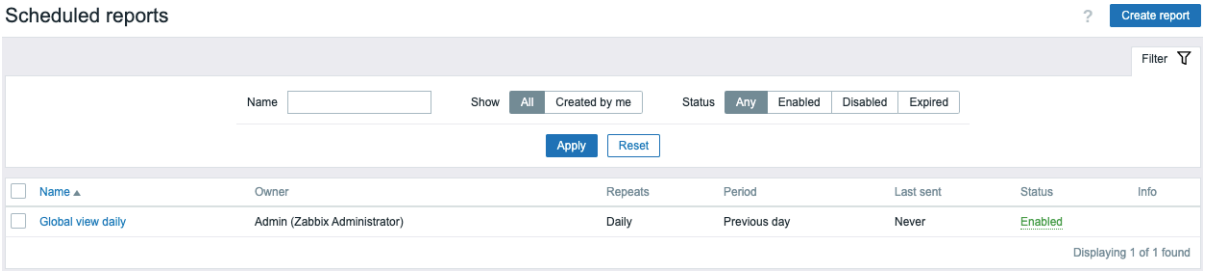
Spalte	Beschreibung
Name	Knotenname, wie in der Server-Konfiguration definiert.
Adresse	IP-Adresse und Port des Knotens.
Letzter Zugriff	Zeitpunkt des letzten Zugriffs auf den Knoten. Wenn Sie den Mauszeiger über die Zelle bewegen, wird der Zeitstempel des letzten Zugriffs im langen Format angezeigt.

Spalte	Beschreibung
Status	Knotenstatus. Die Tabellenzeilen werden nach diesen Statuswerten in folgender Prioritätsreihenfolge sortiert: **Aktiv** - Knoten ist in Betrieb und funktioniert **Nicht verfügbar** - Knoten wurde länger als die Failover-Verzögerung nicht gesehen (Sie sollten die Ursache prüfen) **Gestoppt** - Knoten wurde gestoppt oder konnte nicht gestartet werden (Sie sollten ihn starten oder löschen) **Standby** - Knoten ist in Betrieb und wartet

2 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, die zur einfacheren Navigation gefiltert werden können – siehe den Abschnitt Filter verwenden unten.

Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Berichts. Ein Klick darauf öffnet das Konfigurationsformular des Berichts.
Eigentümer	Benutzer, der den Bericht erstellt hat.
Wiederholung	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 Anklicken ändern – von „Aktiviert“ zu „Deaktiviert“ (und zurück); von „Abgelaufen“ zu „Deaktiviert“ (und zurück). Für Benutzer mit unzureichenden Rechten ist der Status nicht anklickbar.
Info	Zeigt Informationssymbole an: Ein rotes Symbol zeigt an, dass die Berichtserstellung fehlgeschlagen ist; beim Darüberfahren wird ein Tooltip mit Fehlerinformationen angezeigt. Ein gelbes Symbol zeigt an, dass ein Bericht erstellt wurde, das Senden an einige (oder alle) Empfänger jedoch fehlgeschlagen ist oder dass ein Bericht abgelaufen ist; beim Darüberfahren wird ein Tooltip 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:

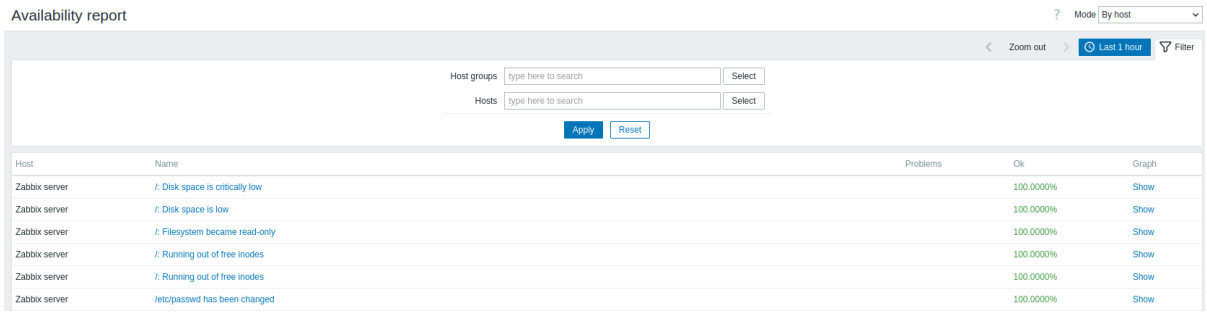
- Aktivieren Sie in der Liste die Kontrollkästchen der zu aktualisierenden Berichte
- Klicken Sie unter der Liste auf die gewünschte Schaltfläche, um die Änderungen vorzunehmen (*Aktivieren*, *Deaktivieren* oder *Löschen*)

3 Verfügbarkeitsbericht

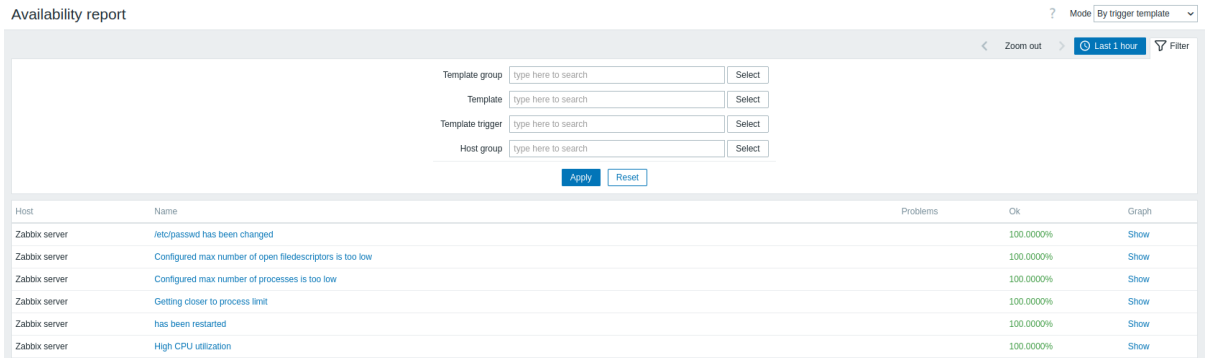
Übersicht

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 den Reiter *Filter* auf der rechten Seite 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.

Parameter	Beschreibung
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

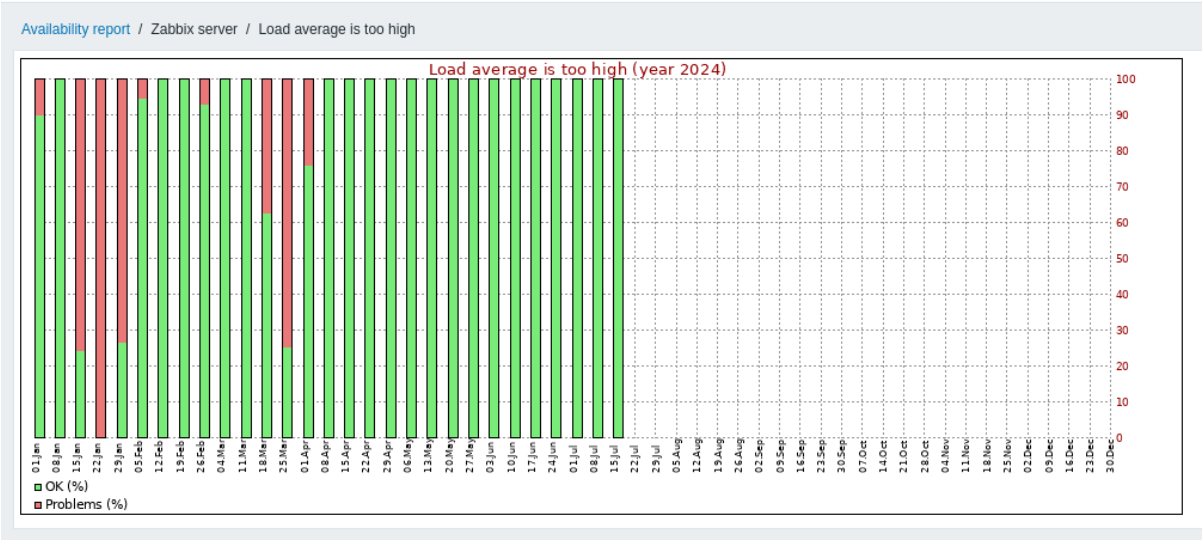
Im Modus *Nach Host* können Ergebnisse nach Host oder Hostgruppe gefiltert werden. Die Angabe einer übergeordneten Hostgruppe wählt implizit alle verschachtelten Hostgruppen aus.

Auswahl des Zeitraums

Der Zeitraumauswahl ermöglicht es, häufig verwendete Zeiträume mit einem Klick auszuwählen. Die Auswahl kann durch Klicken auf die Registerkarte *Zeitraum* neben dem Filter ein- und ausgeblendet werden.

Durch Klicken auf *Anzeigen* in der Spalte *Graph* werden Verfügbarkeitsinformationen in einem Balkendiagramm angezeigt, wobei jeder Balken eine vergangene Woche des aktuellen Jahres darstellt.

Availability report graph



Das Grün eines Balkens steht für die OK-Zeit und Rot für die Problemzeit.

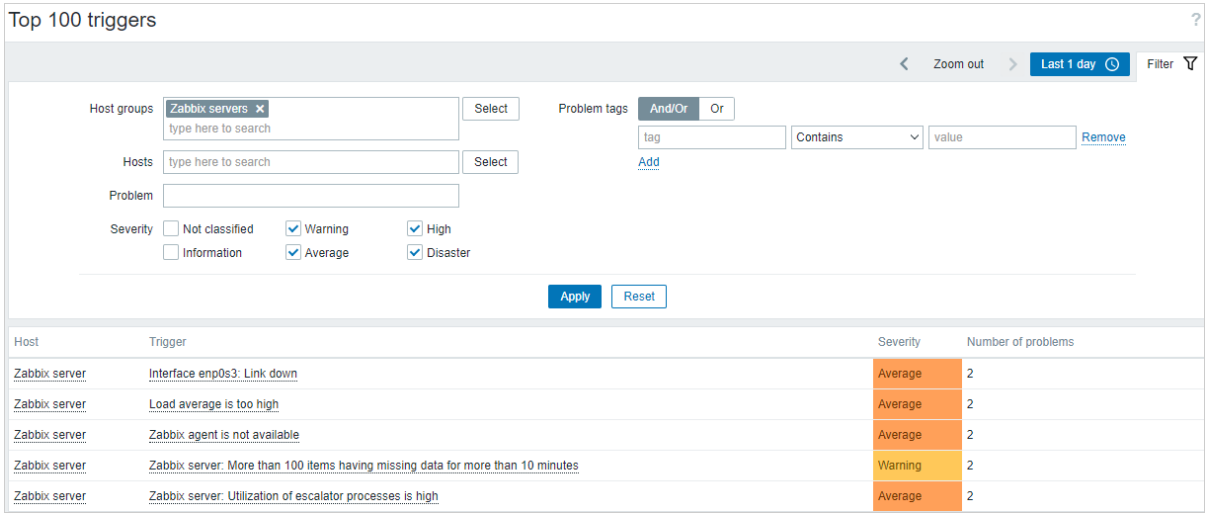
Auswirkung von Wartungszeiträumen

Wartung schließt Zeit nicht automatisch aus dem Verfügbarkeitsbericht aus. Nur eine als *Keine Datenerfassung* konfigurierte Wartung stoppt das Polling (sodass keine Probleme erzeugt werden) und lässt daher den Verfügbarkeitsprozentsatz für betroffene Auslöser unverändert.

4 Top 100 Auslöser

Übersicht

Unter *Berichte* → *Top 100 Auslöser* können Sie die Auslöser mit der höchsten Anzahl erkannter Probleme im ausgewählten Zeitraum sehen.



Sowohl die Einträge in den Spalten Host als auch Auslöser sind Links, die einige nützliche Optionen bieten:

- für Host - ein Klick auf den Hostnamen öffnet das Host-Menü
- für Auslöser - ein Klick auf den Auslösernamen öffnet Links zu den neuesten Ereignissen, zu einem einfachen Diagramm für jeden Datenpunkt des Auslösers sowie zu den Konfigurationsformularen des Auslösers selbst und jedes Datenpunkts des Auslösers

Verwendung des Filters

Sie können den Filter verwenden, um Auslöser nach Hostgruppe, Host, Problemname, Tags oder Auslöser-Schweregrad anzuzeigen. Wenn Sie eine übergeordnete Hostgruppe angeben, werden implizit auch 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.

Zeitraumauswahl

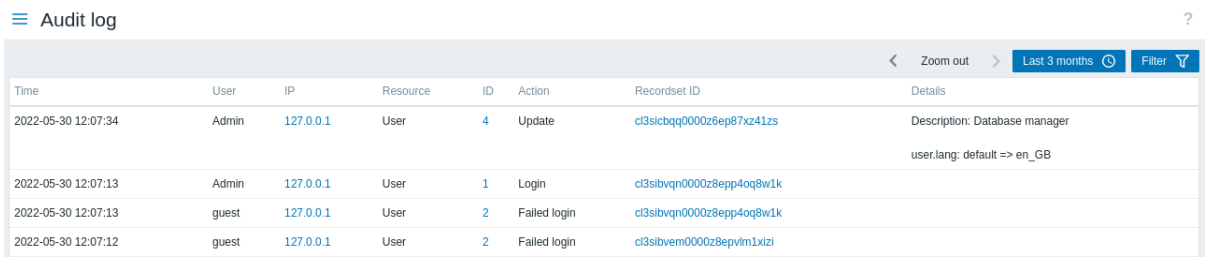
Die Zeitraumauswahl neben dem Filter ermöglicht es, häufig benötigte Zeiträume mit einem Mausklick auszuwählen. Weitere Informationen finden Sie unter Zeitraum- und Host-Auswahl.

5 Audit-Protokoll

Übersicht

Im Abschnitt *Berichte* → *Audit-Log* können die Aufzeichnungen von Benutzer- und Systemaktivitäten eingesehen werden.

Note:
Damit Audit-Einträge erfasst und angezeigt werden, muss das Kontrollkästchen *Audit-Protokollierung aktivieren* im Abschnitt *Administration* → *Audit-Log* aktiviert sein. Ohne diese aktivierte Einstellung wird der Aktivitätsverlauf nicht in der Datenbank gespeichert und nicht im Audit-Log angezeigt.



Das Audit-Log zeigt die folgenden Daten an:

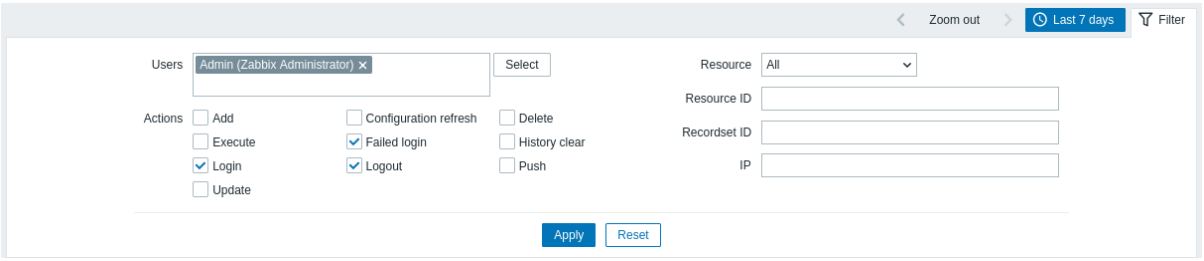
Spalte	Beschreibung
Zeit	Zeitstempel des Audit-Eintrags.
Benutzer	Benutzer, der die Aktivität ausgeführt hat.
IP	IP, von der aus die Aktivität initiiert wurde. Ein Klick auf den Hyperlink filtert die Audit-Log-Einträge nach dieser IP.
Ressource	Typ der betroffenen Ressource (*API-Token*, *Aktion*, *Authentifizierung*, *Autoregistrierung* usw.).

Spalte	Beschreibung
ID	ID der betroffenen Ressource. Ein Klick auf den Hyperlink filtert die Audit-Log-Einträge nach dieser Ressourcen-ID.
Aktion	Typ der Aktivität (*Hinzufügen*, *Konfigurationsaktualisierung*, *Löschen*, *Ausführen*, *Fehlgeschlagene Anmeldung*, *Verlauf löschen*, *Anmeldung*, *Abmeldung*, *Push*, *Aktualisieren*).
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 Vorlagenentität (Datenpunkt, Auslöser usw.) ein separater Audit-Log-Eintrag erstellt – alle diese Einträge haben dieselbe *Recordset-ID*. Ein Klick auf den Hyperlink filtert die Audit-Log-Einträge nach dieser *Recordset-ID*.
Details	Beschreibung der Ressource und detaillierte Informationen über die ausgeführte Aktivität. Wenn ein Eintrag mehr als zwei Zeilen enthält, wird ein zusätzlicher Link *Details* angezeigt. Klicken Sie auf diesen Link, um die vollständige Liste der Änderungen anzuzeigen.

Note:
Wenn ein Trapper-Datenpunkt oder ein HTTP-Agent-Datenpunkt (mit aktiviertem Trapping) Daten empfangen hat, wird nur dann ein Eintrag im Audit-Log hinzugefügt, wenn die Daten mit der API-Methode `history.push` gesendet wurden und nicht mit dem Dienstprogramm Zabbix sender.

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.

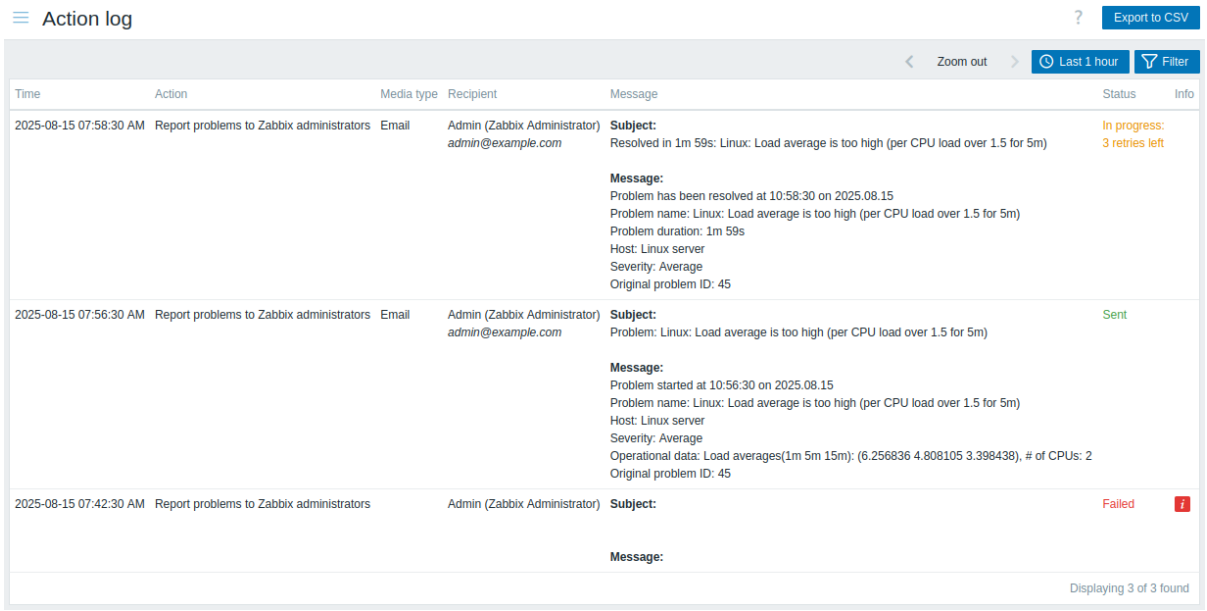
Auswahl des Zeitraums

Die Auswahl des Zeitraums neben dem Filter ermöglicht es, häufig benötigte Zeiträume mit einem Mausklick auszuwählen. Weitere Informationen finden Sie unter Auswahl des Zeitraums und des Hosts.

6 Aktionsprotokoll

Übersicht

Unter *Berichte* → *Aktionsprotokoll* 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 wird. Bei Operationen, die Remote-Befehle ausgeführt haben, bleibt diese Spalte leer.
Recipient	Informationen über den Empfänger der Benachrichtigung – Benutzername, Vorname und Nachname (in Klammern) sowie zusätzliche Informationen abhängig vom 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 der Name 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 folgendes Format: `Zabbix server:<command>`.
Status	Status der Operation: *In Bearbeitung* - Operation zum Senden einer Benachrichtigung wird ausgeführt (die verbleibende Anzahl der Versuche des Servers zum Senden der Benachrichtigung wird ebenfalls angezeigt) *Gesendet* - Benachrichtigung wurde gesendet *Ausgeführt* - Remote-Befehl wurde ausgeführt *Fehlgeschlagen* - Operation wurde nicht abgeschlossen
Info	Fehlerinformationen (falls vorhanden) zur Ausführung der Operation.

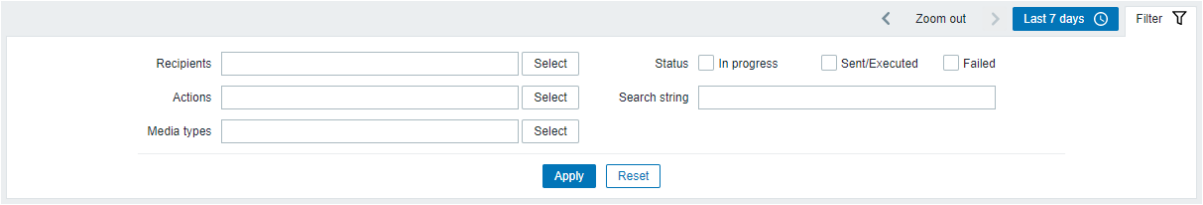
Schaltflächen

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“.

Filter verwenden

Der Filter befindet sich unterhalb der Leiste *Aktionsprotokoll*. 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 (*Suchzeichenfolge*) einzugrenzen. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Auswahl des Zeitraums

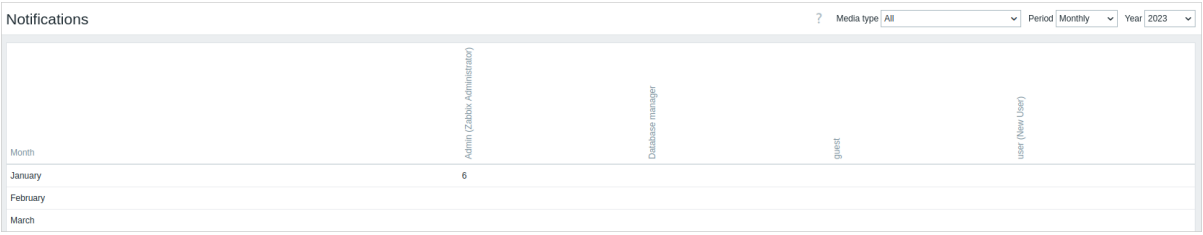
Die Auswahl des Zeitraums neben dem Filter ermöglicht es, häufig benötigte Zeiträume mit einem Mausklick auszuwählen. Weitere Informationen finden Sie unter Zeit- und Host-Auswahl.

7 Benachrichtigungen

Übersicht

Im Abschnitt *Berichte* → *Benachrichtigungen* 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 Gesamtzahl pro Systembenutzer an.

6 Datenerfassung

Übersicht

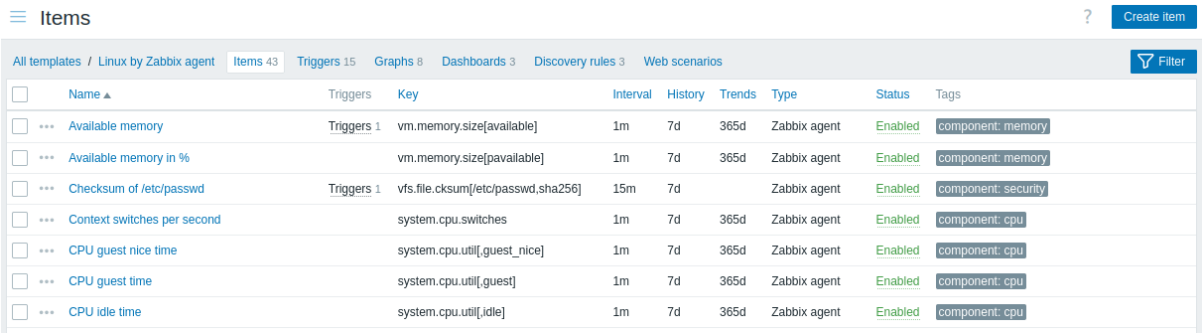
Dieses Menü enthält Abschnitte, die mit der Konfiguration der Datenerfassung zusammenhängen.

1 Datenpunkte

Übersicht

Die Datenpunktliste für eine Vorlage kann unter *Datensammlung* → *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Datenpunkte* klicken.

Eine Liste der vorhandenen Datenpunkte wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Datenpunkt-Menü	Klicken Sie auf das Symbol mit den drei Punkten, um das Datenpunkt-Menü zu öffnen.
Vorlage	Vorlage, zu der der Datenpunkt gehört. Durch Klicken auf den Vorlagennamen wird das Konfigurationsformular der Vorlage geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Vorlagen oder keine Vorlagen ausgewählt sind.
Name	Name des Datenpunkts, angezeigt als blauer Link zu den Details des Datenpunkts. Durch Klicken auf den Link mit dem Datenpunktnamen wird das Konfigurationsformular des Datenpunkts geöffnet. Wenn der Datenpunkt von einer anderen Vorlage geerbt wurde, wird der Vorlagenname vor dem Datenpunktnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagen-Link wird die Datenpunktliste auf dieser Vorlagenebene geöffnet.
Auslöser	Wenn Sie den Mauszeiger über Auslöser bewegen, wird eine Infobox mit den dem Datenpunkt zugeordneten Auslösern angezeigt. Die Anzahl der Auslöser wird grau dargestellt.
Schlüssel	Der Schlüssel des Datenpunkts wird angezeigt.
Intervall	Die Häufigkeit der Prüfung wird angezeigt.
Verlauf	Es wird angezeigt, wie viele Tage der Datenverlauf des Datenpunkts aufbewahrt wird.
Trends	Es wird angezeigt, wie viele Tage die Trendhistorie des Datenpunkts aufbewahrt wird.
Typ	Der Typ des Datenpunkts 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 zurück).
Tags	Die Tags des Datenpunkts werden angezeigt. Es können bis zu drei Tags (Name:Wert-Paare) angezeigt werden. Wenn es mehr Tags gibt, wird ein „...“-Link angezeigt, über den beim Überfahren mit der Maus alle Tags angezeigt werden können.

Um einen neuen Datenpunkt zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Datenpunkt erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

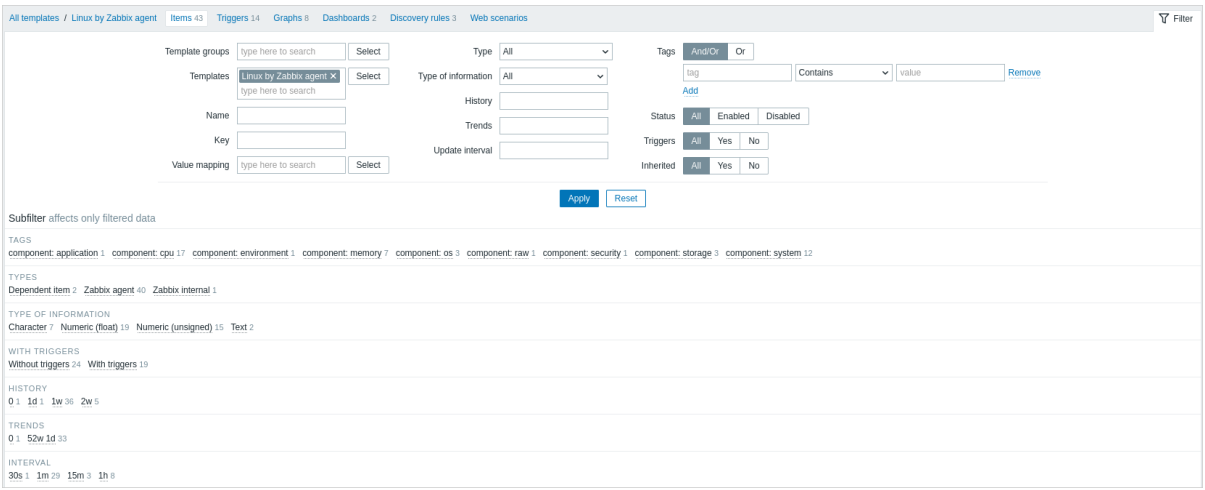
- *Aktivieren* - den Status des Datenpunkts auf *Aktiviert* ändern.
- *Deaktivieren* - den Status des Datenpunkts auf *Deaktiviert* ändern.
- *Kopieren* - die Datenpunkte auf andere Hosts oder Vorlagen kopieren.
- *Massenaktualisierung* - mehrere Eigenschaften für mehrere Datenpunkte gleichzeitig aktualisieren.
- *Löschen* - 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

Die Datenpunktliste kann sehr viele Datenpunkte enthalten. Mithilfe des Filters können Sie einige davon herausfiltern, um die gesuchten Datenpunkte schnell zu finden. 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, öffnet sich ein Filter, in dem Sie die gewünschten Filterkriterien angeben können.

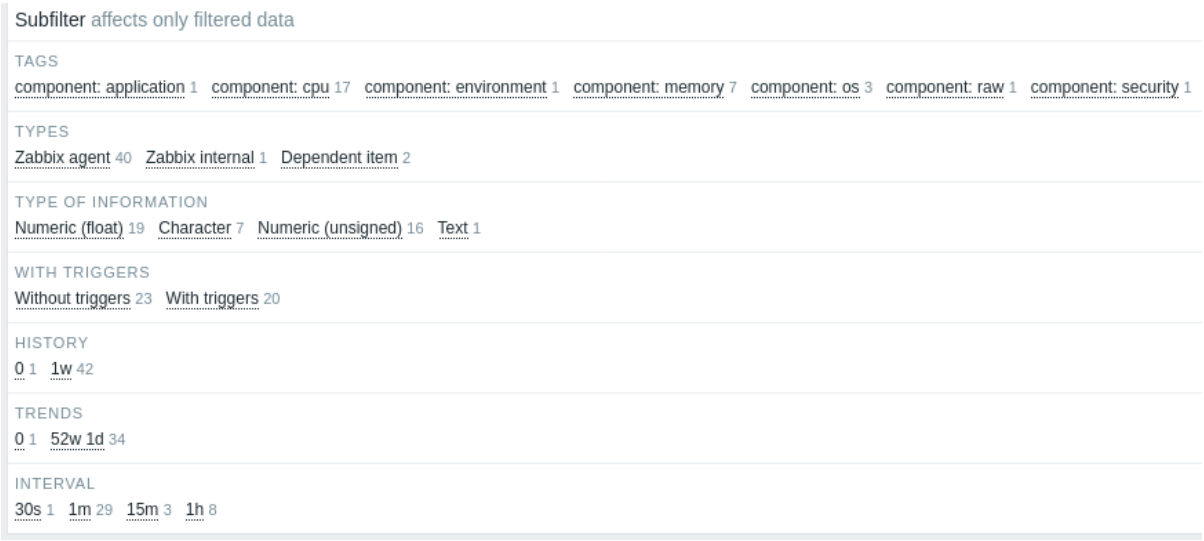


Parameter	Beschreibung
Vorlagengruppen	Nach einer oder mehreren Vorlagengruppen filtern. Wenn eine übergeordnete Vorlagengruppe angegeben wird, werden implizit auch alle verschachtelten Gruppen ausgewählt.
Vorlagen	Nach einer oder mehreren Vorlagen filtern.
Name	Nach dem Namen des Datenpunkts filtern.
Schlüssel	Nach dem Schlüssel des Datenpunkts filtern.
Wertzuordnung	Nach der verwendeten Wertezuordnung filtern. Dieser Parameter wird nicht angezeigt, wenn die Option *Vorlagen* leer ist.
Typ	Nach dem Typ des Datenpunkts filtern (Zabbix Agent, SNMP-Agent usw.).
Informationstyp	Nach dem Informationstyp filtern (Numerisch ohne Vorzeichen, Gleitkommazahl usw.).
Verlauf	Danach filtern, wie lange der Datenpunktverlauf aufbewahrt wird.
Trends	Danach filtern, wie lange die Datenpunkt-Trends aufbewahrt werden.
Aktualisierungsintervall	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 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
Status	Nach dem Status des Datenpunkts filtern - *Aktiviert* oder *Deaktiviert*.
Auslöser	Datenpunkte mit (oder ohne) Auslöser filtern.
Geerbt	Datenpunkte filtern, die aus verknüpften Vorlagen geerbt wurden (oder nicht geerbt wurden).

Verwendung des Subfilters

Der Subfilter ermöglicht es, die Filterung aus dem Hauptfilter weiter anzupassen.

Er enthält anklickbare Links für den schnellen Zugriff auf zugehörige Datenpunkte. Datenpunkte sind über gemeinsame Entitäten verknüpft – Tag, Datenpunkttyp, Aktualisierungsintervall des Datenpunkts usw. Wenn auf eine Entität geklickt wird, wird diese mit einem grauen Hintergrund hervorgehoben, und die Datenpunkte werden sofort gefiltert (es ist nicht nötig, im Hauptfilter auf *Anwenden* zu klicken). Durch Klicken auf eine weitere Entität wird sie zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Klicken auf die Entität wird die Filterung entfernt.



Subfilter werden auf Grundlage der gefilterten Daten erzeugt, die auf 1000 Datensätze begrenzt sind. Wenn Sie im Subfilter mehr Datensätze sehen möchten, müssen Sie den Wert des Parameters *Limit for search and filter results* erhöhen (unter *Administration -> General -> GUI*).

Im Gegensatz zum Hauptfilter wird der Subfilter bei jeder Aktualisierungsanfrage der Tabelle aktualisiert, damit stets aktuelle Informationen über verfügbare Filteroptionen und deren Zählerwerte vorliegen.

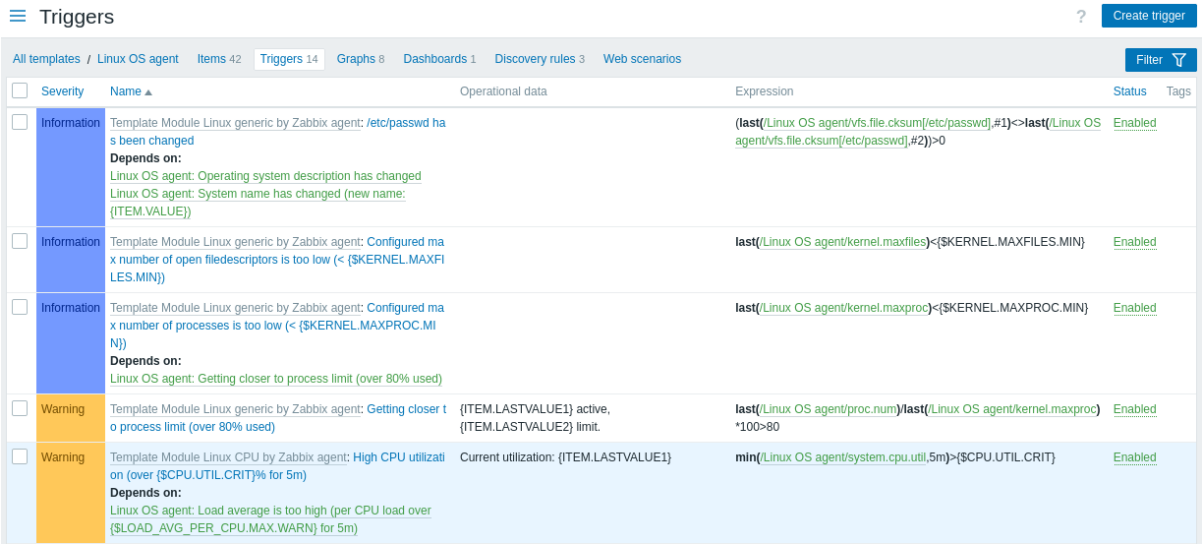
Die Anzahl der angezeigten Entitäten ist horizontal auf 100 begrenzt. Wenn es mehr gibt, wird am Ende ein Symbol mit drei Punkten angezeigt; dieses ist nicht anklickbar.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der darin gruppierten Datenpunkte an (basierend auf den Ergebnissen des Hauptfilters). Wenn auf eine Entität geklickt wird, werden die Zahlen bei anderen verfügbaren Entitäten mit einem Pluszeichen angezeigt, das angibt, wie viele Datenpunkte zur aktuellen Auswahl hinzugefügt werden können. Entitäten ohne Datenpunkte werden nicht angezeigt, es sei denn, sie wurden zuvor im Subfilter ausgewählt.

2 Auslöser

Übersicht

Die Auslöserliste für eine Vorlage kann unter *Datensammlung → Vorlagen* aufgerufen werden, indem Sie bei der jeweiligen 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. Ein Klick auf den Vorlagennamen öffnet das Konfigurationsformular der Vorlage. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Vorlagen oder keine Vorlagen ausgewählt sind.
Name	Der Name des Auslösers wird als blauer Link zu den Auslöserdetails angezeigt. Ein Klick auf den Link mit dem Auslösernamen öffnet das Konfigurationsformular des Auslösers. Wenn der Auslöser von einer anderen Vorlage geerbt wurde, wird der Vorlagenname vor dem Auslösernamen als grauer Link angezeigt. Ein Klick auf den Vorlagenlink öffnet die Auslöserliste auf dieser Vorlagenebene.
Betriebsdaten	Definition der Betriebsdaten des Auslösers, die beliebige Zeichenfolgen und Makros enthält, die in *Überwachung* → *Probleme* dynamisch aufgelöst werden.
Ausdruck	Der Auslöserausdruck wird angezeigt. Der Vorlage-Datenpunkt-Teil des Ausdrucks wird als Link angezeigt, der zum Konfigurationsformular des Datenpunkts führt.
Status	Der Status des Auslösers wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern – von Aktiviert zu Deaktiviert (und zurück).
Tags	Wenn ein Auslöser Tags enthält, werden Tag-Name und -Wert in dieser Spalte angezeigt.

Um einen neuen Auslöser zu konfigurieren, klicken Sie auf die Schaltfläche *Auslöser erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

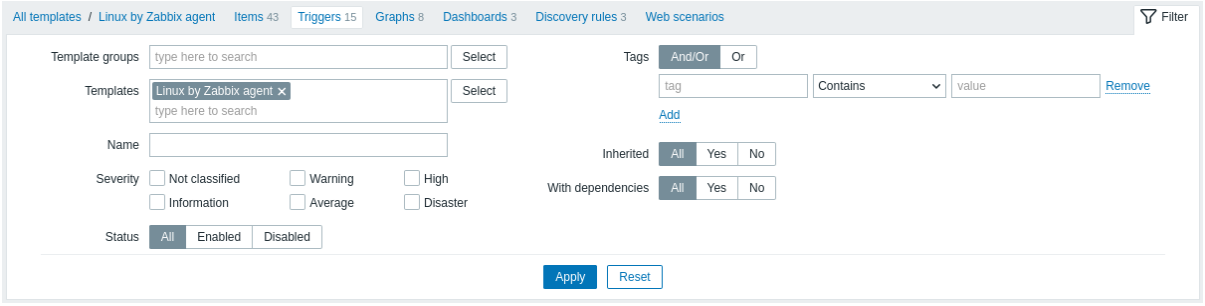
- *Aktivieren* - den Auslöserstatus auf *Aktiviert* ändern
- *Deaktivieren* - den Auslöserstatus auf *Deaktiviert* ändern
- *Kopieren* - die Auslöser auf andere Hosts oder Vorlagen kopieren
- *Massenaktualisierung* - mehrere Eigenschaften für mehrere Auslöser gleichzeitig aktualisieren
- *Löschen* - 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 in der oberen rechten Ecke verfügbar. Ein Klick darauf öffnet einen Filter, in dem Sie die gewünschten Filterkriterien angeben können.



Parameter	Beschreibung
Vorlagengruppen	Nach einer oder mehreren Vorlagengruppen filtern. Wenn eine übergeordnete Vorlagengruppe angegeben wird, werden implizit auch alle untergeordneten Gruppen ausgewählt.
Vorlagen	Nach einer oder mehreren Vorlagen filtern. Wenn oben bereits Vorlagengruppen ausgewählt wurden, ist die Auswahl der Vorlagen auf diese Gruppen beschränkt.
Name	Nach Auslösernamen filtern.
Schweregrad	Auswählen, um nach einem oder mehreren Auslöserschweregraden zu filtern.
Status	Nach dem Aktivierungsstatus des Auslösers filtern (Aktiviert/Deaktiviert).

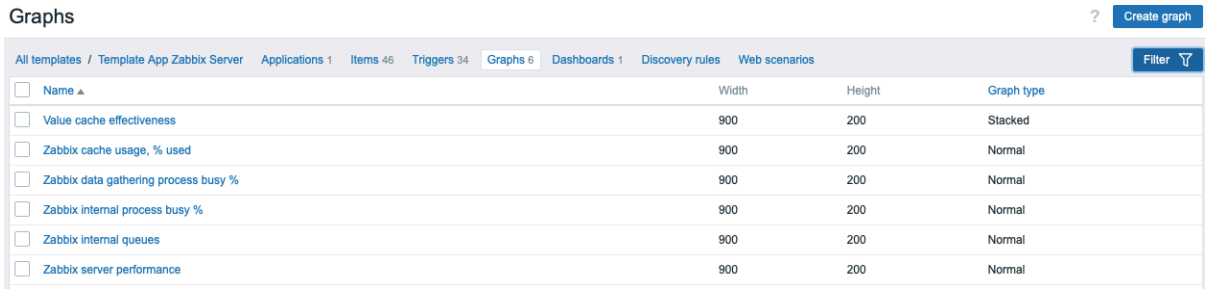
Parameter	Beschreibung
Tags	Nach Tag-Name und -Wert des Auslösers filtern. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung 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 Makros und Makrofunktionen werden in den Feldern für Tag-Namen und Tag-Werte unterstützt.
Vererbt	Auslöser filtern, die von verknüpften Vorlagen geerbt wurden (oder nicht geerbt wurden).
Mit Abhängigkeiten	Auslöser mit (oder ohne) Abhängigkeiten filtern.

3 Diagramme

Übersicht

Die Liste der benutzerdefinierten Graphen für eine Vorlage ist unter *Datensammlung* → *Vorlagen* erreichbar, indem Sie bei der jeweiligen Vorlage auf *Graphen* klicken.

Eine Liste der vorhandenen Graphen wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Vorlage	Vorlage, zu der der Graph gehört. Ein Klick auf den Vorlagennamen öffnet das Konfigurationsformular der Vorlage. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Vorlagen oder keine Vorlagen ausgewählt sind.
Name	Name des benutzerdefinierten Graphen, angezeigt als blauer Link zu den Graphdetails. Ein Klick auf den Link mit dem Graphnamen öffnet das Konfigurationsformular des Graphen. Wenn der Graph von einer anderen Vorlage geerbt wurde, wird der Vorlagenname vor dem Graphnamen als grauer Link angezeigt. Ein Klick auf den Vorlagenlink öffnet die Graphenliste auf der Ebene dieser Vorlage.
Breite	Die Breite des Graphen wird angezeigt.
Höhe	Die Höhe des Graphen wird angezeigt.
Graphtyp	Der Graphtyp wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.

Um einen neuen Graphen zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Graph erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Kopieren* - die Graphen auf andere Hosts oder Vorlagen kopieren
- *Löschen* - die Graphen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Graphen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

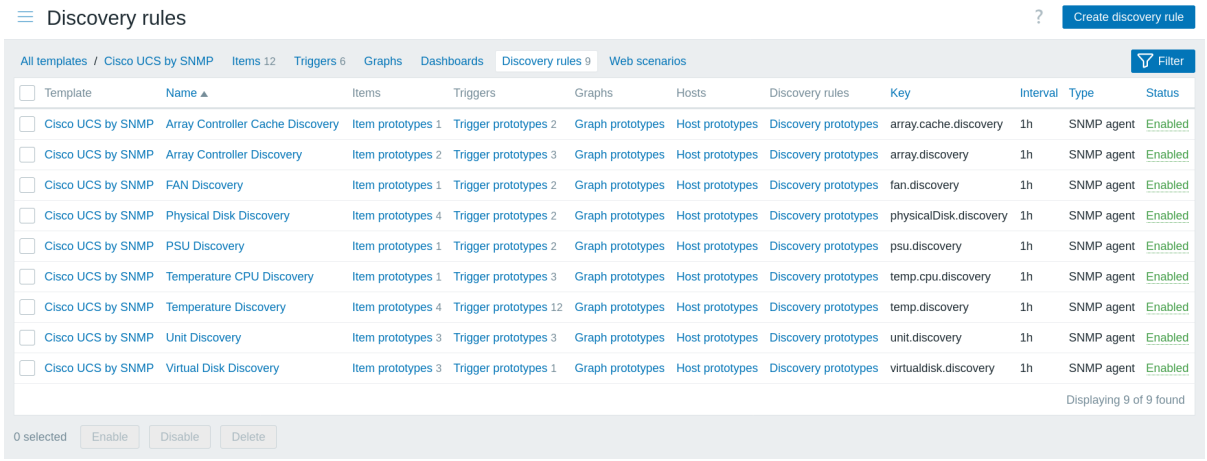
Sie können Graphen nach Vorlagengruppe und Vorlage filtern. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

4 Suchregeln

Übersicht

Die Liste der Low-Level-Discovery-Regeln für eine Vorlage kann über *Datensammlung* → *Vorlagen* aufgerufen werden, indem Sie bei der jeweiligen Vorlage auf *Discovery* klicken.

Eine Liste der vorhandenen Low-Level-Discovery-Regeln wird angezeigt. Es ist auch möglich, alle Discovery-Regeln unabhängig von der Vorlage oder alle Discovery-Regeln einer bestimmten Vorlagengruppe anzuzeigen, indem die Filtereinstellungen geändert werden.



Angezeigte Daten:

Column	Description
Vorlage	Die Vorlage, zu der die Discovery-Regel gehört. Ein Klick auf den Vorlagennamen öffnet das Konfigurationsformular der Vorlage.
Name	Name der Regel, angezeigt als blauer Link. Ein Klick auf den Regelnamen öffnet das Konfigurationsformular der Low-Level-Discovery-Regel. Wenn die Discovery-Regel von einer anderen Vorlage geerbt wird, wird der Vorlagenname vor dem Regelnamen als grauer Link angezeigt. Ein Klick auf den Vorlagenlink öffnet die Liste der Discovery-Regeln auf dieser Vorlagenebene.
Datenpunkte	Ein Link zur Liste der Datenpunktprototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunktprototypen wird grau dargestellt.
Auslöser	Ein Link zur Liste der Auslöserprototypen wird angezeigt. Die Anzahl der vorhandenen Auslöserprototypen wird grau dargestellt.
Diagramme	Ein Link zur Liste der Diagrammprototypen wird angezeigt. Die Anzahl der vorhandenen Diagrammprototypen wird grau dargestellt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau dargestellt.
Schlüssel	Der für die Discovery verwendete Datenpunktschlüssel wird angezeigt.
Intervall	Die Häufigkeit der Discovery-Ausführung wird angezeigt.
Typ	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix Agent, SNMP-Agent usw.).
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 zurück).

Um eine neue Low-Level-Discovery-Regel zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Discovery-Regel erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

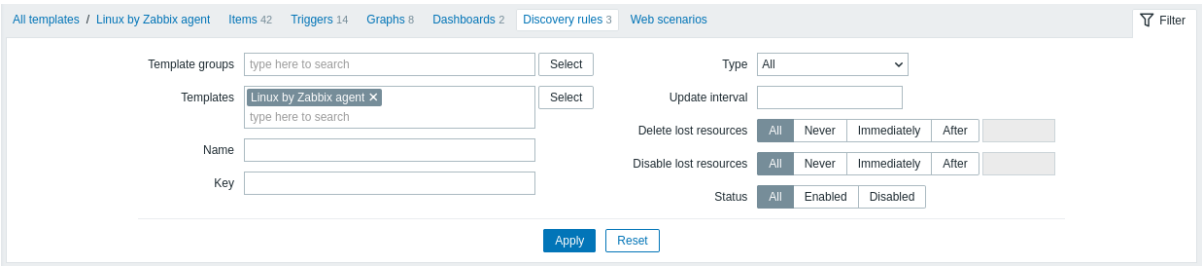
- *Aktivieren* - den Status der Low-Level-Discovery-Regel auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Low-Level-Discovery-Regel auf *Deaktiviert* ändern
- *Löschen* - die Low-Level-Discovery-Regeln löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Discovery-Regeln und klicken Sie dann auf die gewünschte Schaltfläche.

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.

Das Symbol *Filter* befindet sich in der oberen rechten Ecke. Ein Klick darauf öffnet einen Filter, in dem Sie die gewünschten Filterkriterien angeben können, z. B. Vorlage, Name der Discovery-Regel, Schlüssel des Datenpunkts, Typ des Datenpunkts usw.



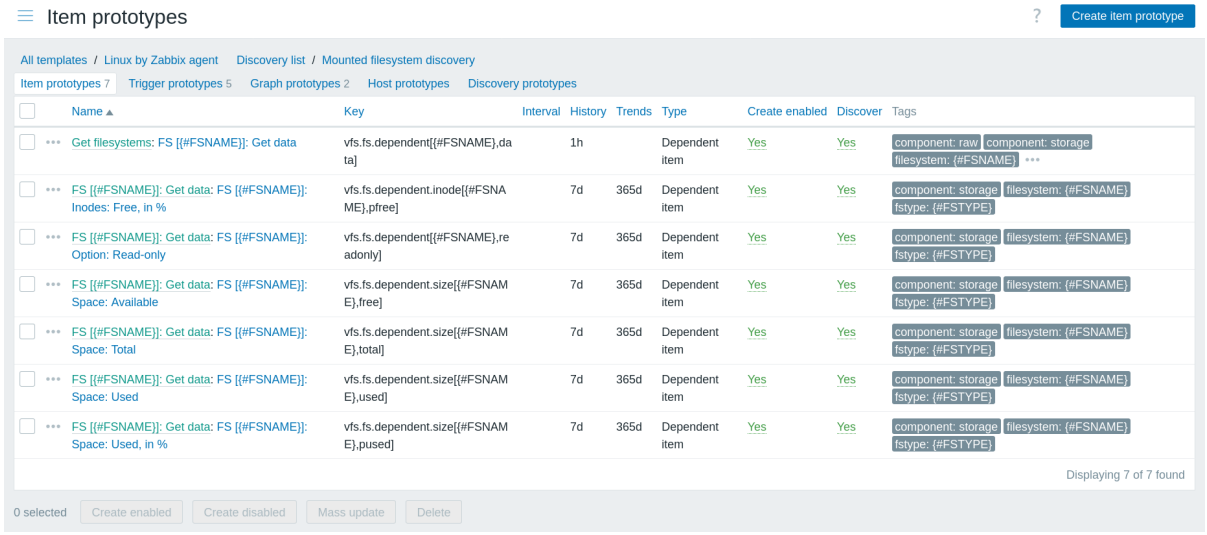
Parameter	Beschreibung
Vorlagengruppen	Nach einer oder mehreren Vorlagengruppen filtern. Wenn Sie eine übergeordnete Vorlagengruppe angeben, werden implizit alle untergeordneten Gruppen ausgewählt.
Vorlagen	Nach einer oder mehreren Vorlagen filtern.
Name	Nach dem Namen der Discovery-Regel filtern.
Schlüssel	Nach dem Schlüssel des Discovery-Datenpunkts filtern.
Typ	Nach dem Typ des Discovery-Datenpunkts filtern.
Aktualisierungsintervall	Nach dem Aktualisierungsintervall filtern. Nicht verfügbar für Zabbix-Trapper und abhängige Datenpunkte.
Verlorene Ressourcen löschen	Nach dem Zeitraum für *Verlorene Ressourcen löschen* filtern.
Verlorene Ressourcen deaktivieren	Nach dem Zeitraum für *Verlorene Ressourcen deaktivieren* filtern.
Status	Nach dem Status der Discovery-Regel filtern (Alle/Aktiviert/Deaktiviert).

1 Datenpunkt-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Datenpunkt-Prototypen einer Low-Level-Discovery-Regel in der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden Datenpunkt-Prototypen die Grundlage für die Erstellung echter Host-Datenpunkte während der Low-Level-Discovery.



Angezeigte Daten:

Column	Description
Name	Der Name des Datenpunkt-Prototyps wird als blauer Link angezeigt. Ein Klick auf den Namen öffnet das Konfigurationsformular des Datenpunkt-Prototyps. Wenn der Datenpunkt-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Namen des Datenpunkt-Prototyps als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Datenpunkt-Prototypen auf der Ebene der verknüpften Vorlage. Wenn der Datenpunkt-Prototyp vom Typ „Abhängiger Datenpunkt“ ist, wird der Name des Master-Datenpunkts vor dem Namen des Datenpunkt-Prototyps in Grün angezeigt (wie bei Datenpunkt-Prototypen auf der Host-Ebene).
Key	Der Schlüssel des Datenpunkt-Prototyps wird angezeigt.
Interval	Die Häufigkeit der Prüfung wird angezeigt.
History	Es wird angezeigt, wie viele Tage die Verlaufsdaten des Datenpunkts aufbewahrt werden.
Trends	Es wird angezeigt, wie viele Tage die Trenddaten des Datenpunkts aufbewahrt werden.
Type	Der Typ des Datenpunkt-Prototyps wird angezeigt (Zabbix-Agent, SNMP-Agent, einfache Prüfung usw.).
Create enabled	Den Datenpunkt basierend auf diesem Prototyp erstellen als: **Ja** - aktiviert **Nein** - deaktiviert. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Discover	Den Datenpunkt basierend auf diesem Prototyp entdecken: **Ja** - entdecken **Nein** - nicht entdecken. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
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 zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Aktiviert erstellen* - diese Datenpunkt-Prototypen als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Datenpunkt-Prototypen als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Datenpunkt-Prototypen gesammelt aktualisieren
- *Löschen* - diese Datenpunkt-Prototypen löschen

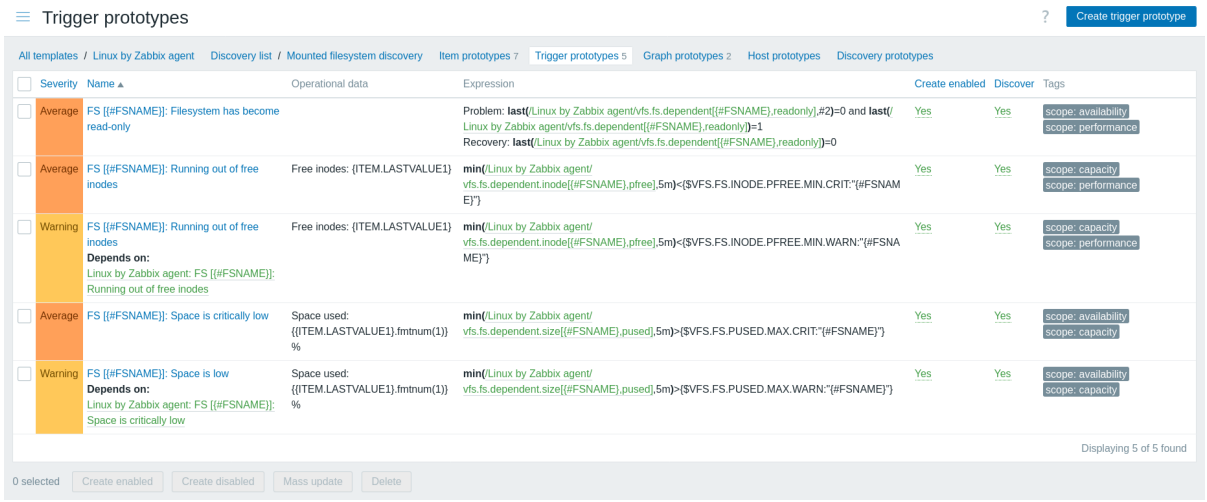
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunkt-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

2 Auslöser-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Auslöser-Prototypen einer Low-Level-Discovery-Regel in der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden Auslöser-Prototypen die Grundlage für die Erstellung echter Host-Auslöser während der Low-Level-Discovery.



Angezeigte Daten:

Spalte	Beschreibung
Schweregrad	Der Schweregrad des Auslösers wird sowohl durch den Namen als auch durch die Hintergrundfarbe der Zelle angezeigt.
Name	Name des Auslöser-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Auslöser-Prototyps. Wenn der Auslöser-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Auslösernamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Auslöser-Prototypen auf der Ebene der verknüpften Vorlage.
Ausdruck	Der Auslöser-Ausdruck wird angezeigt. Der Teil des Ausdrucks, der sich auf den Vorlagen-Datenpunkt bezieht, wird als Link angezeigt und führt zum Konfigurationsformular des Datenpunkts.
Betriebsdaten	Das Format der Betriebsdaten des Auslösers wird angezeigt und enthält beliebige Zeichenfolgen und Makros, die in *Monitoring* → *Probleme* dynamisch aufgelöst werden.
Aktiviert erstellen	Den Auslöser auf Basis dieses Prototyps erstellen als: **Ja** - aktiviert **Nein** - deaktiviert. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Entdecken	Den Auslöser auf Basis dieses Prototyps entdecken: **Ja** - entdecken **Nein** - nicht entdecken. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Tags	Tags des Auslöser-Prototyps werden angezeigt.

Um einen neuen Auslöser-Prototyp zu konfigurieren, klicken Sie auf die Schaltfläche *Auslöser-Prototyp erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Aktiviert erstellen* - diese Auslöser als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Auslöser als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Auslöser-Prototypen gesammelt aktualisieren
- *Löschen* - diese Auslöser-Prototypen löschen

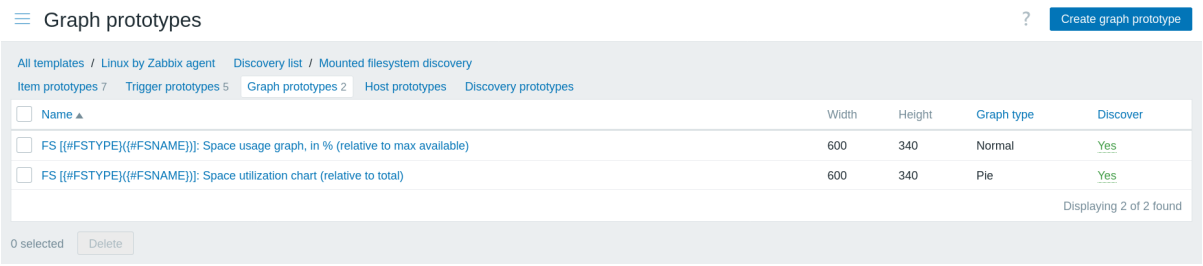
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Auslöser-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

3 Graph-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Graph-Prototypen einer Low-Level-Discovery-Regel in der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden Graph-Prototypen die Grundlage für die Erstellung echter Host-Graphen während der Low-Level-Discovery.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Graph-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Graph-Prototyps. Wenn der Graph-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Graphnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Graph-Prototypen auf Ebene der verknüpften Vorlage.
Breite	Die Breite des Graph-Prototyps wird angezeigt.
Höhe	Die Höhe des Graph-Prototyps wird angezeigt.
Typ	Der Typ des Graph-Prototyps wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.
Entdecken	Den Graphen basierend auf diesem Prototyp entdecken: **Ja** - entdecken **Nein** - nicht entdecken. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.

Um einen neuen Graph-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Graph-Prototyp erstellen*.

Optionen zur Massenbearbeitung

Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Löschen* - diese Graphprototypen löschen

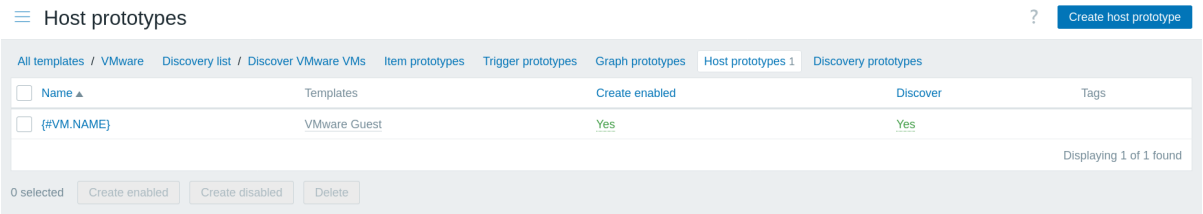
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Graphprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

4 Host-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Host-Prototypen einer Low-Level-Discovery-Regel in der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden Host-Prototypen die Grundlage für die Erstellung echter Hosts während der Low-Level-Discovery.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Host-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Host-Prototyps. Wenn der Host-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Hostnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Host-Prototypen auf der Ebene der verknüpften Vorlage.
Vorlagen	Vorlagen des Host-Prototyps werden angezeigt.

Spalte	Beschreibung
Erstellung aktiviert	Den Host basierend auf diesem Prototyp erstellen als: **Ja** - aktiviert **Nein** - deaktiviert. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Entdecken	Den Host basierend auf diesem Prototyp entdecken: **Ja** - entdecken **Nein** - nicht entdecken. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Tags	Tags des Host-Prototyps werden angezeigt.

Um einen neuen Host-Prototyp zu konfigurieren, klicken Sie auf die Schaltfläche *Host-Prototyp erstellen* in der oberen rechten Ecke.

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

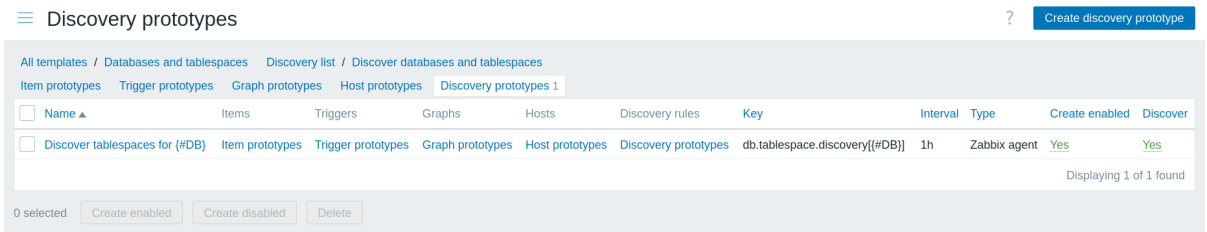
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Host-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

5 Discovery-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Suchprototypen einer Low-Level-Discovery-Regel in der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden Suchprototypen die Grundlage für die Erstellung echter Suchprototypen der übergeordneten Suchregel.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Suchprototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Suchprototyps. Wenn der Suchprototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Namen des Suchprototyps als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Suchprototypen auf der Ebene der verknüpften Vorlage.
Datenpunkte	Ein Link zur Liste der Datenpunktprototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunktprototypen wird grau dargestellt.
Auslöser	Ein Link zur Liste der Auslöserprototypen wird angezeigt. Die Anzahl der vorhandenen Auslöserprototypen wird grau dargestellt.
Diagramme	Ein Link zur Liste der Diagrammprototypen wird angezeigt. Die Anzahl der vorhandenen Diagrammprototypen wird grau dargestellt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau dargestellt.
Suchregeln	Ein Link zur Liste der Suchprototypen wird angezeigt. Die Anzahl der vorhandenen Suchprototypen wird grau dargestellt.
Schlüssel	Der für die Suche verwendete Datenpunktschlüssel wird angezeigt.
Intervall	Die Häufigkeit der Durchführung der Suche wird angezeigt.
Typ	Der für die Suche verwendete Datenpunkttyp wird angezeigt (Zabbix Agent, SNMP-Agent usw.).
Erstellung aktiviert	Der Erstellungsstatus des Suchprototyps 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
Suchen	Der Suchstatus des Suchprototyps wird angezeigt – suchen (*Ja*) oder nicht suchen (*Nein*). Durch Klicken auf den Status können Sie ihn ändern – von Ja zu Nein (und zurück).

Um einen neuen Suchprototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Suchprototyp erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Aktiviert erstellen* - diese Erkennungsprototypen als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Erkennungsprototypen als *Deaktiviert* erstellen
- *Löschen* - diese Erkennungsprototypen löschen

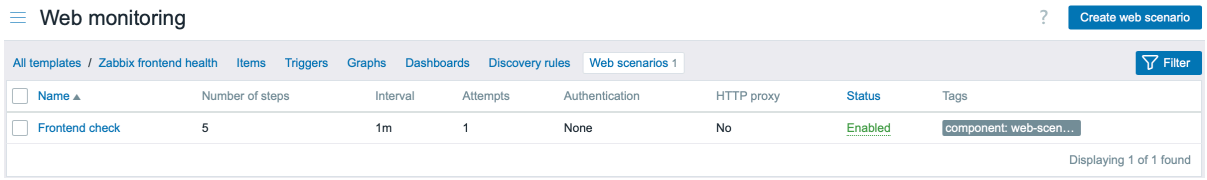
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Erkennungsprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

5 Web-Szenarien

Übersicht

Die Liste der Webszenarien für eine Vorlage kann über *Datensammlung* → *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Web* klicken.

Eine Liste der vorhandenen Webszenarien wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Webszenarios. Durch Klicken auf den Namen des Webszenarios wird das Konfigurationsformular des Webszenarios geöffnet. Wenn das Webszenario von einer anderen Vorlage geerbt wurde, wird der Vorlagenname vor dem Namen des Webszenarios als grauer Link angezeigt. Durch Klicken auf den Vorlagen-Link wird die Liste der Webszenarien auf der Ebene dieser Vorlage geöffnet.
Anzahl der Schritte	Die Anzahl der Schritte, die das Szenario enthält.
Aktualisierungsintervall	Wie oft das Szenario ausgeführt wird.
Versuche	Wie viele Versuche zum Ausführen der Schritte des Webszenarios durchgeführt werden.
Authentifizierung	Die Authentifizierungsmethode wird angezeigt – Basic, NTLM, Kerberos, Digest oder Keine.
HTTP-Proxy	Zeigt den HTTP-Proxy an oder „Nein“, wenn keiner verwendet wird.
Status	Der Status des Webszenarios wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Tags	Die Tags des Webszenarios werden angezeigt. Es können bis zu drei Tags (Name:Wert-Paare) angezeigt werden. Wenn mehr Tags vorhanden sind, wird ein „...“-Link angezeigt, über den Sie alle Tags sehen können, wenn Sie mit der Maus darüberfahren.

Um ein neues Webszenario zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Webszenario erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

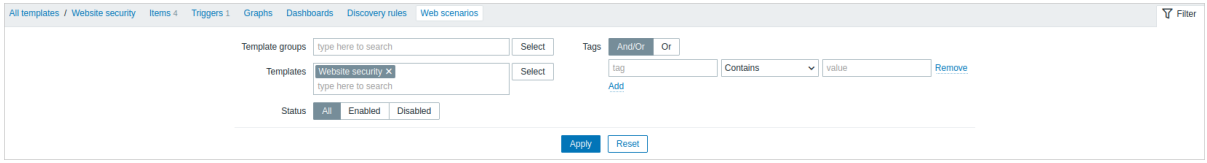
- *Aktivieren* - den Status des Szenarios auf *Aktiviert* ändern
- *Deaktivieren* - den Status des Szenarios auf *Deaktiviert* ändern
- *Löschen* - die Webszenarien löschen

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 Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Webszenarien verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Szenarien nach Vorlagengruppe, Vorlage, Status und Tags filtern können.

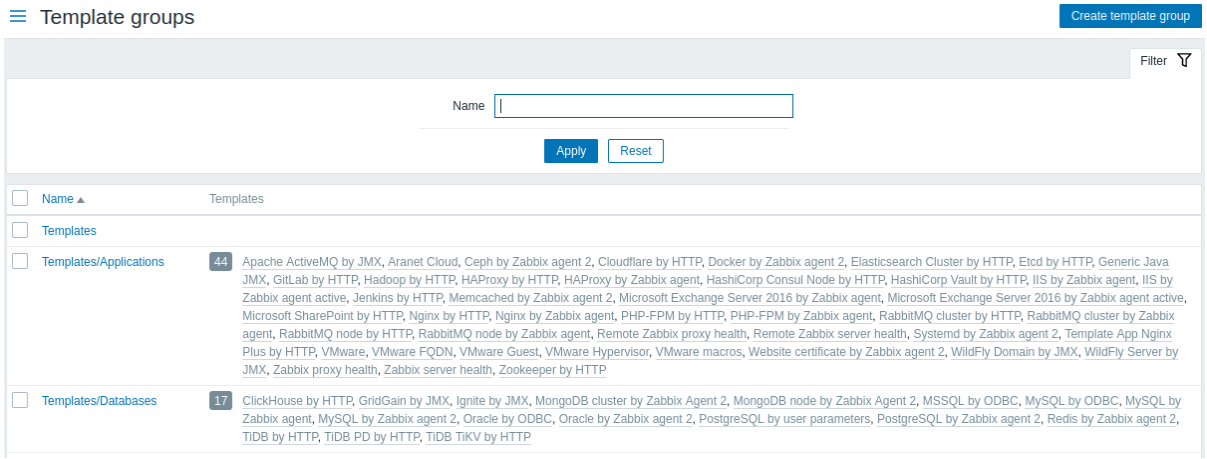


1 Vorlagengruppen

Übersicht

Im Abschnitt *Datenerfassung* → *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. Ein Klick auf den Gruppennamen öffnet das Konfigurationsformular der Gruppe.
Templates	Anzahl der Vorlagen in der Gruppe (grau dargestellt), gefolgt von der Liste der Gruppenmitglieder. Ein Klick auf einen Vorlagennamen öffnet das Konfigurationsformular der Vorlage. Ein Klick auf die Zahl öffnet die Liste der Vorlagen in dieser Gruppe.

Optionen zur 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

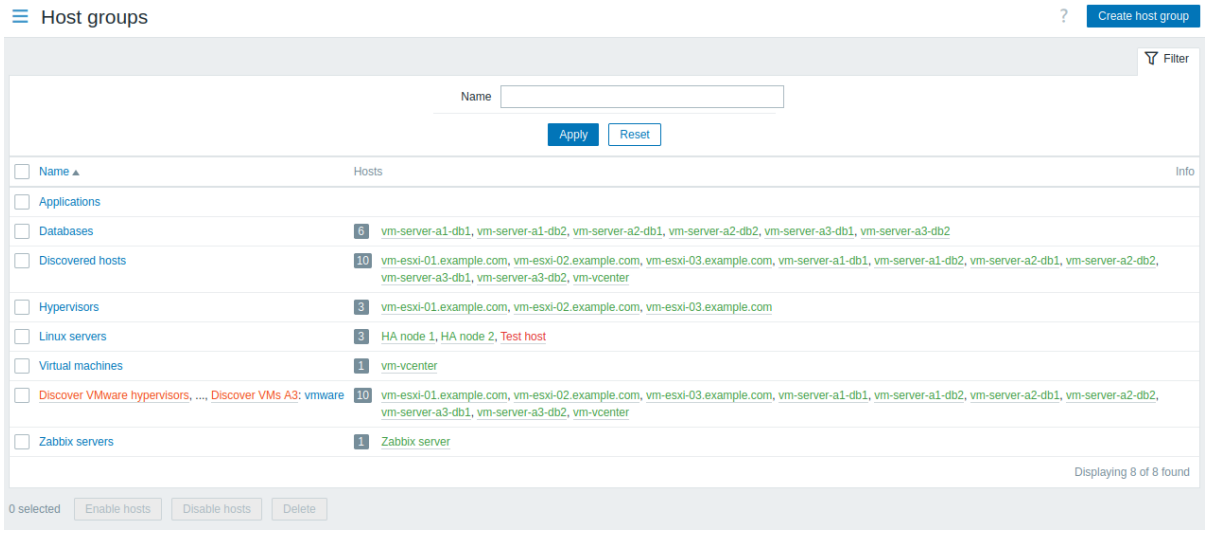
Sie können den Filter verwenden, um nur die Vorlagengruppen anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

2 Host-Gruppen

Übersicht

Im Abschnitt *Datensammlung* → *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:

Column	Description
Name	Name der Host-Gruppe. Durch Klicken auf den Gruppennamen wird das Konfigurationsformular der Gruppe geöffnet. Durch Low-Level-Discovery erkannte Host-Gruppen werden mit den Namen der Low-Level-Discovery-Regeln als Präfix angezeigt. Durch Klicken auf den Namen der LLD-Regel wird das Konfigurationsformular des Host-Prototyps geöffnet. Beachten Sie, dass erkannte Host-Gruppen gelöscht werden, wenn alle LLD-Regeln, die sie erkannt haben, gelöscht werden.
Hosts	Anzahl der Hosts in der Gruppe (grau dargestellt), gefolgt von der Liste der Gruppenmitglieder. Durch Klicken auf einen Host-Namen wird das Host-Konfigurationsformular geöffnet. Durch Klicken auf die Zahl werden in der vollständigen Host-Liste diejenigen herausgefiltert, die zu der Gruppe gehören.
Info	Fehlerinformationen (falls vorhanden) zur Host-Gruppe werden angezeigt.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur 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 entsprechenden Hostgruppen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

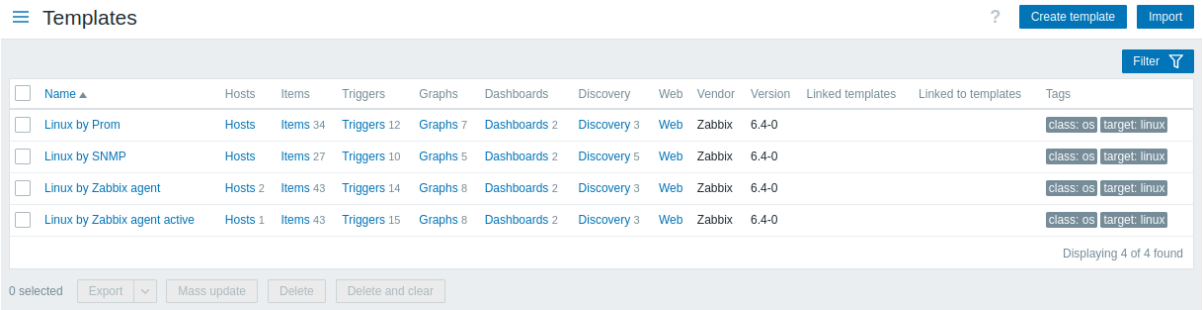
Sie können den Filter verwenden, um nur die Hostgruppen anzuzeigen, an denen Sie interessiert sind. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

3 Vorlagen

Übersicht

Im Abschnitt *Datensammlung → Vorlagen* können Benutzer Vorlagen konfigurieren und verwalten.

Eine Liste der vorhandenen Vorlagen mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Vorlage. Ein Klick auf den Vorlagennamen öffnet das Konfigurationsformular der Vorlage.
Hosts	Anzahl der bearbeitbaren Hosts, mit denen die Vorlage verknüpft ist; schreibgeschützte Hosts sind nicht enthalten. Ein Klick auf *Hosts* öffnet die Host-Liste, in der nur die mit der Vorlage verknüpften Hosts gefiltert angezeigt werden.
Entitäten (Datenpunkte, Auslöser, Diagramme, Dashboards, Discovery, Web)	Anzahl der jeweiligen Entitäten in der Vorlage (grau dargestellt). Ein Klick auf den Namen der Entität filtert in der vollständigen Liste dieser Entität diejenigen heraus, die zur Vorlage gehören.
Verknüpfte Vorlagen	Vorlagen, die mit der Vorlage verknüpft sind.
Mit Vorlagen verknüpft	Vorlagen, mit denen die Vorlage verknüpft ist.
Anbieter, *Version*	Anbieter und Version der Vorlage; werden angezeigt, wenn die Vorlagenkonfiguration solche Informationen enthält, und nur bei mitgelieferten Vorlagen, importierten Vorlagen oder Vorlagen, die über die Template API geändert wurden. Bei mitgelieferten Vorlagen wird die Version wie folgt angezeigt: Hauptversion von Zabbix, Trennzeichen (”-”), Revisionsnummer (wird mit jeder neuen Version der Vorlage erhöht und bei jeder Hauptversion von Zabbix zurückgesetzt). Zum Beispiel 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Tags	Tags der Vorlage, wobei Makros nicht aufgelöst werden.

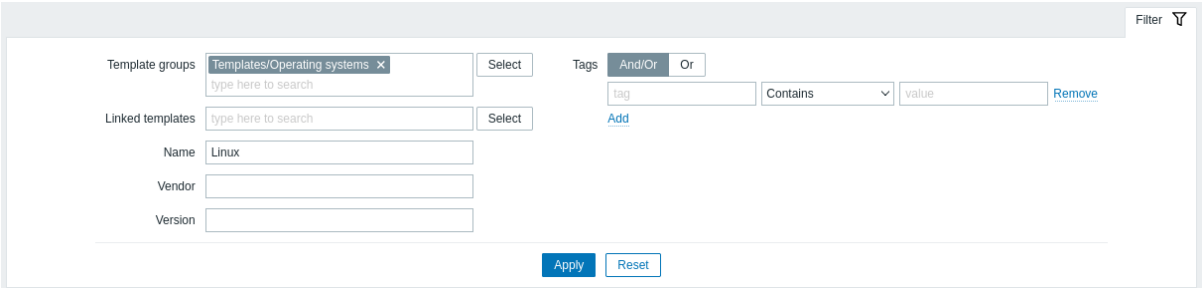
Um eine neue Vorlage zu konfigurieren, klicken Sie auf die Schaltfläche *Vorlage erstellen* in der oberen rechten Ecke.

Um eine Vorlage zu importieren aus einer YAML-, XML- oder JSON-Datei, klicken Sie auf die Schaltfläche *Importieren* in der oberen rechten Ecke.

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 *Create template* und *Import* verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Vorlagen nach Vorlagengruppen, verknüpften Vorlagen, Name und Tags filtern können.



Parameter	Beschreibung
Template groups	Nach einer oder mehreren Vorlagengruppen filtern. Die Angabe einer übergeordneten Vorlagengruppe wählt implizit auch alle verschachtelten Gruppen aus.
Linked templates	Nach direkt verknüpften Vorlagen filtern.
Name	Nach Vorlagenname filtern.
Vendor	Nach dem Anbieter der Vorlage filtern.

Parameter	Beschreibung
Version	Nach der Version der Vorlage filtern.
Tags	Nach Tag-Name und -Wert der Vorlage filtern. Das Filtern ist nur nach Tags auf Vorlagenebene möglich (nicht nach geerbten Tags). 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 (Teilstring-Abgleich, 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 (Teilstring-Abgleich, 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.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Exportieren* - die Vorlage in eine YAML-, XML- oder JSON-Datei exportieren;
- *Massenaktualisierung* - mehrere Eigenschaften für mehrere Vorlagen gleichzeitig aktualisieren;
- *Löschen* - die Vorlage löschen, wobei ihre verknüpften Entitäten (Datenpunkte, Auslöser usw.) bei den Hosts verbleiben;
- *Löschen und bereinigen* - die Vorlage und ihre verknüpften Entitäten von den Hosts löschen.

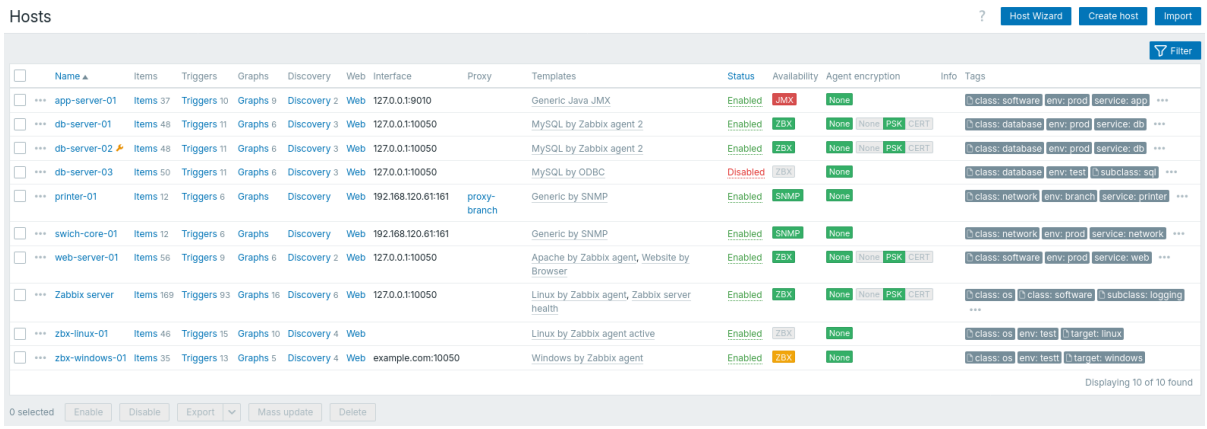
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den entsprechenden Vorlagen und klicken Sie dann auf die gewünschte Schaltfläche.

4 Hosts

Übersicht

Im Abschnitt *Datenerfassung* → *Hosts* können Benutzer Hosts konfigurieren und verwalten.

Eine Liste der vorhandenen Hosts mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Host-Menü	Klicken Sie auf das Symbol mit den drei Punkten, um das Host-Menü zu öffnen.

Spalte	Beschreibung
Name	Name des Hosts. Ein Klick auf den Host-Namen öffnet das Konfigurationsformular des Hosts. Ein oranges Schraubenschlüssel-Symbol nach dem Host-Namen zeigt an, dass sich dieser Host in Wartung befindet. Details zur Wartung werden angezeigt, wenn der Mauszeiger auf das Symbol gesetzt wird.
Entitäten (Datenpunkte, Auslöser, Graphen, Discovery, Web)	Ein Klick auf den Namen der Entität zeigt Datenpunkte, Auslöser usw. des Hosts an. Die Anzahl der jeweiligen Entitäten wird grau dargestellt.
Schnittstelle	Die Hauptschnittstelle des Hosts wird angezeigt.
Proxy	Zugewiesene Proxys werden in dieser Spalte angezeigt: <Proxy-Name> - Host wird von einem eigenständigen Proxy überwacht (auch wenn der Proxy Teil einer Proxy-Gruppe ist); <Name der Proxy-Gruppe: Proxy-Name> - Host wird von einer Proxy-Gruppe überwacht, und der Zabbix Server hat einen Proxy zur Überwachung des Hosts zugewiesen; <Name der Proxy-Gruppe> - Host wird von einer Proxy-Gruppe ohne Proxys überwacht oder wenn der Zabbix Server keinen Proxy zur Überwachung des Hosts zugewiesen hat; Nichts - 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. Ein Klick auf einen Vorlagennamen öffnet dessen Konfigurationsformular.
Status	Der Host-Status wird angezeigt - *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn manuell ändern. Verlorengegangene entdeckte Hosts sind mit einem Info-Symbol markiert. Der Tooltip-Text enthält Details zu ihrem Status.

Spalte	Beschreibung
Verfügbarkeit	Die Verfügbarkeit des Hosts pro konfigurierter Schnittstelle wird angezeigt. Die Verfügbarkeitssymbole stellen den aktuellen Status der Host-Schnittstelle auf dem Zabbix Server dar. Wenn Sie also einen Host im Frontend deaktivieren, wird seine Verfügbarkeit aktualisiert, nachdem der Zabbix Server die Konfigurationsänderungen synchronisiert hat. Ebenso wird bei Aktivierung eines Hosts seine Verfügbarkeit aktualisiert, nachdem der Zabbix Server die Konfigurationsänderungen synchronisiert und den Host abgefragt hat. Die Verfügbarkeitssymbole repräsentieren nur die konfigurierten Schnittstellentypen (Agent, SNMP, IPMI, JMX). 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. Für die Agent-Schnittstelle zeigt das Pop-up Schnittstellen (passiv) und aktive Prüfungen an. Wenn ein Host nur aktive Prüfungen hat, wird das Symbol der 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 nicht versucht, eine Verbindung zum Host herzustellen, 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 einzelnen Typs (Agent, SNMP, IPMI, JMX) wird durch diejenigen 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 auch die gesamte Verfügbarkeit der Agent-Schnittstelle wie oben beschrieben. Zur Bestimmung der Verfügbarkeit aktiver Prüfungen werden Heartbeat-Nachrichten im Thread für aktive Agent-Prüfungen 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 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** - Verwendung eines vorab geteilten Schlüssels; **Zert** - Verwendung eines Zertifikats.
Info	Fehlerinformationen zum Host werden angezeigt (falls vorhanden).
Tags	Tags des Hosts mit nicht aufgelösten Makros.

Um einen neuen Host zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Host Wizard* oder *Host erstellen*. Um einen Host aus einer YAML-, XML- oder JSON-Datei zu importieren, klicken Sie oben rechts auf die Schaltfläche *Importieren*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

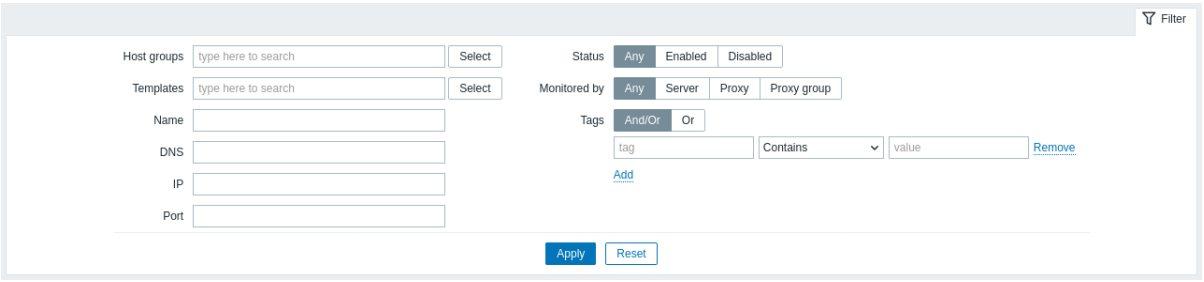
- *Aktivieren* - den Host-Status auf *Überwacht* ändern;
- *Deaktivieren* - den Host-Status auf *Nicht überwacht* ändern;
- *Exportieren* - die Hosts in eine YAML-, XML- oder JSON-Datei exportieren;
- *Massenaktualisierung* - mehrere Eigenschaften für mehrere Hosts gleichzeitig aktualisieren;
- *Löschen* - 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. Ein Klick darauf öffnet einen Filter, in dem Sie die gewünschten Filterkriterien angeben können.



Parameter	Beschreibung
Host-Gruppen	Nach einer oder mehreren Host-Gruppen filtern. Wenn eine übergeordnete Host-Gruppe angegeben wird, werden implizit auch alle untergeordneten Host-Gruppen ausgewählt.
Vorlagen	Nach verknüpften Vorlagen filtern.
Name	Nach dem sichtbaren Host-Namen filtern.
DNS	Nach dem DNS-Namen filtern.
IP	Nach der IP-Adresse filtern.
Port	Nach der Portnummer filtern.
Status	Nach dem Host-Status filtern.
Überwacht von	Hosts filtern, die vom Zabbix Server, Proxy oder der Proxy-Gruppe überwacht werden.
Proxys	Hosts filtern, die von den hier angegebenen Proxys überwacht werden. Dieses Feld ist nur verfügbar, wenn im Feld *Überwacht von* „Proxy“ ausgewählt ist.
Proxy-Gruppen	Hosts filtern, die von den hier angegebenen Proxy-Gruppen überwacht werden. Dieses Feld ist nur verfügbar, wenn im Feld *Überwacht von* „Proxy group“ ausgewählt ist.
Tags	Nach Host-Tag-Name und -Wert 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. 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.

Unbekannter Schnittstellenstatus

Der Zabbix Server zeigt den Status „Unbekannt“ für eine Host-Schnittstelle (Agent, SNMP, IPMI, JMX) in den folgenden Fällen an:

- Der Host ist deaktiviert.
- Der Host ist so eingestellt, dass er durch einen Proxy, einen anderen Proxy oder den Server überwacht wird, wenn er zuvor durch einen Proxy überwacht wurde.
- Der Host wird durch einen Proxy überwacht, der offenbar offline ist (keine Aktualisierungen vom Proxy während des maximalen Heartbeat-Intervalls erhalten – 1 Stunde).
- Alle Host-Datenpunkte mit diesem Schnittstellentyp sind deaktiviert.
- Für diesen Schnittstellentyp wurden keine Poller konfiguriert (zum Beispiel ist der Server-Konfigurationsparameter `StartAgentPollers` oder `StartSNMPPollers` auf 0 gesetzt).

Die Schnittstellenverfügbarkeit wird nach der Synchronisierung des Zabbix-Server-Konfigurationscaches auf „Unbekannt“ gesetzt.

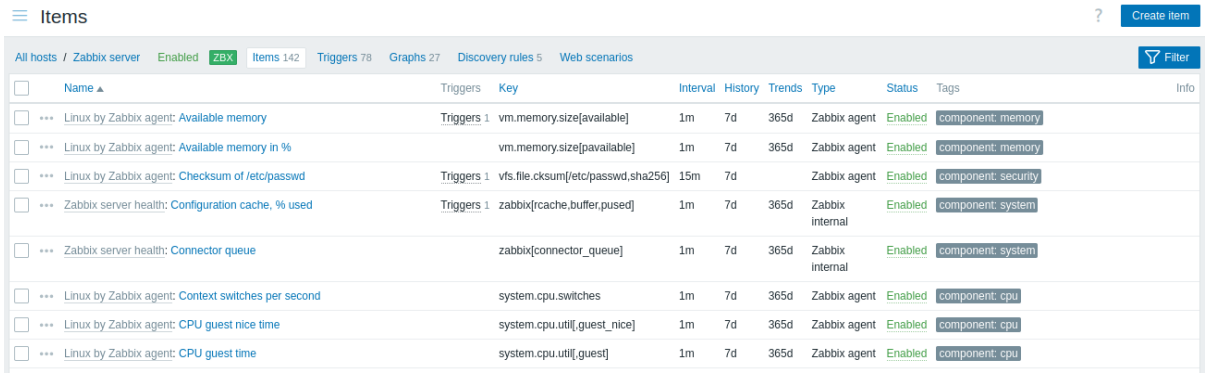
Die Schnittstellenverfügbarkeit (Verfügbar/Nicht verfügbar) auf Hosts, die durch Proxys überwacht werden, wird nach der Synchronisierung des Proxy-Konfigurationscaches wiederhergestellt.

1 Datenpunkte

Übersicht

Die Datenpunktliste für einen Host kann über *Datensammlung* → *Hosts* aufgerufen werden, indem Sie beim jeweiligen Host auf *Datenpunkte* klicken.

Eine Liste der vorhandenen Datenpunkte wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Datenpunkt-Menü	Klicken Sie auf das Symbol mit den drei Punkten, um das Datenpunkt-Menü zu öffnen.
Host	Host des Datenpunkts. Ein Klick auf den Hostnamen öffnet das Konfigurationsformular des Hosts. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Hosts oder keine Hosts ausgewählt sind.
Name	Name des Datenpunkts, angezeigt als blauer Link zu den Datenpunktdetails. Ein Klick auf den Link mit dem Datenpunktnamen öffnet das Konfigurationsformular des Datenpunkts. Wenn der Host-Datenpunkt zu einer Vorlage gehört, wird der Vorlagenname vor dem Datenpunktnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Datenpunktliste auf Vorlagenebene. Wenn der Datenpunkt aus einem Datenpunktprototyp erstellt wurde, wird seinem Namen der Name der Low-Level-Discovery-Regel in Orange vorangestellt. Ein Klick auf den Namen der Discovery-Regel öffnet die Liste der Datenpunktprototypen.
Auslöser	Wenn Sie den Mauszeiger über *Auslöser* bewegen, wird eine Infobox mit den dem Datenpunkt zugeordneten Auslösern angezeigt. Die Anzahl der Auslöser wird grau dargestellt.
Schlüssel	Der Datenpunktschlüssel wird angezeigt.
Intervall	Die Häufigkeit der Prüfung wird angezeigt. Beachten Sie, dass passive Datenpunkte auch sofort geprüft werden können, indem Sie die Schaltfläche *Jetzt ausführen* button betätigen.
Verlauf	Es wird angezeigt, wie viele Tage der Datenpunktverlauf gespeichert wird.
Trends	Es wird angezeigt, wie viele Tage die Trendhistorie des Datenpunkts gespeichert wird.
Typ	Der Datenpunkttyp wird angezeigt (Zabbix-Agent, SNMP-Agent, einfache Prüfung usw.).

Spalte	Beschreibung
Status	Der Datenpunktstatus wird angezeigt – *Aktiviert*, *Deaktiviert* oder *Nicht unterstützt*. Sie können den Status manuell ändern, indem Sie darauf klicken – von Aktiviert zu Deaktiviert (und zurück); von Nicht unterstützt zu Deaktiviert (und zurück). Verlorengegangene entdeckte Datenpunkte sind mit einem Info-Symbol markiert. Der Tooltip-Text enthält Details zu ihrem Status.
Tags	Die Datenpunkt-Tags werden angezeigt. Es können bis zu drei Tags (Name:Wert-Paare) angezeigt werden. Wenn mehr Tags vorhanden sind, wird ein „...“-Link angezeigt, über den beim Überfahren mit der Maus alle Tags sichtbar werden.
Info	Wenn der Datenpunkt korrekt funktioniert, wird in dieser Spalte kein Symbol angezeigt. Im Fehlerfall 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 zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

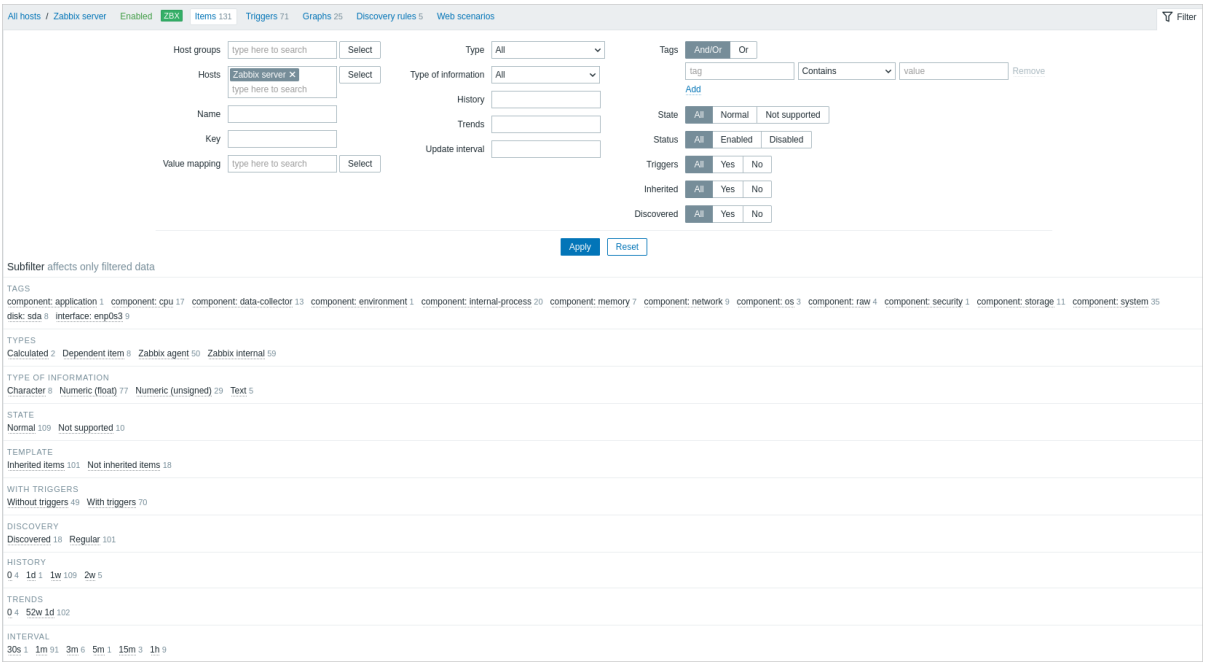
- *Aktivieren* - den Status des Datenpunkts auf *Aktiviert* ändern
- *Deaktivieren* - den Status des Datenpunkts auf *Deaktiviert* ändern
- *Jetzt ausführen* - eine Prüfung auf neue Datenpunktwerte sofort ausführen. Nur für **passive** Prüfungen unterstützt (siehe weitere Details). Beachten Sie, dass beim sofortigen Prüfen auf Werte der Konfigurations-Cache nicht aktualisiert wird; daher spiegeln die Werte keine sehr aktuellen Änderungen an der Datenpunktkonfiguration wider.
- *Verlauf und Trends löschen* - Verlaufs- und Trenddaten für Datenpunkte löschen.
- *Kopieren* - die Datenpunkte auf andere Hosts oder Vorlagen kopieren.
- *Massenaktualisierung* - mehrere Eigenschaften für mehrere Datenpunkte gleichzeitig aktualisieren.
- *Löschen* - 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

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 *Filter* ist in der oberen rechten Ecke verfügbar. Ein Klick darauf öffnet einen Filter, in dem Sie die gewünschten Filterkriterien angeben können.

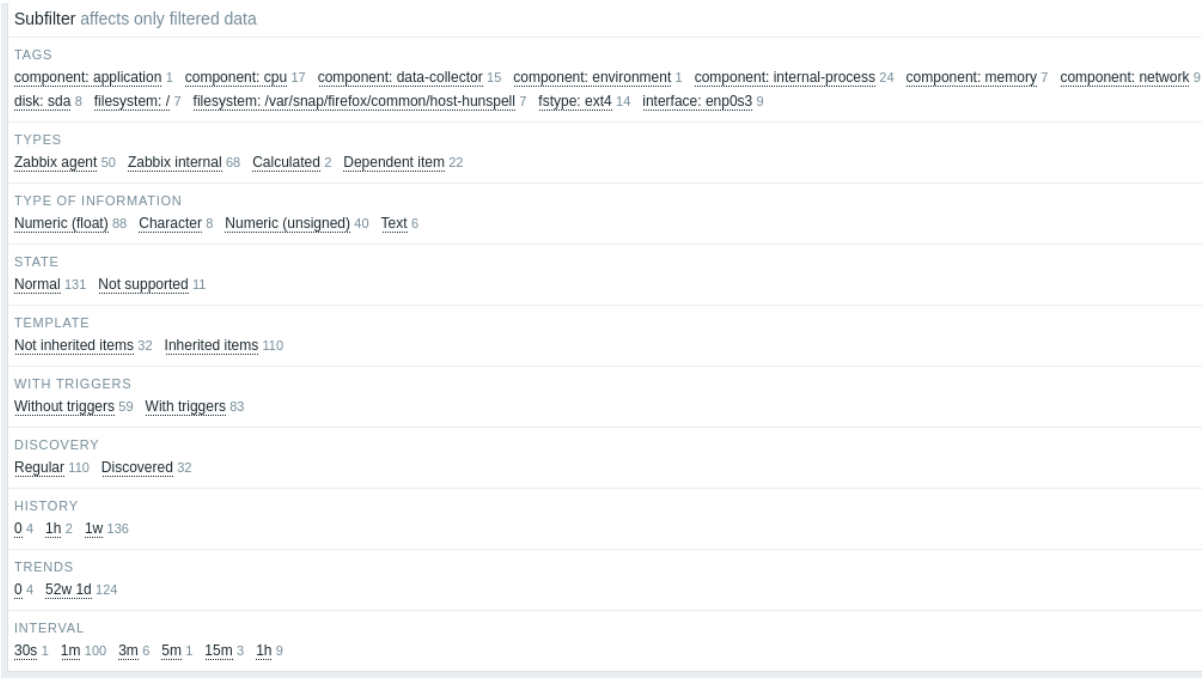


Parameter	Beschreibung
Host-Gruppen	Filtern nach einer oder mehreren Host-Gruppen. Bei Angabe einer übergeordneten Host-Gruppe werden implizit alle untergeordneten Host-Gruppen ausgewählt. Host-Gruppen, die nur Vorlagen enthalten, können nicht ausgewählt werden.
Hosts	Filtern nach einem oder mehreren Hosts.
Name	Filtern nach dem Namen des Datenpunkts.
Schlüssel	Filtern nach dem Schlüssel des Datenpunkts.
Wertzuordnung	Filtern nach der verwendeten Wertezuordnung. Dieser Parameter wird nicht angezeigt, wenn die Option *Hosts* leer ist.
Typ	Filtern nach dem Typ des Datenpunkts (Zabbix-Agent, SNMP-Agent usw.).
Informationstyp	Filtern nach dem Informationstyp (Numerisch ohne Vorzeichen, Gleitkommazahl usw.).
Verlauf	Filtern danach, wie lange der Verlauf des Datenpunkts aufbewahrt wird.
Trends	Filtern danach, wie lange die Trends des Datenpunkts aufbewahrt werden.
Aktualisierungsintervall	Filtern nach dem Aktualisierungsintervall des Datenpunkts.
Tags	Geben Sie Tags an, um die Anzahl der angezeigten Datenpunkte 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
Status	Filtern nach dem Zustand des Datenpunkts - *Normal* oder *Nicht unterstützt*.
Aktiviert	Filtern nach dem Status des Datenpunkts - *Aktiviert* oder *Deaktiviert*.
Auslöser	Filtern von Datenpunkten mit (oder ohne) Auslöser.
Geerbt	Filtern von Datenpunkten, die von einer Vorlage geerbt wurden (oder nicht geerbt wurden).
Discovery	Filtern von Datenpunkten, die durch Low-Level-Discovery erkannt wurden (oder nicht erkannt wurden).

Verwendung des Unterfilters

Der Unterfilter ermöglicht es, die Filterung aus dem Hauptfilter weiter anzupassen.

Er enthält anklickbare Links für einen schnellen Zugriff auf verwandte Datenpunkte. Datenpunkte sind über gemeinsame Entitäten miteinander verknüpft – Tag, Datenpunkttyp, Datenpunktstatus, Aktualisierungsintervall des Datenpunkts usw. Wenn auf eine Entität geklickt wird, wird sie mit einem grauen Hintergrund hervorgehoben, und die Datenpunkte werden sofort gefiltert (es ist nicht nötig, im Hauptfilter auf *Anwenden* zu klicken). Durch Klicken auf eine weitere Entität wird sie zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Klicken auf die Entität wird die Filterung entfernt.



Unterfilter werden auf Grundlage der gefilterten Daten erzeugt, die auf 1000 Datensätze begrenzt sind. Wenn Sie im Unterfilter mehr Datensätze sehen möchten, müssen Sie den Wert des Parameters *Limit for search and filter results* erhöhen (unter *Administration -> General -> GUI*).

Im Gegensatz zum Hauptfilter wird der Unterfilter bei jeder Anforderung zur Tabellenaktualisierung aktualisiert, damit stets aktuelle Informationen über verfügbare Filteroptionen und deren Zählerwerte vorliegen.

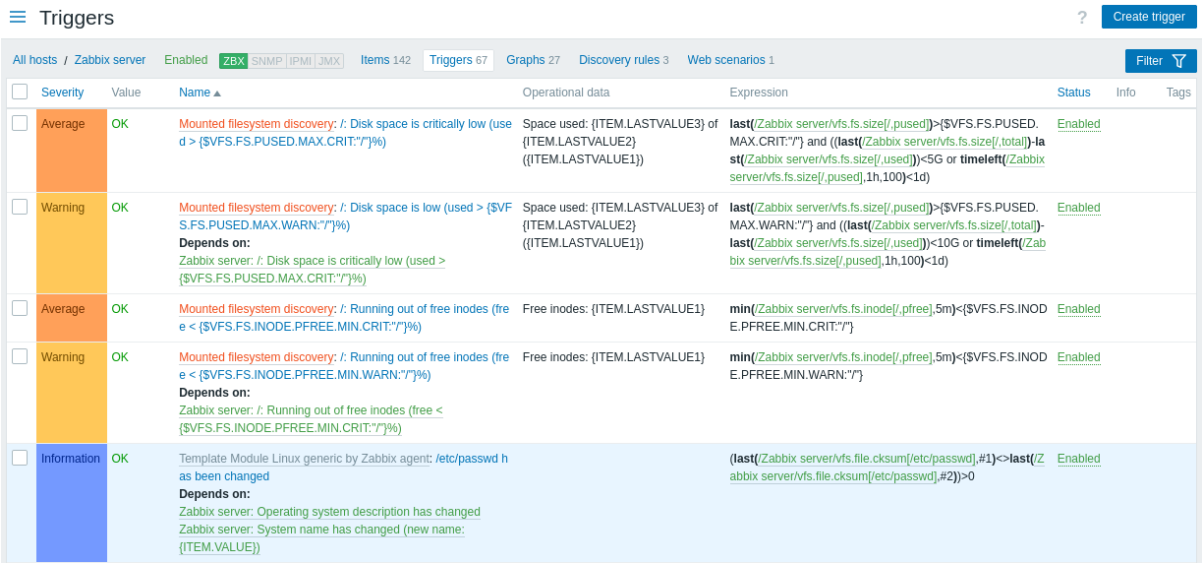
Die Anzahl der angezeigten Entitäten ist horizontal auf 100 begrenzt. Wenn es mehr gibt, wird am Ende ein Symbol mit drei Punkten angezeigt; dieses ist nicht anklickbar.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der darin gruppierten Datenpunkte an (basierend auf den Ergebnissen des Hauptfilters). Wenn auf eine Entität geklickt wird, werden die Zahlen bei anderen verfügbaren Entitäten mit einem Pluszeichen angezeigt, das angibt, wie viele Datenpunkte zur aktuellen Auswahl hinzugefügt werden können. Entitäten ohne Datenpunkte werden nicht angezeigt, es sei denn, sie wurden zuvor im Unterfilter ausgewählt.

2 Auslöser

Übersicht

Die Auslöser-Liste für einen Host kann über *Datensammlung → Hosts* aufgerufen werden, indem Sie beim jeweiligen 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 Zustand des Auslösers wird angezeigt: **OK** - OK-Zustand **PROBLEM** - Problemzustand
Host	Host des Auslösers. Ein Klick auf den Hostnamen öffnet das Konfigurationsformular des Hosts. 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 Auslöser-Details. Ein Klick auf den Link mit dem Auslösernamen öffnet das Konfigurationsformular des Auslösers. Wenn der Host-Auslöser zu einer Vorlage gehört, wird der Vorlagenname vor dem Auslösernamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Auslöser-Liste auf Vorlagenebene. Wenn der Auslöser aus einem Auslöser-Prototyp erstellt wurde, wird seinem Namen der Name der Low-Level-Discovery-Regel in Orange vorangestellt. Ein Klick auf den Namen der Discovery-Regel öffnet die Liste der Auslöser-Prototypen.
Betriebsdaten	Definition der Betriebsdaten des Auslösers, die beliebige Zeichenfolgen und Makros enthält, welche unter *Überwachung* → *Probleme* dynamisch aufgelöst werden.
Ausdruck	Der Auslöser-Ausdruck wird angezeigt. Der Host-Datenpunkt-Teil des Ausdrucks wird als Link angezeigt, der zum Konfigurationsformular des Datenpunkts 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); von Unbekannt zu Deaktiviert (und zurück). Probleme eines deaktivierten Auslösers werden im Frontend nicht mehr angezeigt, aber nicht gelöscht. Verloren gegangene entdeckte Auslöser sind mit einem Info-Symbol markiert. Der Tooltip-Text enthält Details zu ihrem Status.
Info	Wenn alles korrekt funktioniert, wird in dieser Spalte kein Symbol angezeigt. Im Fehlerfall 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 Tag-Name und -Wert in dieser Spalte angezeigt.

Um einen neuen Auslöser zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Auslöser erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

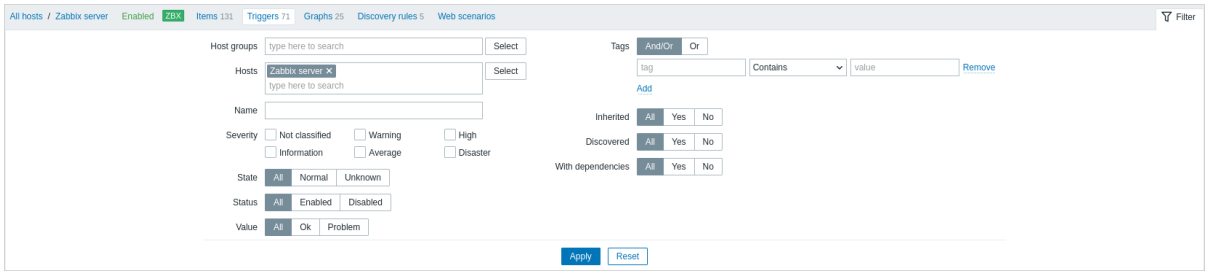
- *Aktivieren* - den Status des Auslösers auf *Aktiviert* ändern.
- *Deaktivieren* - den Status des Auslösers auf *Deaktiviert* ändern.
- *Kopieren* - die Auslöser auf andere Hosts oder Vorlagen kopieren.
- *Massenaktualisierung* - mehrere Eigenschaften für mehrere Auslöser gleichzeitig aktualisieren.
- *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* befindet sich in der oberen rechten Ecke. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien festlegen können.



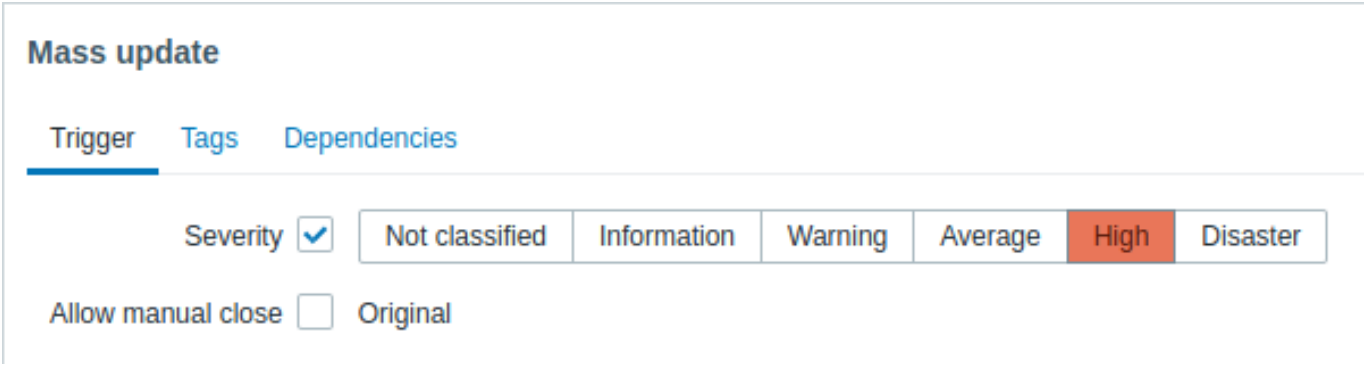
Parameter	Beschreibung
Host groups	Nach einer oder mehreren Host-Gruppen filtern. Wenn eine übergeordnete Host-Gruppe angegeben wird, werden implizit auch alle untergeordneten Host-Gruppen ausgewählt. Host-Gruppen, die nur Vorlagen enthalten, können nicht ausgewählt werden.
Hosts	Nach einem oder mehreren Hosts filtern. Wenn oben bereits Host-Gruppen ausgewählt wurden, ist die Host-Auswahl auf diese Gruppen beschränkt.
Name	Nach dem Namen des Auslösers filtern.
Severity	Auswählen, um nach einem oder mehreren Schweregraden von Auslösern zu filtern.
State	Nach dem Ausdruckszustand des Auslösers filtern (Normal/Unbekannt).
Status	Nach dem Aktivierungsstatus des Auslösers filtern (Aktiviert/Deaktiviert).
Value	Nach dem Wert des Auslösers filtern.
Tags	Nach Tag-Namen und Tag-Wert des Auslösers 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. 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 Makros und Makrofunktionen werden sowohl in den Feldern für Tag-Namen als auch für Tag-Werte unterstützt.
Inherited	Auslöser filtern, die von einer Vorlage geerbt wurden (oder nicht geerbt wurden).
Discovered	Auslöser filtern, die durch Low-Level-Discovery erkannt wurden (oder nicht erkannt wurden).
With dependencies	Auslöser mit (oder ohne) Abhängigkeiten filtern.

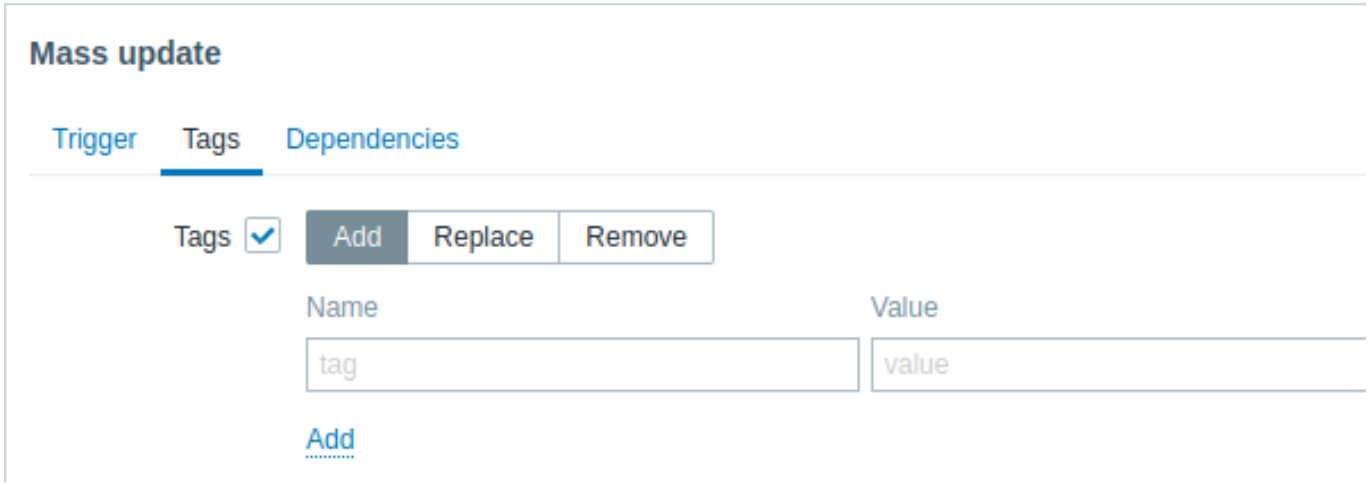
Massenaktualisierung verwenden

Mit der Massenaktualisierung können Sie bestimmte Attribute für mehrere Auslöser gleichzeitig ändern. Dadurch müssen Sie nicht jeden einzelnen Auslöser zur Bearbeitung öffnen.

Um mehrere Auslöser per Massenaktualisierung zu ändern, gehen Sie wie folgt vor:

- Aktivieren Sie in der Liste die Kontrollkästchen der Auslöser, die Sie aktualisieren möchten
- Klicken Sie unterhalb der Liste auf *Massenaktualisierung*
- Wechseln Sie zur Registerkarte mit den erforderlichen Attributen (*Auslöser*, *Tags* oder *Abhängigkeiten*)
- Aktivieren Sie die Kontrollkästchen der Attribute, die aktualisiert werden sollen

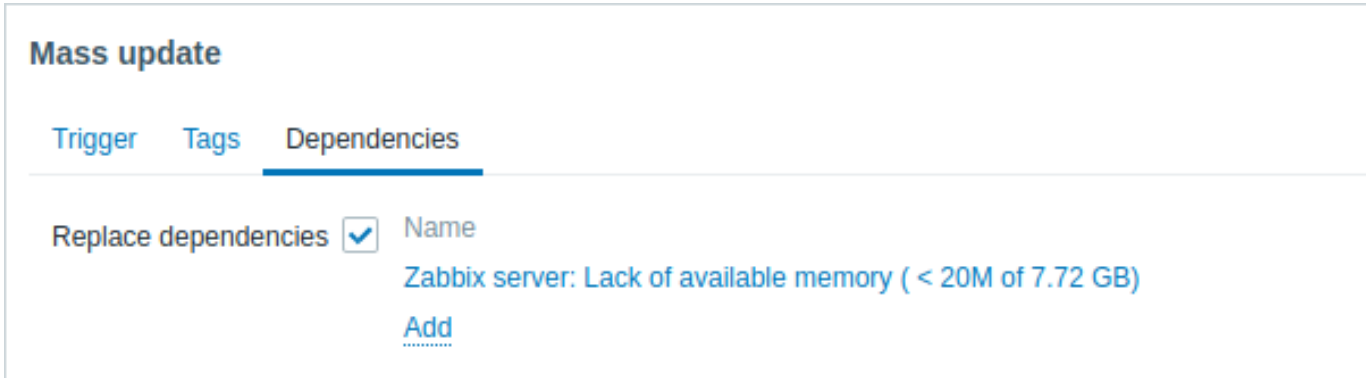




Für die Tag-Aktualisierung stehen bei Auswahl der entsprechenden Schaltfläche folgende Optionen zur Verfügung:

- *Hinzufügen* - ermöglicht das Hinzufügen neuer Tags zu den Auslösern;
- *Ersetzen* - entfernt alle vorhandenen Tags aus dem Auslöser und ersetzt sie durch die unten angegebenen Tags;
- *Entfernen* - entfernt die angegebenen Tags aus den Auslösern.

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.

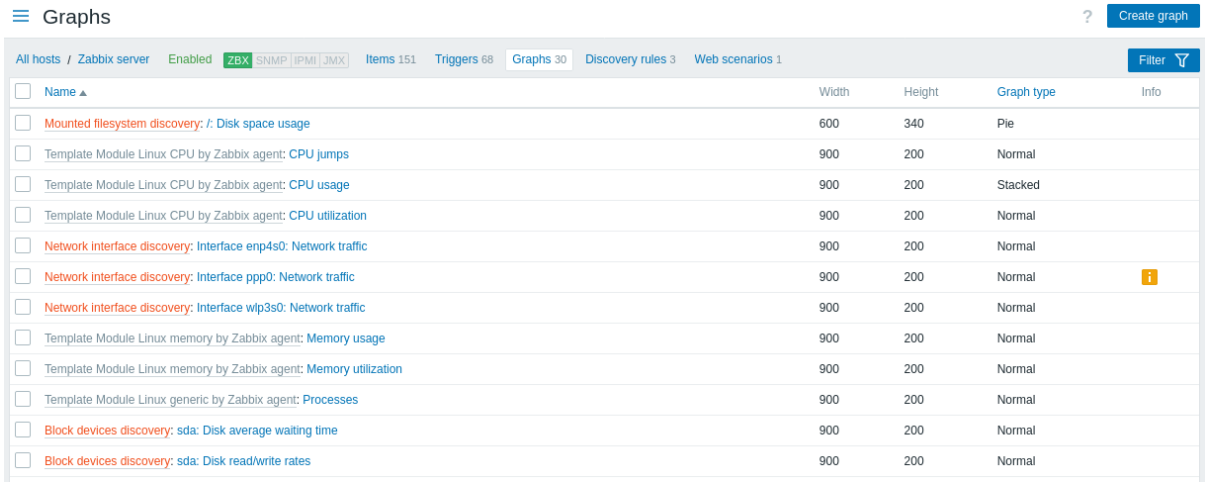
Klicken Sie auf *Aktualisieren*, um die Änderungen anzuwenden.

3 Diagramme

Übersicht

Die Liste der benutzerdefinierten Graphen für einen Host kann unter *Datensammlung* → *Hosts* aufgerufen werden, indem Sie beim jeweiligen Host auf *Graphen* klicken.

Eine Liste der vorhandenen Graphen wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Host	Host des Graphen. Ein Klick auf den Hostnamen öffnet das Konfigurationsformular des Hosts. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Hosts oder keine Hosts ausgewählt sind.
Name	Name des benutzerdefinierten Graphen, angezeigt als blauer Link zu den Graphdetails. Ein Klick auf den Link mit dem Graphnamen öffnet das Konfigurationsformular des Graphen. Wenn der Host-Graph zu einer Vorlage gehört, wird der Vorlagenname vor dem Graphnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Graphenliste auf Vorlagenebene. Wenn der Graph aus einem Graphprototyp erstellt wurde, wird seinem Namen der Name der Low-Level-Discovery-Regel in Orange vorangestellt. Ein Klick auf den Namen der Discovery-Regel öffnet die Liste der Graphprototypen.
Breite	Die Breite des Graphen wird angezeigt.
Höhe	Die Höhe des Graphen wird angezeigt.
Graphtyp	Der Graphtyp wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.
Info	Wenn der Graph korrekt funktioniert, wird in dieser Spalte kein Symbol angezeigt. Im Fehlerfall 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 Graphen zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Graph erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Kopieren* - die Graphen auf andere Hosts oder Vorlagen kopieren
- *Löschen* - die Graphen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Graphen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

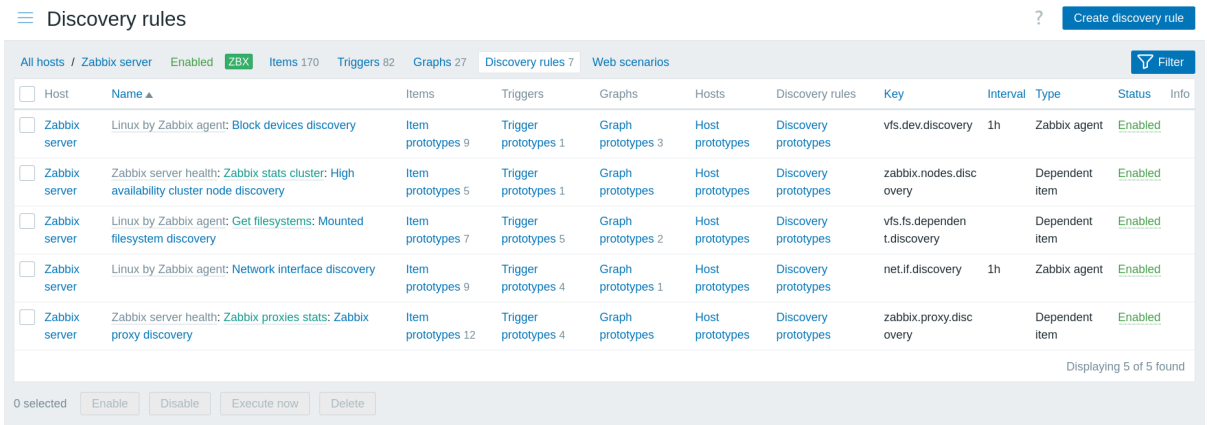
Sie können Graphen nach Host-Gruppe und Host filtern. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

4 Discovery-Regeln

Übersicht

Die Liste der Low-Level-Discovery-Regeln für einen Host kann über *Datensammlung* → *Hosts* aufgerufen werden, indem Sie beim jeweiligen Host auf *Discovery* klicken.

Es wird eine Liste der vorhandenen Low-Level-Discovery-Regeln angezeigt. Außerdem ist es möglich, alle Discovery-Regeln unabhängig vom Host oder alle Discovery-Regeln einer bestimmten Hostgruppe anzuzeigen, indem die Filtereinstellungen geändert werden.



Angezeigte Daten:

Spalte	Beschreibung
Host	Der sichtbare Hostname wird angezeigt. Ein Klick auf den Hostnamen öffnet das Konfigurationsformular des Hosts. Falls kein sichtbarer Hostname vorhanden ist, wird der technische Hostname angezeigt.
Name	Der Name der Regel wird als blauer Link angezeigt. Ein Klick auf den Regelnamen öffnet das Konfigurationsformular der Low-Level-Discovery-Regel. Wenn die Discovery-Regel zu einer Vorlage gehört, wird der Vorlagenname vor dem Regelnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Regelliste auf Vorlagenebene.
Items	Ein Link zur Liste der Datenpunkt-Prototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunkt-Prototypen wird grau dargestellt.
Triggers	Ein Link zur Liste der Auslöser-Prototypen wird angezeigt. Die Anzahl der vorhandenen Auslöser-Prototypen wird grau dargestellt.
Graphs	Ein Link zur Liste der Graph-Prototypen wird angezeigt. Die Anzahl der vorhandenen Graph-Prototypen wird grau dargestellt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau dargestellt.
Key	Der für die Discovery verwendete Datenpunktschlüssel wird angezeigt.
Interval	Die Häufigkeit der Discovery-Ausführung wird angezeigt. Beachten Sie, dass die Discovery auch sofort ausgeführt werden kann, indem Sie unterhalb der Liste auf die Schaltfläche *Execute now* klicken.
Type	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix-Agent, SNMP-Agent usw.).
Status	Der Status der Discovery-Regel wird angezeigt – *Enabled*, *Disabled* oder *Not supported*. Durch Klicken auf den Status können Sie ihn ändern – von Enabled zu Disabled (und zurück) sowie von Not supported zu Disabled (und zurück).
Info	Wenn alles in Ordnung ist, wird in dieser Spalte kein Symbol angezeigt. Im Fehlerfall wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Fehlerbeschreibung anzuzeigen.

Um eine neue Low-Level-Discovery-Regel zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create discovery rule*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

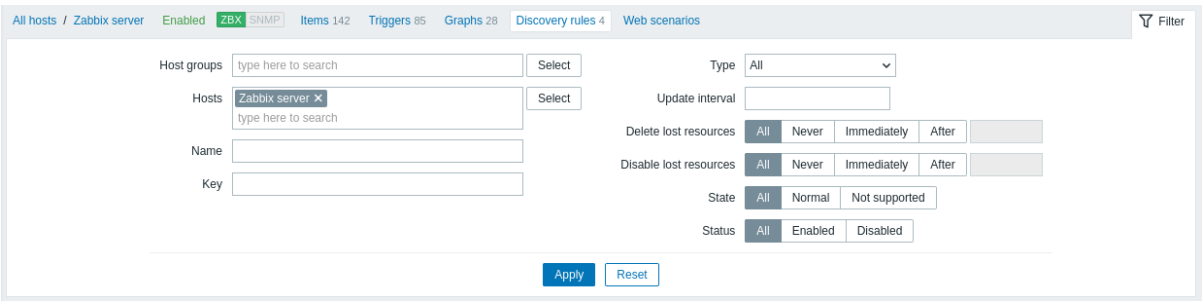
- *Aktivieren* - den Status der Low-Level-Discovery-Regel auf *Aktiviert* ändern.
- *Deaktivieren* - den Status der Low-Level-Discovery-Regel auf *Deaktiviert* ändern.
- *Jetzt ausführen* - die Discovery basierend auf den Discovery-Regeln sofort durchführen. Siehe weitere Details. Beachten Sie, dass bei sofortiger Durchführung der Discovery der Konfigurations-Cache nicht aktualisiert wird; daher spiegelt das Ergebnis keine sehr aktuellen Änderungen an der Konfiguration der Discovery-Regeln wider.
- *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.

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.

Der Link *Filter* ist oberhalb der Liste der Discovery-Regeln verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Discovery-Regeln nach Hostgruppe, Host, Name, Datenpunktschlüssel, Datenpunkttyp und anderen Parametern filtern können.

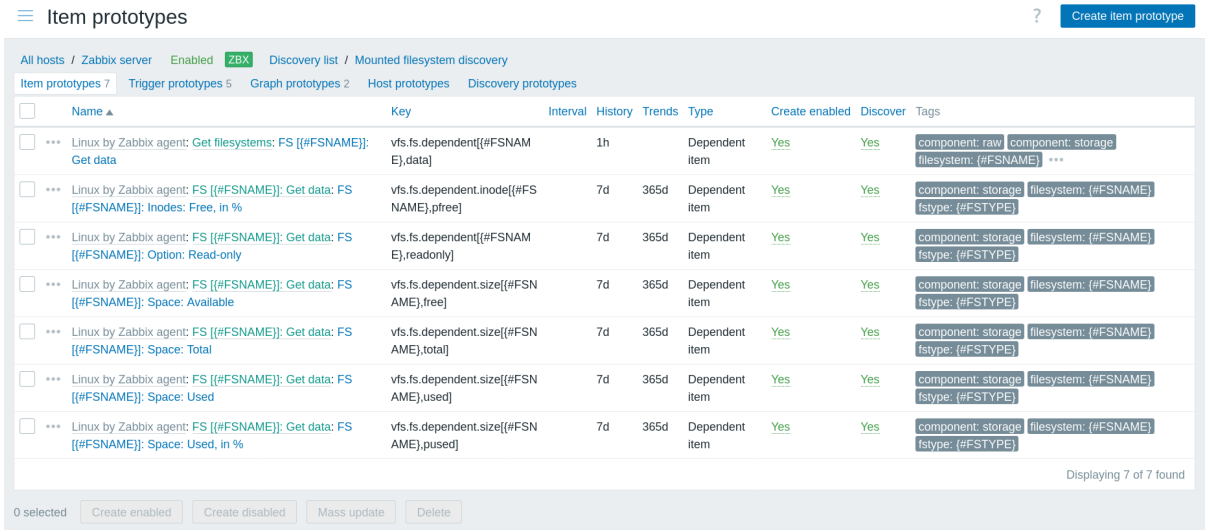


Parameter	Beschreibung
Host groups	Nach einer oder mehreren Hostgruppen filtern. Die Angabe einer übergeordneten Hostgruppe wählt implizit auch alle untergeordneten Hostgruppen aus.
Hosts	Nach einem oder mehreren Hosts filtern.
Name	Nach dem Namen der Discovery-Regel filtern.
Key	Nach dem Schlüssel des Discovery-Datenpunkts filtern.
Type	Nach dem Typ des Discovery-Datenpunkts filtern.
Update interval	Nach Aktualisierungsintervall filtern. Nicht verfügbar für Zabbix-Trapper und abhängige Datenpunkte.
Delete lost resources	Nach dem Zeitraum für *Delete lost resources* filtern.
Disable lost resources	Nach dem Zeitraum für *Disable lost resources* filtern.
SNMP OID	Nach SNMP OID filtern. Nur verfügbar, wenn *SNMP agent* als Typ ausgewählt ist.
State	Nach dem Status der Discovery-Regel filtern (Alle/Normal/Nicht unterstützt).
Status	Nach dem Status der Discovery-Regel filtern (Alle/Aktiviert/Deaktiviert).

1 Datenpunkt-Prototypen

Übersicht

In diesem Abschnitt werden die Datenpunkt-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Datenpunkt-Prototypen sind die Grundlage für echte Host-Datenpunkte, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Column	Description
Name	Name des Datenpunkt-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Datenpunkt-Prototyps. Wenn der Datenpunkt-Prototyp zu einer Vorlage gehört, wird der Vorlagenname vor dem Regelnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Datenpunkt-Prototypen auf Vorlagenebene. 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	Es wird angezeigt, wie viele Tage die Verlaufsdaten des Datenpunkts aufbewahrt werden.
Trends	Es wird angezeigt, wie viele Tage die Trenddaten des Datenpunkts aufbewahrt werden.
Type	Der Typ des Datenpunkt-Prototyps wird angezeigt (Zabbix-Agent, SNMP-Agent, einfache Prüfung usw.).
Create enabled	Den Datenpunkt basierend auf diesem Prototyp erstellen als: **Yes** - aktiviert **No** - deaktiviert. Sie können zwischen „Yes“ und „No“ wechseln, indem Sie darauf klicken.

Column	Description
Discover	Den Datenpunkt basierend auf diesem Prototyp entdecken: **Yes** - entdecken **No** - nicht entdecken. Sie können zwischen „Yes“ und „No“ wechseln, indem Sie darauf klicken.
Tags	Tags des Datenpunkt-Prototyps werden angezeigt.

Um einen neuen Datenpunkt-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create item prototype*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

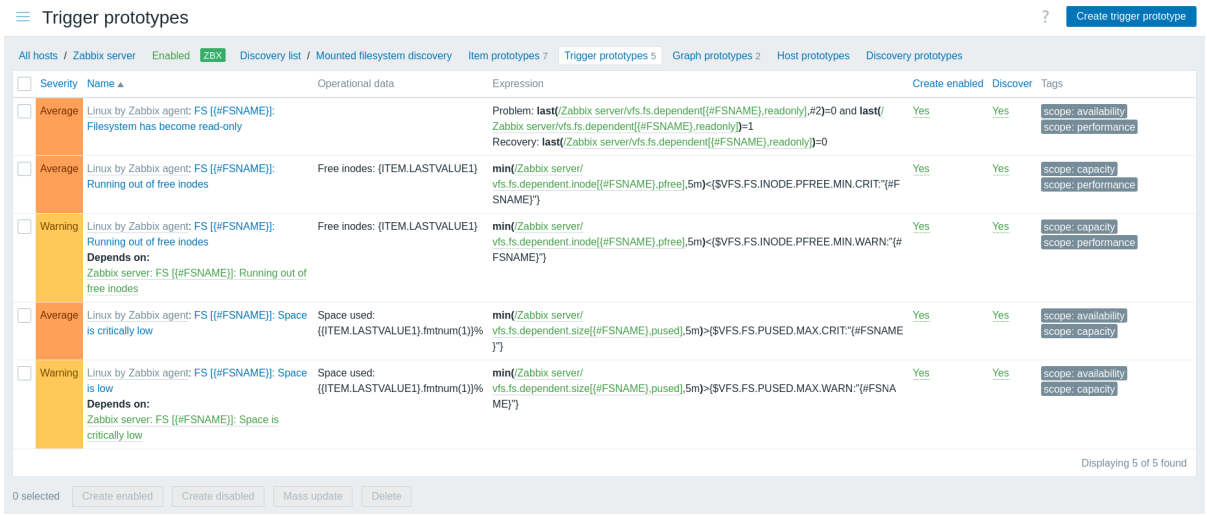
- *Aktiviert erstellen* - diese Datenpunkte als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Datenpunkte als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Datenpunktprototypen gesammelt aktualisieren
- *Löschen* - diese Datenpunktprototypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunktprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

2 Auslöser-Prototypen

Übersicht

In diesem Abschnitt werden die Auslöser-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Auslöser-Prototypen sind die Grundlage für echte Host-Auslöser, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Auslöser-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Auslöser-Prototyps. Wenn der Auslöser-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Auslösernamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Auslöser-Prototypen auf der Ebene der verknüpften Vorlage.
Betriebsdaten	Das Format der Betriebsdaten des Auslösers wird angezeigt und enthält beliebige Zeichenfolgen und Makros, die in *Monitoring* → *Probleme* dynamisch aufgelöst werden.
Aktiviert erstellen	Den Auslöser basierend auf diesem Prototyp erstellen als: **Ja** - aktiviert **Nein** - deaktiviert. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Entdecken	Den Auslöser basierend auf diesem Prototyp entdecken: **Ja** - entdecken **Nein** - nicht entdecken. Sie können zwischen „Ja“ und „Nein“ wechseln, indem Sie darauf klicken.
Tags	Tags des Auslöser-Prototyps werden angezeigt.

Um einen neuen Auslöser-Prototyp zu konfigurieren, klicken Sie auf die Schaltfläche *Auslöser-Prototyp erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

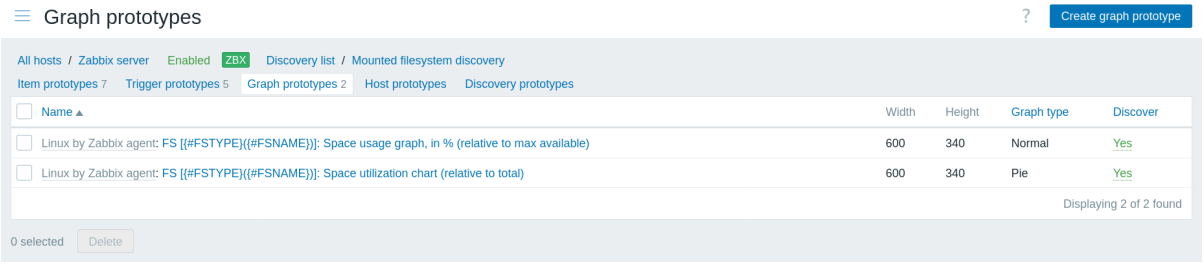
- *Aktiviert erstellen* - diese Auslöser als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Auslöser als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Auslöser-Prototypen gesammelt aktualisieren
- *Löschen* - diese Auslöser-Prototypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den entsprechenden Auslöser-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

3 Graph-Prototypen

Übersicht

In diesem Abschnitt werden die Graph-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Graph-Prototypen sind die Grundlage für echte Host-Graphen, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Column	Description
Name	Der Name des Graph-Prototyps wird als blauer Link angezeigt. Ein Klick auf den Namen öffnet das Konfigurationsformular des Graph-Prototyps. Wenn der Graph-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Graphnamen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Graph-Prototypen auf der Ebene der verknüpften Vorlage.
Width	Die Breite des Graph-Prototyps wird angezeigt.
Height	Die Höhe des Graph-Prototyps wird angezeigt.
Type	Der Typ des Graph-Prototyps wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.
Discover	Den Graphen auf Basis dieses Prototyps ermitteln: **Yes** - ermitteln **No** - nicht ermitteln. Sie können zwischen „Yes“ und „No“ wechseln, indem Sie darauf klicken.

Um einen neuen Graph-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create graph prototype*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

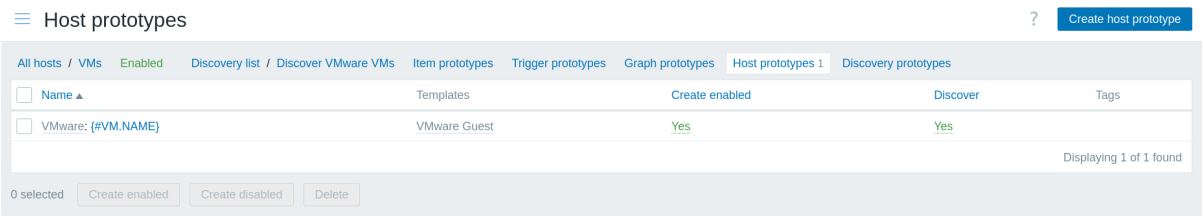
- *Löschen* - diese Graphprototypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Graphprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

4 Host-Prototypen

Übersicht

In diesem Abschnitt werden die Host-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Host-Prototypen sind die Grundlage für echte Hosts, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Column	Description
Name	Name des Host-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Host-Prototyps. Wenn der Host-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Host-Namen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Host-Prototypen auf der Ebene der verknüpften Vorlage.
Templates	Vorlagen des Host-Prototyps werden angezeigt.
Create enabled	Den Host basierend auf diesem Prototyp erstellen als: **Yes** - aktiviert **No** - deaktiviert. Sie können zwischen „Yes“ und „No“ wechseln, indem Sie darauf klicken.
Discover	Den Host basierend auf diesem Prototyp entdecken: **Yes** - entdecken **No** - nicht entdecken. Sie können zwischen „Yes“ und „No“ wechseln, indem Sie darauf klicken.
Tags	Tags des Host-Prototyps werden angezeigt.

Um einen neuen Host-Prototyp zu konfigurieren, klicken Sie auf die Schaltfläche *Create host prototype* oben rechts.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

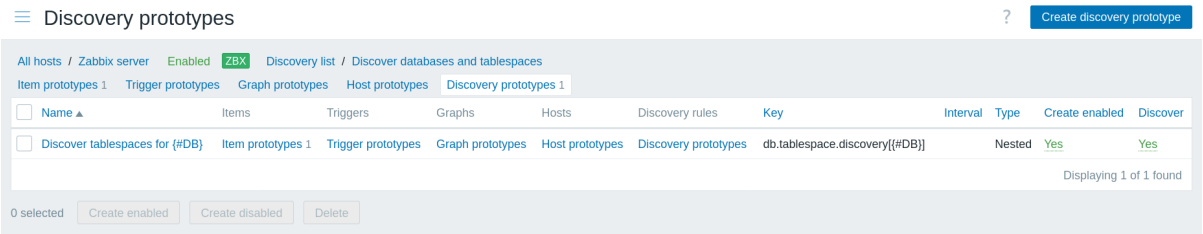
- *Aktiviert erstellen* - diese Hosts als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Hosts als *Deaktiviert* erstellen
- *Löschen* - diese Host-Prototypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Host-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

5 Discovery-Prototypen

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	Name des Discovery-Prototyps, angezeigt als blauer Link. Ein Klick auf den Namen öffnet das Konfigurationsformular des Discovery-Prototyps. Wenn der Discovery-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Host-Namen als grauer Link angezeigt. Ein Klick auf den Vorlagen-Link öffnet die Liste der Discovery-Prototypen auf der Ebene der verknüpften Vorlage.
Items	Ein Link zur Liste der Datenpunkt-Prototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunkt-Prototypen wird grau angezeigt.

Spalte	Beschreibung
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.
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. Beachten Sie, dass die Discovery auch sofort durch Klicken auf die Schaltfläche *Execute now* unterhalb der Liste ausgeführt werden kann.
Type	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix Agent, SNMP-Agent usw.).
Create enabled	Der Erstellungsstatus 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 zurück).
Discover	Der Discovery-Status des Discovery-Prototyps wird angezeigt – entdecken (*Yes*) oder nicht entdecken (*No*). Durch Klicken auf den Status können Sie ihn ändern – von Yes zu No (und zurück).

Um einen neuen Discovery-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create discovery prototype*.

Optionen zur Massenbearbeitung

Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Aktiviert erstellen* - diese Erkennungsprototypen als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Erkennungsprototypen als *Deaktiviert* erstellen
- *Löschen* - diese Erkennungsprototypen löschen

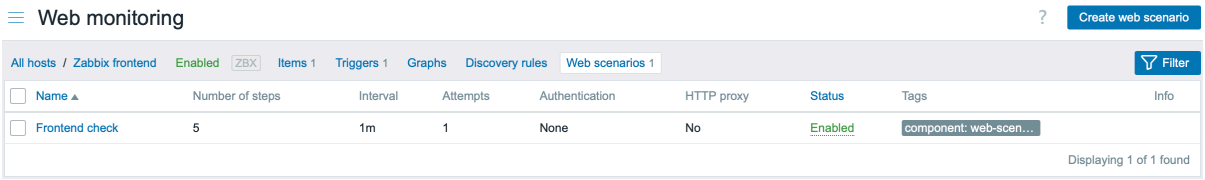
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Erkennungsprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

5 Web-Szenarien

Übersicht

Die Liste der Webszenarien für einen Host kann über *Datensammlung* → *Hosts* aufgerufen werden, indem Sie beim jeweiligen Host auf *Web* klicken.

Eine Liste der vorhandenen Webszenarien wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Webszenarios. Durch Klicken auf den Namen des Webszenarios wird das Konfigurationsformular des Webszenarios geöffnet. Wenn das Webszenario des Hosts zu einer Vorlage gehört, wird der Vorlagenname vor dem Namen des Webszenarios als grauer Link angezeigt. Durch Klicken auf den Vorlagen-Link wird die Liste der Webszenarien auf Vorlagenebene geöffnet.
Anzahl der Schritte	Die Anzahl der Schritte, die das Szenario enthält.
Aktualisierungsintervall	Wie oft das Szenario ausgeführt wird.
Versuche	Wie viele Versuche zur Ausführung der Schritte des Webszenarios durchgeführt werden.
Authentifizierung	Die Authentifizierungsmethode wird angezeigt – Basic, NTLM, Kerberos, Digest oder Keine.
HTTP-Proxy	Zeigt den HTTP-Proxy an oder „Nein“, wenn keiner verwendet wird.

Spalte	Beschreibung
Status	Der Status des Webszenarios wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Tags	Die Tags des Webszenarios werden angezeigt. Es können bis zu drei Tags (Name:Wert-Paare) angezeigt werden. Wenn mehr Tags vorhanden sind, wird ein „...“-Link angezeigt, über den beim Überfahren mit der Maus alle Tags angezeigt werden können.
Info	Wenn alles korrekt funktioniert, wird in dieser Spalte kein Symbol angezeigt. Im Fehlerfall wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Fehlerbeschreibung anzuzeigen.

Um ein neues Webszenario zu konfigurieren, klicken Sie auf die Schaltfläche *Webszenario erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

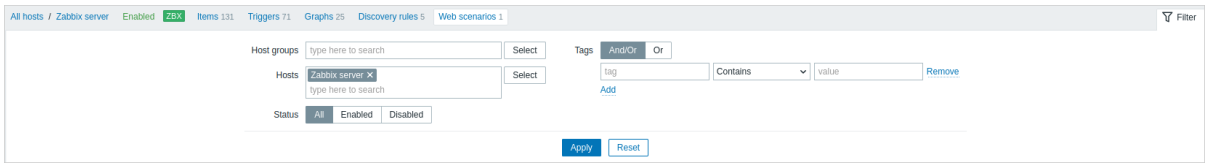
- *Aktivieren* - den Szenariostatus auf *Aktiviert* ändern
- *Deaktivieren* - den Szenariostatus auf *Deaktiviert* ändern
- *Verlauf und Trends löschen* - Verlaufs- und Trenddaten für die Szenarien löschen
- *Löschen* - die Webszenarien löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Webszenarien und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Szenarien anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Webszenarien verfügbar. Ein Klick darauf öffnet einen Filter, in dem Sie die gewünschten Filterkriterien angeben können.



Parameter	Beschreibung
Host-Gruppen	Filtern nach einer oder mehreren Host-Gruppen. Wenn eine übergeordnete Host-Gruppe angegeben wird, werden implizit auch alle darunterliegenden Host-Gruppen ausgewählt.
Hosts	Filtern nach einem oder mehreren Hosts. Wenn oben bereits Host-Gruppen ausgewählt wurden, ist die Host-Auswahl auf diese Gruppen beschränkt.
Status	Filtern nach dem Status des Webszenarios.

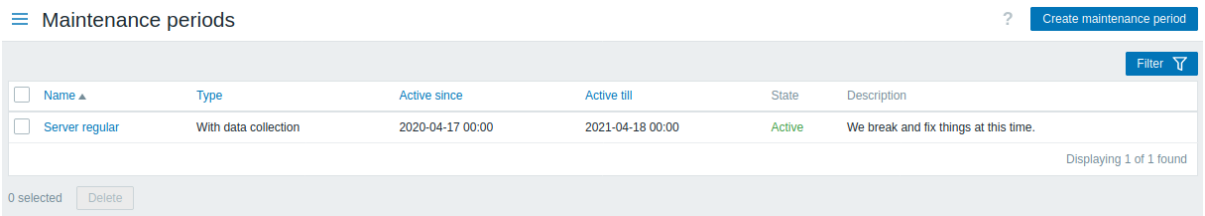
Parameter	Beschreibung
Tags	Filtern nach Tag-Namen und -Wert des Webszenarios. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen erfolgt immer unter Beachtung der Groß-/Kleinschreibung. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (Groß-/Kleinschreibung beachten); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, ohne Beachtung der Groß-/Kleinschreibung); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (Groß-/Kleinschreibung beachten); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, ohne Beachtung 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 genügt, wenn eine Bedingung erfüllt ist.

5 Wartung

Übersicht

Im Abschnitt *Datensammlung* → *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. Ein Klick auf den Namen des Wartungszeitraums öffnet das Konfigurationsformular des Wartungszeitraums.
Typ	Der Typ der Wartung wird angezeigt: *Mit Datensammlung* oder *Ohne Datensammlung*
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, zu denen die Ausführung von Wartungszeiträumen nicht mehr aktiv ist.
Status	Der Status des Wartungszeitraums: **Bevorstehend** - 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 auf die Schaltfläche *Wartungszeitraum erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Eine Schaltfläche unterhalb der Liste bietet eine Option zur Massenbearbeitung:

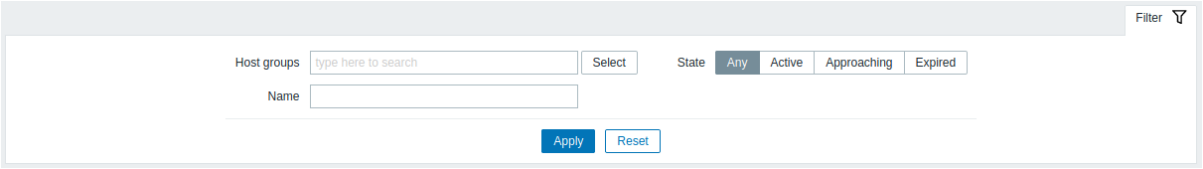
- *Löschen* - die Wartungszeiträume löschen

Um diese Option zu verwenden, markieren Sie die Kontrollkästchen vor den entsprechenden Wartungszeiträumen und klicken Sie auf *Löschen*.

Filter verwenden

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 eingeblendet, mit dem Sie Wartungszeiträume nach Hostgruppe, Name und Status filtern können.



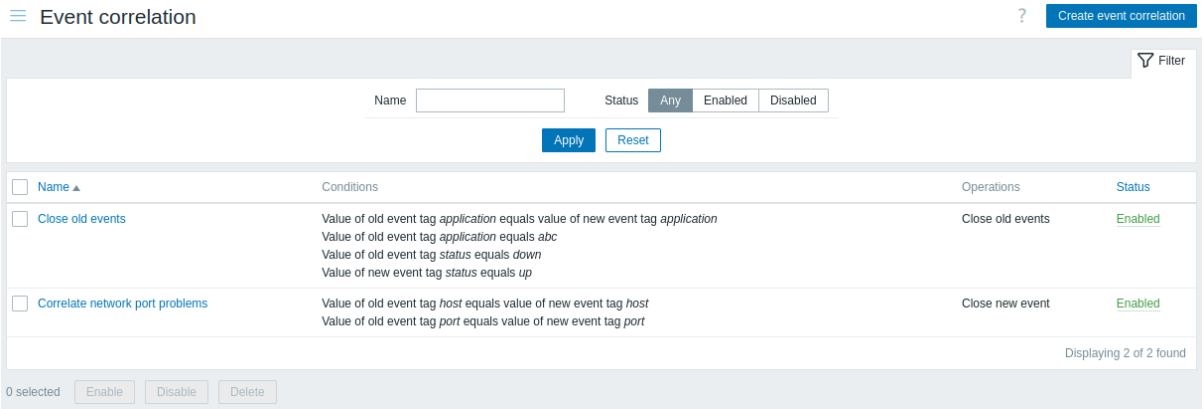
Berechnung von Warteschlangen während der Wartung

Attention:
Der Zabbix Proxy kennt keine Wartungszeiträume; siehe Berechnung von Warteschlangen während der Wartung für Details.

6 Ereigniskorrelation

Übersicht

Im Abschnitt *Datenerfassung* → *Ereigniskorrelation* können Benutzer globale Korrelations-Regeln für Zabbix-Ereignisse konfigurieren und verwalten.



Angezeigte Daten:

Column	Description
Name	Name der Korrelationsregel. Durch Klicken auf den Namen der Korrelationsregel wird das Konfigurationsformular der Regel geöffnet.
Conditions	Die Bedingungen der Korrelationsregel werden angezeigt.
Operations	Die Operationen der Korrelationsregel werden angezeigt.
Status	Der Status der Korrelationsregel wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.

Um eine neue Korrelationsregel zu konfigurieren, klicken Sie auf die Schaltfläche *Ereigniskorrelation erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

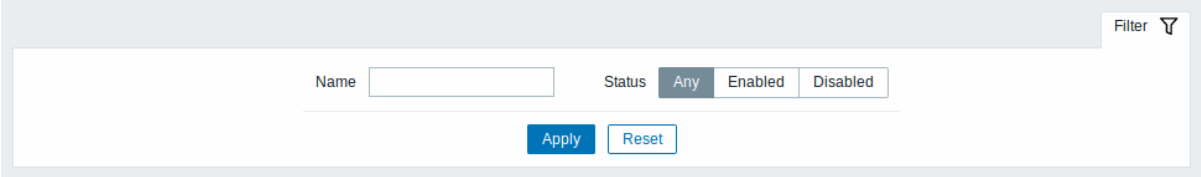
- *Aktivieren* - den Status der Korrelationsregel auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Korrelationsregel auf *Deaktiviert* ändern
- *Löschen* - die Korrelationsregeln löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Korrelationsregeln und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Korrelationsregeln anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Korrelationsregeln verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Korrelationsregeln nach Name und Status filtern können.

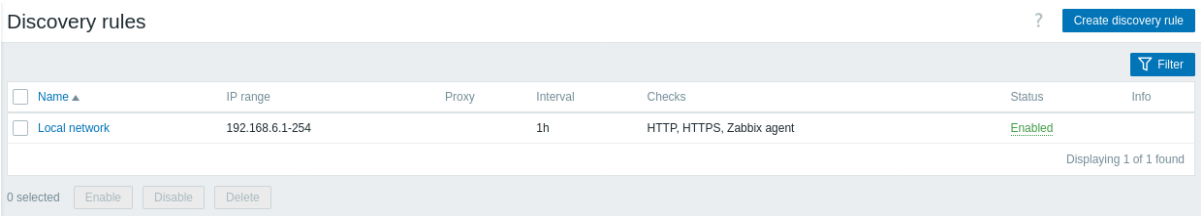


7 Discovery

Übersicht

Im Abschnitt *Datensammlung → Discovery* können Benutzer Discovery-Regeln konfigurieren und verwalten.

Eine Liste der vorhandenen Discovery-Regeln mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Discovery-Regel. Durch Klicken auf den Namen der Discovery-Regel wird das Konfigurationsformular der Discovery-Regel geöffnet.
IP range	Der für das Netzwerk-Scanning zu verwendende IP-Adressbereich wird angezeigt.
Proxy	Der Name des Proxy wird angezeigt, wenn die Discovery durch den Proxy durchgeführt wird.
Interval	Die Häufigkeit der Durchführung der Discovery wird angezeigt.
Checks	Die für die Discovery verwendeten Prüfungsarten werden angezeigt.
Status	Der Status der Discovery-Regel wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Info	Wenn alles korrekt funktioniert, wird in dieser Spalte nichts angezeigt. Im Fehlerfall wird ein rotes Info-Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um eine Kurzinfo mit der Fehlerbeschreibung anzuzeigen.

Um eine neue Discovery-Regel zu konfigurieren, klicken Sie auf die Schaltfläche *Discovery-Regel erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

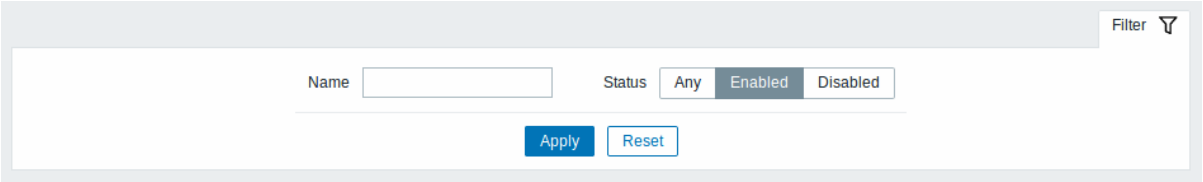
- *Aktivieren* - den Status der Discovery-Regel auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Discovery-Regel auf *Deaktiviert* ändern
- *Löschen* - die Discovery-Regeln löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Discovery-Regeln und klicken Sie dann auf die gewünschte Schaltfläche.

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.

Der Link *Filter* ist oberhalb der Liste der Discovery-Regeln verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Discovery-Regeln nach Name und Status filtern können.



7 Warnmeldungen

Übersicht

Dieses Menü enthält Abschnitte, die mit der Konfiguration von Warnmeldungen in Zabbix zusammenhängen.

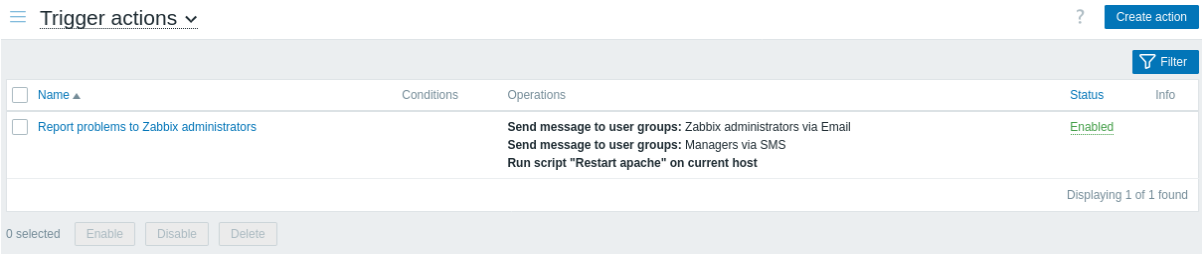
1 Aktionen

Übersicht

Im Abschnitt *Benachrichtigungen* → *Aktionen* können Benutzer Aktionen konfigurieren und verwalten.

Die angezeigten Aktionen sind Aktionen, die der ausgewählten Ereignisquelle zugewiesen sind (Auslöser, Services, Discovery, Autoregistrierung, interne Aktionen).

Um zu einer anderen Ereignisquelle zu wechseln, klicken Sie im Menüabschnitt *Benachrichtigungen* auf *Aktionen*. Es ist auch möglich, zwischen Quellen über die Titelauswahl in der oberen linken Ecke zu wechseln.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Aktion. Durch Klicken auf den Aktionsnamen wird das Konfigurationsformular der Aktion 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 Details dazu, was passiert, wenn eine Aktion während einer laufenden Eskalation deaktiviert wird, finden Sie im Abschnitt Eskalationen.
Info	Wenn alles korrekt funktioniert, wird in dieser Spalte kein Symbol angezeigt. Im Fehlerfall, z. B. bei fehlenden Aktionsoperationen oder Bedingungen nach Upgrade/Low-Level-Discovery, wird ein Warnsymbol angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Fehlerbeschreibung anzuzeigen.

Um eine neue Aktion zu konfigurieren, klicken Sie auf die Schaltfläche *Aktion erstellen* in der oberen rechten Ecke.

Für Benutzer ohne Super-Admin-Rechte werden Aktionen entsprechend den Berechtigungseinstellungen angezeigt. Das bedeutet, dass ein Benutzer ohne Super-Admin-Rechte in einigen Fällen aufgrund bestimmter Berechtigungseinschränkungen nicht die vollständige Aktionsliste anzeigen kann. Eine Aktion wird einem Benutzer ohne Super-Admin-Rechte angezeigt, wenn die folgenden Bedingungen erfüllt sind:

- Der Benutzer hat Lese-/Schreibzugriff auf Hostgruppen, Hosts, Vorlagen und Auslöser in Aktionsbedingungen
- Der Benutzer hat Lese-/Schreibzugriff auf Hostgruppen, Hosts und Vorlagen in Aktionsoperationen, Wiederherstellungsoperationen und Aktualisierungsoperationen
- Der Benutzer hat Lesezugriff auf Benutzergruppen und Benutzer in Aktionsoperationen, Wiederherstellungsoperationen und Aktualisierungsoperationen

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

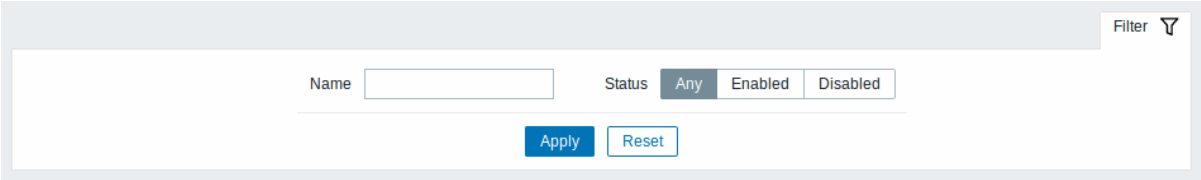
- *Aktivieren* - den Aktionsstatus auf *Aktiviert* ändern
- *Deaktivieren* - den Aktionsstatus auf *Deaktiviert* ändern
- *Löschen* - die Aktionen löschen

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.



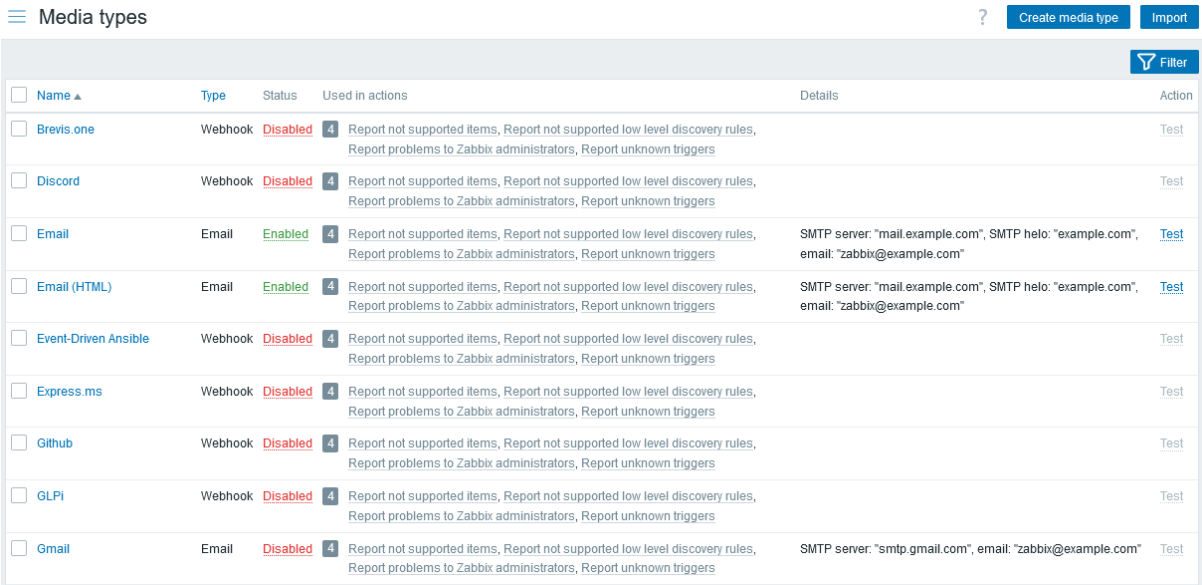
2 Medientypen

Übersicht

Im Abschnitt *Warnungen* → *Medientypen* können Benutzer Medientyp-Informationen konfigurieren und verwalten.

Medientyp-Informationen enthalten allgemeine Anweisungen für die Verwendung eines Mediums als Zustellkanal für Benachrichtigungen. Spezifische Details, wie z. B. die einzelnen E-Mail-Adressen, an die eine Benachrichtigung gesendet werden soll, werden bei den jeweiligen Benutzern hinterlegt.

Eine Liste der vorhandenen Medientypen mit ihren Details wird angezeigt.



Angezeigte Daten:

Column	Description
Name	Name des Medientyps. Ein Klick auf den Namen öffnet das Konfigurationsformular des Medientyps.

Column	Description
Type	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.
Used in actions	Aktionen, in denen der Medientyp verwendet wird, werden angezeigt, zusammen mit der Gesamtzahl dieser Aktionen. Ein Klick auf den Aktionsnamen öffnet das Aktions-Konfigurationsformular. Wenn der Benutzer keine Berechtigungen für die Aktion hat, ist der Name nicht anklickbar.
Details	Detaillierte Informationen zum Medientyp werden angezeigt.
Actions	Die folgende Aktion ist verfügbar: **Test** – klicken Sie hier, um ein Testformular zu öffnen, in dem Sie Medientyp-Parameter eingeben können (z. B. eine Empfängeradresse mit Testbetreff und Nachrichtentext) 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 auf die Schaltfläche *Medientyp erstellen* in der oberen rechten Ecke.

Um einen Medientyp zu importieren, klicken Sie auf die Schaltfläche *Importieren* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

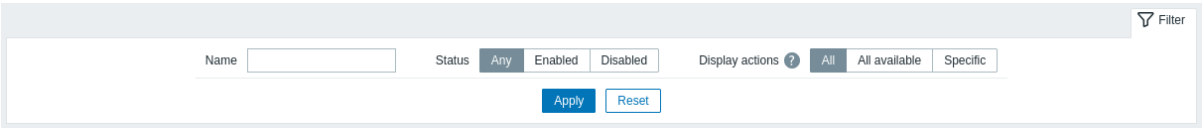
- *Aktivieren* - den Status des Medientyps auf *Aktiviert* ändern
- *Deaktivieren* - den Status des Medientyps auf *Deaktiviert* ändern
- *Exportieren* - die Medientypen in eine YAML-, XML- oder JSON-Datei exportieren
- *Löschen* - die Medientypen löschen

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 Medientypen anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Medientypen verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Medientypen 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 Medientypverwendung anzuzeigen (definiert durch den Parameter *Send to media type* in den Details zur Operation einer Aktion).



3 Skripte

Übersicht

Im Abschnitt *Benachrichtigungen* > *Skripte* können benutzerdefinierte globale Skripte konfiguriert und verwaltet werden.

Globale Skripte stehen je nach konfiguriertem Geltungsbereich und den Benutzerberechtigungen zur Ausführung zur Verfügung:

- über das Host-Menü an verschiedenen Stellen im Frontend (*Dashboard*, *Probleme*, *Letzte Daten*, *Karten* usw.)
- über das Ereignismenü
- sie können als Aktionsoperation ausgeführt werden

Die Skripte werden auf dem Zabbix Agent, dem Zabbix Server (Proxy) oder nur auf dem Zabbix Server ausgeführt. Siehe auch Befehlsausführung.

Sowohl auf dem Zabbix Agent als auch auf dem Zabbix Proxy sind Remote-Skripte standardmäßig deaktiviert. Sie können wie folgt aktiviert werden:

- Für Remote-Befehle, die auf dem Zabbix Agent ausgeführt werden:
  - durch Hinzufügen eines Parameters `AllowKey=system.run[<command>,\*]` für jeden erlaubten Befehl in der Agent-Konfiguration; * steht für den Modus wait und nowait;
- Für Remote-Befehle, die auf dem Zabbix Proxy ausgeführt werden:

- **Warnung: Es ist nicht erforderlich, Remote-Befehle auf dem Zabbix Proxy zu aktivieren, wenn Remote-Befehle auf dem Zabbix Agent ausgeführt werden, der vom Zabbix Proxy überwacht wird.** Falls es jedoch erforderlich ist, Remote-Befehle auf dem Zabbix Proxy auszuführen, setzen Sie den Parameter *EnableRemoteCommands* in der Proxy-Konfiguration auf '1'.

Die Ausführung globaler Skripte auf dem Zabbix Server kann durch Setzen von `EnableGlobalScripts=0` in der Server-Konfiguration deaktiviert werden. Bei Neuinstallationen ist die Ausführung globaler Skripte auf dem Zabbix Server seit Zabbix 7.0 standardmäßig deaktiviert.

Eine Liste der vorhandenen Skripte mit ihren Details wird angezeigt.

Angezeigte Daten:

Column	Description
Name	Name des Skripts. Ein Klick auf den Skriptnamen öffnet das Konfigurationsformular des Skripts.
Scope	Geltungsbereich des Skripts – Aktionsoperation, manuelle Host-Aktion oder manuelle Ereignisaktion. Diese Einstellung bestimmt, wo das Skript verfügbar ist.
Used in actions	Alle Aktionen, in denen das Skript verwendet wird, werden angezeigt, zusammen mit der Gesamtzahl dieser Aktionen. Ein Klick auf den Aktionsnamen öffnet das Aktionskonfigurationsformular. Wenn der Benutzer keine Berechtigungen für die Aktion hat, ist der Name nicht anklickbar.
Type	Der Skripttyp wird angezeigt – *URL*, *Webhook*, *Skript*, *SSH-*, *Telnet-* oder *IPMI*-Befehl.
Execute on	Es wird angezeigt, ob das Skript auf dem Zabbix Agent, dem Zabbix Proxy oder Server oder nur auf dem Zabbix Server ausgeführt wird.
Commands	Alle Befehle, die innerhalb des Skripts ausgeführt werden, werden angezeigt. Für webhooks wird hier nichts angezeigt.
User group	Die Benutzergruppe, für die das Skript verfügbar ist, wird angezeigt (oder *All* für alle Benutzergruppen).
Host group	Die Host-Gruppe, für die das Skript verfügbar ist, wird angezeigt (oder *All* für alle Host-Gruppen).
Host access	Die Berechtigungsstufe für die Host-Gruppe wird angezeigt – *Read* oder *Write*. 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 zur Massenbearbeitung

Eine Schaltfläche unterhalb der Liste bietet eine Option zur Massenbearbeitung:

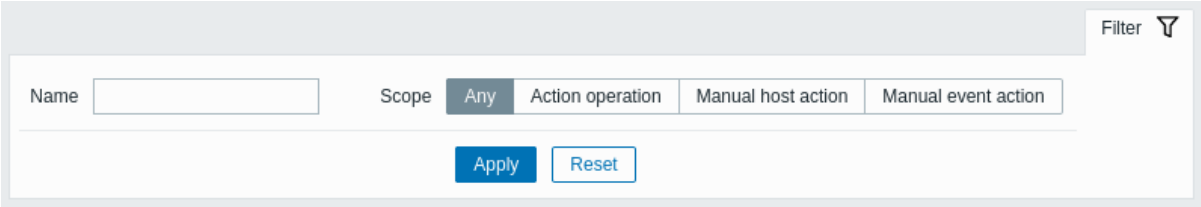
- *Löschen* - die Skripte löschen

Um diese Option zu verwenden, markieren Sie die Kontrollkästchen vor den entsprechenden Skripten und klicken Sie auf *Löschen*.

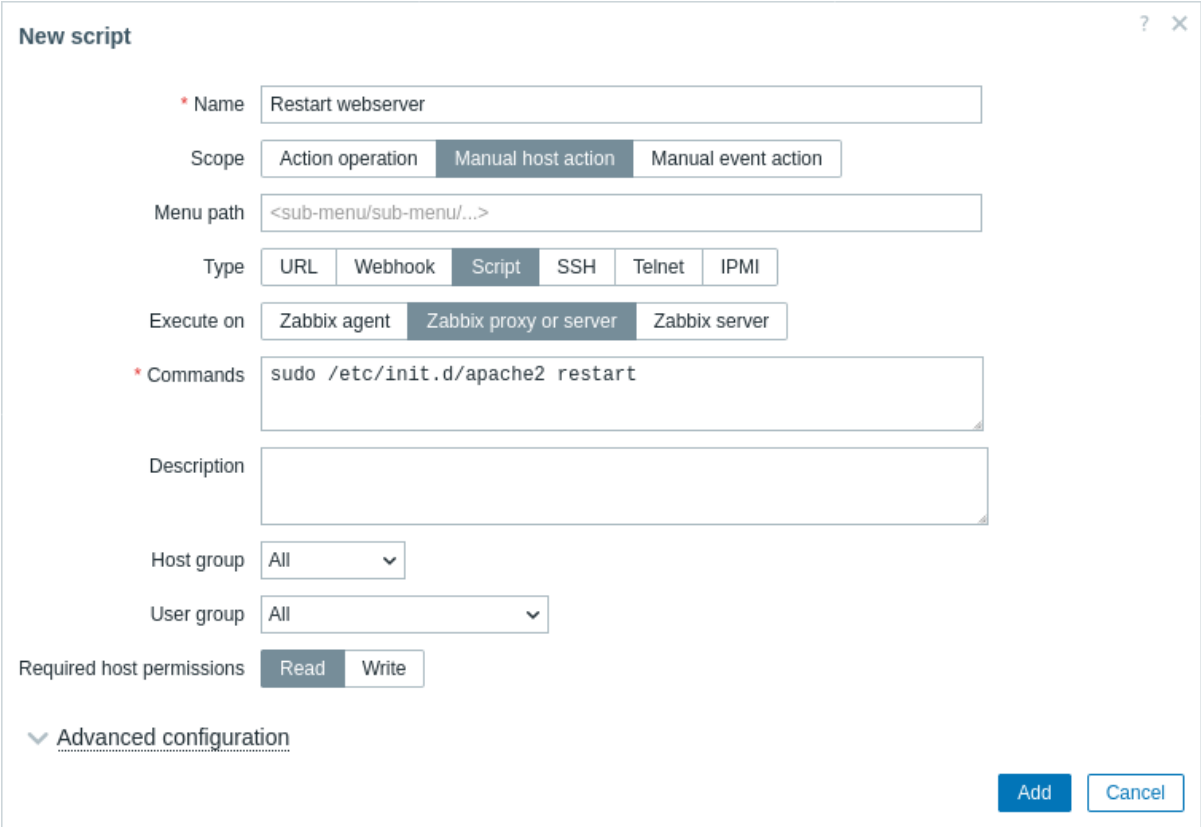
Filter verwenden

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 eingeblendet, mit dem Sie Skripte nach Name und Geltungsbereich filtern können.



Konfiguration eines globalen Skripts



Note:
Es wird empfohlen, in globalen Skripten statt einfacher Makros Makrofunktionen zu verwenden, um die Sicherheit zu erhöhen, da Makros nicht automatisch maskiert werden.

Skriptattribute:

Parameter	Beschreibung
Name	Eindeutiger Name des Skripts. Z. B. `Clear /tmp filesystem`

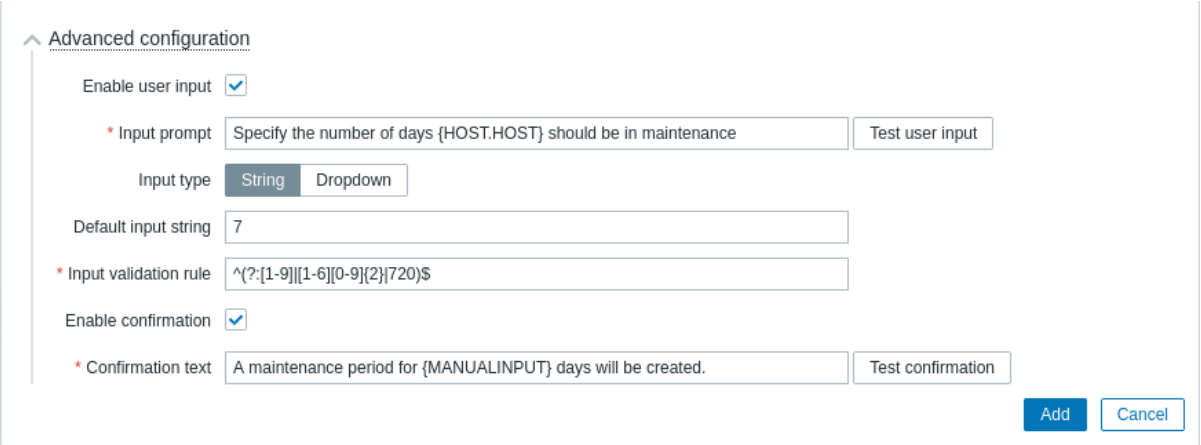
Parameter	Beschreibung
Scope	Geltungsbereich des Skripts – Aktionsoperation, manuelle Host-Aktion oder manuelle Ereignisaktion. Diese Einstellung bestimmt, wo das Skript verwendet werden kann – in Remote-Befehlen von Aktionsoperationen, im Host-Menü bzw. im Ereignismenü. Wenn der Geltungsbereich auf „Aktionsoperation“ gesetzt ist, ist das Skript für alle Benutzer mit Zugriff auf *Benachrichtigungen* > *Aktionen* verfügbar. Wird ein Skript tatsächlich in einer Aktion verwendet, kann sein Geltungsbereich nicht von „Aktionsoperation“ 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 Skript-Geltungsbereich eine Aktionsoperation ist, da Aktionsoperationen automatisch ausgeführt werden. Mit dem Makro {MANUALINPUT} kann bei der Ausführung des Skripts eine manuelle Eingabe angegeben werden. Es wird für Skripte mit manueller Host-Aktion und manueller Ereignisaktion 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 möglicherweise zu einem Wert mit Leerzeichen aufgelöst werden (zum Beispiel Host-Name), bei Bedarf in Anführungszeichen setzen müssen.
Menu path	Der gewünschte Menüpfad zum Skript. Zum Beispiel wird das Skript bei `Default` oder `Default/` im entsprechenden Verzeichnis angezeigt. Menüs können verschachtelt sein, z. B. `Main menu/Sub menu1/Sub menu2`. Beim Zugriff auf Skripte über das Host-/Ereignismenü in Monitoring-Bereichen werden sie entsprechend den angegebenen Verzeichnissen organisiert. Dieses Feld wird nur angezeigt, wenn als *Scope* „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ ausgewählt ist.
Type	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 als *Scope* „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ ausgewählt ist.
Skripttyp: **URL**	
URL	Geben Sie die URL für den Schnellzugriff aus dem Host-Menü oder Ereignismenü an. Makros und benutzerdefinierte Benutzermakros werden unterstützt. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Scope* oben). Verwenden Sie in diesem Feld das Makro {MANUALINPUT}, um bei der Ausführung des Skripts eine manuelle Eingabe angeben zu können, zum Beispiel: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Makrowerte dürfen nicht URL-kodiert sein.
Open in new window	Legt fest, ob die URL in einem neuen oder im selben Browser-Tab geöffnet werden soll.
Skripttyp: **Webhook**	
Parameters	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 *Scope* oben).
Script	Geben Sie den JavaScript-Code im modalen Editor ein, der sich öffnet, wenn Sie in das Parameterfeld oder auf das Stiftsymbol daneben klicken. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Scope* oben). Siehe auch: Medienkonfiguration Webhook, Zusätzliche JavaScript-Objekte.
Timeout	Zeitlimit für die JavaScript-Ausführung (1–60 s, Standard 30 s). Zeitsuffixe werden unterstützt, z. B. 30s, 1m.
Skripttyp: **Skript**	

Parameter	Beschreibung
Execute on	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 Datenpunkt system.run erlaubt ist) **Zabbix Proxy oder Server** – das Skript wird von Zabbix Proxy oder Server ausgeführt, je nachdem, ob der Host vom Proxy oder vom 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 Server-Parameter EnableGlobalScripts aktiviert sind. **Zabbix Server** – das Skript wird nur vom Zabbix Server ausgeführt. Diese Option ist nicht verfügbar, wenn globale Skripte durch den Server-Parameter EnableGlobalScripts deaktiviert sind.
Commands	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 *Scope* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Skripttyp: **SSH**	
Authentication method	Wählen Sie die Authentifizierungsmethode – Passwort oder öffentlicher Schlüssel.
Username	Geben Sie den Benutzernamen ein.
Password	Geben Sie das Passwort ein. Dieses Feld ist verfügbar, wenn als Authentifizierungsmethode „Passwort“ ausgewählt ist.
Public key file	Geben Sie den Pfad zur Datei mit dem öffentlichen Schlüssel ein. Dieses Feld ist verfügbar, wenn als Authentifizierungsmethode „Öffentlicher Schlüssel“ ausgewählt ist.
Private key file	Geben Sie den Pfad zur Datei mit dem privaten Schlüssel ein. Dieses Feld ist verfügbar, wenn als Authentifizierungsmethode „Öffentlicher Schlüssel“ ausgewählt ist.
Passphrase	Geben Sie die Passphrase ein. Dieses Feld ist verfügbar, wenn als Authentifizierungsmethode „Öffentlicher Schlüssel“ ausgewählt ist.
Port	Geben Sie den Port des entfernten SSH-Dienstes auf dem Ziel-Host ein, zu dem Zabbix eine Verbindung herstellt.
Commands	Geben Sie die Befehle ein. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Scope* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Skripttyp: **Telnet**	
Username	Geben Sie den Benutzernamen ein.
Password	Geben Sie das Passwort ein.
Port	Geben Sie den Port des entfernten Telnet-Dienstes auf dem Ziel-Host ein, zu dem Zabbix eine Verbindung herstellt.
Commands	Geben Sie die Befehle ein. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Scope* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Skripttyp: **IPMI**	
Command	Geben Sie den IPMI-Befehl ein. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Scope* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Description	Geben Sie eine Beschreibung für das Skript ein.
Host group	Wählen Sie die Host-Gruppe aus, für die das Skript verfügbar sein soll (oder *Alle* für alle Host-Gruppen).
User group	Wählen Sie die Benutzergruppe aus, für die das Skript verfügbar sein soll (oder *Alle* für alle Benutzergruppen). Dieses Feld wird nur angezeigt, wenn als *Scope* „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ ausgewählt ist.
Required host permissions	Wählen Sie die Berechtigungsstufe für die Host-Gruppe – *Lesen* oder *Schreiben*. Nur Benutzer mit der erforderlichen Berechtigungsstufe haben Zugriff auf die Ausführung des Skripts. Dieses Feld wird nur angezeigt, wenn als *Scope* „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ ausgewählt ist.

Parameter	Beschreibung
Advanced configuration	Klicken Sie auf die Bezeichnung *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen. Dieses Feld wird nur angezeigt, wenn als *Scope* „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ ausgewählt ist.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind in einem einklappbaren 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 Wert des Makros {MANUALINPUT} im Skript. Siehe auch: Manuelle Benutzereingabe.
Eingabeaufforderung	Geben Sie einen benutzerdefinierten Text ein, der zur benutzerdefinierten Benutzereingabe auffordert. Dieser Text wird oberhalb des Eingabefelds im Popup *Manuelle Eingabe* angezeigt. Um eine Vorschau des Popups *Manuelle Eingabe* anzuzeigen, klicken Sie auf *Benutzereingabe testen*. In der Vorschau können Sie auch prüfen, ob die Eingabezeichenfolge der Eingabevalidierungsregel entspricht (siehe Parameter unten). Die Unterstützung für Makros und Benutzermakros hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* in den allgemeinen Konfigurationsparametern des Skripts).
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 der Benutzereingabe als kommagetrennte Liste ein. Um eine leere Option in das Dropdown aufzunehmen, fügen Sie am Anfang, in der Mitte oder am Ende der Liste ein zusätzliches Komma ein. Dieses Feld wird nur angezeigt, wenn als *Eingabetyp* „Dropdown“ 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 als *Eingabetyp* „Zeichenfolge“ ausgewählt ist.
Eingabevalidierungsregel	Geben Sie einen regulären Ausdruck ein, um die Benutzereingabezeichenfolge zu validieren. Globale reguläre Ausdrücke werden unterstützt. Dieses Feld wird nur angezeigt, wenn als *Eingabetyp* „Zeichenfolge“ 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 (wie einem Neustartskript) oder solchen, die lange dauern können, nützlich sein.

Parameter	Beschreibung
Bestätigungstext	Geben Sie einen benutzerdefinierten Bestätigungstext für das mit dem obigen Kontrollkästchen aktivierte Bestätigungs-Popup ein (zum Beispiel *Das entfernte System wird neu gestartet. Sind Sie sicher?*). Um zu sehen, wie der Text aussehen wird, klicken Sie neben dem Feld auf *Bestätigung testen*. Makros und benutzerdefinierte Benutzermakros werden unterstützt. *Hinweis:* Die Makros werden beim Testen der Bestätigungsmeldung nicht erweitert.

Wenn sowohl eine manuelle Benutzereingabe als auch eine Bestätigungsmeldung konfiguriert sind, werden sie in aufeinanderfolgenden Popup-Fenstern angezeigt.

Manuelle Benutzereingabe

Die manuelle Benutzereingabe ermöglicht es, bei jeder Ausführung des Skripts einen benutzerdefinierten Parameter anzugeben. Dadurch entfällt die Notwendigkeit, mehrere ähnliche Benutzerskripte zu erstellen, die sich nur in einem einzelnen Parameter unterscheiden.

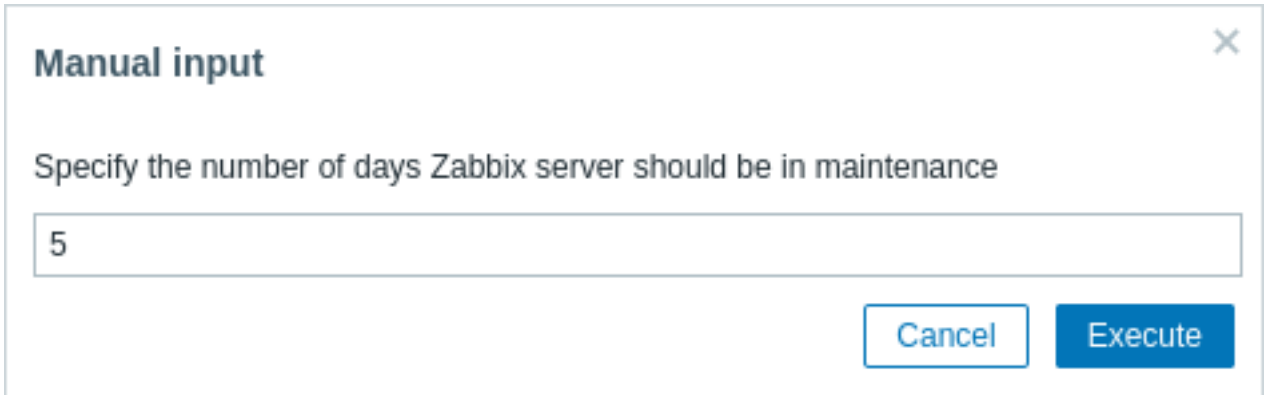
Beispielsweise möchten Sie dem Skript während der Ausführung möglicherweise eine andere Ganzzahl oder eine andere URL-Adresse übergeben.

So aktivieren Sie die manuelle Benutzereingabe:

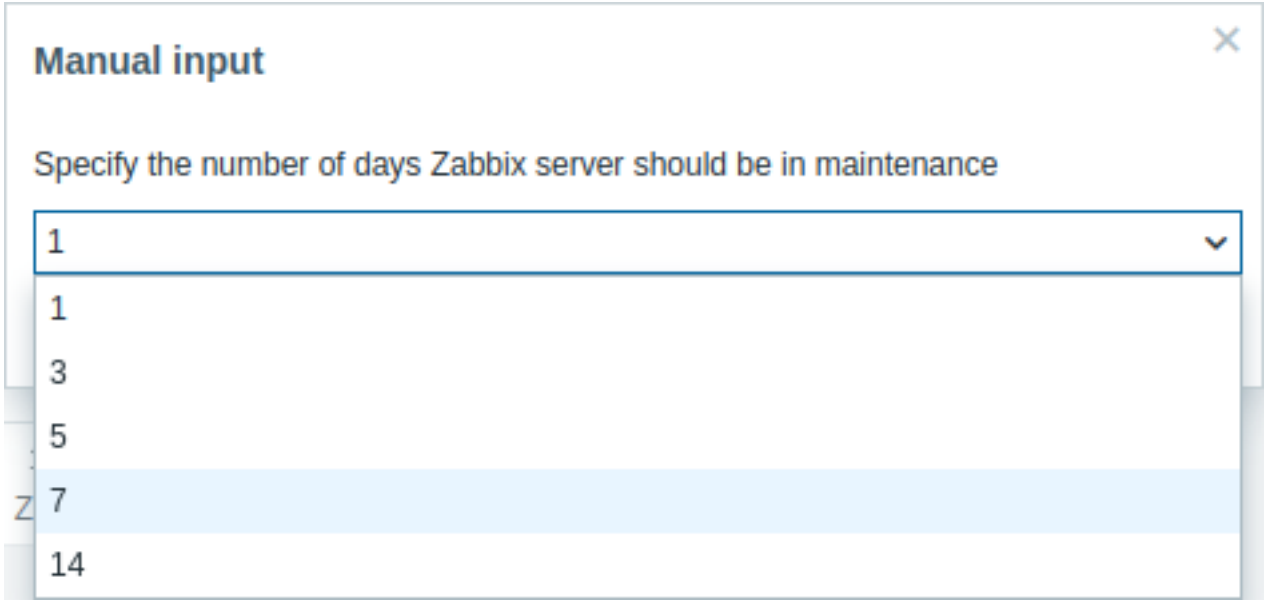
- verwenden Sie das Makro {MANUALINPUT} im Skript (Befehle, Skript, Skriptparameter), wo erforderlich; oder im URL-Feld von URL-Skripten;
- aktivieren Sie in der erweiterten Skriptkonfiguration die manuelle Benutzereingabe und konfigurieren Sie die Eingabeoptionen.

Wenn die Benutzereingabe aktiviert ist, wird dem Benutzer vor der Ausführung des Skripts ein Popup *Manuelle Eingabe* angezeigt, in dem er aufgefordert wird, einen benutzerdefinierten Wert anzugeben. Der angegebene Wert ersetzt {MANUALINPUT} im Skript.

Abhängig von der Konfiguration wird der Benutzer aufgefordert, einen Zeichenfolgenwert einzugeben:



Oder einen 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“ ist.

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:

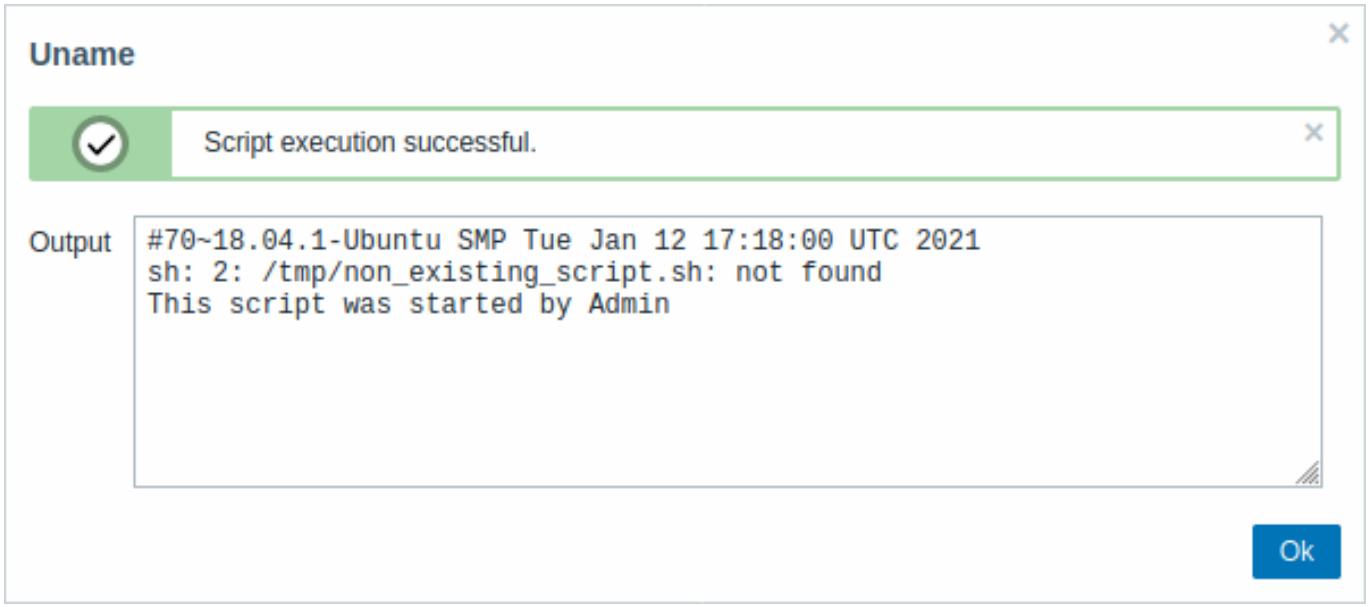
- Wenn das Skript erfolgreich abgeschlossen wird (Exit-Code 0), ist der Rückgabewert auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden).
- Wenn das Skript mit einem Fehler beendet wird (Exit-Code ungleich null), ist der Rückgabewert eine Standardfehlerausgabe, die 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.

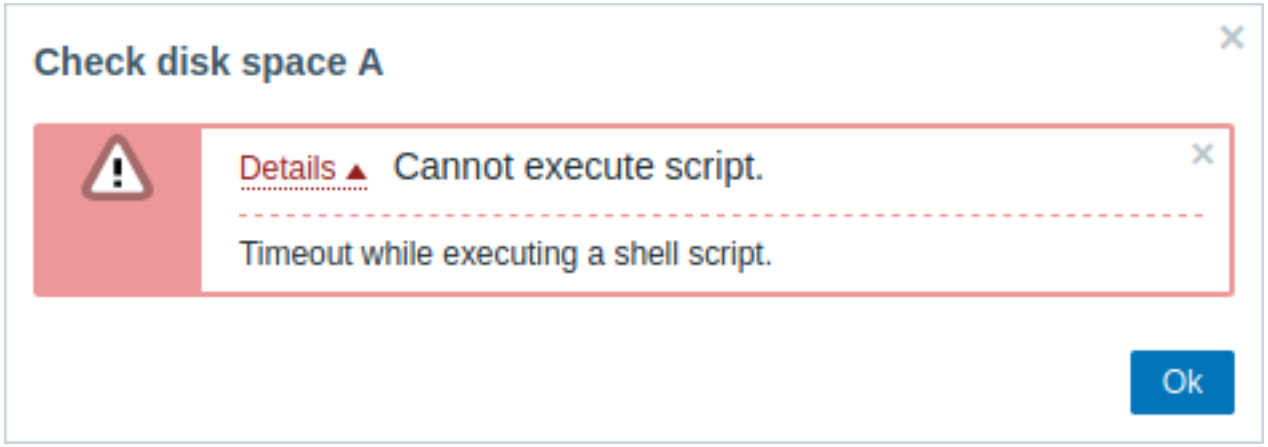
Skript-Timeout

Zabbix Agent

Es kann vorkommen, dass beim Ausführen eines Skripts ein Timeout auftritt.

Nachfolgend sehen Sie ein Beispiel für ein Skript, das auf dem Zabbix Agent ausgeführt wird, sowie das Ergebnisfenster:

```
sleep 5
df -h
```



Die Fehlermeldung lautet in diesem Fall wie folgt:

```
Timeout while executing a shell script.
```

Um solche Situationen zu vermeiden, wird empfohlen, das Skript selbst zu optimieren (im obigen Beispiel „5“), anstatt den Parameter `Timeout` in der Zabbix-Agent-Konfiguration und der Zabbix-Server-Konfiguration anzupassen. Für den Zabbix Agent im aktiven Modus sollte der Parameter `Timeout` in der Zabbix-Server-Konfiguration jedoch mindestens einige Sekunden länger sein als der Parameter `RefreshActiveChecks` in der Zabbix-Agent-Konfiguration. Dadurch wird sichergestellt, dass der Server genügend Zeit hat, die Ergebnisse der aktiven Prüfungen vom Agent zu empfangen. Beachten Sie, dass die Skriptausführung auf einem aktiven Agent seit Zabbix Agent 7.0 unterstützt wird.

Falls der Parameter `Timeout` in der Zabbix-Agent-Konfiguration geändert wurde, erscheint die folgende Fehlermeldung:

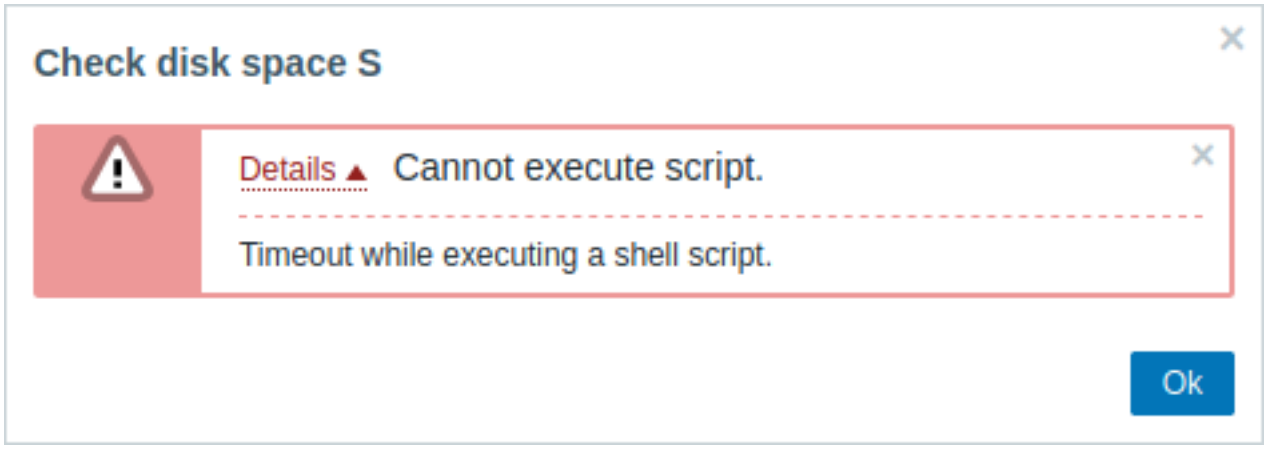
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Das bedeutet, dass die Änderung in der Zabbix-Agent-Konfiguration vorgenommen wurde, der Parameter `Timeout` jedoch zusätzlich auch in der Zabbix-Server-Konfiguration geändert werden muss.

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` durch Änderung der Zabbix Server-Konfiguration auf einen entsprechenden Wert anzupassen (in unserem Fall > 11)).

8 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.

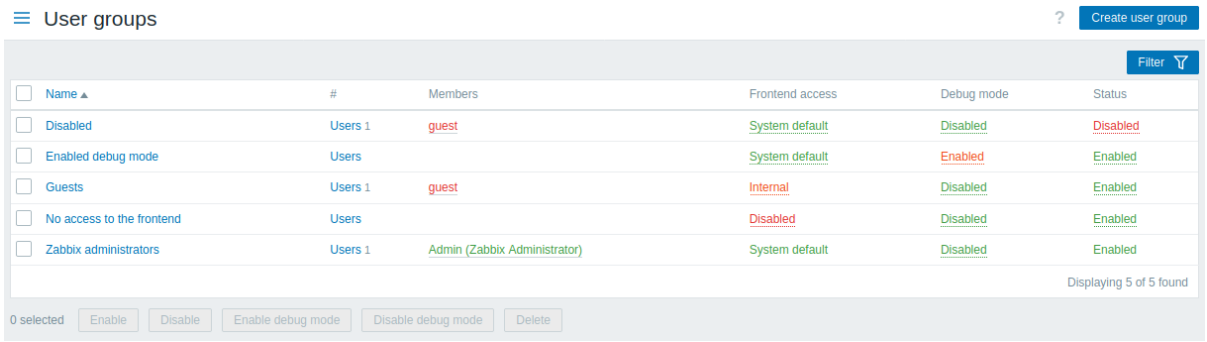
1 Benutzergruppen

Übersicht

Im Abschnitt *Benutzer* → *Benutzergruppen* 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. Durch Klicken auf den Namen der Benutzergruppe wird das Konfigurationsformular der Benutzergruppe geöffnet.
#	Die Anzahl der Benutzer in der Gruppe. Durch Klicken auf *Benutzer* werden die entsprechenden Benutzer gefiltert in der Benutzerliste angezeigt.
Mitglieder	Benutzernamen einzelner Benutzer in der Benutzergruppe (mit Vor- und Nachnamen in Klammern). Durch Klicken auf den Benutzernamen wird das Benutzer-Konfigurationsformular geöffnet. Benutzer aus deaktivierten Gruppen werden rot angezeigt.
Frontend-Zugriff	Die Frontend-Zugriffsebene wird angezeigt: **Systemstandard** - Benutzer werden durch Zabbix, LDAP oder HTTP authentifiziert (abhängig von der global festgelegten Authentifizierungsmethode); **Intern** - Benutzer werden durch Zabbix authentifiziert; wird ignoriert, wenn die HTTP-Authentifizierung global als Standard festgelegt ist; **LDAP** - Benutzer werden durch LDAP authentifiziert; wird ignoriert, wenn die HTTP-Authentifizierung global als Standard festgelegt 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 auf die Schaltfläche *Benutzergruppe erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

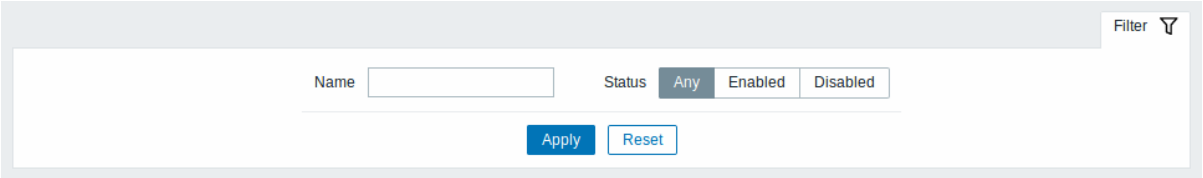
- *Aktivieren* - den Status der Benutzergruppe auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Benutzergruppe auf *Deaktiviert* ändern
- *Debug-Modus aktivieren* - den Debug-Modus für die Benutzergruppen aktivieren
- *Debug-Modus deaktivieren* - den Debug-Modus für die Benutzergruppen deaktivieren
- *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.

Filter verwenden

Sie können den Filter verwenden, um nur die Benutzergruppen anzuzeigen, die Sie interessieren. 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 eingeblendet, mit dem Sie Benutzergruppen nach Name und Status filtern können.



2 Benutzerrollen

Übersicht

Im Abschnitt *Benutzer* > *Benutzerrollen* können Sie Benutzerrollen erstellen.

Mit Benutzerrollen lassen sich fein abgestufte Berechtigungen auf Grundlage des anfänglich ausgewählten Benutzertyps (*Benutzer*, *Admin*, *Super admin*) erstellen.

Bei 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 das hinaus erweitert werden, was für den Benutzertyp verfügbar ist.

Kontrollkästchen für nicht verfügbare Berechtigungen sind ausgegraut; Benutzer können nicht auf das Element zugreifen, selbst wenn sie eine direkte URL zu diesem Element im Browser eingeben.

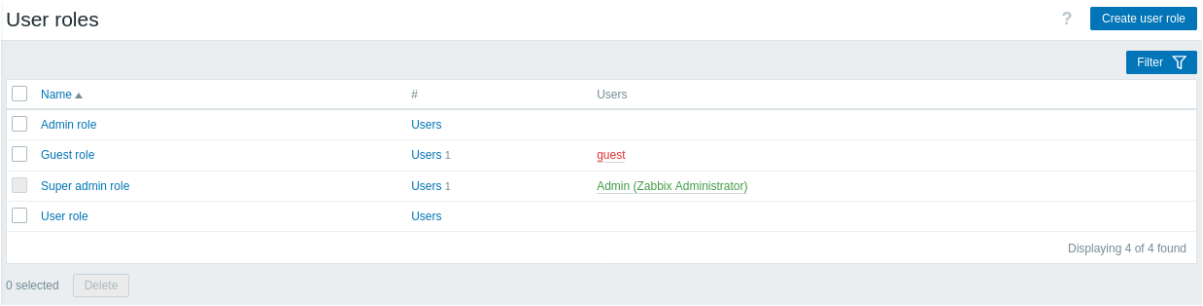
Note:
Die Einschränkung des Zugriffs auf einige UI-Elemente verhindert nur das Öffnen dieser Seite – sie hebt nicht die Möglichkeit auf, auf zugrunde liegende Daten in anderen Teilen der Oberfläche zuzugreifen.

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 Satz von Berechtigungen verfügen:

- Gastrolle
- Benutzerrolle
- Adminrolle
- Super-Admin-Rolle

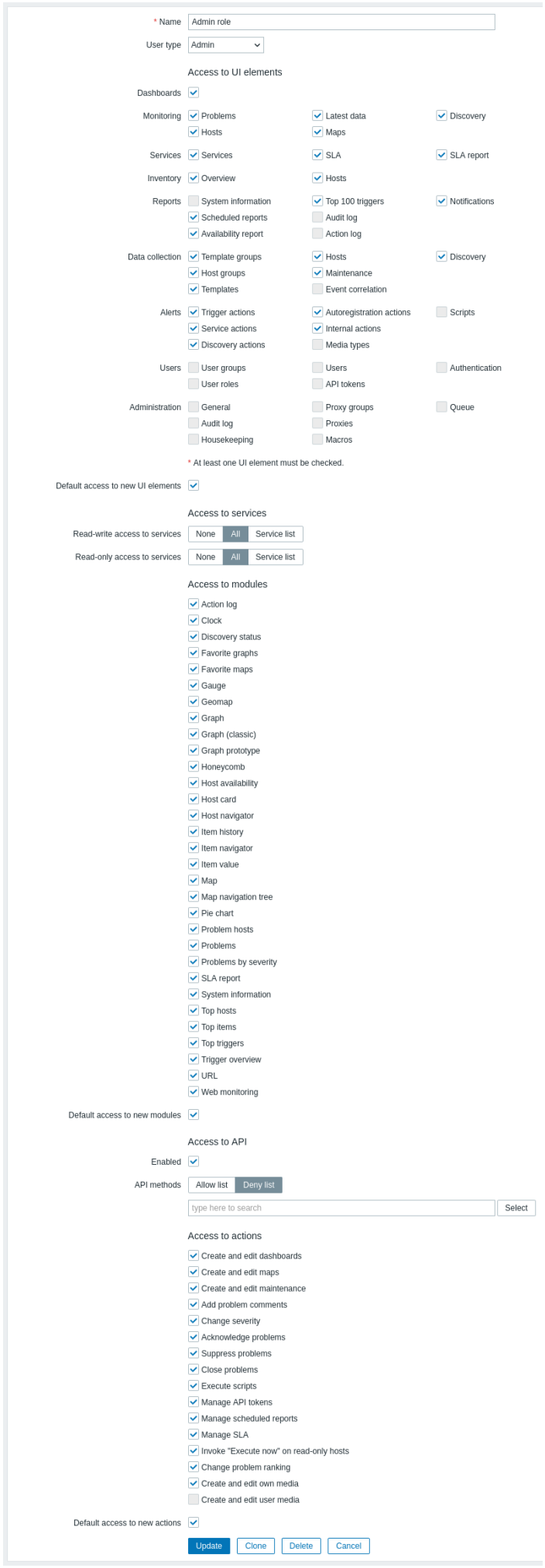


Diese basieren auf den Hauptbenutzertypen in Zabbix. Die Liste aller Benutzer, denen die jeweilige Rolle zugewiesen ist, wird angezeigt. Benutzer, die in deaktivierten Gruppen enthalten sind, werden rot dargestellt. Die *Gastrolle* ist eine Benutzertyprolle, deren einzige Berechtigung darin besteht, einige Bereiche des Frontend anzuzeigen.

Note:
Die standardmäßige *Super-Admin-Rolle* kann nicht geändert oder gelöscht werden, da in Zabbix mindestens ein Super-Admin-Benutzer mit uneingeschränkten Rechten vorhanden sein muss. Benutzer vom Typ *Super admin* können Einstellungen ihrer eigenen Rolle ändern, jedoch nicht den Benutzertyp.

Konfiguration

Um eine neue Rolle zu erstellen, klicken Sie auf die Schaltfläche *Benutzerrolle erstellen* in der oberen rechten Ecke. Um eine vorhandene Rolle zu aktualisieren, klicken Sie auf den Rollennamen, um das Konfigurationsformular zu öffnen.


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
type here to search
Select
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Default access to new actions
Update
Clone
Delete
Cancel

Verfügbare Berechtigungen werden angezeigt. Um eine bestimmte Berechtigung zu entziehen, deaktivieren Sie das entsprechende Kontrollkästchen.

Verfügbare Berechtigungen sowie die Standardwerte für jede vorkonfigurierte Benutzerrolle in Zabbix werden unten beschrieben.

Standardberechtigungen

Zugriff auf UI-Elemente

Der Standardzugriff auf Menüabschnitte hängt vom Benutzertyp ab. Siehe die Seite Berechtigungen für Details.

Zugriff auf andere Optionen

Parameter	**Beschreibung**	**Standardbenutzerrollen**			
		Super-Admin-Rolle	*Admin-Rolle*	*Benutzerrolle*	*Gastrolle*
Standardzugriff auf neue UI-Elemente	Diese Option legt fest, wie auf neue Menüabschnitte nach einem Zabbix-Upgrade zugegriffen werden kann. Bereits vorhandene Menüabschnitte von Modulen bleiben davon unberührt.	Ja	Ja	Ja	Ja
Zugriff auf Services					
Lese-/Schreibzugriff auf Services	Wählen Sie den Lese-/Schreibzugriff auf Services aus: **Keine** - überhaupt kein Zugriff **Alle** - Zugriff auf alle Services mit Lese-/Schreibrechten **Serviceliste** - Services für den Lese-/Schreibzugriff auswählen Der Lese-/Schreibzugriff hat, falls gewährt, 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 den Tag-Namen und optional den 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 Lese-/Schreibzugriff hat, falls gewährt, 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** - Zugriff auf alle Services nur lesend **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 den Tag-Namen und optional den 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 Module					
<Modulname>	Zugriff auf ein bestimmtes Modul erlauben/verbieten. 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 auf neue Module und Widgets nach einem Zabbix-Upgrade zugegriffen werden kann. Sie gilt auch für Module und Widgets, die im Abschnitt *Administration > Allgemein > Module* hinzugefügt wurden.				
Zugriff auf API					
Aktiviert	Zugriff auf API aktivieren/deaktivieren.	Ja	Ja	Ja	Nein
API-Methoden	Wählen Sie entweder *Erlaubnisliste*, um die im Suchfeld angegebenen API-Methoden zu erlauben, oder *Sperrliste*, um sie zu verweigern. Beachten Sie, dass es nicht möglich ist, einige API-Methoden zu erlauben und andere zu verweigern. Beginnen Sie im Suchfeld mit der Eingabe des Methodennamens und wählen Sie dann die Methode aus der Liste mit automatischer Vervollständigung aus. Sie können auch die Schaltfläche Auswählen drücken und Methoden aus der vollständigen Liste auswählen, die für diesen Benutzertyp verfügbar ist. Beachten Sie, dass Benutzer API-Methoden, die mit einer Aktion verknüpft sind, nicht verwenden können, wenn die 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 wurden, wird die Regel *Erlaubnis-/Sperrliste* ignoriert.				
Zugriff auf Aktionen					
Dashboards erstellen und bearbeiten	Wenn dieses Kontrollkästchen deaktiviert wird, werden auch die Rechte 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 dieses Kontrollkästchen deaktiviert wird, werden auch die Rechte entzogen, die entsprechende Aktion über die API-Methode `event.acknowledge` auszuführen.			Ja	

Schweregrad ändern Probleme bestätigen Probleme unterdrücken Probleme schließen			
Skripte ausführen	Wenn dieses Kontrollkästchen deaktiviert wird, werden auch die Rechte zur Verwendung der API-Methode `script.execute` entzogen.		
API-Tokens verwalten	Wenn dieses Kontrollkästchen deaktiviert wird, werden auch die Rechte zur Verwendung aller API-Methoden `token.` entzogen.		
Geplante Berichte verwalten	Wenn dieses Kontrollkästchen deaktiviert wird, werden auch die Rechte zur Verwendung aller API-Methoden `report.` entzogen.		Nein
SLA verwalten	Rechte zum Verwalten von SLA aktivieren/deaktivieren.		
„Jetzt ausführen“ auf Nur-Lese-Hosts aufrufen	Erlaubt die Verwendung der Option „Jetzt ausführen“ in den letzten Daten für Datenpunkte von Nur-Lese-Hosts.		Ja
Problemrangfolge ändern	Erlaubt das Ändern der Problemrangfolge von Ursache zu Symptom und umgekehrt.		
Eigene Medien erstellen und bearbeiten	Erlaubt das Erstellen/Bearbeiten eigener Medien.		
Benutzermedien erstellen und bearbeiten	Erlaubt das Erstellen/Bearbeiten von Medien für Benutzer. Diese Option ist nur für Super-Admin-Benutzer verfügbar.	Nein	Nein
Standardzugriff auf neue Aktionen	Diese Option legt fest, wie auf neue Aktionen nach einem Zabbix-Upgrade zugegriffen werden kann.	Ja	Ja

Siehe auch:

- Konfigurieren eines Benutzers

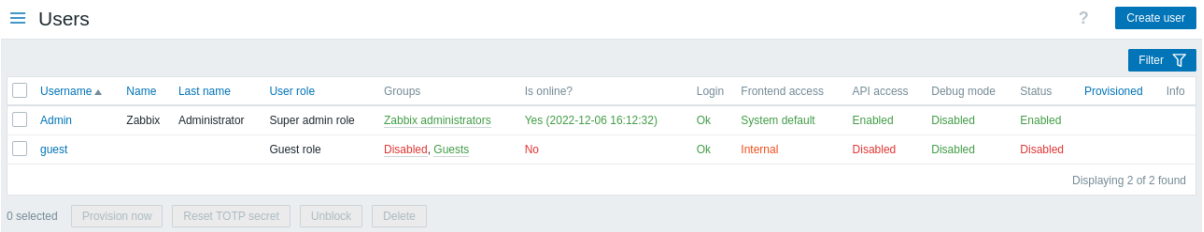
3 Benutzer

Übersicht

Im Abschnitt *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. Durch Klicken auf den Benutzernamen wird das Konfigurationsformular des Benutzers geöffnet.
Name	Vorname des Benutzers.
Nachname	Nachname des Benutzers.

Spalte	Beschreibung
Benutzerrolle	Die Benutzerrolle wird angezeigt.
Gruppen	Die Gruppen, deren Mitglied der Benutzer ist, werden aufgelistet. Durch Klicken auf den Namen der Benutzergruppe wird das Konfigurationsformular der Benutzergruppe 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 *Blockiert*. Ein Benutzer kann vorübergehend blockiert werden, wenn die im Abschnitt *Administration → Allgemein → Sonstiges* festgelegte Anzahl erfolgloser Anmeldeversuche überschritten wird (standardmäßig fünf). Durch Klicken auf *Blockiert* können Sie die Blockierung des Benutzers aufheben.
Frontend-Zugriff	Die Zugriffsebene für das Frontend wird angezeigt – *Systemstandard*, *Intern*, *LDAP* oder *Deaktiviert*, abhängig von der für die gesamte Benutzergruppe festgelegten Einstellung.
API-Zugriff	Der API-Zugriffsstatus wird angezeigt – *Aktiviert* oder *Deaktiviert*, abhängig von der für die Benutzerrolle festgelegten Einstellung.
Debug-Modus	Der Status des Debug-Modus wird angezeigt – *Aktiviert* oder *Deaktiviert*, abhängig von der für die gesamte Benutzergruppe festgelegten Einstellung.
Status	Der Benutzerstatus wird angezeigt – *Aktiviert* oder *Deaktiviert*, abhängig von der für die gesamte Benutzergruppe festgelegten Einstellung.
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 auf die Schaltfläche *Benutzer erstellen* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

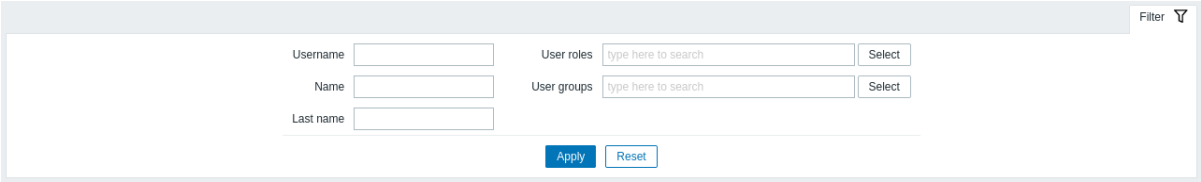
- *Jetzt bereitstellen* - Benutzerinformationen aus LDAP aktualisieren (diese Option ist nur aktiviert, wenn ein LDAP-Benutzer ausgewählt ist)
- *TOTP-Geheimnis zurücksetzen* - die TOTP-Geheimnisse des Benutzers für alle TOTP-Methoden zurücksetzen und die Benutzersitzung löschen (diese Option ist nur aktiviert, wenn MFA aktiviert ist; bei Benutzern ohne TOTP-Geheimnisse wird ihre Sitzung nicht gelöscht)
- *Entsperren* - den Systemzugriff für blockierte Benutzer wieder aktivieren
- *Löschen* - die Benutzer löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Benutzern und klicken Sie dann 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.

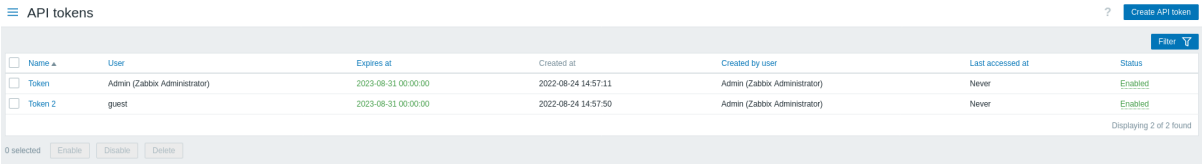
Der Link *Filter* ist oberhalb der Benutzerliste verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Benutzer nach Benutzername, Vorname, Nachname, Benutzerrolle und Benutzergruppe filtern können.



4 API-Tokens

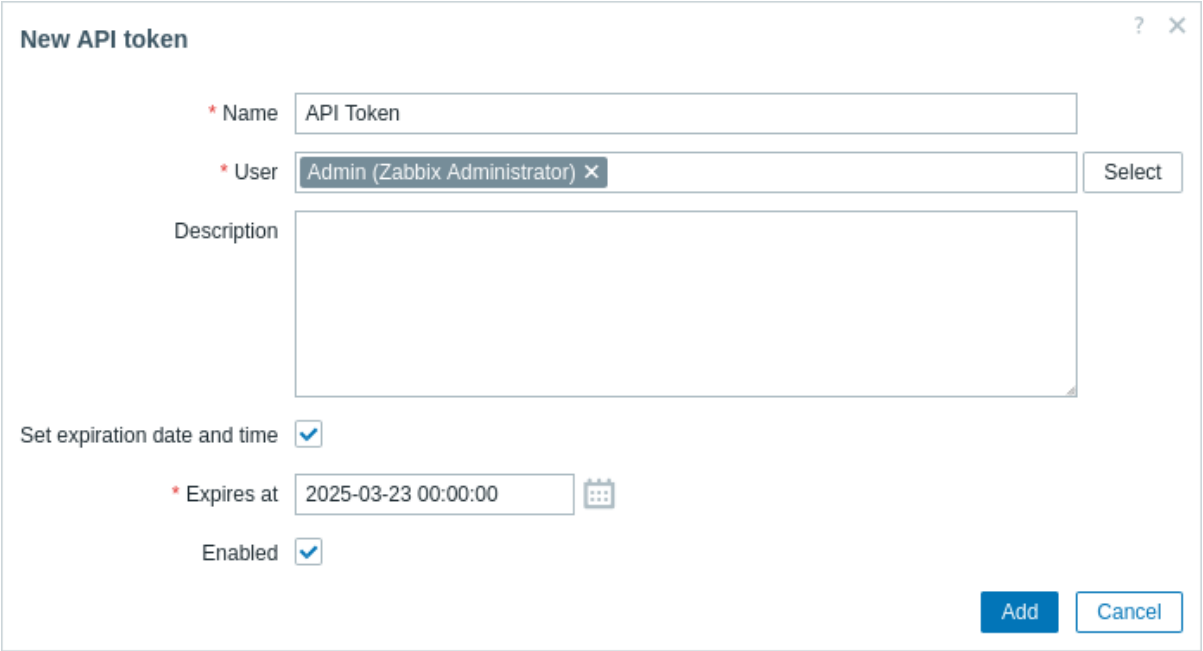
Übersicht

In diesem Abschnitt können API-Tokens erstellt und verwaltet werden.



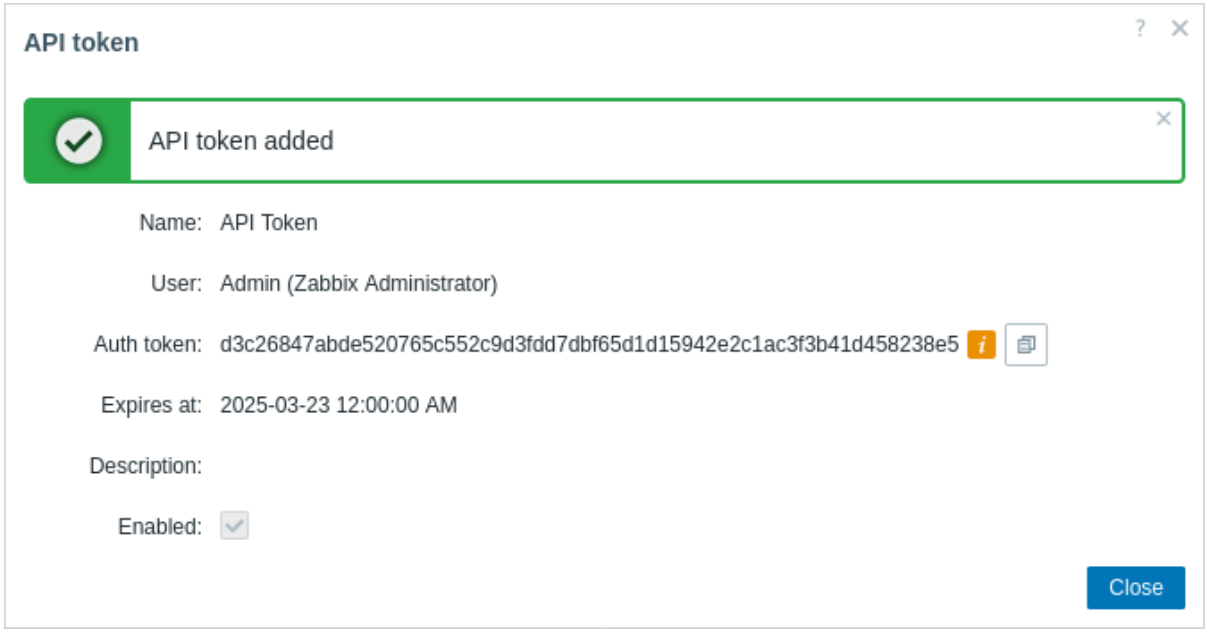
Sie können API-Tokens nach Name, den Benutzern, denen die Tokens zugewiesen sind, Ablaufdatum, den Benutzern, die Tokens erstellt haben, oder Status (aktiviert/deaktiviert) filtern. Klicken Sie in der Liste auf den Token-Status, um ein Token schnell zu aktivieren/deaktivieren. Sie können Tokens auch gesammelt aktivieren/deaktivieren, indem Sie sie in der Liste auswählen und dann unterhalb der Liste auf die Schaltflächen *Aktivieren/Deaktivieren* klicken.

Um ein neues Token zu erstellen, klicken Sie oben rechts auf die Schaltfläche *API-Token erstellen* und füllen Sie dann die erforderlichen Felder im Token-Konfigurationsbildschirm aus:



Parameter	Beschreibung
Name	Sichtbarer Name des Tokens.
User	Benutzer, dem das Token zugewiesen werden soll. Um schnell einen Benutzer auszuwählen, beginnen Sie mit der Eingabe des Benutzernamens, Vor- oder Nachnamens und wählen Sie dann den gewünschten Benutzer aus der Autovervollständigungsliste aus. Alternativ können Sie auf die Schaltfläche *Auswählen* 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 ein 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 ein Token im deaktivierten Zustand erstellen möchten.

Klicken Sie auf *Hinzufügen*, um ein Token zu erstellen.



Kopieren Sie den Wert des *Auth token* und speichern Sie ihn an einem sicheren Ort **bevor Sie die Seite schließen**. Klicken Sie anschließend auf Schließen. Das Token wird in der Liste angezeigt.

Warning:
Der Wert des *Auth token* kann später nicht erneut angezeigt werden. Er ist nur unmittelbar nach dem Erstellen eines Tokens verfügbar. Wenn Sie ein gespeichertes Token verlieren, müssen Sie es neu generieren; dabei wird eine neue Autorisierungszeichenfolge erstellt.

Klicken Sie auf den Tokennamen, um Name, Beschreibung, Ablaufdatumseinstellungen oder den Token-Status zu bearbeiten. Beachten Sie, dass es nicht möglich ist, den Benutzer zu ändern, dem das Token zugewiesen ist. Klicken Sie auf die Schaltfläche *Aktualisieren*, um die Änderungen zu speichern. Wenn ein Token verloren gegangen ist oder offengelegt wurde, können Sie auf die Schaltfläche *Neu generieren* klicken, um einen neuen Token-Wert zu erzeugen. Es wird ein Bestätigungsdialog angezeigt, in dem Sie aufgefordert werden, diesen Vorgang zu bestätigen, da das zuvor erzeugte Token nach dem Fortfahren ungültig wird.

Benutzer ohne Zugriff auf den Menüabschnitt *Administration* können Details von ihnen zugewiesenen Tokens nur im Abschnitt *Benutzerprofil* → *API-Tokens* section anzeigen und ändern, wenn *API-Tokens verwalten* in den Berechtigungen ihrer Benutzerrolle erlaubt ist.

5 Authentifizierung

Übersicht

Im Abschnitt *Benutzer* → *Authentifizierung* können die Benutzerauthentifizierungsmethode für Zabbix und die internen Anforderungen an Passwörter festgelegt werden.

Die verfügbaren Authentifizierungsmethoden sind interne, HTTP-, LDAP-, SAML- und MFA-Authentifizierung.

Standardauthentifizierung

Standardmäßig verwendet Zabbix die **interne** Zabbix-Authentifizierung für alle Benutzer.

Es ist möglich, die Standardauthentifizierungsmethode systemweit auf LDAP umzustellen. Navigieren Sie dazu zur Registerkarte *LDAP* und konfigurieren Sie die LDAP-Parameter. Kehren Sie anschließend zur Registerkarte *Authentication* zurück und setzen Sie den Selektor *Default authentication* auf LDAP.

Beachten Sie, dass die Authentifizierungsmethode auf Ebene der Benutzergruppe fein abgestimmt werden kann. Selbst wenn die LDAP-Authentifizierung global festgelegt ist, können einige Benutzergruppen weiterhin durch Zabbix authentifiziert werden. Für diese Gruppen muss der Frontend-Zugriff auf Internal 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 für bestimmte Benutzergruppen angegeben und verwendet werden, deren Frontend-Zugriff dann auf LDAP gesetzt sein muss. Wenn ein Benutzer in mindestens einer Benutzergruppe mit LDAP-Authentifizierung enthalten ist, kann dieser Benutzer die interne Authentifizierungsmethode nicht verwenden.

Die Authentifizierungsmethoden HTTP, SAML 2.0 und MFA können zusätzlich zur Standardauthentifizierungsmethode verwendet werden.

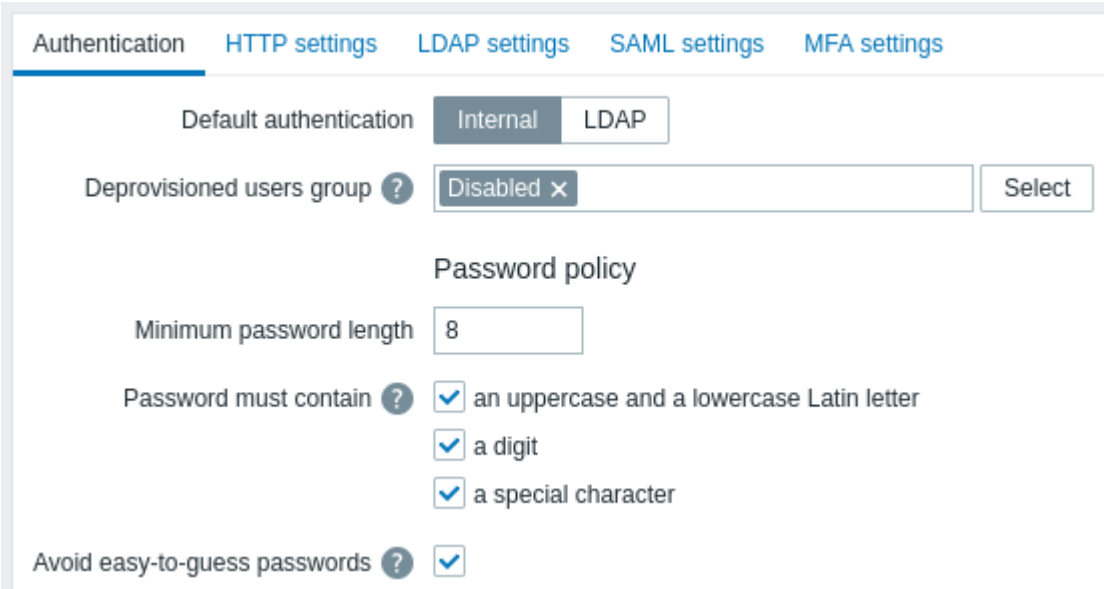
Zabbix unterstützt die Just-in-Time-(JIT)-Bereitstellung, mit der Benutzerkonten in Zabbix erstellt werden können, wenn sich ein externer Benutzer zum ersten Mal authentifiziert, und diese Benutzerkonten bereitgestellt werden. Die JIT-Bereitstellung wird für LDAP und SAML unterstützt.

Siehe auch:

- HTTP-Authentifizierung
- LDAP-Authentifizierung
- SAML-Authentifizierung
- MFA-Authentifizierung

Konfiguration

Die Registerkarte *Authentifizierung* ermöglicht es, die Standardauthentifizierungsmethode festzulegen, eine Gruppe für deprovisionierte Benutzer anzugeben und Anforderungen an die Passwortkomplexität für Zabbix-Benutzer festzulegen.



Konfigurationsparameter:

Parameter	Beschreibung
Standardauthentifizierung	Wählen Sie die Standardauthentifizierungsmethode für Zabbix aus - *Intern* oder *LDAP*.
Gruppe für deprovisionierte Benutzer	Geben Sie eine Benutzergruppe für deprovisionierte Benutzer an. Diese Einstellung ist nur für die JIT-Bereitstellung erforderlich, in Bezug auf Benutzer, die in Zabbix aus LDAP- oder SAML-Systemen erstellt wurden, aber nicht länger bereitgestellt werden müssen. Es muss eine deaktivierte Benutzergruppe angegeben werden.
Minimale Passwortlänge	Standardmäßig ist die minimale Passwortlänge auf 8 gesetzt. Unterstützter Bereich: 1-70. Beachten Sie, dass Passwörter mit mehr als 72 Zeichen abgeschnitten werden.
Passwort muss enthalten	Aktivieren Sie ein oder mehrere Kontrollkästchen, um die Verwendung bestimmter 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 Zeichenliste 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 Vornamen, Nachnamen oder Benutzernamen des Benutzers enthalten - darf keines der gängigen oder kontextspezifischen Passwörter sein. Die Liste der gängigen und kontextspezifischen Passwörter wird automatisch aus der NCSC-Liste „Top 100k passwords“, der SecLists-Liste „Top 1M passwords“ und der Zabbix-Liste kontextspezifischer Passwörter erstellt. Interne Benutzer dürfen keine in dieser Liste enthaltenen Passwörter festlegen, da solche Passwörter aufgrund ihrer weiten Verbreitung als schwach gelten.

Änderungen an den Anforderungen an die Passwortkomplexität wirken sich nicht auf bestehende Benutzerpasswörter aus. Wenn

sich jedoch ein bestehender Benutzer dazu entscheidet, ein Passwort zu ändern, muss das neue Passwort die aktuellen Anforderungen erfüllen. Ein Hinweis mit der Liste der Anforderungen wird neben dem Feld *Passwort* im Benutzerprofil und im über das Menü *Benutzer* → *Benutzer* zugänglichen Benutzerkonfigurationsformular angezeigt.

1 HTTP

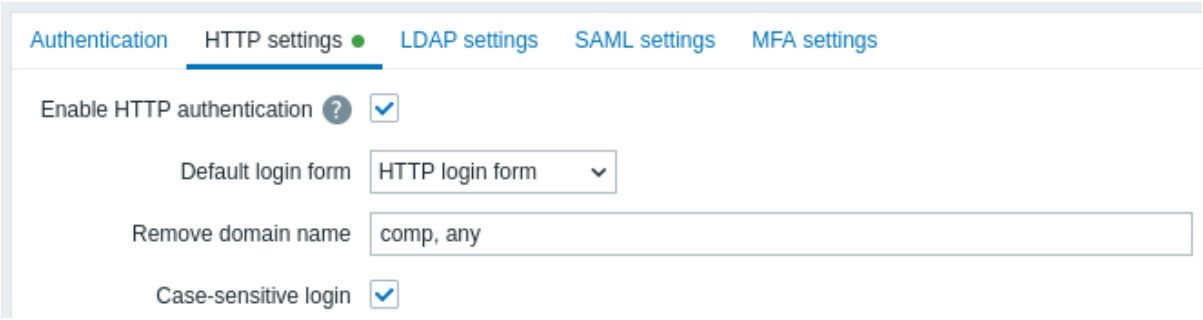
Übersicht

HTTP- oder webserverbasierte Authentifizierung (zum Beispiel: BasicAuthentication, NTLM/Kerberos) kann verwendet werden, um Benutzernamen und Passwörter zu prüfen. Beachten Sie, dass ein Benutzer auch in Zabbix vorhanden sein muss, sein Zabbix-Passwort jedoch nicht verwendet wird.

Attention:
Vorsicht! Stellen Sie sicher, dass die Webserver-Authentifizierung konfiguriert ist und ordnungsgemäß funktioniert, bevor Sie sie aktivieren.

Die HTTP-Authentifizierung kann im Frontend deaktiviert werden, indem die entsprechende Option auf der Registerkarte *HTTP-Einstellungen* im Abschnitt *Benutzer > Authentifizierung* konfiguriert wird. Wenn die HTTP-Authentifizierung deaktiviert ist, wird die Registerkarte mit den HTTP-Authentifizierungsoptionen im Frontend nicht angezeigt. Beachten Sie, dass eine Neuinstallation des Frontends (Ausführen von setup.php) die Authentifizierungseinstellungen zurücksetzt, einschließlich der Konfiguration der HTTP-Authentifizierung.

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 hinweist, dass bei Webserver-Authentifizierung alle Benutzer (auch wenn der Frontend-Zugriff auf LDAP/Intern gesetzt ist) durch den Webserver und nicht durch Zabbix authentifiziert werden.
Standard-Anmeldeformular	Geben Sie an, ob nicht authentifizierte Benutzer weitergeleitet werden sollen zu: **Zabbix-Anmeldeformular** - standardmäßige Zabbix-Anmeldeseite. **HTTP-Anmeldeformular** - HTTP-Anmeldeseite. Es wird empfohlen, die Webserver-basierte 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 Webserver-Authentifizierungsmodul einen gültigen Benutzer-Login in der Variablen `$_SERVER` setzt. Unterstützte `$_SERVER`-Schlüssel sind `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Domänennamen entfernen	Eine durch Kommas getrennte Liste von Domänennamen, die aus dem Benutzernamen entfernt werden sollen. Z. B. `comp,any` - wenn der Benutzername „Admin@any“ oder „comp\Admin“ ist, wird der Benutzer als „Admin“ angemeldet; wenn der Benutzername „notacompany\Admin“ ist, wird die Anmeldung verweigert.

Parameter	Beschreibung
Groß-/Kleinschreibung bei der Anmeldung beachten	Deaktivieren Sie das Kontrollkästchen, um die Beachtung der Groß-/Kleinschreibung bei Benutzernamen zu deaktivieren (standardmäßig aktiviert). Wenn die Beachtung der Groß-/Kleinschreibung deaktiviert ist, kann man sich beispielsweise als „admin“ anmelden, auch wenn der Zabbix-Benutzer „Admin“ oder „ADMIN“ ist. Bitte beachten Sie, dass bei deaktivierter Beachtung 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 mit HTTP- Anmeldedaten nicht anmelden können (wenn das HTTP- Anmeldeformular als Standard festgelegt ist), was zum Fehler 401 führt, können Sie eine Zeile `ErrorDocument 401 /index.php?form=default` zu den Direktiven der Basisauthentifizierung hinzufügen, die zur normalen Zabbix-Anmeldeseite umleitet.

2 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, JIT (Just-in-Time) **Benutzerbereitstellung** für LDAP-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.

Wenn ein LDAP-Benutzer seinen LDAP-Benutzernamen und sein Passwort eingibt, prüft Zabbix den *standardmäßigen* LDAP-Server darauf, ob dieser Benutzer existiert. Wenn der Benutzer existiert und noch kein Konto in Zabbix hat, wird in Zabbix ein neuer Benutzer 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. Eine spätere Änderung des standardmäßigen LDAP-Servers ändert oder aktualisiert den mit bereits bereitgestellten Benutzern verknüpften LDAP-Server nicht.

Attention:
Wenn die JIT-Bereitstellung aktiviert ist, muss im Reiter *Authentifizierung* eine Benutzergruppe für deprovisionierte Benutzer angegeben werden.

Die JIT-Bereitstellung ermöglicht außerdem, bereitgestellte Benutzerkonten anhand 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 der Benutzer auch in Zabbix aus der Gruppe entfernt und, falls er keiner anderen Gruppe angehört, der Benutzergruppe für deprovisionierte Benutzer hinzugefügt. Bereitgestellte Benutzerkonten werden basierend auf dem konfigurierten Bereitstellungsintervall oder bei der Anmeldung des Benutzers bei Zabbix aktualisiert.

Beachten Sie, dass die Hintergrundbereitstellung durch das Zabbix Frontend erfolgt, während der Benutzer damit interagiert oder zumindest eine Frontend-Seite im Browser geöffnet hat. Es gibt keine dedizierten Hintergrundprozesse für die Benutzerbereitstellung.

LDAP unterstützt drei Möglichkeiten, sich zur Authentifizierung und für Suchvorgänge an das Verzeichnis zu binden:

- Anonyme Bindung — es werden kein *Bind DN* / *Bind password* angegeben und der LDAP-Server erlaubt anonyme Abfragen.
- Dedizierter Bind-Benutzer (Servicekonto) — ein bestimmtes LDAP-Konto wird in *Bind DN* / *Bind password* festgelegt und von Zabbix für Suche und Bereitstellung verwendet. Dies ist die empfohlene und flexibelste Option, da Zabbix Suchvorgänge und Hintergrundbereitstellung ohne Endbenutzer-Anmeldedaten durchführen kann.

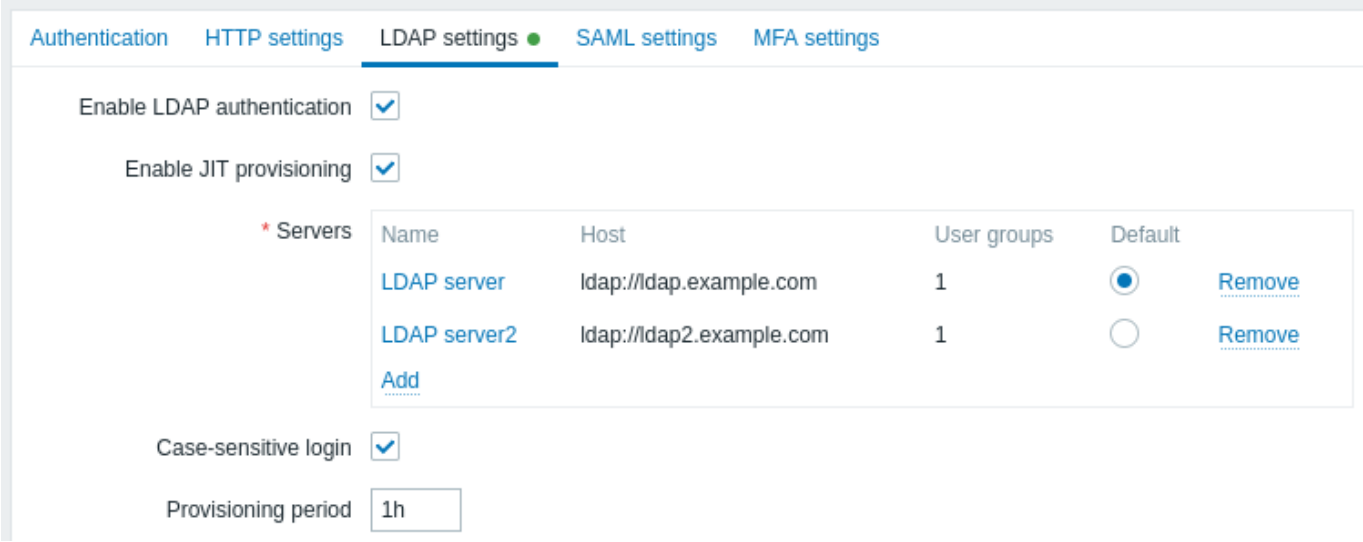
- Direkte Benutzerbindung — Zabbix führt die Bindung mit den Anmeldedaten durch, die der Benutzer bei der Anmeldung eingibt (es ist kein *Bind DN* / *Bind password* konfiguriert); dies wird durch Einfügen eines Platzhalters wie `uid=%{user}` in die Base DN konfiguriert. In diesem Modus hat Zabbix nur während der interaktiven Anmeldung Zugriff auf das Passwort des Benutzers. Daher können Bereitstellungsaktionen, die eine Authentifizierung bei LDAP außerhalb der Anmeldesitzung des Benutzers erfordern (zum Beispiel die Verwendung der Schaltfläche *Provision now* oder die Ausführung der Hintergrundbereitstellung, wenn der Benutzer nicht aktiv angemeldet ist), nicht authentifiziert werden und funktionieren daher nicht. Bereitstellung und Aktualisierungen bei direkter Benutzerbindung erfolgen nur in dem Moment, in 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.

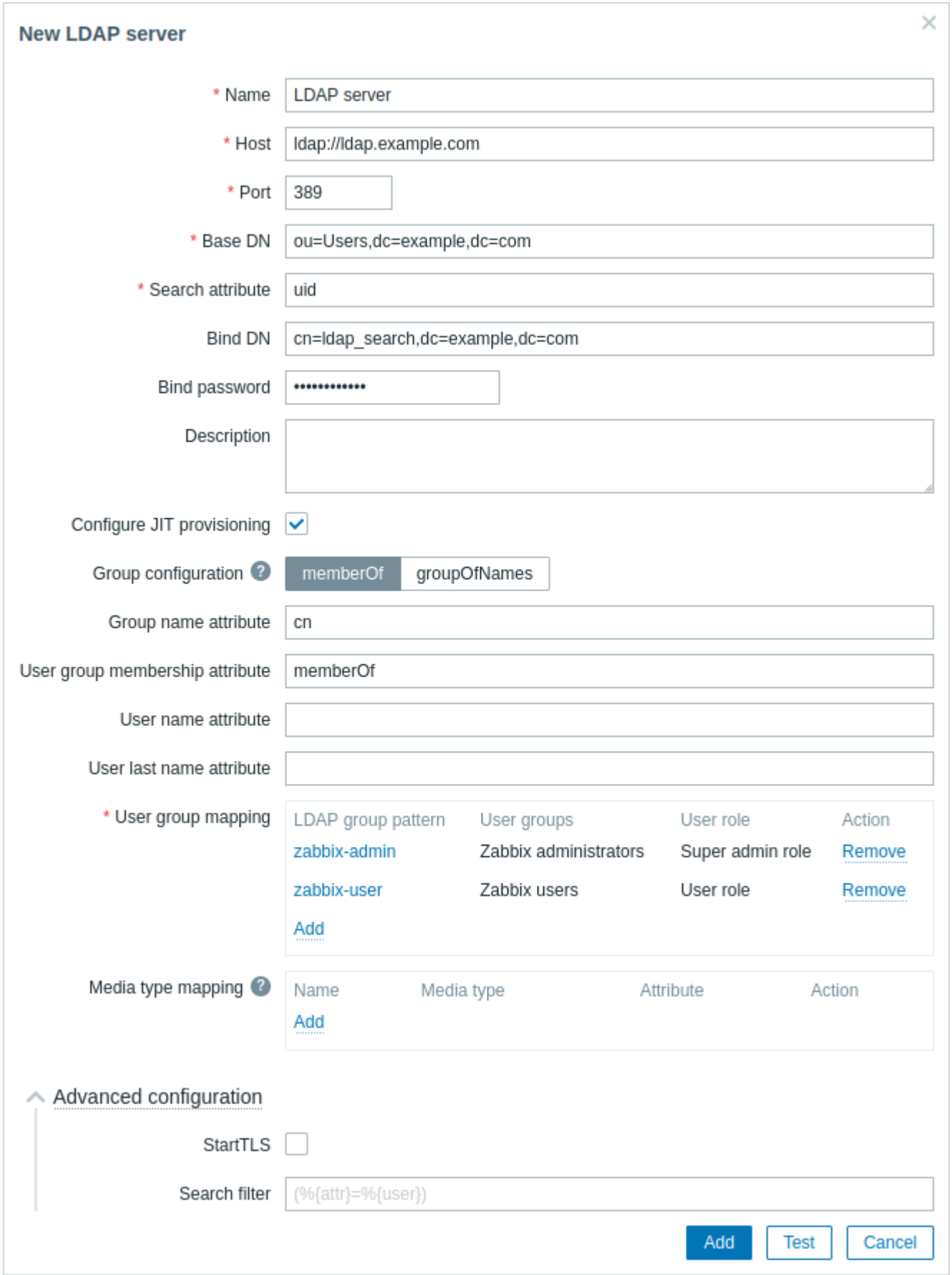
Konfiguration



Konfigurationsparameter:

Parameter	Beschreibung
Enable LDAP authentication	Aktivieren Sie das Kontrollkästchen, um die LDAP-Authentifizierung zu aktivieren.
Enable JIT provisioning	Aktivieren Sie das Kontrollkästchen, um die JIT-Bereitstellung zu aktivieren.
Servers	Klicken Sie auf *Add*, um einen LDAP-Server zu konfigurieren (siehe LDAP-Server-Konfiguration unten).
Case-sensitive login	Deaktivieren Sie das Kontrollkästchen, um die Groß-/Kleinschreibung bei der Anmeldung für Benutzernamen zu deaktivieren (standardmäßig aktiviert). Wenn die Groß-/Kleinschreibung bei der Anmeldung deaktiviert ist, kann man sich beispielsweise als „admin“ anmelden, auch wenn der Zabbix-Benutzer „Admin“ oder „ADMIN“ ist. Bitte beachten Sie, dass bei deaktivierter Groß-/Kleinschreibung bei der Anmeldung 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.“
Provisioning period	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:

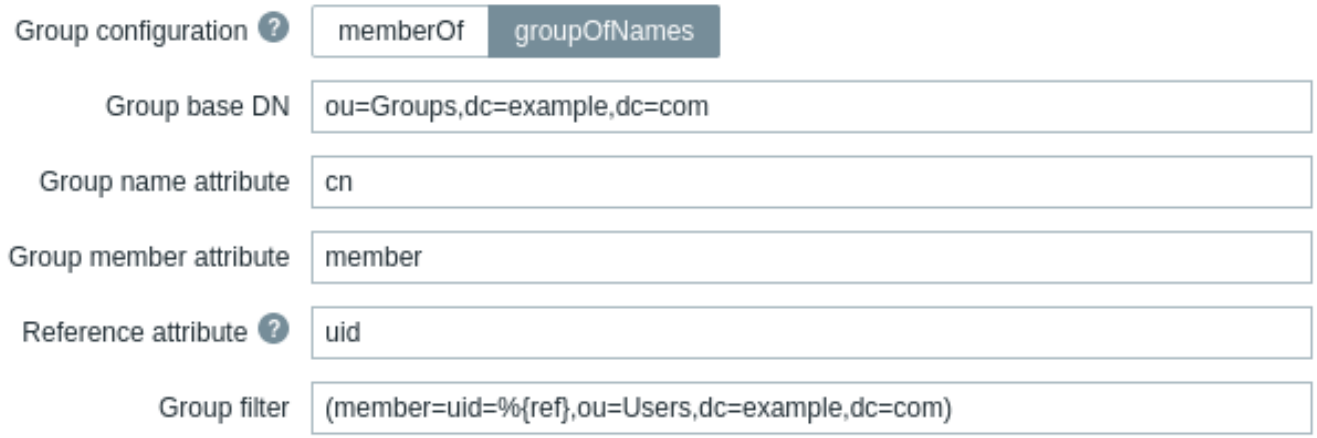
Parameter	Beschreibung
Name	Name des LDAP-Servers in der Zabbix-Konfiguration.

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 den Hostnamen. Beispiel: ldaps://ldap.example.com Mit OpenLDAP 2.x.x und höher kann eine vollständige LDAP-URI im Format 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	Basispfad zu Benutzerkonten im 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	Für die Suche verwendetes LDAP-Kontoattribut: uid (für OpenLDAP), sAMAccountName (für Microsoft Active Directory)
Bind DN	LDAP-Konto für Bindung und Suche über den 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 die Domänenkonfiguration potenziell für unbefugte Benutzer öffnen kann (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 Bindung und Suche über den LDAP-Server.
Description	Beschreibung des LDAP-Servers.
Configure JIT provisioning	Aktivieren Sie dieses Kontrollkästchen, um Optionen für JIT-Provisionierung anzuzeigen.
Group configuration	Wählen Sie die Methode zur Gruppenkonfiguration aus: **memberOf** - durch Suche nach Benutzern und ihrem Attribut für Gruppenzugehörigkeit **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 Gruppenfilterung erforderlich ist.
Group name attribute	Geben Sie das Attribut an, aus dem der Gruppenname aus allen Objekten im Attribut `memberOf` abgerufen wird (siehe Feld *User group membership attribute*) Der Gruppenname ist für die Zuordnung 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 Benutzerrolle zu. Dies ist erforderlich, um zu bestimmen, welche Benutzergruppe/-rolle der bereitgestellte Benutzer in Zabbix erhält. Klicken Sie auf *Add*, um eine Zuordnung 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 in allen diesen Gruppen. Wenn ein Benutzer mehreren Zabbix-Benutzerrollen entspricht, erhält der Benutzer die Rolle mit der höchsten Berechtigungsstufe darunter.
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 Beschriftung *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 des zu authentifizierenden Benutzers Um beispielsweise in einer LDAP- oder Microsoft-Active-Directory-Umgebung ohne Berücksichtigung der Groß-/Kleinschreibung eine Suche mit Berücksichtigung der Groß-/Kleinschreibung durchzuführen, kann die Zeichenfolge wie folgt definiert werden: `(%{attr}:caseExactMatch:=%{user})`. Wenn der Filter nicht angepasst wird, verwendet LDAP den Standardwert: `(%{attr}=%{user})`.

Note:
Um einen LDAP-Server für **direkte Benutzerbindung** zu konfigurieren, hängen Sie ein Attribut uid=%{user} an den Parameter *Base DN* an (zum Beispiel *uid=%{user},dc=example,dc=com*) und lassen Sie die Parameter *BindDN* und *Bind password* leer. Bei der Authentifizierung wird ein Platzhalter %{user} durch den beim Login eingegebenen Benutzernamen ersetzt.
 Bei direkter Benutzerbindung hat Zabbix nur während des interaktiven Logins Zugriff auf die Anmeldedaten des Benutzers. Daher ignorieren Provisionierungsaufgaben, die ohne interaktiven Login arbeiten (zum Beispiel die Schaltfläche *Provision now* oder die API-Methode `user.provision`), Benutzer, die sich mit direkter Benutzerbindung authentifizieren, da Zabbix nicht in ihrem Namen binden kann. Verwenden Sie anonyme Bindung oder einen dedizierten Bind-Benutzer (Dienstkonto), wenn Provisionierung und LDAP-Suchen im Frontend funktionieren sollen, ohne dass der Endbenutzer angemeldet sein muss.

Die folgenden Felder sind spezifisch für "groupOfNames" als Methode für *Group configuration*:



Parameter	Beschreibung
Group base DN	Basispfad zu den Gruppen im LDAP-Server.
Group name attribute	Geben Sie das Attribut an, aus dem der Gruppenname im angegebenen Basispfad zu den Gruppen abgerufen wird. Der Gruppenname ist für die Zuordnung 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, deren Mitglied der Benutzer ist. Zum Beispiel entspricht `(member=uid=%{ref},ou=Users,dc=example,dc=com)` dem Benutzer "User1", wenn das member-Attribut der Gruppe `uid=User1,ou=Users,dc=example,dc=com` ist, und gibt die Gruppe zurück, deren Mitglied "User1" ist.

Warning:
Bei Problemen mit Zertifikaten 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*) mit minimalen Berechtigungen im LDAP für Bindung und Suche über den LDAP-Server zu erstellen, anstatt echte Benutzerkonten zu verwenden (die für die Anmeldung im Zabbix Frontend verwendet werden).
Ein solcher Ansatz bietet mehr Sicherheit und erfordert keine Änderung des Parameters *Bind password*, wenn der Benutzer sein eigenes Passwort auf dem LDAP-Server ändert.
In der obigen Tabelle ist dies der Kontoname *ldap_search*.

Zugriff testen

Mit der Schaltfläche *Test* kann der Benutzerzugriff getestet werden:

Parameter	Beschreibung
Login	LDAP-Benutzername zum Testen (vorausgefüllt mit dem aktuellen Benutzernamen aus dem Zabbix Frontend). Dieser Benutzername muss auf dem LDAP-Server vorhanden sein. Zabbix aktiviert die LDAP-Authentifizierung nicht, wenn der Testbenutzer nicht authentifiziert werden kann.
User password	LDAP-Benutzerpasswort zum Testen.

3 SAML

Übersicht

Die SAML 2.0-Authentifizierung kann verwendet werden, um sich bei Zabbix anzumelden.

Wenn nur die SAML-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 SAML zurückgegebenen Benutzernamenattribut 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`

Einrichten von Zabbix

Attention:
Es ist erforderlich, php-openssl zu installieren, wenn Sie die SAML-Authentifizierung im Frontend verwenden möchten.

Um die SAML-Authentifizierung zu verwenden, sollte Zabbix wie folgt konfiguriert werden:

1. Stellen Sie den privaten SP-Schlüssel und die Zertifikate bereit. Ein IdP-Zertifikat muss bereitgestellt werden (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 zabbix.conf.php gesetzt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen — Dateien im Dateisystem sind nicht erforderlich.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in zabbix.conf.php gesetzt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in ui/conf/certs oder im in zabbix.conf.php konfigurierten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank. Beachten Sie, dass Dateispeicherung angenommen wird, wenn `$SSO['CERT_STORAGE']` nicht gesetzt oder auskommentiert ist.

Standardmäßig sucht Zabbix an den folgenden Speicherorten:

- ui/conf/certs/sp.key - Datei mit privatem 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 der Einfachheit halber in einigen Umgebungen zulässig sind, wird die Verwendung von verschlüsselter Übertragung 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 notwendige Minimum.

2. Alle wichtigsten Einstellungen können im Zabbix-Frontend konfiguriert werden. Es ist jedoch möglich, zusätzliche Einstellungen in der Konfigurationsdatei anzugeben.

Konfigurationsparameter, die im Zabbix-Frontend verfügbar sind:

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 sie leer bleibt, wird der SLO-Dienst nicht verwendet.

Parameter	Beschreibung
Username attribute	SAML-Attribut, das bei der Anmeldung an Zabbix als Benutzername verwendet wird. 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 Service Providers (wenn sie nicht übereinstimmt, wird der Vorgang abgelehnt). Es kann eine URL oder eine beliebige Zeichenfolge angegeben werden.
SP name ID format	Fordern Sie 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 Service-Provider-(SP-)Zertifikats für die Einrichtung des SAML-Single-Sign-On-(SSO-)Dienstes.
SP private key	Inhalt des privaten Schlüssels des Service Providers (SP) für die Einrichtung des SAML-Single-Sign-On-(SSO-)Dienstes. Ermöglicht sichere Authentifizierung und sicheren Datenaustausch mit dem Identitätsanbieter (IdP).
SP certificate	Inhalt des Service-Provider-(SP-)Zertifikats für die Einrichtung des SAML-Single-Sign-On-(SSO-)Dienstes.
Sign	Aktivieren Sie die Kontrollkästchen, um Entitäten auszuwählen, für die die SAML-Signatur aktiviert werden soll: *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	Aktivieren Sie die Kontrollkästchen, um 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 das Deaktivieren der Groß-/Kleinschreibung bei der Anmeldung ist es beispielsweise möglich, sich als ”admin” anzumelden, auch wenn der Zabbix-Benutzer ”Admin” oder ”ADMIN” ist. Bitte beachten Sie, dass bei deaktivierter 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: ”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 Gruppenname-Attribut für die JIT-Benutzerbereitstellung an.
User name attribute	Geben Sie das Benutzernamen-Attribut für die JIT-Benutzerbereitstellung an.
User last name attribute	Geben Sie das Nachnamen-Attribut des Benutzers für die JIT-Benutzerbereitstellung an.

Parameter	Beschreibung
User group mapping	Ordnen Sie ein SAML-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 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 in allen diesen Gruppen. Wenn ein Benutzer mehreren Zabbix-Benutzerrollen entspricht, erhält der Benutzer die höchste Berechtigungsstufe unter ihnen.
Media type mapping	Ordnen Sie die SAML-Medienattribute des Benutzers (z. B. E-Mail) den Zabbix-Benutzermedien für den Versand von Benachrichtigungen zu.
Enable SCIM provisioning	Aktivieren Sie dieses Kontrollkästchen, um die SCIM-2.0-Bereitstellung zu aktivieren.

Siehe Beispiele für die Konfiguration von SAML-Identitätsanbietern für die Anmeldung und Benutzerbereitstellung in Zabbix mit:

- 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, d. h.:

`https://<path-to-zabbix-ui>/api_scim.php`

Benutzerattribute, die in Zabbix verwendet werden (Benutzername, Vorname des Benutzers, Nachname des Benutzers 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 Zabbix-Frontend-Konfigurationsdatei (*zabbix.conf.php*) konfiguriert werden:

- $SSO['SP_KEY'] = '<Pfad zur privaten SP-Schlüsseldatei>';
- $SSO['SP_CERT'] = '<Pfad zur SP-Zertifikatsdatei>';
- $SSO['IDP_CERT'] = '<Pfad zur IDP-Zertifikatsdatei>';
- $SSO['SETTINGS']

Note:
Das Array `$SSO['SETTINGS']` muss derselben Struktur folgen, die von der Bibliothek *SAML PHP Toolkit* erwartet wird (mitgeliefert mit Zabbix). Eine vollständige Beschreibung der verfügbaren Konfigurationsoptionen finden Sie in der offiziellen Bibliotheksdokumentation.

Nur die folgenden Optionen können als Teil von $SSO['SETTINGS'] gesetzt 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 UI außerdem hinter einem Proxy oder Load Balancer befindet, kann die benutzerdefinierte Option *use_proxy_headers* verwendet werden:

- *false* (Standard) - Option ignorieren;
- *true* - X-Forwarded-*-HTTP-Header zum Erstellen der Basis-URL verwenden.

Wenn ein Load Balancer für die Verbindung zur Zabbix-Instanz verwendet wird, wobei der Load Balancer TLS/SSL verwendet und Zabbix nicht, müssen die Parameter 'baseurl', 'strict' und 'use_proxy_headers' wie folgt angegeben werden:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Konfigurationsbeispiel:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Frontend-Konfiguration mit Kerberos/ADFS

Die Zabbix-Frontend-Konfigurationsdatei (*zabbix.conf.php*) kann verwendet werden, um SSO mit Kerberos-Authentifizierung und ADFS zu konfigurieren:

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

Setzen Sie in diesem Fall im SAML-Konfigurationsfeld *SP name ID*:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

4 MFA

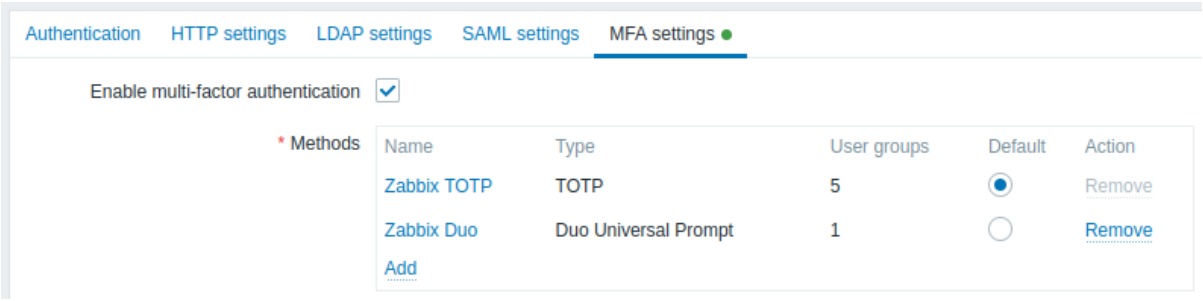
Übersicht

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.

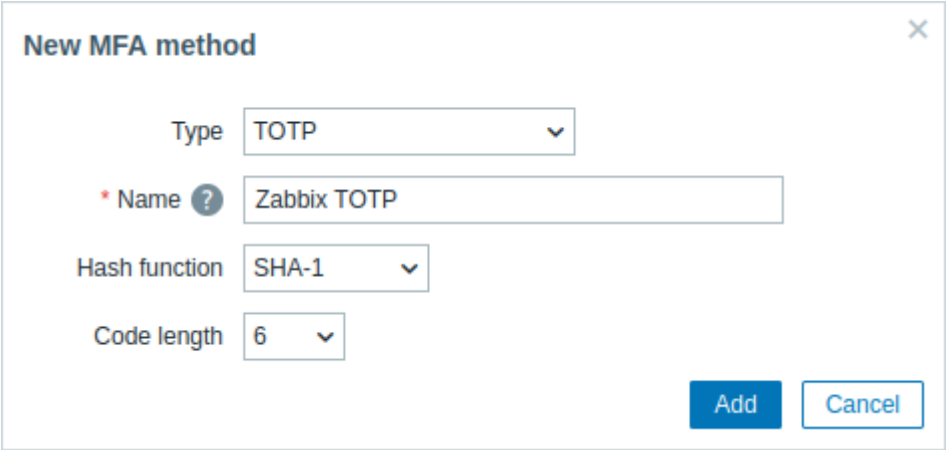
Konfiguration



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.

Parameter	Beschreibung
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 mit einer Authenticator-App verifizieren, zum Beispiel mit der App Google Authenticator.

1. Gehen Sie in Zabbix zu den MFA-Einstellungen unter *Benutzer* → *Authentifizierung* 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
- Codelänge: 6

3. Klicken Sie auf die Schaltfläche *Hinzufügen* und anschließend auf die Schaltfläche *Aktualisieren*.

4. Gehen Sie zu *Benutzer* → *Benutzergruppen* und erstellen Sie eine neue Benutzergruppe mit der folgenden Konfiguration:

- Gruppenname: TOTP-Gruppe
- Benutzer: Admin
- Multi-Faktor-Authentifizierung: Standard (oder „Zabbix TOTP“, falls diese nicht als Standard festgelegt ist)

5. Melden Sie sich von Zabbix ab und 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 erzeugt einen Verifizierungscode, den Sie eingeben müssen, um den Anmeldevorgang abzuschließen.

7. Rufen Sie bei nachfolgenden Anmeldungen 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, den Zugriff auf Zabbix über HTTPS sowie 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 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 Zugangsdaten (Client ID, Client secret, API hostname), die für die Konfiguration der MFA-Methode in Zabbix erforderlich sind.

4. Gehen Sie in Zabbix zu den MFA-Einstellungen unter *Users* → *Authentication* und aktivieren Sie die Multi-Faktor-Authentifizierung.

5. Fügen Sie eine neue MFA-Methode mit der folgenden Konfiguration hinzu:

   - Typ: Duo Universal Prompt
   - Name: Zabbix Duo
   - API hostname: (verwenden Sie den API hostname von Duo)
   - Client ID: (verwenden Sie die Client ID von Duo)
   - Client secret: (verwenden Sie das Client secret von Duo)

6. Klicken Sie auf die Schaltfläche *Add* und dann auf die Schaltfläche *Update*.

7. Gehen Sie zu *Users* → *User groups* und erstellen Sie eine neue Benutzergruppe mit der folgenden Konfiguration:

   - Gruppenname: Duo group
   - Benutzer: Admin
   - Multi-Faktor-Authentifizierung: Standard (oder "Zabbix Duo", falls dies nicht als Standard festgelegt ist)

8. Melden Sie sich von Zabbix ab und 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 nachfolgenden Anmeldungen die entsprechende MFA-Methode, die von der Duo-App bereitgestellt wird (z. B. Abrufen eines Bestätigungscodes, Reagieren auf Push-Benachrichtigungen oder Verwenden von Hardwareschlüsseln), und geben Sie die erforderlichen Informationen während der Anmeldung ein.

9 Administration

Übersicht

Das Menü **Administration** ist für administrative Funktionen von Zabbix vorgesehen. Dieses Menü ist nur für Benutzer des Benutzertyps SuperAdmin verfügbar.

1 Allgemein

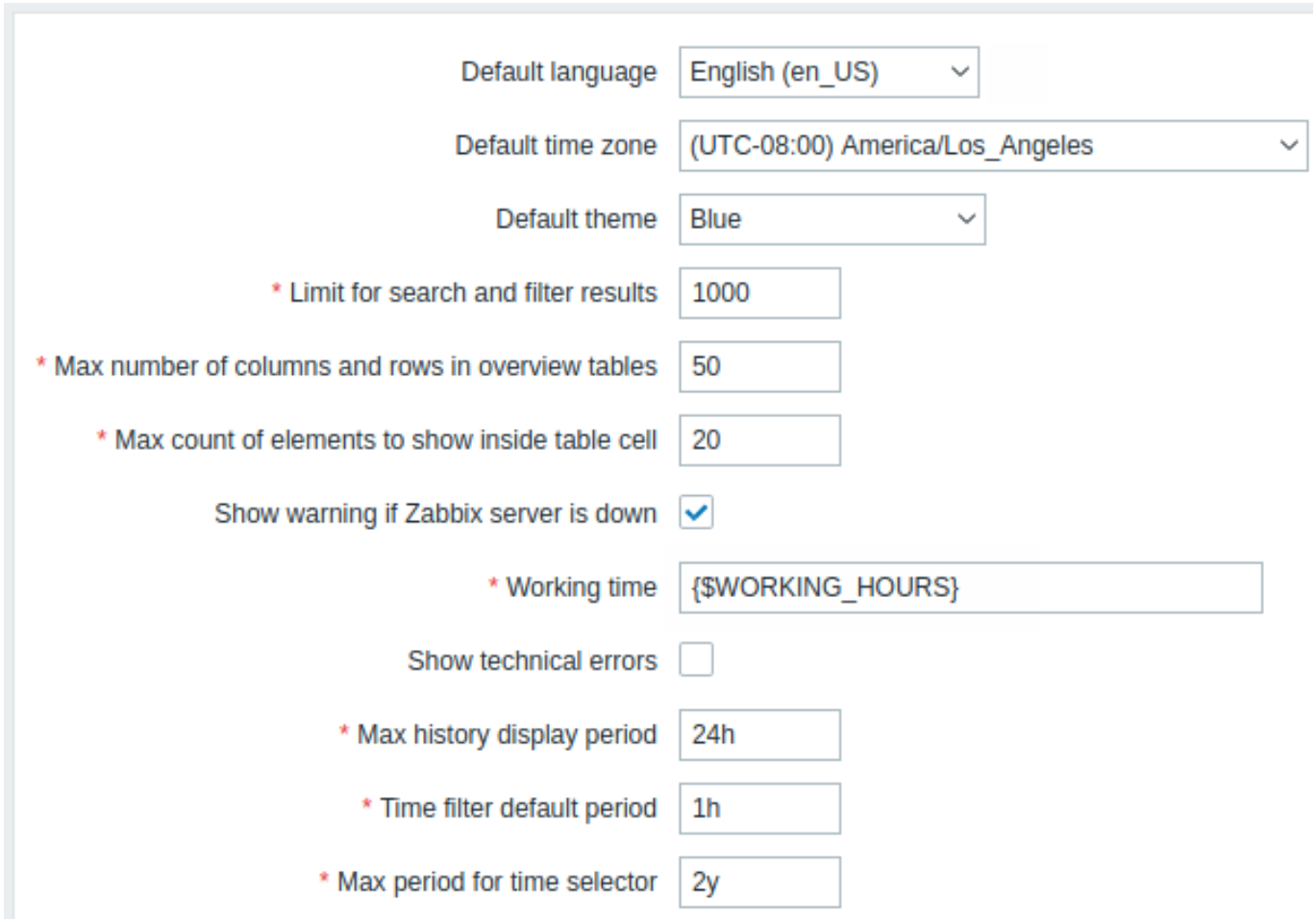
Übersicht

Der Abschnitt *Administration* → *General* enthält eine Reihe von Unterabschnitten zum Festlegen von Frontend-bezogenen Standardwerten und zur Anpassung von Zabbix.

Die Liste der verfügbaren Unterabschnitte wird angezeigt, wenn Sie im Menübereich *Administration* auf *General* klicken. Es ist auch möglich, zwischen den Unterabschnitten zu wechseln, indem Sie die Titelauswahlliste in der oberen linken Ecke verwenden.

GUI

Dieser Abschnitt ermöglicht die Anpassung mehrerer Standardwerte des Frontends.



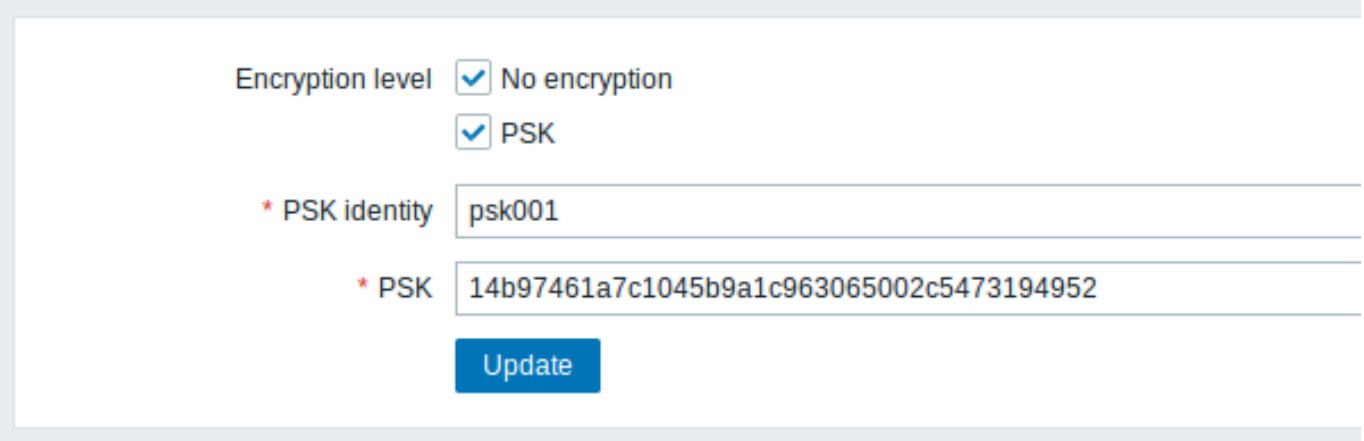
Konfigurationsparameter:

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.
Standard-Theme	Standard-Theme für Benutzer, die in ihren Profilen kein Theme angegeben haben, sowie für Gastbenutzer.
Limit für Such- und Filterergebnisse	Maximale Anzahl von Elementen (Zeilen), die in einer Liste der Weboberfläche angezeigt werden, zum Beispiel unter *Datensammlung* > *Hosts*. *Hinweis*: Wenn der Wert beispielsweise auf „50“ gesetzt ist, werden in allen betroffenen Frontend-Listen nur die ersten 50 Elemente angezeigt. Wenn eine Liste mehr als fünfzig Elemente enthält, wird dies durch das Zeichen „+“ in *„Displaying 1 to 50 of **50+** found“* angezeigt. Auch wenn ein Filter verwendet wird und weiterhin mehr als 50 Treffer vorhanden sind, werden nur die ersten 50 angezeigt. Beachten Sie, dass eine Erhöhung dieses Parameters zu geringerer Leistung und höherem Speicherverbrauch auf Seiten des Webservers führen kann.
Maximale Anzahl von Spalten und Zeilen in Übersichtstabellen	Maximale Anzahl von Spalten und Zeilen, die im Dashboard-Widget *Auslöserübersicht* angezeigt werden. Dasselbe Limit gilt sowohl für Spalten als auch für Zeilen. Wenn mehr Zeilen und/oder Spalten vorhanden sind als angezeigt werden, zeigt das System am unteren Rand der Tabelle eine Warnung an: „Not all results are displayed. Please provide more specific search criteria.“
Maximale Anzahl von Elementen zur Anzeige innerhalb einer Tabellenzelle	Bei Einträgen, die in einer einzelnen Tabellenzelle angezeigt werden, wird nicht mehr als die hier konfigurierte Anzahl angezeigt.
Warnung anzeigen, wenn der Zabbix server nicht verfügbar ist	Dieser Parameter aktiviert die Anzeige einer Warnmeldung im Browserfenster, wenn der Zabbix server nicht erreichbar ist (möglicherweise ausgefallen). Die Meldung bleibt sichtbar, auch wenn der Benutzer auf der Seite nach unten scrollt. Wenn der Mauszeiger darüber bewegt wird, wird die Meldung vorübergehend ausgeblendet, damit der darunterliegende Inhalt sichtbar wird.

Parameter	Beschreibung
Arbeitszeit	Dieser systemweite Parameter definiert die Arbeitszeiten. In Diagrammen wird die Arbeitszeit mit weißem Hintergrund und die Nicht-Arbeitszeit grau dargestellt. Eine Beschreibung des Zeitformats finden Sie auf der Seite Angabe von Zeiträumen. 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 diese Informationen nur für Zabbix Super Admins und Benutzer verfügbar, die Benutzergruppen mit aktiviertem Debug-Modus angehören.
Maximaler Anzeigezeitraum für Verlauf	Maximaler Zeitraum, für den Verlaufsdaten unter *Überwachung* > *Letzte Daten*, Informationen zu Host-web-Szenarien unter *Überwachung* > *Hosts* sowie im Dashboard-Widget *Top items* angezeigt werden. Zulässiger Bereich: 24 Stunden (Standard) bis 1 Woche. Zeitsuffixe, z. B. 1w (eine Woche), 36h (36 Stunden), werden unterstützt.
Standardzeitraum für Zeitfilter	Zeitraum, der standardmäßig in Diagrammen und Dashboards verwendet wird. Zulässiger Bereich: 1 Minute bis 10 Jahre (Standard: 1 Stunde). Zeitsuffixe, z. B. 10m (zehn Minuten), 5w (fünf Wochen), werden unterstützt. Hinweis: Wenn ein Benutzer beim Anzeigen eines Diagramms den Zeitraum ändert, wird dieser Zeitraum als Benutzereinstellung gespeichert und ersetzt den globalen Standard oder eine vorherige Benutzerauswahl.
Maximaler Zeitraum für Zeitauswahl	Maximal verfügbarer Zeitraum für Diagramme und Dashboards. Benutzer können keine älteren Daten visualisieren. Zulässiger Bereich: 1 Jahr bis 10 Jahre (Standard: 2 Jahre). Zeitsuffixe, 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.

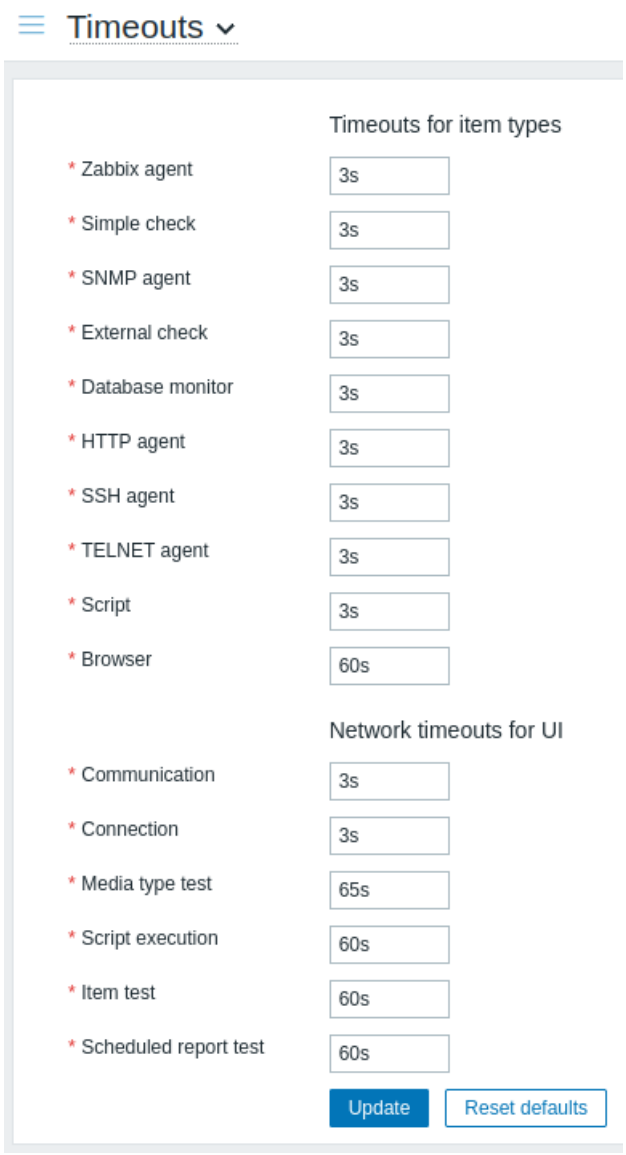
Konfigurationsparameter:

Parameter	Beschreibung
Verschlüsselungsstufe	Wählen Sie eine oder beide Optionen für die Verschlüsselungsstufe aus: **Keine Verschlüsselung** - unverschlüsselte Verbindungen sind zulässig **PSK** - TLS-verschlüsselte Verbindungen mit einem vorab geteilten Schlüssel sind zulässig
PSK-Identität	Geben Sie die Identitätszeichenfolge des vorab geteilten 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 dem Empfänger mitzuteilen, welcher PSK verwendet werden soll.
PSK	Geben Sie den vorab geteilten Schlüssel ein (eine gerade Anzahl hexadezimaler Zeichen). Maximale Länge: 512 Hex-Ziffern (256-Byte-PSK), wenn Zabbix die Bibliothek GnuTLS oder OpenSSL verwendet, 64 Hex-Ziffern (32-Byte-PSK), wenn Zabbix die Bibliothek mbed TLS (PolarSSL) 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 Datenpunkt-Typen und Netzwerk-Timeouts festgelegt werden. Alle Felder in diesem Formular sind Pflichtfelder.

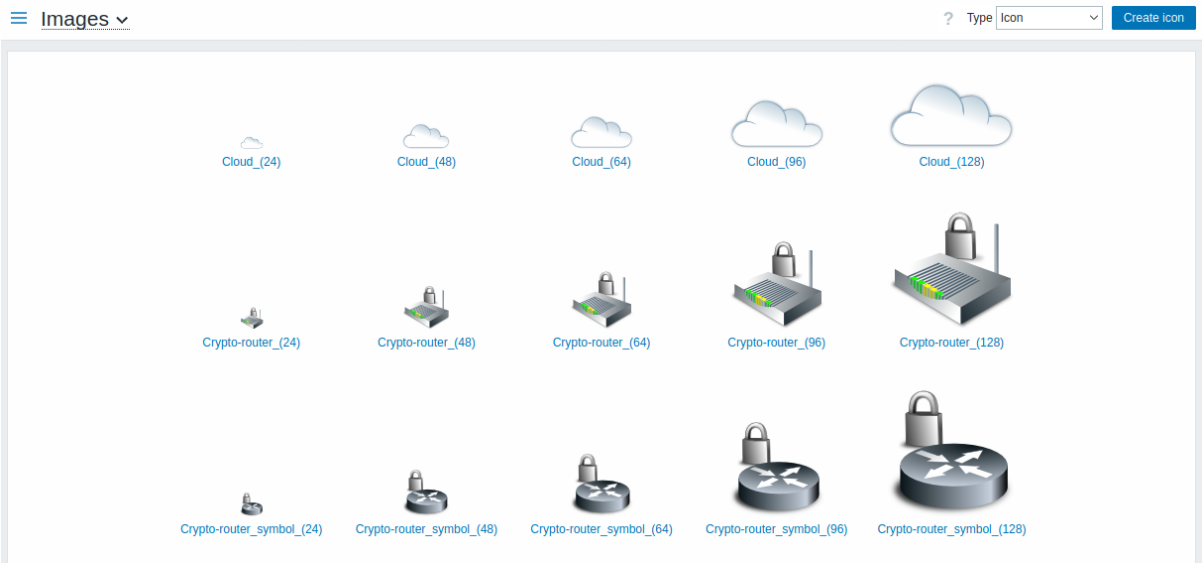


Parameter		Beschreibung
Timeouts für Datenpunkt-Typen		Wie viele Sekunden Zabbix warten soll, bevor die Prüfung abgebrochen wird (abhängig von ihrem Typ). Weitere Details und Einschränkungen finden Sie unter Datenpunkt-Timeout. Zulässiger Bereich: 1 - 600s (Standard: 3s; Standard für Browser-Datenpunkte: 60s). Zeitsuffixe, z. B. 30s, 1m, und Benutzermakros werden unterstützt. Beachten Sie, dass auf Proxy-Ebene konfigurierte Timeouts diese globalen Einstellungen überschreiben. Wenn außerdem Timeouts für bestimmte Datenpunkte konfiguriert sind, überschreiben diese sowohl die globalen als auch die Proxy-Einstellungen.
Netzwerk-Timeouts für die UI		
	Kommunikation	Wie viele Sekunden gewartet werden soll, bevor ein inaktiver Socket geschlossen wird (wenn zuvor bereits eine Verbindung zum Zabbix-Server hergestellt wurde, das Frontend jedoch innerhalb dieser Zeit das Lesen/Senden von Daten nicht abschließen kann, wird die Verbindung getrennt). Zulässiger Bereich: 1 - 300s (Standard: 3s).
	Verbindung	Wie viele Sekunden gewartet werden soll, bevor ein Verbindungsversuch zum Zabbix-Server abgebrochen wird. Zulässiger Bereich: 1 - 30s (Standard: 3s).
	Medientyp-Test	Wie viele Sekunden bei einem Medientyp-Test auf eine Antwort gewartet werden soll. Zulässiger Bereich: 1 - 300s (Standard: 65s).

Parameter	Beschreibung
Skriptausführung	Wie viele Sekunden bei der Ausführung eines Skripts auf eine Antwort gewartet werden soll. Zulässiger Bereich: 1 - 300s (Standard: 60s). Dieses Timeout gilt für die gesamte Skriptausführungskette, die unterschiedlich lang sein kann. Wenn das Skript beispielsweise auf dem Agent ausgeführt wird, erfolgt ein Roundtrip über den Server (möglicherweise auch über den 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 Server/Proxy wahrscheinlich erhöht werden. Das Server/Proxy-Timeout muss höher sein als die Aktualisierungsfrequenz der aktiven Prüfung, andernfalls kann das Server/Proxy-Timeout überschritten werden, bevor der aktive Agent das Skript überhaupt erhält. Siehe auch: Skript-Timeout
Datenpunkt-Test	Wie viele Sekunden beim Testen eines Datenpunkts auf zurückgegebene Daten gewartet werden soll. Zulässiger Bereich: 1 - 600s (Standard: 60s).
Test für geplanten Bericht	Wie viele Sekunden beim Testen eines geplanten Berichts auf zurückgegebene Daten gewartet werden soll. Zulässiger Bereich: 1 - 300s (Standard: 60s).

Bilder

Der Abschnitt **Bilder** zeigt alle in Zabbix verfügbaren Bilder an. Bilder werden in der Datenbank gespeichert.

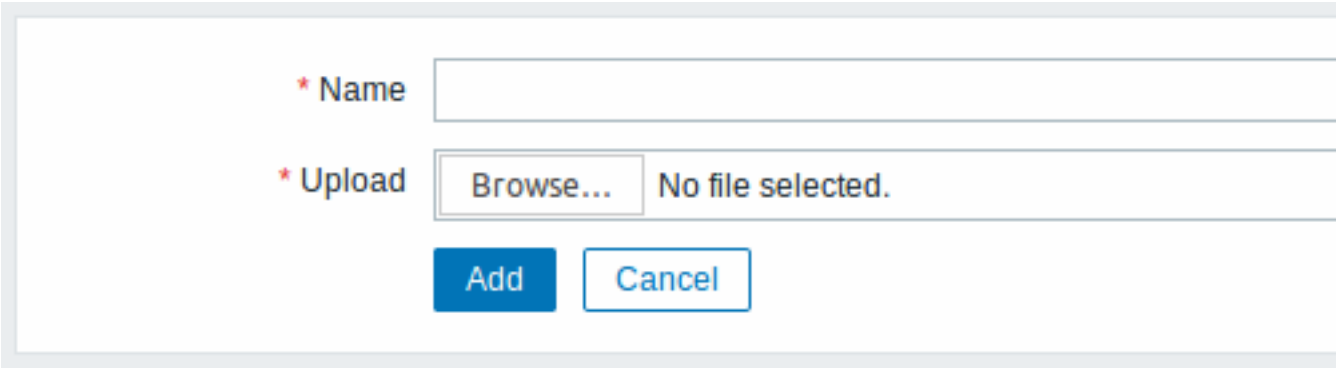


Über das Dropdown-Menü *Typ* können Sie zwischen Symbol- und Hintergrundbildern wechseln:

- Symbole werden zur Anzeige von Elementen in der Netzwerkkarte 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, um sie nach Zabbix hochzuladen. *Beachten Sie*, dass möglicherweise auch andere Formate hochgeladen werden können, 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 (2.0.28 oder höher wird von Zabbix benötigt).

Note:
Die maximale Größe der Upload-Datei ist durch den Wert von ZBX_MAX_IMAGE_SIZE begrenzt, der 1024x1024 Byte bzw. 1 MB beträgt.

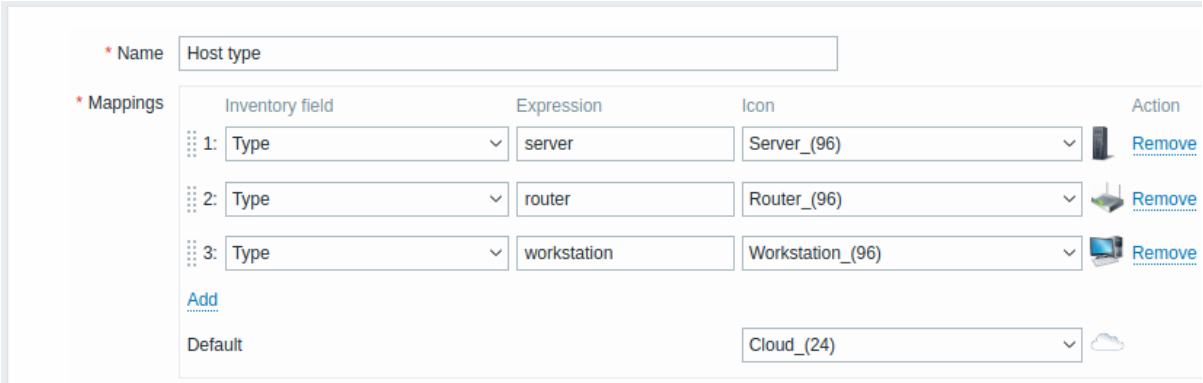
Der Upload eines Bildes kann fehlschlagen, wenn die Bildgröße nahe bei 1 MB liegt und der MySQL-Konfigurationsparameter `max_allowed_packet` auf dem Standardwert von 1 MB steht. Erhöhen Sie in diesem Fall den Parameter max_allowed_packet.

Symbolzuordnung

In diesem Abschnitt können Zuordnungen bestimmter Hosts zu bestimmten Symbolen erstellt werden. Informationen aus den Host-Inventarfeldern werden verwendet, um die Zuordnung zu erstellen.

Die Zuordnungen können dann in der Konfiguration von Netzwerkkarten verwendet werden, um passenden Hosts automatisch geeignete Symbole zuzuweisen.

Um eine neue Symbolzuordnung zu erstellen, klicken Sie oben rechts auf *Symbolzuordnung erstellen*.



Konfigurationsparameter:

Parameter	Beschreibung
Name	Eindeutiger Name der Symbolzuordnung.
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	Host-Inventarfeld, das auf eine Übereinstimmung geprüft wird.
Expression	Regulärer Ausdruck, der die Übereinstimmung beschreibt.
Icon	Zu verwendendes Symbol, wenn eine Übereinstimmung mit dem Ausdruck gefunden wird.
Default	Standardmäßig zu verwendendes Symbol.

Reguläre Ausdrücke

In diesem Abschnitt können benutzerdefinierte reguläre Ausdrücke erstellt werden, die an mehreren Stellen im Frontend verwendet werden können. Weitere Einzelheiten finden Sie im Abschnitt Reguläre Ausdrücke.

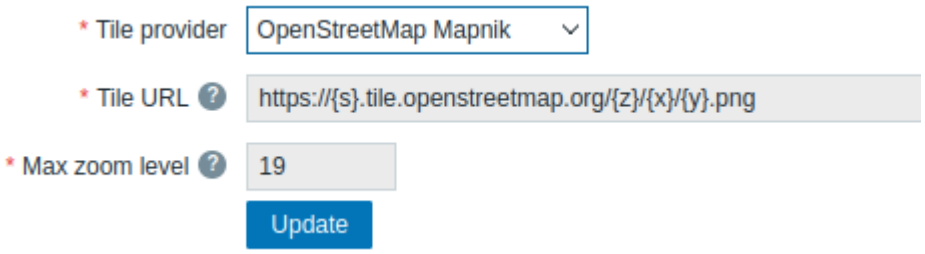
Optionen zur Anzeige von Auslösern

In diesem Abschnitt können Sie anpassen, wie der Auslöserstatus im Frontend sowie die Namen und Farben der Auslöser-Schweregrade angezeigt werden.

Parameter	Beschreibung
Benutzerdefinierte Farben für Ereignisstatus verwenden	Wenn Sie diesen Parameter aktivieren, wird die Anpassung der Farben für bestätigte/nicht bestätigte Probleme eingeschaltet.
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 aus der bereitgestellten Palette eine neue auszuwählen. Wenn das Kontrollkästchen *blinking* markiert ist, blinken Auslöser nach einer Statusänderung für некоторое Zeit, damit sie besser sichtbar sind.
OK-Auslöser anzeigen für	Zeitraum für die Anzeige von OK-Auslösern. Zulässiger Bereich: 0 - 24 Stunden. Zeitsuffixe, z. B. 5m, 2h, 1d, werden unterstützt.
Bei Statusänderung blinken Auslöser für	Dauer des Blinkens von Auslösern. Zulässiger Bereich: 0 - 24 Stunden. Zeitsuffixe, z. B. 5m, 2h, 1d, werden unterstützt.
Nicht klassifiziert, *Information*, *Warnung*, *Durchschnittlich*, *Hoch*, *Katastrophe*	Benutzerdefinierte Schweregradnamen und/oder Farben, die anstelle der Systemstandardwerte angezeigt werden. Geben Sie einen neuen Farbcode ein oder klicken Sie auf die Farbe, um aus der bereitgestellten Palette eine neue auszuwählen. Beachten Sie, dass hier eingegebene benutzerdefinierte Schweregradnamen in allen Gebietsschemas verwendet werden. Wenn Sie sie für bestimmte Benutzer in andere Sprachen übersetzen müssen, siehe die Seite Anpassen von Auslöser-Schweregraden.

Geografische Karten

In diesem Abschnitt können Sie den Anbieter des Kachel-Dienstes für geografische Karten auswählen und die Einstellungen des Anbieters für das Geomap-Dashboard-Widget konfigurieren. Für die Visualisierung mit geografischen Karten verwendet Zabbix die Open-Source-JavaScript-Bibliothek Leaflet für interaktive Karten. Bitte beachten Sie, dass Zabbix keinen Einfluss auf die Qualität der von Drittanbieter-Kachel-Anbietern bereitgestellten Bilder hat, einschließlich der vordefinierten Kachel-Anbieter.



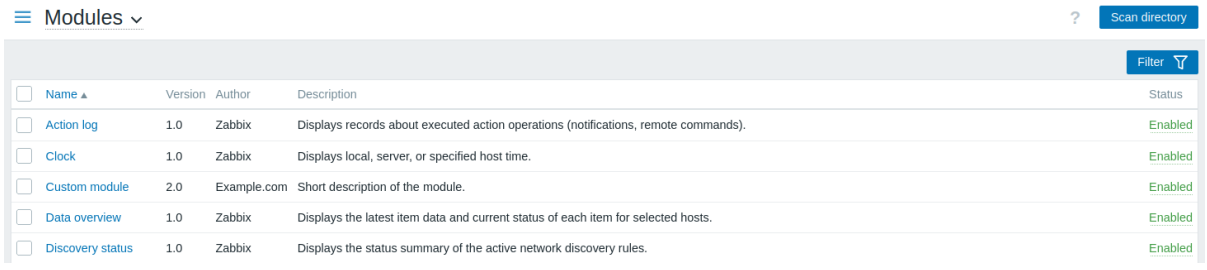
Parameter	Beschreibung
Tile provider	Wählen Sie einen der verfügbaren Kachel-Dienstanbieter aus oder wählen Sie *Other*, um einen weiteren Kachel-Anbieter oder selbst gehostete Kacheln hinzuzufügen (siehe Verwendung eines benutzerdefinierten Kachel-Dienstanbieters).
Tile URL	Die URL-Vorlage (bis zu 2048 Zeichen) zum Laden und Anzeigen der Kachel-Ebene 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 Zoomstufen-Parameter 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`
Attribution text	Der Quellenhinweis des Kachel-Anbieters, der in einem kleinen Textfeld auf der Karte angezeigt wird. Dieses Feld ist nur sichtbar, wenn *Tile provider* auf *Other* gesetzt ist.
Max zoom level	Die maximale Zoomstufe der Karte. Dieses Feld ist nur bearbeitbar, wenn *Tile provider* auf *Other* gesetzt ist.

Verwenden eines benutzerdefinierten Kacheldienstanbieters

Das Geomap-Widget kann Raster-Kachelbilder von einem benutzerdefinierten selbst gehosteten oder einem Drittanbieter-Kacheldienst laden. Um einen benutzerdefinierten Kacheldienst eines Drittanbieters oder einen selbst gehosteten Kachelordner oder Server zu verwenden, wählen Sie *Other* im Feld *Tile provider* aus und geben Sie die benutzerdefinierte URL im Feld *Tile URL* unter Verwendung der entsprechenden Platzhalter an.

Module

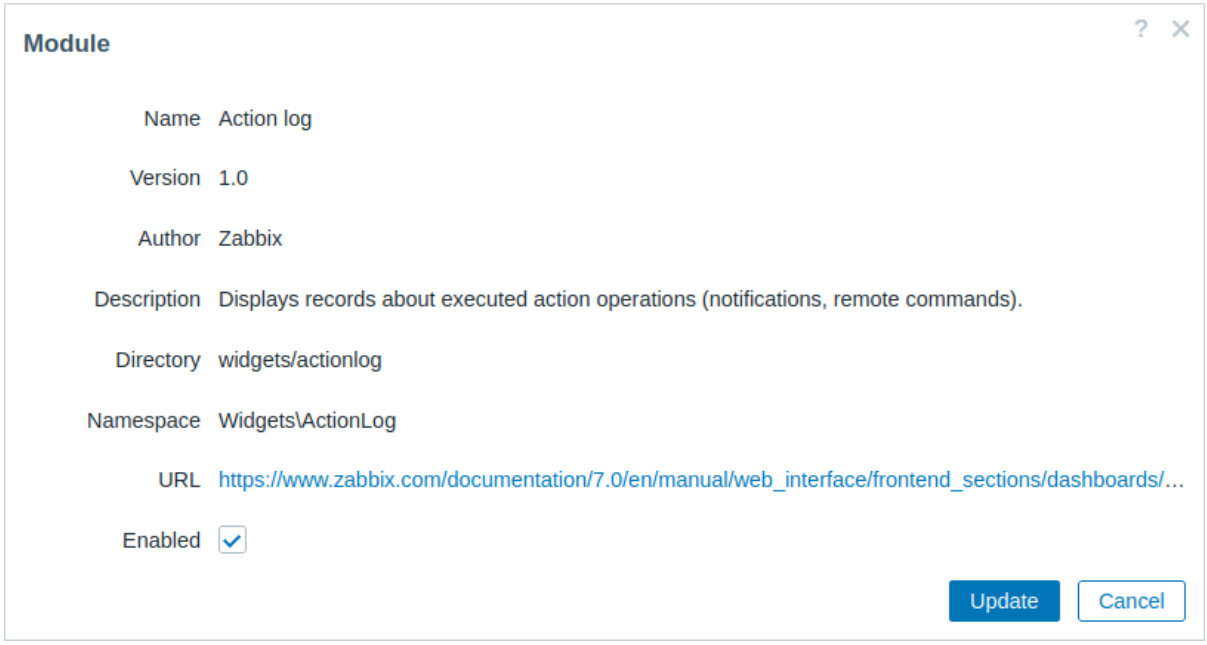
In diesem Abschnitt können benutzerdefinierte sowie integrierte Frontend-Module verwaltet werden.



Klicken Sie auf *Scan directory*, um benutzerdefinierte Module zu registrieren bzw. die Registrierung aufzuheben. Registrierte Module werden in der Liste angezeigt; nicht registrierte Module werden aus der Liste entfernt.

Klicken Sie in der Liste auf den Modulstatus, um ein Modul zu aktivieren/deaktivieren. Sie können Module auch gesammelt aktivieren/deaktivieren, indem Sie sie in der Liste auswählen und anschließend auf die Schaltflächen *Enable/Disable* unterhalb der Liste klicken.

Klicken Sie in der Liste auf den Modulnamen, um dessen Details in einem Pop-up-Fenster anzuzeigen.

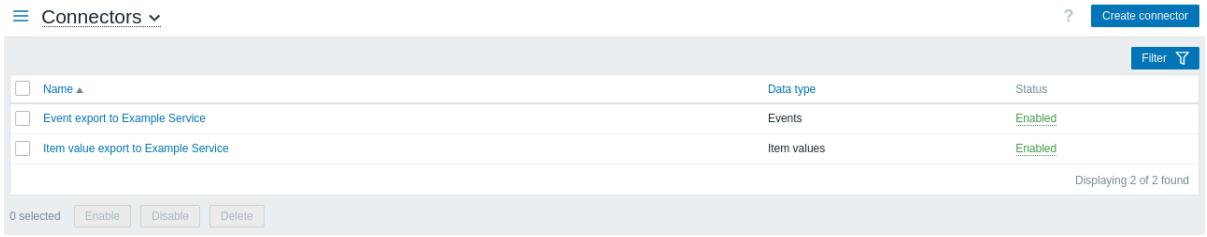


Der Modulstatus kann auch im Pop-up-Fenster mit den Moduldetails aktualisiert werden; markieren bzw. deaktivieren Sie dazu das Kontrollkästchen *Enabled* und klicken Sie anschließend auf *Update*.

Sie können Module nach Name oder Status (aktiviert/deaktiviert) filtern.

Konnektoren

In diesem Abschnitt können Konnektoren für das Streaming von Zabbix-Daten an externe Systeme über HTTP konfiguriert werden.



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 des Konnektors, um ihn zu aktivieren/deaktivieren.
Sie können Konnektoren auch gesammelt aktivieren/deaktivieren, indem Sie sie in der Liste auswählen und dann unterhalb der Liste auf die Schaltflächen *Aktivieren/Deaktivieren* 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.
Group for discovered hosts	Hosts, die durch Netzwerkerkennung und Agent-Autoregistrierung erkannt wurden, werden automatisch in die hier ausgewählte Hostgruppe aufgenommen.
Default host inventory mode	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 *Set host inventory mode* überschrieben wird.

Parameter	Beschreibung
User group for database down message	Benutzergruppe für das Senden einer Alarmmeldung oder 'None'. 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 Alarm-Skripte. 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.
Log unmatched SNMP traps	SNMP-Trap protokollieren, wenn keine entsprechenden SNMP-Schnittstellen gefunden wurden.

Autorisierung

Parameter	Beschreibung
Anmeldeversuche *Sperrintervall für Anmeldungen*	Anzahl erfolgloser Anmeldeversuche, bevor die Möglichkeit zur Anmeldung gesperrt wird. Zeitraum, für den die Anmeldung untersagt wird, wenn das Limit für *Anmeldeversuche* überschritten wird. Zulässiger Bereich: 0 - 3600 Sekunden. Zeitsuffixe, z. B. 90s, 5m, 1h, werden unterstützt.

Speicherung von Geheimnissen

Parameter	Beschreibung
Vault-Anbieter	Wählen Sie die Software zur Verwaltung von Geheimnissen für die Speicherung von Werten der Benutzermakros aus - *HashiCorp Vault* (Standard) oder *CyberArk Vault*.
Secret-Vault-Makros auflösen durch	Secret-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 diese Makros unabhängig auflösen können.

Siehe auch: Speicherung von Geheimnissen.

Sicherheit

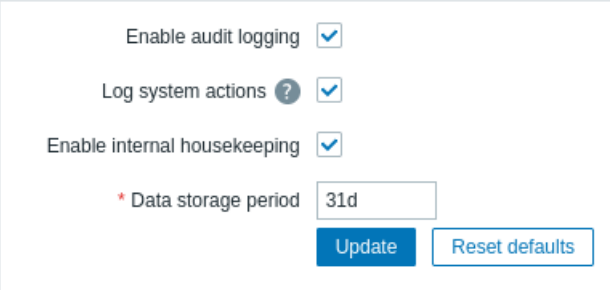
Parameter	Beschreibung
URI-Schemata validieren	Deaktivieren Sie dieses Kontrollkästchen, um die Validierung von URI-Schemata zu deaktivieren (standardmäßig aktiviert). Wenn es aktiviert ist, können Sie eine kommagetrennte Liste zulässiger URI-Schemata angeben (Standard: http,https,ftp,file,mailto,tel,ssh). 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`.

2 Audit-Protokoll

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.

3 Bereinigung

Übersicht

Im Abschnitt *Administration* > *Housekeeping* können Sie das interne Housekeeping von Zabbix konfigurieren, das automatisch veraltete oder vom Benutzer gelöschte Daten aus der Datenbank entfernt. Dadurch wird verhindert, dass die Datenbank unbegrenzt anwächst, was unnötigen Speicherplatz verbrauchen und die Leistung der Datenbank beeinträchtigen würde.

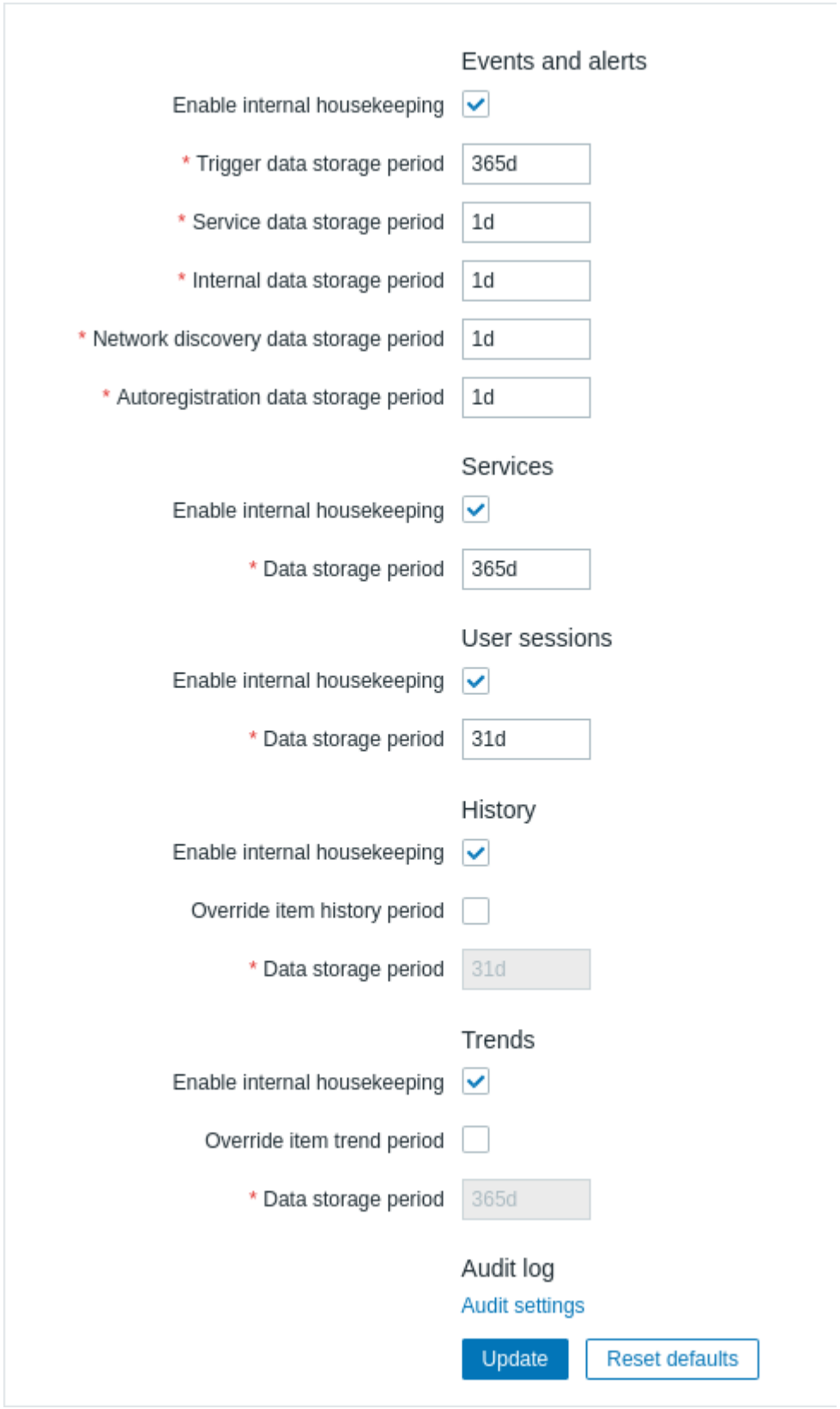
Housekeeping kann für die folgenden Datentypen aktiviert und konfiguriert werden:

- Ereignisse und Warnungen aus Auslösern, Services, internen Daten, der Netzwerkerkennung und der automatischen Registrierung
- Services
- Benutzersitzungen
- Verlauf und Trends für Datenpunkte

Note:
Das Housekeeping des Auditprotokolls wird separat konfiguriert.

Konfiguration

Die folgenden Parameter sind verfügbar:



Parameter	Beschreibung
Interne Bereinigung aktivieren	Aktiviert oder deaktiviert die interne Bereinigung (standardmäßig aktiviert). Wenn aktiviert, entfernt das Bereinigungsverfahren automatisch Daten aus der Datenbank, die den *Datenspeicherzeitraum* überschreiten.

Parameter	Beschreibung
Datenspeicherzeitraum	Geben Sie an, wie lange Daten aufbewahrt werden, bevor sie von der Bereinigung entfernt werden. Erforderlich, wenn die interne Bereinigung aktiviert ist. Bereich: 1 Tag (1 Stunde für Verlauf) - 25 Jahre; oder „0“. Zeitsuffixe (z. B. 1d, 1w) werden unterstützt. Für *Ereignisse und Warnungen* wird der Datenspeicherzeitraum separat für Auslöser, Services, interne Daten, Netzwerkdiscovery und Autoregistrierung festgelegt. Die Bereinigung entfernt nur Ereignisse, die nicht mit Problemen verknüpft sind. Beispielsweise wird ein Problem-/Wiederherstellungsereignis, das älter als der *Datenspeicherzeitraum* ist, nicht entfernt, wenn es mit einem ungelösten Problem verknüpft ist. Wenn die Bereinigung veraltete Entitäten entfernt, entfernt sie zuerst Probleme und dann Ereignisse. Für *Verlauf* und *Trends* bestimmen die Datenspeicherzeiträume auch, wie lange Daten unter *Überwachung* > *Letzte Daten* sichtbar bleiben, selbst wenn die interne Bereinigung deaktiviert ist. Beachten Sie, dass beim Löschen eines Datenpunkts oder Auslösers auch dessen Probleme und zugehörige Ereignisse gelöscht werden.
Datenpunkt-Verlaufszeitraum überschreiben	Wenn aktiviert, wird der in der Datenpunkt-Konfiguration angegebene Speicherzeitraum für Verlauf/Trends durch die Einstellung *Datenspeicherzeitraum* überschrieben (außer bei Datenpunkten, für die die Option *Nicht speichern* aktiviert ist). Diese Option kann auch verwendet werden, wenn die interne Bereinigung deaktiviert ist und eine externe Bereinigung verwendet wird.
Datenpunkt-Trendzeitraum überschreiben	

Bei Verwendung von **TimescaleDB** wird der Abschnitt *Komprimierung von Verlauf, Trends und Audit-Log* verfügbar.

Note:
Für TimescaleDB aktivieren Sie *Datenpunkt-Verlaufszeitraum überschreiben*, *Datenpunkt-Trendzeitraum überschreiben* und *Interne Bereinigung aktivieren* für Verlauf und Trends, um den vollen Nutzen aus der automatischen Partitionierung zu ziehen. Wenn diese Optionen deaktiviert sind, werden die in den Tabellen für Verlauf und Trends gespeicherten Daten weiterhin partitioniert, aber die Bereinigung 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 der Verlauf für diese Datenpunkte wird gelöscht, wenn eine veraltete Partition entfernt wird.

Mit der Schaltfläche *Standardwerte zurücksetzen* können Sie alle vorgenommenen Änderungen rückgängig machen.

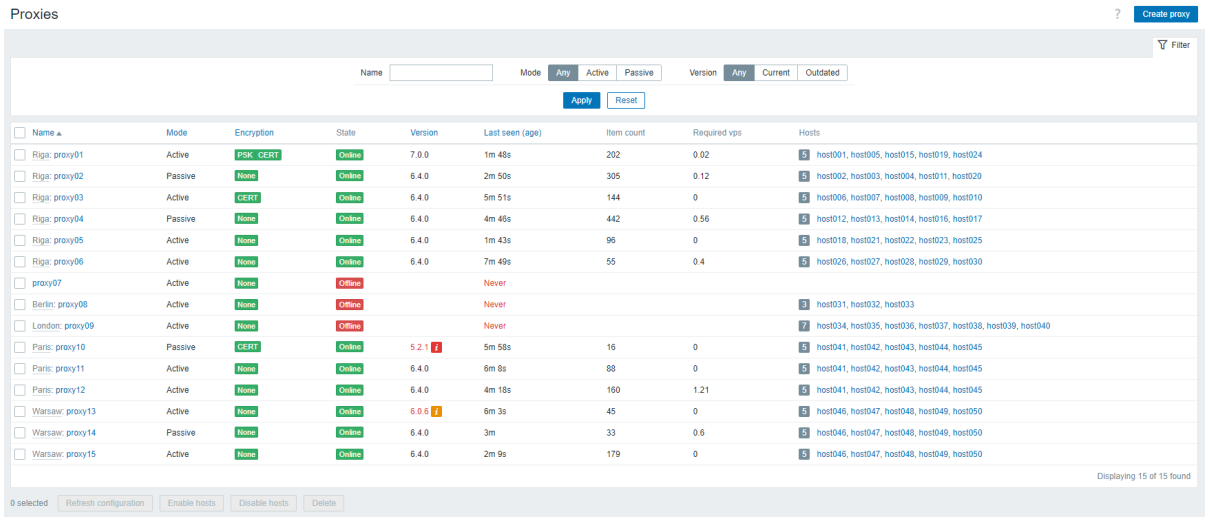
4 Proxys

Übersicht

Im Abschnitt *Administration* → *Proxies* können Proxys für das verteilte Monitoring im Zabbix Frontend konfiguriert werden.

Proxys

Eine Liste der vorhandenen Proxys mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Proxy. Ein Klick auf den Proxy-Namen öffnet das Konfigurationsformular des Proxy. Wenn der Proxy zu einer Proxy-Gruppe gehört, wird der Gruppenname vor dem Proxy-Namen als grauer Link angezeigt. Ein Klick auf den Gruppennamen öffnet das Konfigurationsformular der Proxy-Gruppe.
Mode	Proxy-Modus – *Aktiv* oder *Passiv*.
Encryption	Verschlüsselungsstatus für Verbindungen vom Proxy: **None** – keine Verschlüsselung; **PSK** – Verwendung eines vorinstallierten Schlüssels; **Cert** – Verwendung eines Zertifikats.
State	Status des Proxy: **Unknown** – 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 (ein passiver Proxy hat auf eine Server-Anfrage geantwortet; ein aktiver 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.
Last seen (age)	Der Zeitpunkt, zu dem der Proxy zuletzt vom Server gesehen wurde.
Item count	Die Anzahl aktivierter Datenpunkte auf aktivierten Hosts, die dem Proxy zugewiesen sind.
Required vps	Erforderliche Proxy-Leistung (die Anzahl der Werte, die pro Sekunde erfasst werden müssen).
Hosts	Anzahl aktivierter Hosts, die dem Proxy zugewiesen sind, sowie eine Liste der vom Proxy überwachten Hosts. Ein Klick auf den Host-Namen öffnet das Host-Konfigurationsformular.

Um einen neuen Proxy zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create proxy*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

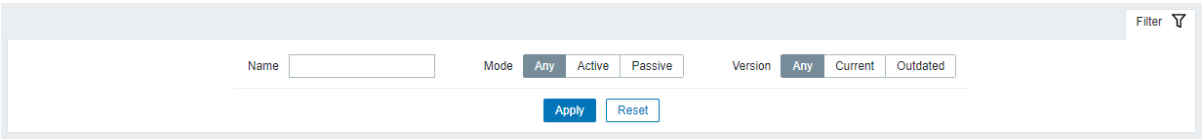
- *Konfiguration aktualisieren* - die Konfiguration der Proxys aktualisieren;
- *Hosts aktivieren* - den Status der vom Proxy überwachten Hosts auf *Überwacht* ändern;
- *Hosts deaktivieren* - den Status der vom Proxy überwachten Hosts auf *Nicht überwacht* ändern;
- *Löschen* - die Proxys löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Proxys 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 die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Proxy-Liste verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Proxys nach Name, Modus und Version filtern können. Beachten Sie, dass die Filteroption *Outdated* sowohl veraltete (teilweise unterstützte) als auch nicht unterstützte Proxys anzeigt.



5 Proxy-Gruppen

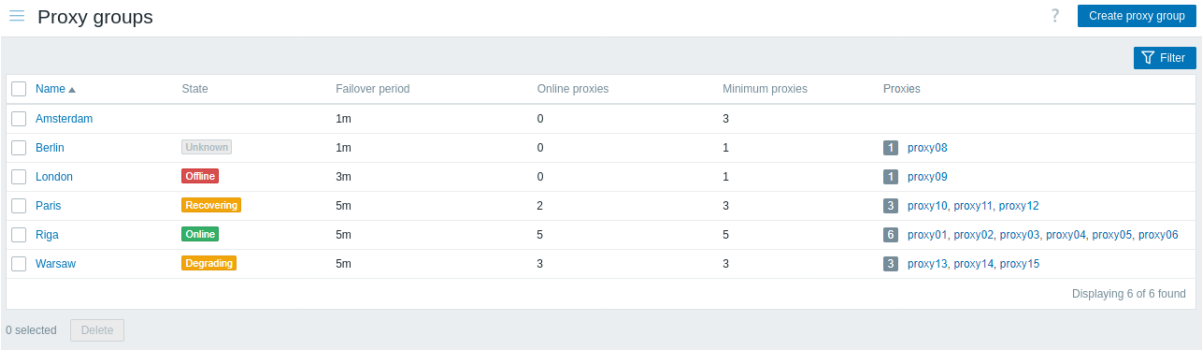
Übersicht

Unter *Administration* → *Proxy groups* können Proxy-Gruppen konfiguriert werden.

Proxy-Gruppen werden für Lastausgleich von Proxys mit automatisierter Verteilung von Hosts zwischen Proxys und Hochverfügbarkeit zwischen Proxys verwendet.

Proxy-Gruppen

Eine Liste der vorhandenen Proxy-Gruppen mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Proxy-Gruppe. Ein Klick auf den Namen der Proxy-Gruppe öffnet das Konfigurationsformular der Proxy-Gruppe.
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; **Verschlechternd** - die Gruppe wird in Kürze offline gehen, da die Anzahl der Online-Proxys unter den Schwellenwert *Minimale Proxys* fällt; **Offline** - weniger als die Mindestanzahl an Proxys hat innerhalb des Failover-Zeitraums mit dem Zabbix Server kommuniziert; **Wiederherstellend** - die Gruppe wird in Kürze wieder online kommen, da die Anzahl der Online-Proxys den Schwellenwert *Minimale Proxys* erreicht. 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 Proxys	Mindestanzahl an Online-Proxys, die erforderlich ist, damit die Proxy-Gruppe online bleibt.
Proxys	Anzahl der Proxys in der Gruppe und eine Liste der Proxys in der Gruppe, mit Links zum Proxy-Konfigurationsformular. Die maximale Anzahl der aufgelisteten Proxys ist durch den Wert *Max count of elements to show inside table cell* begrenzt.

Um eine neue Proxy-Gruppe zu konfigurieren, klicken Sie auf die Schaltfläche *Create proxy groups* in der oberen rechten Ecke.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

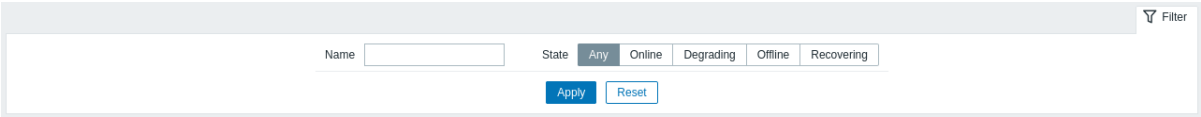
- *Löschen* - die Proxy-Gruppen löschen.

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.

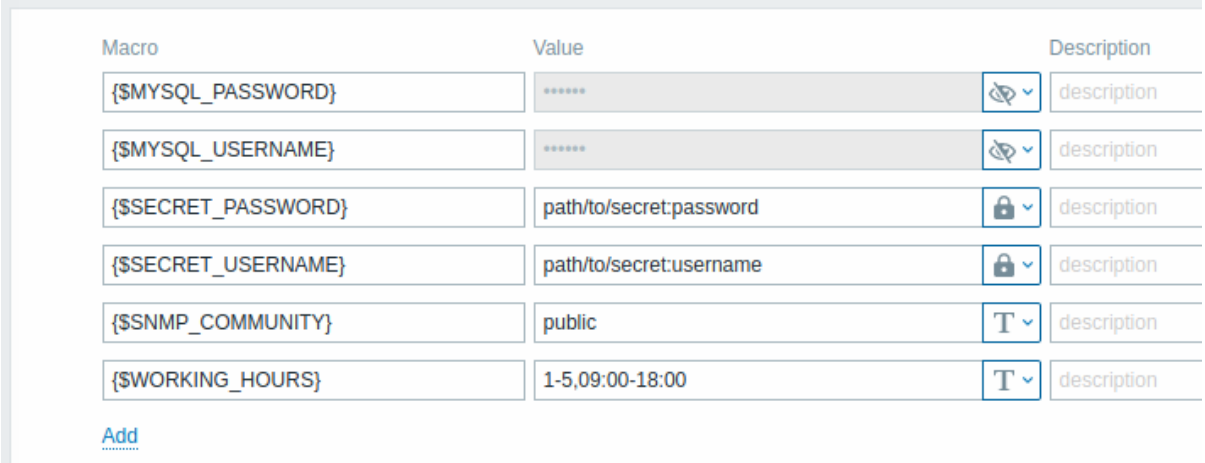
Der Link *Filter* ist oberhalb der Liste der Proxy-Gruppen verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Proxy-Gruppen nach Name und Status filtern können.



6 Makros

Übersicht

In diesem Abschnitt können systemweite Benutzer-Makros als Name-Wert-Paare definiert werden. Beachten Sie, dass Makrowerte als Klartext, geheimer Text oder Vault-Geheimnis gespeichert werden können. Das Hinzufügen einer Beschreibung wird ebenfalls unterstützt.



7 Warteschlange

Übersicht

Im Abschnitt *Administration* → *Queue* werden Datenpunkte angezeigt, die auf eine Aktualisierung warten.

Idealerweise sollte beim Öffnen dieses Abschnitts alles „grün“ sein, was bedeutet, dass sich keine Datenpunkte in der Queue befinden. Wenn alle Datenpunkte ohne Verzögerung aktualisiert werden, wartet keiner. Aufgrund unzureichender Server-Leistung können sich jedoch einige Datenpunkte verzögern, und diese Information wird in diesem Abschnitt angezeigt. Weitere Details finden Sie im Abschnitt Queue.

Note:
Der Zabbix Proxy berücksichtigt keine Wartungszeiträume; siehe Berechnung von Queues während der Wartung für Details.

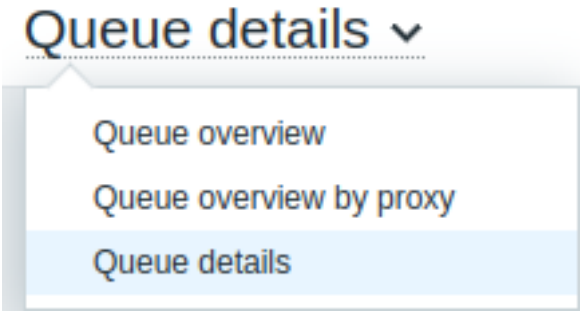
Note:
Die Queue ist nur verfügbar, wenn der Zabbix Server läuft. Datenpunkte werden in der Queue nicht gezählt, wenn die Schnittstelle des Datenpunkts aufgrund von Verbindungsproblemen nicht verfügbar wird oder der Agent nicht ordnungsgemäß funktioniert.

Der Abschnitt *Administration* → *Queue* enthält die folgenden Seiten:

- Queue-Übersicht — zeigt die Queue nach Datenpunkttyp an;
- Queue-Übersicht nach Proxy — zeigt die Queue nach Proxy an;
- Queue-Details — zeigt eine Liste verzögerter Datenpunkte an.

Die Liste der verfügbaren Seiten erscheint, wenn Sie im Menüabschnitt *Administration* auf *Queue* klicken. Es ist auch möglich, zwischen den Seiten zu wechseln, indem Sie die Titelauswahl in der oberen linken Ecke verwenden.

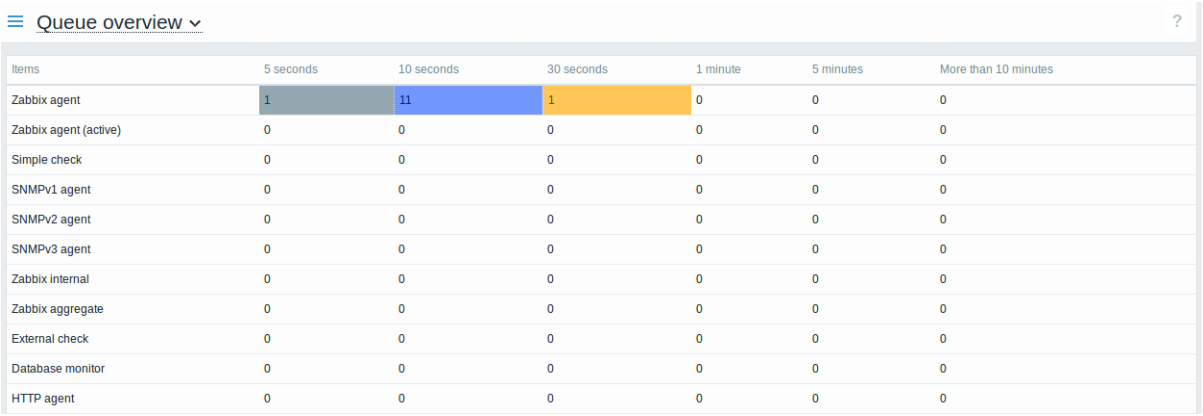
Menü der dritten Ebene.



Titelauswahl.

Übersicht nach Datenpunkttyp

In diesem Bildschirm lässt sich leicht erkennen, ob das Problem mit einem oder mehreren Datenpunkttypen zusammenhängt.

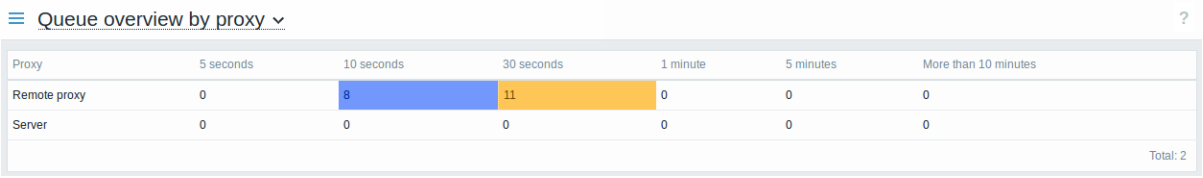


Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External 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.



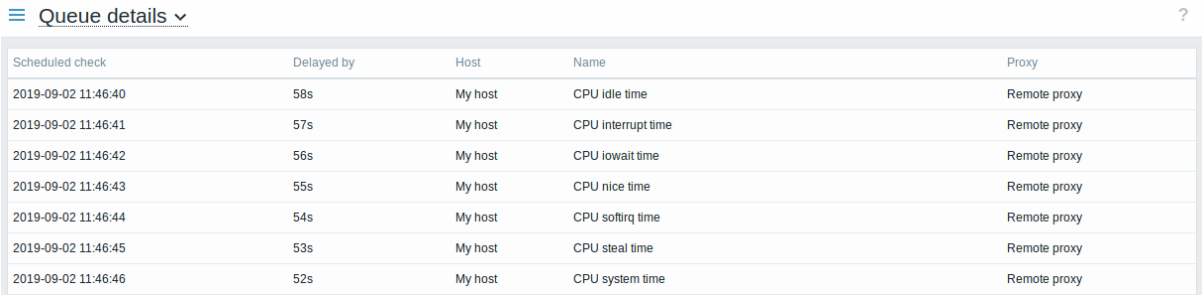
Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

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.

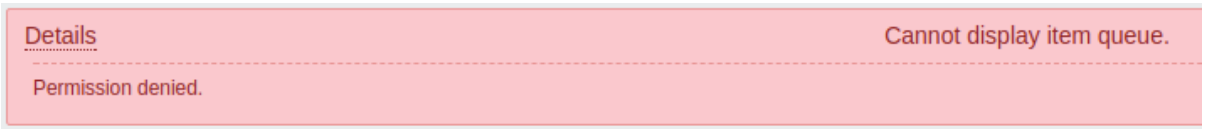


Angezeigte Daten:

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 wie folgt:

```
Cannot display item queue. Permission denied
```

Dies passiert, wenn die PHP-Konfigurationsparameter in der Datei *zabbix.conf.php* – `$ZBX_SERVER` oder sowohl `$ZBX_SERVER` als auch `$ZBX_SERVER_PORT` – auf einen vorhandenen Zabbix-Server verweisen, der eine andere Datenbank verwendet.

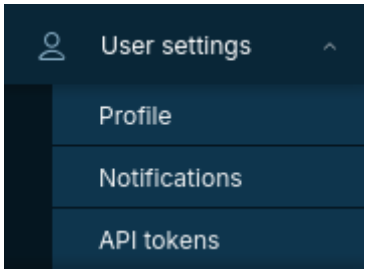
3 Benutzereinstellungen

Übersicht

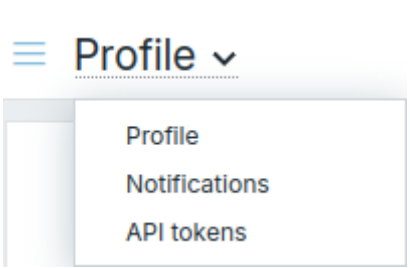
Abhängig von den Berechtigungen der Benutzerrolle kann der Abschnitt *Benutzereinstellungen* die folgenden Seiten enthalten:

- *Profil* oder *Benutzerprofil* – zum Anpassen bestimmter Funktionen des Zabbix Frontends;
- *Benachrichtigungen* – zum Anpassen der Benachrichtigungen des aktuellen Benutzers;
- *API-Tokens* – zum Verwalten der dem aktuellen Benutzer zugewiesenen API-Tokens.

Die Liste der verfügbaren Seiten wird angezeigt, wenn auf das Benutzersymbol nahe dem unteren Rand des Zabbix-Menüs geklickt wird (für den Benutzer *guest* nicht verfügbar). Es ist auch möglich, mithilfe der Titelauswahl in der oberen linken Ecke zwischen den Seiten zu wechseln.



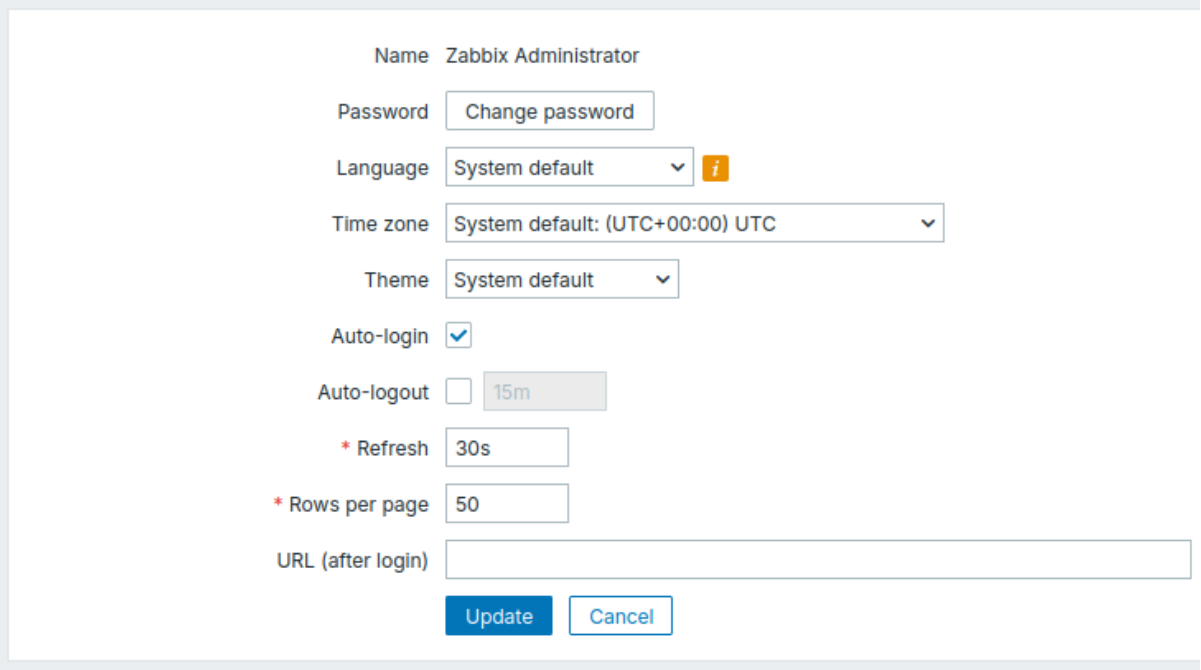
Menü der dritten Ebene.



Titelauswahl.

Benutzerprofil

Der Abschnitt *Benutzerprofil* bietet Optionen zum Festlegen einer benutzerdefinierten Oberflächensprache, eines Farbschemas, der Anzahl der in Listen angezeigten Zeilen usw. Die hier vorgenommenen Änderungen werden nur auf den aktuellen Benutzer angewendet.



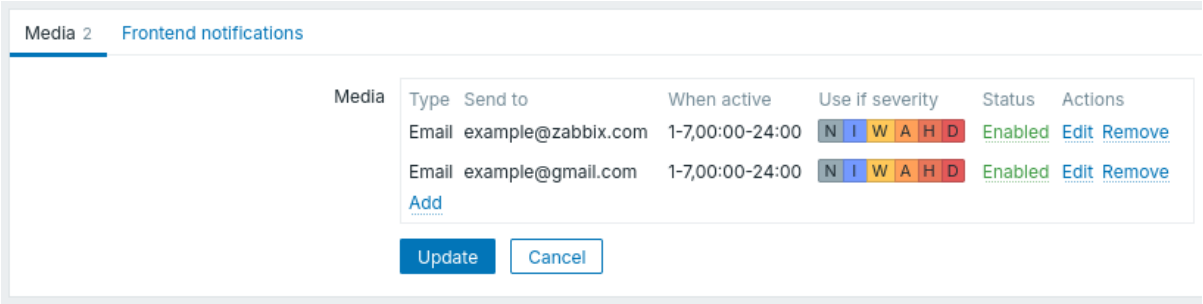
Parameter	Beschreibung
Name	Vorname und Nachname des Benutzers werden angezeigt. Falls kein Vor- und Nachname vorhanden sind, wird der Benutzername angezeigt.
Password	Klicken Sie auf die Schaltfläche *Passwort ändern*, um drei Felder zu öffnen: *Altes Passwort*, *Neues Passwort*, *Neues Passwort (noch einmal)*. Nach einer erfolgreichen Passwortänderung wird der Benutzer aus allen aktiven Sitzungen abgemeldet. Beachten Sie, dass das Passwort nur für Benutzer geändert werden kann, die die Zabbix-interne Authentifizierung verwenden.
Language	Wählen Sie die gewünschte Oberflächensprache aus oder wählen Sie **Systemstandard**, um die Standard-Systemeinstellungen zu verwenden. Die Auswahl von *English (en_US)* aktiviert außerdem das US-Zeit-/Datumsformat im Frontend. Weitere Informationen finden Sie unter Installation zusätzlicher Frontend-Sprachen.
Time zone	Wählen Sie die Zeitzone aus, um die globale Zeitzone auf Benutzerebene zu überschreiben, oder wählen Sie **Systemstandard**, um die globalen Zeitzoneneinstellungen zu verwenden.
Theme	Wählen Sie ein Farbschema speziell für Ihr Profil aus: **Systemstandard** - Standard-Systemeinstellungen verwenden; **Blue** - standardmäßiges blaues Farbschema; **Blue (classic)** - standardmäßiges blaues Farbschema mit Legacy-Schriftarten; **Dark** - alternatives dunkles Farbschema; **Dark (classic)** - alternatives dunkles Farbschema mit Legacy-Schriftarten; **High-contrast light** - helles Farbschema mit hohem Kontrast; **High-contrast dark** - dunkles Farbschema mit hohem Kontrast.
Auto-login	Aktivieren Sie dieses Kontrollkästchen, damit Zabbix sich an Sie erinnert und Sie 30 Tage lang automatisch anmeldet. Bei der Anmeldung mit aktivierter Option *Remember for 30 days*: - Der Timer für die automatische Abmeldung wird zurückgesetzt (die Sitzung bleibt bestehen, bis der Zeitraum von 30 Tagen endet). - Die automatische Anmeldung ist bei nachfolgenden Besuchen innerhalb von 30 Tagen aktiviert. Bei der Anmeldung ohne *Remember for 30 days*: - Der Parameter für die automatische Anmeldung wird gelöscht. - Die automatische Abmeldung erfolgt wie gewohnt gemäß dem konfigurierten Timeout. Hierfür werden Browser-Cookies verwendet.

Parameter	Beschreibung
Auto-logout	Wenn dieses Kontrollkästchen aktiviert ist, werden Sie nach der festgelegten Anzahl von Sekunden automatisch abgemeldet (mindestens 90 Sekunden, maximal 1 Tag). Zeitsuffixe werden unterstützt, zum Beispiel: 90s, 5m, 2h, 1d. Beachten Sie, dass diese Option in den folgenden Fällen nicht funktioniert: * Wenn Seiten im Menü *Monitoring* Hintergrundaktualisierungen von Informationen durchführen. Wenn Seiten, die Daten in einem bestimmten Zeitintervall aktualisieren (Dashboards, Diagramme, letzte Daten usw.), geöffnet bleiben, wird die Sitzungsdauer entsprechend verlängert, wodurch die Funktion zur automatischen Abmeldung deaktiviert wird. * Wenn die Anmeldung mit aktivierter Option *Remember me for 30 days* erfolgt; in diesem Fall wird der Timer für die automatische Abmeldung überschrieben, sodass Ihre Sitzung für den gesamten Zeitraum aktiv bleibt. * Wenn die Authentifizierung über einen SSO-Identity-Provider (IdP) erfolgt, beendet die automatische Abmeldung nur die Zabbix-Sitzung. Die SSO-Sitzung mit dem IdP bleibt aktiv, sodass sich der Benutzer möglicherweise erneut anmelden kann, ohne ein Passwort einzugeben. *Auto-logout* kann auch den Wert ”0” annehmen; das bedeutet, dass die Funktion zur automatischen Abmeldung nach der Aktualisierung der Profileinstellungen deaktiviert wird.
Refresh	Legen Sie fest, wie oft die Informationen auf den Seiten des Menüs *Monitoring* aktualisiert werden (mindestens 0 Sekunden, maximal 1 Stunde). Zeitsuffixe werden unterstützt, zum Beispiel: 30s, 90s, 1m, 1h.
Rows per page	Legen Sie fest, wie viele Zeilen pro Seite in den Listen angezeigt werden. Weniger Zeilen (und weniger anzuzeigende Datensätze) führen zu schnelleren Ladezeiten.
URL (after login)	Legen Sie eine bestimmte URL fest, die nach der Anmeldung angezeigt werden soll. Anstelle des standardmäßigen *Dashboards* kann dies zum Beispiel die URL von *Monitoring* > *Auslöser* sein.

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*)

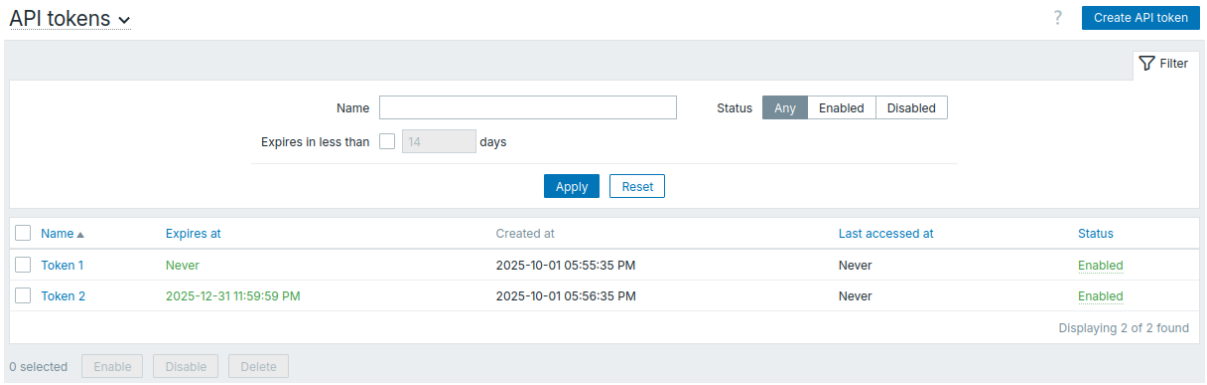
Beachten Sie, dass für bereitgestellte Benutzer gilt:

- 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-Token

Im Abschnitt *API-Token* können Sie dem Benutzer zugewiesene Token anzeigen, Tokendetails bearbeiten und neue Token erstellen. Dieser Abschnitt ist für einen Benutzer nur verfügbar, wenn die Aktion *API-Token verwalten* in den Einstellungen der Benutzerrolle erlaubt ist.



Sie können API-Token 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 auch mehrere Token gleichzeitig aktivieren/deaktivieren, indem Sie sie in der Liste auswählen und dann unterhalb der Liste auf die Schaltflächen *Aktivieren/Deaktivieren* klicken.

Attention:
Benutzer können den Wert des *Auth token* der ihnen in Zabbix zugewiesenen Token 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.

1 Globale Benachrichtigungen

Übersicht

Globale Benachrichtigungen bieten eine Möglichkeit, aktuelle Probleme in Echtzeit direkt auf Ihrem aktuellen Bildschirm im Zabbix Frontend anzuzeigen.

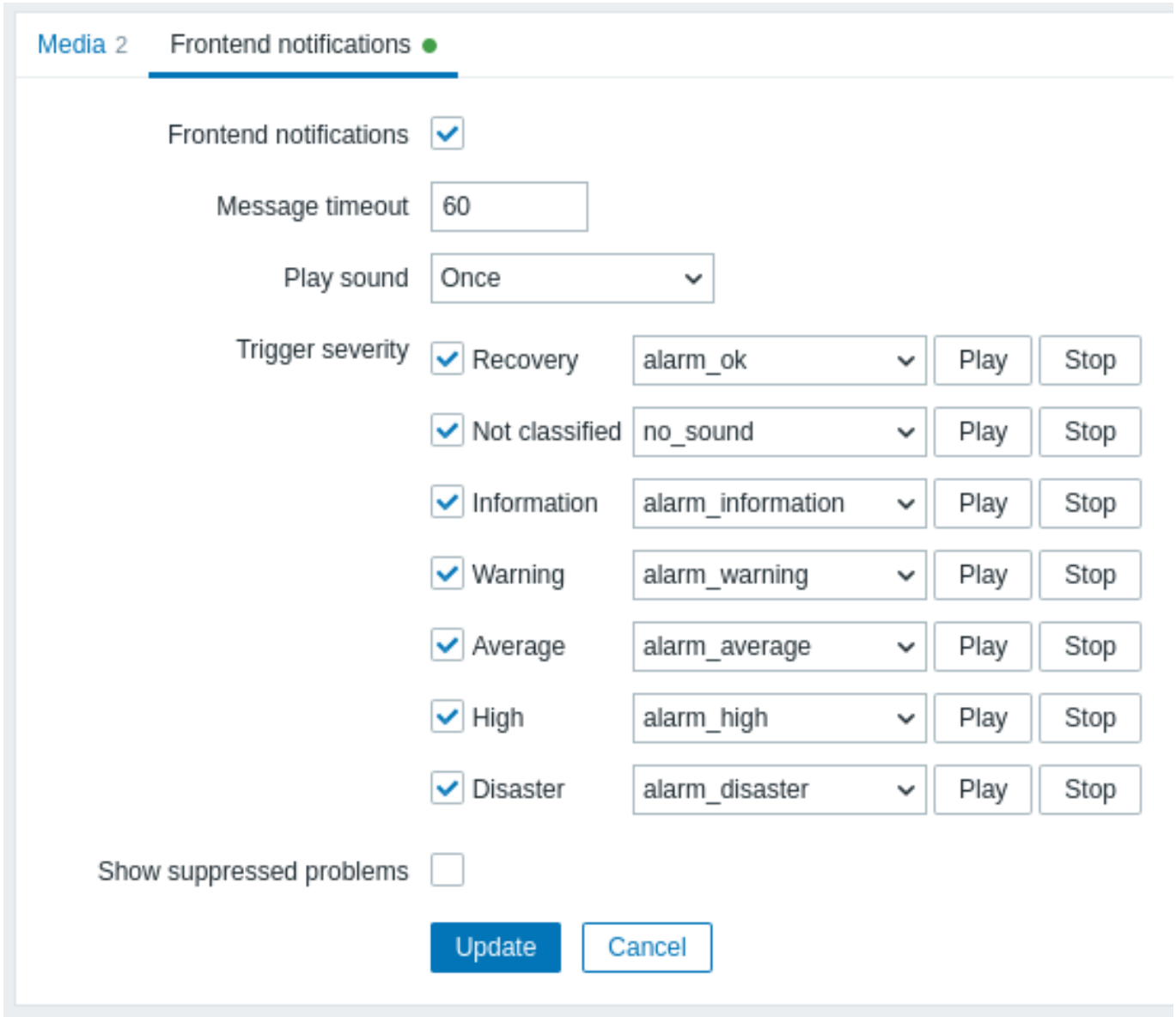
Ohne globale Benachrichtigungen würden Sie bei der Arbeit außerhalb der Bereiche *Probleme* oder *Dashboard* keine Informationen über aktuelle Probleme erhalten. Globale Benachrichtigungen stellen sicher, dass diese Informationen unabhängig von Ihrer aktuellen Position im Zabbix Frontend angezeigt werden.

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

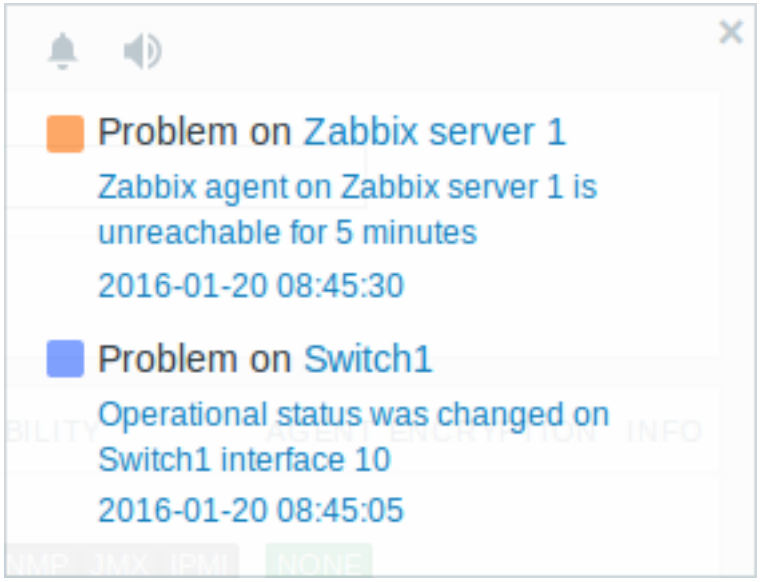
Globale Benachrichtigungen können pro Benutzer im Reiter **Frontend-Benachrichtigungen** im Abschnitt Benachrichtigungen aktiviert werden.



Parameter	Beschreibung
Frontend-Benachrichtigungen	Aktivieren Sie das Kontrollkästchen, um globale Benachrichtigungen zu aktivieren.
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.

2 Ton 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.

4 Globale Suche

Es ist möglich, im Zabbix Frontend nach Hosts, Hostgruppen, Vorlagen und Vorlagengruppen zu suchen.

Das Sucheingabefeld befindet sich im Menü unterhalb des Zabbix-Logos. Die Suche kann durch Drücken von *Enter* oder durch Klicken auf das Suchsymbol gestartet werden.

Wenn es einen Host gibt, dessen Name die eingegebene Zeichenfolge an beliebiger Stelle enthält, wird eine Dropdown-Liste angezeigt, in der alle solchen Hosts aufgeführt sind (wobei der übereinstimmende Teil orange hervorgehoben wird). Die Dropdown-Liste führt einen Host auch dann auf, wenn der sichtbare Name dieses Hosts mit dem als Suchzeichenfolge eingegebenen technischen Namen übereinstimmt; der entsprechende Host wird aufgelistet, jedoch ohne Hervorhebung.

Durchsuchbare Attribute

Hosts können nach den folgenden Eigenschaften durchsucht werden:

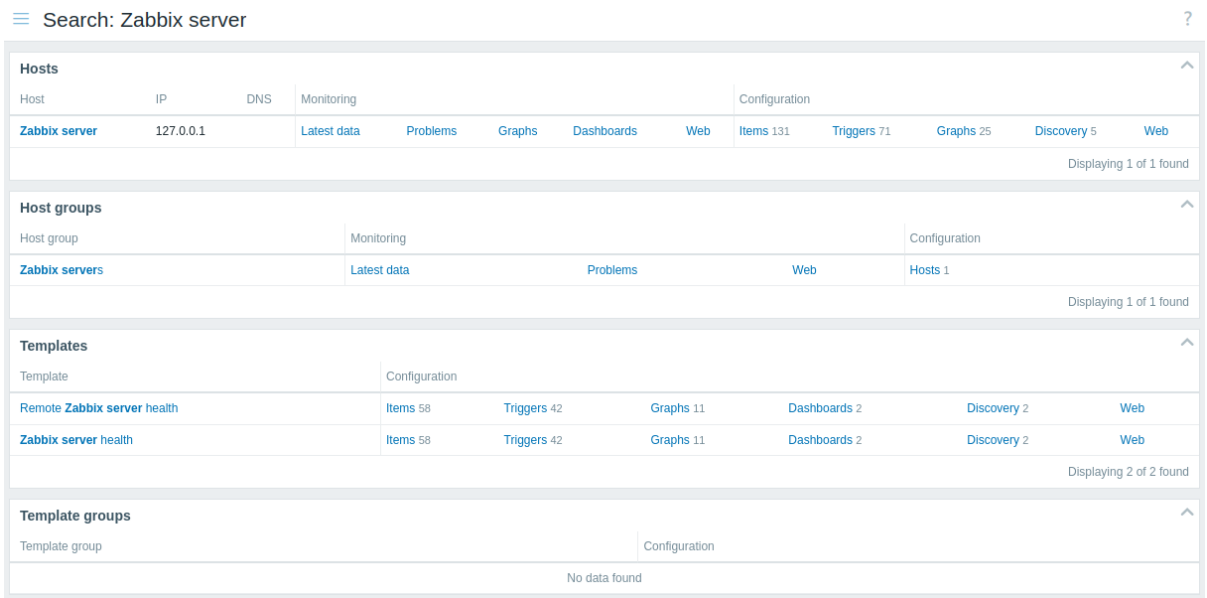
- Host-Name
- Sichtbarer Name
- IP-Adresse
- DNS-Name

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 Vorlagengruppen.



Jeder einzelne Block kann ein- oder ausgeklappt werden. Die Anzahl der Einträge wird am unteren Rand jedes Blocks angezeigt, zum Beispiel *Anzeige 13 von 13 gefundenen*. 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, Graphen) wird die Anzahl der gefundenen Entitäten als graue Zahl neben dem Entitätsnamen 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

5 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` (im Verzeichnis `/conf` des Zabbix-HTML-Dokumentverzeichnisses auf dem Webserver) und entfernen Sie die Auskommentierung der 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. Bitte haben Sie etwas Geduld..
```

Note:
In den meisten Fällen befindet sich die Datei `maintenance.inc.php` im Verzeichnis `/conf` des Zabbix-HTML-Dokumentverzeichnisses auf dem Webserver. Einige Betriebssysteme und Webserver können jedoch einen anderen Speicherort verwenden.
Zum Beispiel ist der Speicherort für:

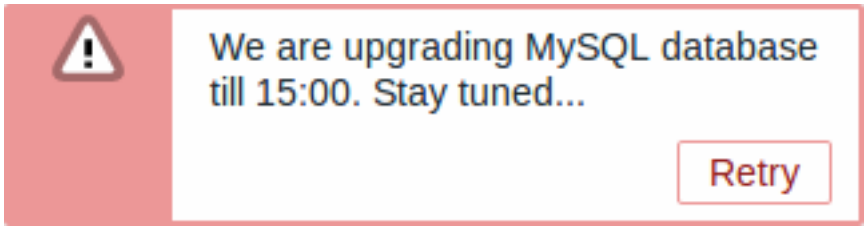
- SUSE und RedHat `/etc/zabbix/web/maintenance.inc.php`.
- Debian-basierte Systeme `/usr/share/zabbix/conf/`.

Siehe auch Kopieren von PHP-Dateien.

Parameter	Details
ZBX_DENY_GUI_ACCESS	Wenn mit einem beliebigen Wert definiert, wird der Wartungsmodus aktiviert. Zum Deaktivieren des Wartungsmodus auskommentieren oder löschen.
ZBX_GUI_ACCESS_IP_RANGE	Array von IP-Adressen, die eine Verbindung zum Frontend herstellen dürfen (optional). Zum Beispiel: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Eine Nachricht, um Benutzer über die Wartung zu informieren (optional). Falls nicht definiert, wird die Standardmeldung *'Zabbix is under maintenance'* 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.

6 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.

Überwachung > 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*: Freitextzeichenfolge.	name=Zabbix agent

Parameter	Beschreibung	Beispiel
severities	Filteroption *Schweregrad*: Array der ausgewählten Schweregrade im Format `severities[*]=*` (ersetzen Sie * durch die Schweregradstufe). 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*: Auswertungsmethode für Tags. Mögliche Werte: 0 - und/oder; 2 - oder.	evaltype=0
tags	Filteroption *Tags*: Array definierter Tags `[tag], [operator], [value]` Mögliche Werte für operator: 0 - enthält; 1 - gleich; 2 - enthält nicht; 3 - ungleich; 4 - existiert; 5 - existiert nicht.	tags[0][tag]=target&tags[0][ope
show_tags	Filteroption *Tags anzeigen*. Mögliche Werte: 0 - keine; 1 - eins; 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 *Priorität der Tag-Anzeige*: kommagetrennte Zeichenfolge der Priorität für die Tag-Anzeige	tag_priority=customer, target
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 - unbestätigt; 2 - bestätigt.	acknowledgement_status=0
acknowledged_by_me	Filteroption *Von mir*; wird nur mit `acknowledgement_status=2` unterstützt. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	acknowledged_by_me=1
compact_view	Filteroption *Kompaktansicht*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	compact_view=1

Parameter	Beschreibung	Beispiel
highlight_row	Filteroption *Gesamte Zeile hervorheben*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	highlight_row=1
filter_name	Option der Filtereigenschaften *Name*: Freitextzeichenfolge	filter_name=Databases
filter_show_counter	Option der Filtereigenschaften *Anzahl der Datensätze anzeigen*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	filter_show_counter=1
filter_custom_time	Option der Filtereigenschaften *Benutzerdefinierten Zeitraum festlegen*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	filter_custom_time=1
sort	Spalte, nach der sortiert werden soll. 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 *Alter kleiner als*; wird nur mit `show`=3 unterstützt. Mögliche Werte: 0 - Parameter age deaktivieren; 1 - Parameter age aktivieren.	age_state=1
age	Filteroption *Alter kleiner als*: integer, Anzahl der Tage; wird 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
show_timeline	Filteroption *Zeitachse anzeigen*; wird mit `compact_view`=1 nicht 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); wird nur mit `filter_custom_time`=1 unterstützt	from=now-2h
to	Ende des Datumsbereichs, kann relativ sein (z. B. now-1m); wird nur mit `filter_custom_time`=1 unterstützt	to=now

Siehe auch: Filteroptionen der Seite „Probleme“.

Kiosk-Modus

Der Kiosk-Modus kann auf unterstützten Frontend-Seiten mithilfe von URL-Parametern aktiviert werden. Zum Beispiel in Dashboards:

- `/zabbix.php?action=dashboard.view&kiosk=1` - Kiosk-Modus aktivieren
- `/zabbix.php?action=dashboard.view&kiosk=0` - normalen Modus aktivieren

Diashow

Es ist möglich, eine Diashow im Dashboard zu aktivieren:

- `/zabbix.php?action=dashboard.view&slideshow=1` - Diashow aktivieren

7 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 Session-Cookies des Zabbix Frontends verwendet wird.

Standard: zbx_sessionid

- ZBX_DATA_CACHE_TTL

TTL-Timeout in Sekunden, das zur Invalidierung des Datencaches der Vault-Antwort verwendet wird. Setzen Sie 0, um das Caching von Vault-Antworten zu deaktivieren.

Standard: 60

- SUBFILTER_VALUES_PER_GROUP

Anzahl der Subfilter-Werte pro Gruppe (zum Beispiel im Subfilter der letzten Daten).

Standard: 1000

- ZBX_MAX_WIDGET_LINES

Maximale Anzahl der anzuzeigenden Widget-Zeilen.

Standard: 1000

8 Erstellen Ihres eigenen Themes

Überblick

Standardmäßig stellt Zabbix eine Reihe vordefinierter Themes bereit. Sie können der hier beschriebenen Schritt-für-Schritt-Anleitung folgen, um Ihr eigenes zu erstellen. Teilen Sie das Ergebnis Ihrer Arbeit gern mit der Zabbix-Community, wenn Sie etwas Schönes erstellt haben.

Schritt 1

Um Ihr eigenes Theme zu definieren, müssen Sie eine CSS-Datei erstellen und im Ordner `assets/styles/` speichern (zum Beispiel *custom-theme.css*). Sie können entweder die Dateien aus einem anderen Theme kopieren und Ihr Theme darauf aufbauen oder von Grund auf neu beginnen.

Schritt 2

Fügen Sie Ihr Theme zur Liste der Themes hinzu, die von der Methode APP::getThemes() zurückgegeben werden. Sie können dies tun, indem Sie die Methode ZBase::getThemes() in der Klasse APP überschreiben. Dies kann durch Hinzufügen des folgenden Codes vor der schließenden Klammer in *include/classes/core/APP.php* erfolgen:

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme')
    ]);
}
```

Attention:
Beachten Sie, dass der Name, den Sie innerhalb des ersten Anführungszeichenpaars angeben, mit dem Namen der Theme-Datei ohne Erweiterung übereinstimmen muss.

Um mehrere Themes hinzuzufügen, listen Sie sie einfach unter dem ersten Theme auf, zum Beispiel:

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme'),
        'anothertheme' => _('Another theme'),
        'onemoretheme' => _('One more theme')
    ]);
}
```

Beachten Sie, dass jedes Theme außer dem letzten mit einem nachgestellten Komma versehen sein muss.

Note:
Um Graphfarben zu ändern, muss der Eintrag in der Datenbanktabelle *graph_theme* hinzugefügt werden.

Schritt 3

Aktivieren Sie das neue Theme.

Im Zabbix Frontend können Sie dieses Theme entweder als Standard festlegen oder Ihr Theme im Benutzerprofil ändern.

Viel Freude mit dem neuen Erscheinungsbild!

9 Debug-Modus

Übersicht

Der Debug-Modus kann verwendet werden, um Leistungsprobleme bei Frontend-Seiten zu diagnostizieren.

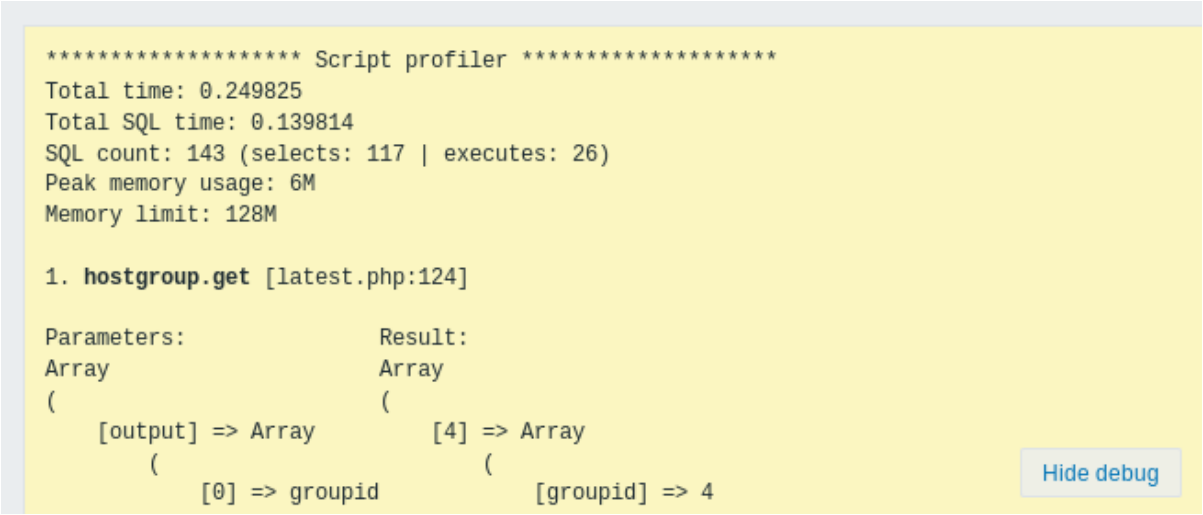
Konfiguration

Der Debug-Modus kann für einzelne Benutzer aktiviert werden, die zu einer Benutzergruppe gehören:

- bei der Konfiguration einer Benutzergruppe;
- bei der Anzeige konfigurierter Benutzergruppen.

Wenn der *Debug-Modus* für eine Benutzergruppe aktiviert ist, sehen deren Benutzer in der unteren rechten Ecke des Browserfensters eine Schaltfläche *Debug*:

Durch Klicken auf die Schaltfläche *Debug* wird unterhalb des Seiteninhalts ein neues Fenster geöffnet, das die SQL-Statistiken der Seite sowie eine Liste von API-Aufrufen 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-Modus* wirkt sich negativ auf die Leistung des Frontends aus.

10 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 Web-server 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	+	-

Name	Beschreibung	Werte	Läuft ab/Max-Age	HttpOnly[1]	Secure[2]
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

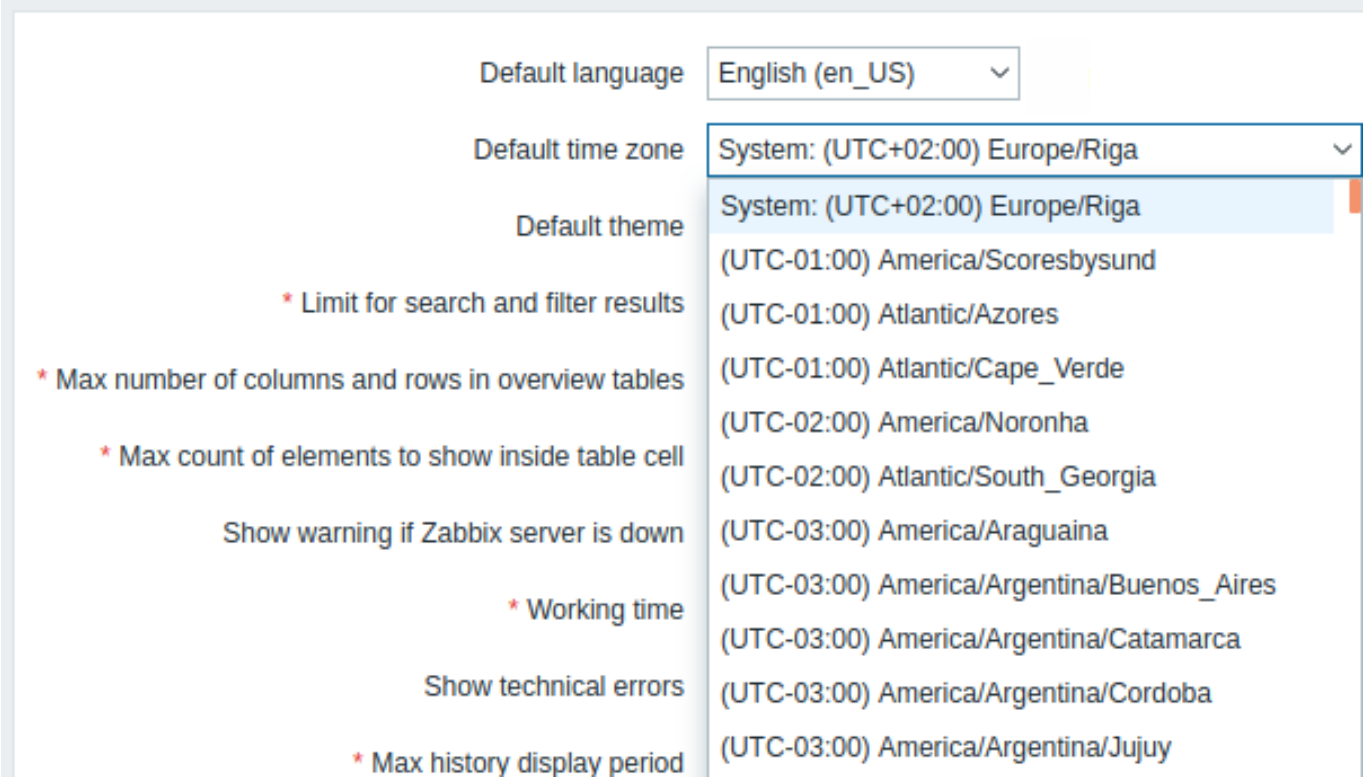
[1] Wenn `HttpOnly` auf 'true' gesetzt ist, ist das Cookie nur über das HTTP-Protokoll zugänglich. Das bedeutet, dass das Cookie nicht für Skriptsprachen wie JavaScript zugänglich ist. Diese Einstellung kann wirksam dazu beitragen, Identitätsdiebstahl durch XSS-Angriffe zu verringern (obwohl sie nicht 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.

11 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, falls gesetzt), während der Zabbix Server die Zeitzone des Systems verwendet, auf dem er ausgeführt wird.

Note:
Der Zabbix Server verwendet die angegebene globale/Benutzer- Zeitzone nur beim Erweitern von Makros in Benachrichtigungen (z. B. kann {EVENT.TIME} je nach Benutzer in eine andere Zeitzone expandiert werden) sowie für die Zeitbegrenzung beim Versand von Benachrichtigungen (siehe die Einstellung „When active" in der Medienkonfiguration des Benutzers).

Die Wahl der Zeitzone beeinflusst nicht das Zeit-/Datumsformat des Frontends. Stattdessen können Sie die Sprache der Benutzeroberfläche anpassen (entweder bei der Installation oder unter Benutzereinstellungen) – die Auswahl von *English (en_US)* aktiviert im Frontend auch das US-Zeit-/Datumsformat.

Konfiguration

Die globale Zeitzone:

- kann bei der Installation des Frontend manuell festgelegt werden
- kann unter *Administration* → *General* → *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: Ausrichten von Zeitzonen bei der Verwendung von Planungsintervallen.

12 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 Benutzer mit Super-Administrator-Rechten kann Passwörter für alle 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-Admin-Benutzer anzuwenden (ersetzen Sie 'Admin' durch den entsprechenden Super-Admin-Benutzernamen):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Nach dem Ausführen dieser Abfrage wird das Benutzerpasswort auf *zabbix* gesetzt. Stellen Sie sicher, dass Sie das Standardpasswort bei der ersten Anmeldung ändern.

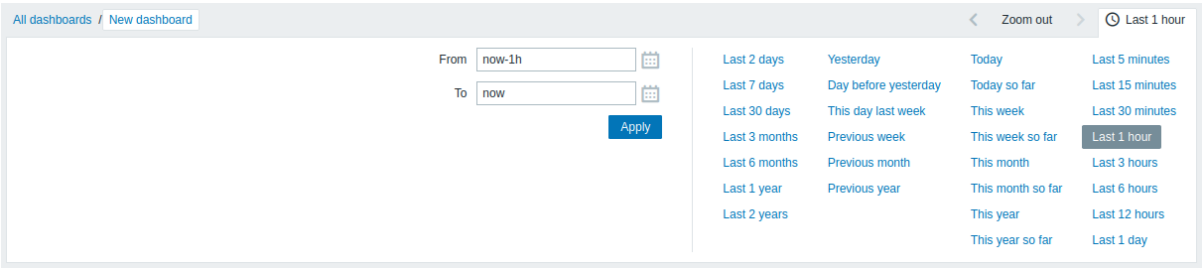
13 Zeitraumauswahl und Host-Auswahl

Übersicht

Diese Seite beschreibt die auf Dashboards verfügbaren Selektoren für Zeitraum und Host. Mit diesen Selektoren können die angezeigten Daten in allen kompatiblen Dashboard-Widgets dynamisch angepasst werden, ohne die Einstellungen einzelner Widgets zu bearbeiten.

Auswahl des Zeitraums

Die Auswahl des Zeitraums ermöglicht es, häufig benötigte Zeiträume mit einem Mausklick auszuwählen. Sie kann durch Klicken auf die Schaltfläche zur Auswahl des Zeitraums in der oberen rechten Ecke des Dashboards ein- oder ausgeblendet werden.



Optionen wie *Heute*, *Diese Woche* usw. zeigen den gesamten Zeitraum an, einschließlich der Stunden/Tage in der Zukunft. Optionen wie *Heute bisher*, *Diese Woche bisher* usw. zeigen nur die bereits vergangenen Stunden an.

Sobald ein Zeitraum ausgewählt ist, kann er durch Klicken auf die Pfeilschaltflächen  zeitlich vor- und zurückverschoben werden. Mit der Schaltfläche *Herauszoomen* kann der Zeitraum in jede Richtung um 50 % vergrößert werden.

Note:
Bei Graphen (außer bei solchen mit dem Zeitraum *Benutzerdefiniert*) kann der angezeigte Zeitraum auch ausgewählt werden, indem mit der linken Maustaste ein Bereich im Graphen markiert wird. Sobald Sie die linke Maustaste loslassen, wird in den markierten Bereich des Graphen hineingezoomt. Ein Herauszoomen ist auch durch Doppelklicken im Graphen 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`.

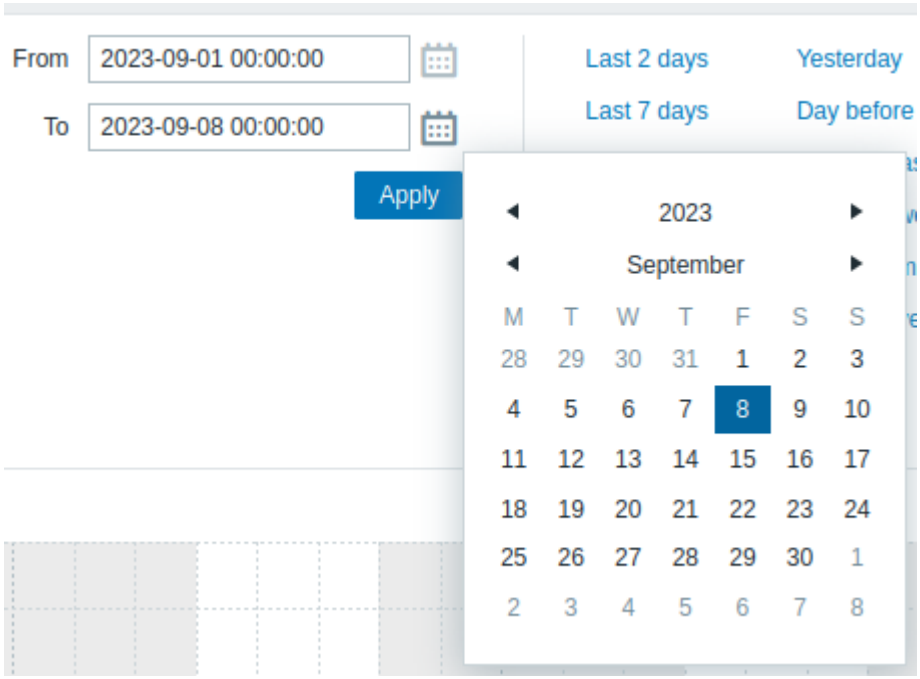
Die folgenden Abkürzungen für relative Zeit werden unterstützt:

- `now`
- `s` (Sekunden)
- `m` (Minuten)
- `h` (Stunden)
- `d` (Tage)
- `w` (Wochen)
- `M` (Monate)
- `y` (Jahre)

In der Auswahl des Zeitraums wird auch Genauigkeit unterstützt (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
`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

Es ist auch möglich, einen Zeitraum mit der Datumsauswahl auszuwählen. Um sie zu öffnen, klicken Sie auf das Kalendersymbol neben den Feldern *Von/Bis*.



Note:
Innerhalb der Datumsauswahl können Sie mit `Tab`, `Shift+Tab` und den Pfeiltasten der Tastatur zwischen Jahr/Monat/Datum navigieren. Durch Drücken von `Enter` wird die Auswahl bestätigt.

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, damit dessen Daten in kompatiblen Dashboard-Widgets angezeigt werden.

Sie wird verfügbar, nachdem ein Dashboard bearbeitet und gespeichert wurde, sofern mindestens ein Widget mit dem Parameter *Host überschreiben* so konfiguriert ist, dass das Dashboard als Datenquelle verwendet wird.

17 Best Practices

Überblick Dieser Abschnitt beschreibt bewährte Verfahren für die Einrichtung von Zabbix.

Diese Verfahren sind zwar keine Voraussetzung für den Betrieb von Zabbix, ihre Umsetzung wird jedoch für eine optimale und sichere Nutzung dringend empfohlen.

1 Best Practices für die Sicherheit

Übersicht

Dieser Abschnitt enthält Best Practices für die sichere Einrichtung von Zabbix.

Die in diesem Abschnitt beschriebenen Praktiken sind für die Funktionsweise von Zabbix nicht erforderlich, werden jedoch für eine bessere Systemsicherheit empfohlen.

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 bestehen.

Windows-Installer-Pfade

Bei der Verwendung von Windows-Installern wird empfohlen, die vom Installer bereitgestellten Standardpfade zu verwenden. Die Verwendung benutzerdefinierter Pfade ohne entsprechende Berechtigungen könnte die Sicherheit der Installation beeinträchtigen.

Makros in benutzerdefinierten globalen Skripten

Zur Erhöhung der Sicherheit wird empfohlen, in benutzerdefinierten globalen Skripten Makrofunktionen anstelle von einfachen Makros zu verwenden, da Makros nicht automatisch maskiert werden.

Zabbix-Sicherheitshinweise und CVE-Datenbank

Siehe *Zabbix-Sicherheitshinweise und CVE-Datenbank*.

HTML-E-Mail-Vorlagen

Beim Erstellen oder Bearbeiten von Nachrichtenvorlagen, die für HTML-E-Mails verwendet werden, umschließen Sie jedes Makro immer mit der Makrofunktion `htmlencode()`. 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 (zum Beispiel, wenn ein Angreifer einen bösartigen Phishing-Link in eine Benachrichtigung einfügt).

Hinweis: Die von Zabbix bereitgestellten Standardnachrichten für HTML-E-Mails wenden `htmlencode()` bereits auf Makros an. Diese Empfehlung gilt beim Bearbeiten vorhandener Vorlagen oder beim Erstellen neuer Vorlagen — vergewissern Sie sich, dass Makros kodiert sind, bevor Sie eine Vorlage zum Senden von HTML-E-Mails verwenden.

1 Zugriffskontrolle

Überblick

Dieser Abschnitt enthält Best Practices für die sichere Einrichtung der Zugriffskontrolle.

Prinzip der geringsten Rechte

Benutzerkonten sollten jederzeit mit so wenigen Rechten wie möglich ausgeführt werden. Das bedeutet, dass Benutzerkonten im Zabbix Frontend, Datenbankbenutzer oder der Benutzer für Zabbix Server-/Proxy-/Agent-Prozesse nur die Rechte haben sollten, die für die Ausführung der vorgesehenen Funktionen unbedingt erforderlich sind.

> **Attention:**
> Die Vergabe zusätzlicher Rechte an 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 Benutzerrechten sollten die Zabbix-Frontend-Benutzertypen berücksichtigt werden. Beachten Sie, dass der Benutzertyp *Admin* zwar weniger Rechte als der Benutzertyp *Super Admin* hat, aber dennoch die Konfiguration verwalten und benutzerdefinierte Skripte ausführen kann.

> **Note:**
> Einige Informationen sind auch für nicht privilegierte Benutzer verfügbar. Zum Beispiel ist *Warnungen → Skripte* nur für Benutzer vom Typ *Super Admin* verfügbar, Skripte können jedoch auch über die Zabbix API abgerufen werden. In diesem Fall können die Einschränkung von Skriptberechtigungen und der Ausschluss sensibler Informationen aus Skripten (zum Beispiel Zugangsdaten) dazu beitragen, die Offenlegung sensibler Informationen zu vermeiden, die in globalen Skripten verfügbar sind.

Sicherer Benutzer für Zabbix Agent

Standardmäßig verwenden die Prozesse von Zabbix Server, Proxy und Agent (oder Agent 2) gemeinsam einen `zabbix`-Benutzer. Um zu verhindern, dass Zabbix Agent/Agent 2 (der auf derselben Maschine wie Server/Proxy läuft) auf vertrauliche Details in der Server-/Proxy-Konfiguration zugreift (zum Beispiel Datenbank-Zugangsdaten), 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. Legen Sie diesen Benutzer im Parameter User der Agent-Konfigurationsdatei fest.
3. Starten Sie den Agent neu, damit die Berechtigungen auf den neuen Benutzer reduziert werden.

Für Zabbix Agent 2 muss die Konfiguration auf der Ebene des Dienstes angewendet werden, da die Agent-2-Konfigurationsdatei den Parameter `User` nicht unterstützt. Ein Beispiel finden Sie unter ZBX-26442.

Schreibzugriff auf die SSL-Konfiguration widerrufen (Windows)

Wenn Sie den Zabbix Agent unter Windows kompiliert haben und sich OpenSSL 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 Sie Nicht-Administrator-Benutzern den Schreibzugriff auf dieses Verzeichnis entziehen. Andernfalls lädt der Agent SSL-Einstellungen aus einem Pfad, der von nicht privilegierten Benutzern geändert werden kann, was zu einer potenziellen Sicherheitslücke führt.

Absicherung der Sicherheit von Zabbix-Komponenten

Einige Funktionen können deaktiviert werden, um die Sicherheit von Zabbix-Komponenten zu erhöhen:

- die Ausführung globaler Skripte auf dem Zabbix Server kann durch Setzen von EnableGlobalScripts=0 in der Server-Konfiguration deaktiviert werden;
- die Ausführung globaler Skripte auf dem Zabbix Proxy ist standardmäßig deaktiviert (kann durch Setzen von EnableRemoteCommands=1 in der Proxy-Konfiguration aktiviert werden);
- die Ausführung globaler Skripte auf Zabbix Agents ist standardmäßig deaktiviert (kann durch Hinzufügen eines Parameters AllowKey=system.run[<command>,*] für jeden erlaubten Befehl in der Agent-Konfiguration aktiviert werden);
- die HTTP-Benutzerauthentifizierung kann durch Setzen von `$ALLOW_HTTP_AUTH=false` in der Frontend-Konfigurationsdatei (zabbix.conf.php) deaktiviert werden. Beachten Sie, dass bei einer Neuinstallation des Frontends (durch Ausführen von setup.php) dieser Parameter entfernt wird.

Zugriff auf UNC-Pfade 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 manchen Kontexten ein Sicherheitsrisiko darstellen.

Wenn Windows aufgefordert wird, auf einen UNC-Pfad zuzugreifen, versucht es, sich auf diesem Server zu authentifizieren. Das bedeutet, dass eine bösartige Anfrage an den Zabbix Agent den NTLM-Hash gegenüber dem Server des Anfragenden offenlegen kann. Benutzer können dies bei Bedarf mit den Konfigurationsparametern AllowKey und DenyKey abmildern.

1 Absicherung von MySQL/MariaDB

Überblick

Dieser Abschnitt enthält Best Practices für die Absicherung einer MySQL-/MariaDB-Datenbank.

> **Note:**
> Für eine grundlegende Einrichtung siehe die standardmäßigen Anweisungen zur Erstellung einer MySQL-/MariaDB-Datenbank, die auch die Erstellung des Benutzers „zabbix“ mit vollständigen Berechtigungen für die Zabbix-Datenbank umfassen. Dieser Benutzer ist der Eigentümer der Datenbank und verfügt außerdem über die erforderlichen Berechtigungen, um bei einem Upgrade von Zabbix die Datenbankstruktur zu ändern.

Zur Verbesserung der Sicherheit wird empfohlen, zusätzliche Datenbankrollen und Benutzer mit minimalen Berechtigungen zu erstellen. Diese Rollen und Benutzer sollten gemäß dem Prinzip der geringsten Privilegien konfiguriert werden, das heißt, sie sollten nur über die 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 zum Ausführen von Zabbix Server und Proxy:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - Rolle zum Ausführen 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 Eigentümer der Datenbank durchgeführt werden.

- **zbx_part** - Rolle mit einem reduzierten Satz an 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';
-- Für MariaDB: die nächste Zeile überspringen (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 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>, <host>, <role> und <password> nach Bedarf.

```
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;
```

Zum Beispiel, um die Rolle für den Betrieb von Zabbix Server und Proxy zu erstellen und zuzuweisen:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 Absicherung von PostgreSQL/TimescaleDB

Überblick

Dieser Abschnitt enthält Best Practices für die Absicherung einer PostgreSQL-Datenbank.

> **Note:**
> Für eine grundlegende Einrichtung siehe die standardmäßigen Anweisungen zur Erstellung einer PostgreSQL-Datenbank, die auch die Erstellung des Benutzers „zabbix“ mit vollständigen Berechtigungen für die Zabbix-Datenbank umfassen. Dieser Benutzer ist der Eigentümer der Datenbank und verfügt außerdem über die erforderlichen Berechtigungen, um die Datenbankstruktur bei einem Upgrade von Zabbix zu ändern.

Zur Verbesserung der Sicherheit wird empfohlen, ein sicheres Nutzungsmuster für Schemas zu konfigurieren sowie zusätzliche Datenbankrollen und Benutzer mit minimalen Berechtigungen zu erstellen. Diese Rollen und Benutzer sollten gemäß dem Prinzip 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 Eigentümer der Datenbank sein wird, und erstellen Sie die Zabbix-Datenbank; der Datenbankeigentümer ist der Benutzer, der bei der Erstellung der Datenbank 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 durch den Datenbankeigentümer durchgeführt werden. Der Grund dafür ist, dass das Recht, ein Datenbankobjekt zu löschen oder seine Definition zu ändern, ein Privileg ist, das dem Datenbankeigentümer inhärent ist und nicht gewährt oder entzogen werden kann.

> **Attention:**
> Die folgenden Befehle auf dieser Seite müssen ausgeführt werden, während die Verbindung zu PostgreSQL speziell mit der Datenbank `zabbix` hergestellt ist.

Erstellen Sie das Schema zabbix und legen Sie den Datenbankeigentümer (usr_owner) als Eigentümer dieses Schemas fest:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Konfigurieren Sie ein sicheres Nutzungsmuster für Schemas:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Note: search_path should point to the "zabbix" schema:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Fahren Sie nach der Einrichtung der Datenbank mit der Erstellung von Benutzerrollen fort.

Erstellen von Benutzerrollen

Erstellen Sie die folgenden Rollen mit den entsprechenden Berechtigungen:

- **zbx_srv** - Rolle zum Ausführen 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 zum Ausführen 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 die Tabellensicherung:

```
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 Eigentümer der Datenbank 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> nach Bedarf.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Um beispielsweise die Rolle für den Betrieb von Zabbix Server und Proxy zu erstellen und zuzuweisen:

```
CREATE USER usr_srv WITH ENCRYPTED password 'password';
GRANT zbx_srv TO usr_srv;
```

Datenbankpartitionierung mit TimescaleDB

Die Datenbankpartitionierung wird durch TimescaleDB erleichtert. Um TimescaleDB zu nutzen, benötigt der Zabbix Server Datenbankeigentümerrechte.

Wenn das PostgreSQL-Schema zabbix bereits in der Datenbank zabbix erstellt wurde, können Sie TimescaleDB mit folgendem Befehl aktivieren:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

2 Kryptografie

Überblick

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, in der nach dem `Common Name` gefragt wird. Sie müssen den Domainnamen eingeben, der mit Ihrem Server verknüpft werden soll. Falls Sie keinen Domainnamen haben, können Sie stattdessen die öffentliche IP-Adresse eingeben.

```
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, damit die Änderungen übernommen werden:

```
systemctl restart httpd.service
```

3 Webserver

Überblick

Dieser Abschnitt enthält Best Practices für die sichere Einrichtung des Webservers.

Erzwingen der Weiterleitung der Root-URL zu Zabbix SSL

Fügen Sie auf RHEL-basierten Systemen einen virtuellen Host zur Apache-Konfiguration (`/etc/httpd/conf/httpd.conf`) hinzu und legen Sie eine permanente Weiterleitung für das Dokumenten-Root auf die Zabbix-SSL-URL fest. 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 anzuwenden:

```
systemctl restart httpd.service
```

Aktivieren von HTTP Strict Transport Security (HSTS) auf dem Webserver

Um das Zabbix Frontend vor Protokoll-Downgrade-Angriffen zu schützen, empfehlen wir, die HSTS-Richtlinie auf dem Webserver zu aktivieren.

Gehen Sie wie folgt vor, um die HSTS-Richtlinie für Ihr Zabbix Frontend in der Apache-Konfiguration zu aktivieren:

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 Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Starten Sie den Apache-Dienst neu, um die Änderungen anzuwenden:

```
#### Auf RHEL-basierten Systemen:
systemctl restart httpd.service

#### Auf Debian/Ubuntu
systemctl restart apache2.service
```

Erzwingen von sicheren und SameSite-Session-Cookies in Zabbix

Bei der Konfiguration von Zabbix ist es wichtig, sichere und SameSite-Attribute 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 „Benutzer nicht angemeldet“ an, wenn iframes derselben Domain eingebettet werden.
- Benutzer, die über HTTP statt über HTTPS auf das Dashboard zugreifen, können Anmeldeprobleme haben.
- URLs zu bestimmten Zabbix-Menüabschnitten oder Hosts können nicht geteilt werden.

Um diese Probleme zu entschärfen, 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 und nicht über unverschlüsselte Verbindungen offengelegt werden.

Um sichere Cookies in Zabbix zu aktivieren, fügen Sie die folgende Einstellung in der Webserver-Konfiguration hinzu oder ändern Sie sie entsprechend:

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. Konfiguration 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 ~ für mehr Genauigkeit durch 'zbx_session'
```

Aktivieren der Content Security Policy (CSP) auf dem Webserver

Um das Zabbix Frontend vor Cross Site Scripting (XSS), Dateninjektion und ähnlichen Angriffen zu schützen, empfehlen wir, die 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 alle Inhalte von der Domain der Website stammen (ausgenommen Subdomains). Eine andere CSP-Header-Konfiguration kann erforderlich sein, wenn Sie beispielsweise das Widget *URL* so konfigurieren, dass Inhalte von den Subdomains der Website oder von externen Domains angezeigt werden, von *OpenStreetMap* zu einer anderen Karten-Engine wechseln oder externes CSS oder Widgets hinzufügen. Wenn Sie für die Multi-Faktor-Authentifizierung die Methode Duo Universal Prompt 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>
```

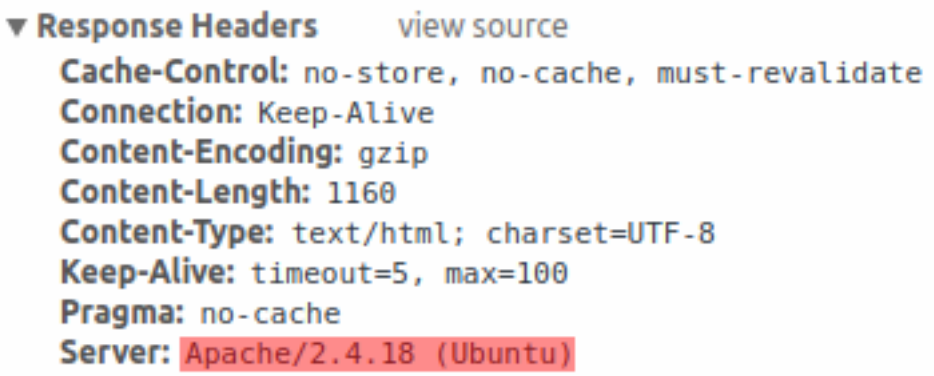
3. Starten Sie den Apache-Dienst neu, um die Änderungen anzuwenden:

```
#### 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 Webserver-Signaturen zu deaktivieren.

Standardmäßig legt der Webserver die Software-Signatur 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 wirksam werden, ist ein Neustart des Webservers erforderlich.

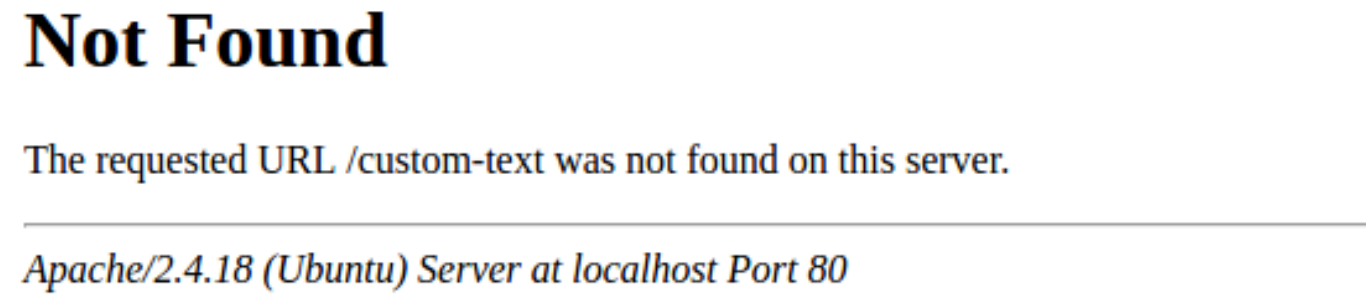
Für zusätzliche Sicherheit können Sie das Werkzeug *mod_security* mit Apache verwenden (Paket *libapache2-mod-security2*). Mit diesem Werkzeug kann die Server-Signatur vollständig entfernt werden, anstatt nur die Versionsangabe aus der Server-Signatur zu entfernen. Die Server-Signatur kann nach der Installation von *mod_security* durch Setzen von "SecServerSignature" auf einen beliebigen gewünschten Wert geändert werden.

Weitere Informationen zum Entfernen/Ändern von Software-Signaturen finden Sie in der Dokumentation Ihres Webservers.

Standard-Fehlerseiten des Webservers deaktivieren

Um die Offenlegung von Informationen zu vermeiden, wird empfohlen, die Standard-Fehlerseiten zu deaktivieren.

Standardmäßig verwendet ein Webserver integrierte Fehlerseiten:



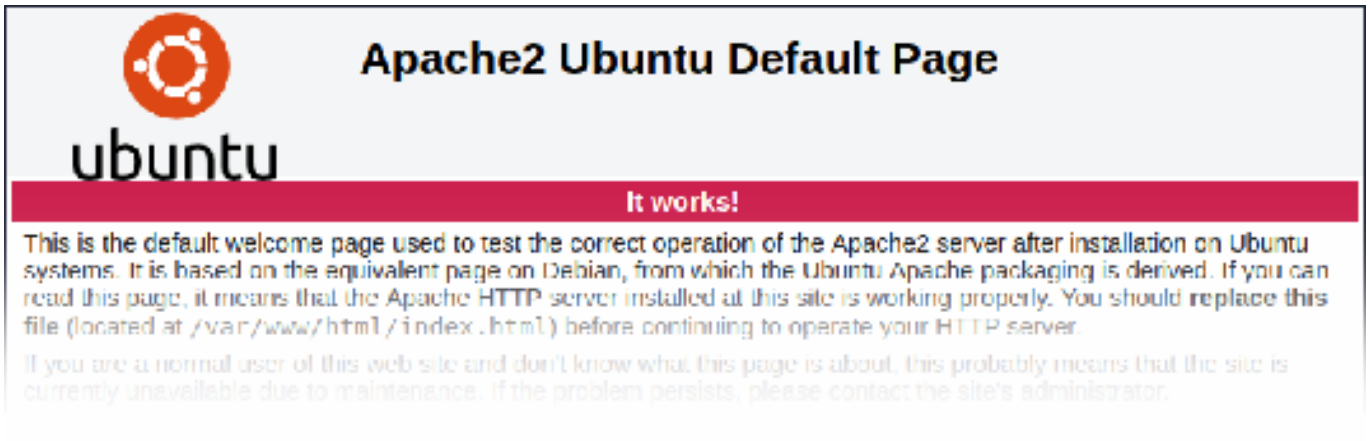
Diese Standard-Fehlerseiten sollten ersetzt/entfernt werden. Zum Beispiel kann die Direktive "ErrorDocument" verwendet werden, um für den Apache-Webserver eine benutzerdefinierte Fehlerseite bzw. einen benutzerdefinierten Fehlertext festzulegen.

Bitte lesen Sie in der Dokumentation Ihres Webservers nach, wie Sie Standard-Fehlerseiten ersetzen/entfernen können.

Entfernen der Testseite des Web-Servers

Um die Offenlegung von Informationen zu vermeiden, wird empfohlen, die Testseite des Web-Servers zu entfernen.

Standardmäßig enthält das Webroot des Apache-Web-Servers die Testseite `index.html`:



Bitte lesen Sie in der Dokumentation Ihres Web-Servers nach, um Hilfe dazu zu finden, wie Standard-Testseiten entfernt werden können.

X-Frame-Options-HTTP-Response-Header festlegen

Standardmäßig ist Zabbix so konfiguriert, dass der Parameter *Use X-Frame-Options HTTP header* auf `SAMEORIGIN` gesetzt ist. Das bedeutet, dass Inhalte nur in einem Frame geladen werden können, der dieselbe Herkunft wie die Seite selbst hat.

Zabbix-Frontend-Elemente, die Inhalte von externen URLs abrufen (nämlich das URL-Dashboard-Widget), zeigen die abgerufenen Inhalte in einer Sandbox mit aktivierten vollständigen 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, iframe-Sandboxing oder den X-Frame-Options-HTTP-Header vollständig zu deaktivieren.

Verbergen der Datei mit der Liste gängiger 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 am häufigsten verwendeten und kontextspezifischen Passwörter und verhindert, dass Benutzer solche Passwörter festlegen können (wenn der Parameter *Leicht zu erratende Passwörter vermeiden* in der Passwortrichtlinie aktiviert ist).

Um den Zugriff auf die Datei `top_passwords.txt` einzuschränken, ändern Sie die Konfiguration Ihres Webservers.

Unter Apache kann der Dateizugriff mit der Datei `.htaccess` eingeschränkt werden:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

Unter NGINX kann der Dateizugriff mit der Direktive `location` eingeschränkt werden:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

2 Best Practices für die Konfiguration

Überblick

Dieser Abschnitt beschreibt bewährte Verfahren für die Konfiguration von Zabbix, um optimale Leistung und eine einfache Nutzung zu erreichen. Die Empfehlungen basieren auf den Ratschlägen der Zabbix-Entwickler sowie auf den praktischen Erfahrungen von Zabbix-Trainern und Support-Ingenieuren.

Jede Zabbix-Installation ist einzigartig, und einige dieser Richtlinien sind möglicherweise nicht für Ihre spezifische Konfiguration geeignet. Es wird jedoch empfohlen, diese Richtlinien so weit wie möglich einzuhalten, um häufige potenzielle Probleme zu vermeiden.

> **Note:**
> Wenn Sie der Meinung sind, dass diese Seite verbessert werden könnte, würden wir uns freuen, von Ihnen zu hören! Bitte markieren Sie den betreffenden Text und drücken Sie **Strg+Enter**, um einen Fehler zu melden oder uns Ihr Feedback mitzuteilen.

Hosts und Datenpunkte Definieren eines Hosts

Ein Host in Zabbix ist keine physische Maschine oder kein physisches Gerät, sondern eine logische Entität. Zu Überwachungszwecken können Sie separate Hosts für eine Datenbank oder beispielsweise eine virtuelle Maschine erstellen. Alternativ können Sie einen generischen Host *Johns Laptop* erstellen und alle Metriken unter diesem Host überwachen.

Es empfiehlt sich, für jede unabhängige Instanz wie eine virtuelle Maschine, eine Datenbank, einen Container oder einen Netzwerk-Switch einen separaten Host zu erstellen. Mit diesem Ansatz werden Sie:

1. Unübersichtlichkeit in den Überwachungsdaten vermeiden, da Sie für jeden Host separate Datenpunkte, Auslöser und Alarmbenachrichtigungen haben.
2. Benutzerzugriffsebenen fein abstimmen. Sie können Benutzerrollen konfigurieren, um Zugriff auf die Anzeige und/oder Konfiguration nur bestimmter Hosts zu gewähren. Siehe auch das Prinzip der geringsten Rechte.

Hosts mit duplizierten Datenpunkten

Wenn Sie mehrere ähnliche Hosts haben, wie z. B. *Network switch 1* und *Network switch 2*, bietet Zabbix mehrere Möglichkeiten, den Host schnell neu zu erstellen. Sie können einen Host einfach mit all seinen Metriken klonen, indem Sie auf die Schaltfläche „Clone“ klicken. In diesem Fall müssen Sie jedoch einen Datenpunkt später auf jedem Host manuell aktualisieren.

Die empfohlene Vorgehensweise ist, eine Vorlage mit allen erforderlichen Metriken zu erstellen, zum Beispiel *Network switch template*. Gruppieren Sie dann die ähnlichen Hosts in einer Host-Gruppe; im obigen Beispiel könnte dies *Network switches* sein. Nun können Sie im Abschnitt *Data Collection -> Hosts* alle Hosts nach Host-Gruppe filtern und die Schaltfläche *Mass update* verwenden, um die Vorlage mit all Ihren Netzwerk-Switches zu verknüpfen.

Abhängige Datenpunkte

Um die Anzahl der Anfragen an die Zielentität zu minimieren, ermöglicht Zabbix die Erstellung von Master- und abhängigen Datenpunkten. In diesem Fall sammelt der Master-Datenpunkt in einer einzigen Anfrage eine große Menge an Informationen. Anschließend können abhängige Datenpunkte so konfiguriert werden, dass sie über die Vorverarbeitung bestimmte Datenteile aus dieser Sammlung extrahieren und als einzelne Metriken speichern.

Beispielsweise kann der Master-Datenpunkt eine JSON- oder XML-Antwort mit mehreren Metriken erfassen oder eine Datenbankabfrage ausführen, die mehrere Datenspalten zurückgibt (z. B. Anzahl offener Verbindungen, abgebrochene Verbindungen, maximal zulässige gleichzeitige Verbindungen und gesamte kumulative Verbindungen seit dem Start). Die abhängigen Datenpunkte analysieren dann jeden benötigten Wert und speichern ihn separat.

Als Best Practice für diese Konfiguration empfiehlt es sich, den Verlauf des Master-Datenpunkts direkt nach der Erfassung zu verwerfen und nur die Daten der abhängigen Datenpunkte aufzubewahren.

Server und Proxys

Wenn sich alle Hosts im selben lokalen Netzwerk wie der Zabbix Server befinden und es keine Bedenken hinsichtlich Skalierbarkeit oder Leistung gibt, benötigen Sie möglicherweise keinen Proxy. In größeren oder komplexeren Umgebungen reicht die direkte Überwachung von Hosts mit dem Zabbix Server möglicherweise nicht aus. Das Hinzufügen eines Proxys und die Zuweisung eines Teils der Hosts zu diesem Proxy ermöglichen eine gleichmäßigere Lastverteilung.

Es empfiehlt sich, einen Zabbix Proxy hinzuzufügen, wenn:

1. Sie mehrere Hosts mit verschiedenen Methoden zur Erfassung von Metriken hinter einer Firewall überwachen. Der Proxy sammelt Daten von den Hosts und leitet sie an den Zabbix Server weiter, wodurch die Notwendigkeit verringert wird, mehrere Firewall-Ports zu öffnen.
2. Sie entfernte Standorte, Niederlassungen und/oder Netzwerke überwachen. Im Falle einer Netzwerkunterbrechung zwischen dem Zabbix Server und Ihren entfernten Standorten setzen die an den entfernten Standorten eingesetzten Zabbix Proxys die Datenerfassung fort und senden die gesammelten Daten an den Zabbix Server zurück, sobald die Netzwerkverbindung wiederhergestellt ist.
3. Sie eine groß angelegte Bereitstellung haben und die Last auf dem Zabbix Server reduzieren sowie die Leistung verbessern möchten. Die Definition einer groß angelegten Bereitstellung ist sehr weit gefasst und hängt nicht nur von der Anzahl der Hosts ab, sondern auch von der Anzahl der pro Sekunde erfassten Werte.

Geheime Makros

Möglicherweise möchten Sie geheime Benutzermakros entweder als geheimen Text oder als geheime Vault-Makros verwenden.

Für erhöhte Sicherheit bei der Verwendung geheimer Vault-Makros wird empfohlen, Makrowerte so zu konfigurieren, dass sie vom Zabbix Server und von Zabbix Proxys unabhängig voneinander abgerufen werden. Standardmäßig werden die Werte geheimer Makros vom Zabbix Server abgerufen und an Zabbix Proxys weitergegeben.

18 API

Überblick Die Zabbix-API ermöglicht es Ihnen, die Konfiguration von Zabbix programmgesteuert abzurufen und zu ändern, und bietet Zugriff auf Verlaufsdaten. 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 Web-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 kodiert.

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-Funktionalität in Ihre Python-Anwendungen finden Sie unter Python library for Zabbix.

> **Note:**
> Der Benutzerzugriff in Zabbix, einschließlich Konfigurations- und Verlaufsdaten, 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 zum Erstellen neuer Hosts verwendet. In der Vergangenheit wurden APIs manchmal als "Klassen" bezeichnet.

> **Note:**
> Die meisten APIs enthalten mindestens vier Methoden: `get`, `create`, `update` und `delete` zum Abrufen, Erstellen, Aktualisieren und Löschen von Daten, aber einige APIs können auch einen völlig anderen Satz von Methoden bereitstellen.

Anfragen ausführen Sobald 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` auf einen der folgenden Werte gesetzt haben: `application/json-rpc`, `application/json` oder `application/jsonrequest`.

Das Anfrageobjekt muss die folgenden Eigenschaften enthalten:

- `jsonrpc` - die Version des JSON-RPC-Protokolls, die von der API verwendet wird (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 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.

Beispiel-Workflow Im folgenden Abschnitt werden Sie einige Beispiele für die Verwendung genauer kennenlernen.

Authentifizierung Um auf beliebige Daten in Zabbix zuzugreifen, müssen Sie entweder:

- ein vorhandenes API-Token verwenden (erstellt im Zabbix Frontend oder mithilfe der Token API);
- ein Authentifizierungs-Token verwenden, das mit der Methode user.login abgerufen wurde.

Wenn Sie beispielsweise ein neues Authentifizierungs-Token erhalten möchten, indem Sie sich als standardmäßiger *Admin*-Benutzer anmelden, 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 Zugangsdaten korrekt angegeben haben, sollte die von der API zurückgegebene Antwort das Benutzerauthentifizierungs-Token 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).

Hosts abrufen Jetzt haben Sie ein gültiges Benutzerauthentifizierungs-Token (in den folgenden Beispielen als Variable dargestellt), das für den Zugriff auf die Daten in Zabbix verwendet werden kann. Sie können zum Beispiel die Methode host.get verwenden, um die IDs, Host-Namen 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 Objekteigenschaften aufzulisten, die Sie abrufen möchten. So vermeiden Sie, alles abzurufen.

Erstellen eines neuen Datenpunkts Erstellen Sie nun einen neuen Datenpunkt auf dem Host „Zabbix server“ mithilfe der Daten, die Sie aus der vorherigen Anfrage `host.get` 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-Methoden* können auch Arrays von Objekten akzeptieren und mit einem API-Aufruf mehrere Datenpunkte erstellen.

Mehrere Auslöser erstellen Wenn *create-Methoden* 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":"Die Prozessorlast ist auf {
```

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 API-Aufruf aktualisieren.

Mehrere Auslöser aktualisieren 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 senden würden? Versuchen Sie, einen weiteren Host zu erstellen, indem Sie host.create aufrufen, dabei jedoch den verpflichtenden 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, diese Arten von Fehlern zuverlässig 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 für die Verwendung versionsspezifischer Funktionen anzupassen.

Zabbix garantiert die Abwärtskompatibilität von Funktionen innerhalb einer Hauptversion. Wenn zwischen Hauptversionen nicht abwärtskompatible Änderungen vorgenommen werden, lässt Zabbix die alten Funktionen in der Regel in der nächsten Version als veraltet bestehen und entfernt sie erst in der darauffolgenden Version. Gelegentlich kann Zabbix jedoch Funktionen zwischen Hauptversionen entfernen, ohne irgendeine Form von Abwärtskompatibilität bereitzustellen. Es ist wichtig, dass Sie sich niemals auf veraltete Funktionen verlassen und so bald wie möglich auf neuere Alternativen migrieren.

> **Note:**
> Sie können alle an der API vorgenommenen Änderungen im API-Änderungsprotokoll verfolgen.

Weiterführende Lektüre Jetzt verfügen Sie über genügend Wissen, um mit der Zabbix-API zu arbeiten. Hören Sie hier jedoch nicht auf. Für weiterführende Informationen empfiehlt es sich, einen Blick auf die Liste der verfügbaren APIs zu werfen.

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

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

Dienste

Verwalten Sie Dienste für die Service-Level-Überwachung und rufen Sie detaillierte SLA-Informationen zu jedem Dienst ab.

Dienst-API

Datensammlung Die Zabbix-API ermöglicht es Ihnen, die Konfiguration Ihres Überwachungssystems zu verwalten.

Hosts und Hostgruppen

Verwalten Sie Hostgruppen, Hosts und alles, was damit zusammenhängt, einschließlich Host-Schnittstellen, Host-Makros und Wartungszeiträume.

Host-API | Hostgruppen-API API | Host-Schnittstellen-API API | Benutzer-Makro-API API | Wertzuordnung-API API | Wartungs-API 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 Auslöser-Abhängigkeiten.

Auslöser-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

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

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

API-Informationen Rufen Sie die Version der Zabbix-API ab, damit Ihre Anwendung versionsspezifische Funktionen nutzen kann.

API-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 hat die folgenden Eigenschaften.

Property	Type	Description
actionid	ID	ID der Aktion. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
esc_period	string	Standarddauer des Operationsschritts. Muss mindestens 60 Sekunden betragen. Akzeptiert Sekunden, eine Zeiteinheit mit Suffix oder ein Benutzermakro. Property behavior: - *unterstützt*, wenn `eventsource` auf „durch einen Auslöser erstelltes Ereignis“, „internes Ereignis“ oder „bei einer Service-Statusaktualisierung erstelltes Ereignis“ gesetzt ist
eventsource	integer	Typ der Ereignisse, die von der Aktion verarbeitet werden. Eine Liste der unterstützten Ereignistypen finden Sie in der Ereignis-Eigenschaft `source`. Property behavior: - *konstant* - *erforderlich* für Erstellungsvorgänge
name	string	Name der Aktion. Property behavior: - *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 das Ereignis ein Symptomereignis ist. Mögliche Werte: 0 - Eskalation für Symptomprobleme nicht pausieren; 1 - *(Standard)* Eskalation für Symptomprobleme pausieren. Property behavior: - *unterstützt*, wenn `eventsource` auf „durch einen Auslöser erstelltes Ereignis“ gesetzt ist
pause_suppressed	integer	Gibt an, ob die Eskalation während Wartungszeiträumen pausiert werden soll oder nicht. Mögliche Werte: 0 - Eskalation nicht pausieren; 1 - *(Standard)* Eskalation pausieren. Property behavior: - *unterstützt*, wenn `eventsource` auf „durch einen Auslöser erstelltes Ereignis“ gesetzt ist
notify_if_canceled	integer	Gibt an, ob benachrichtigt werden soll, wenn die Eskalation abgebrochen wird. Mögliche Werte: 0 - Nicht benachrichtigen, wenn die Eskalation abgebrochen wird; 1 - *(Standard)* Benachrichtigen, wenn die Eskalation abgebrochen wird. Property behavior: - *unterstützt*, wenn `eventsource` auf „durch einen Auslöser erstelltes Ereignis“ gesetzt ist

Aktionsoperation

Das Objekt der Aktionsoperation definiert eine Operation, die ausgeführt wird, wenn eine Aktion ausgeführt wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
operationtype	integer	Typ der Operation. Mögliche Werte: 0 - Nachricht senden; 1 - globales Skript; 2 - Host hinzufügen; 3 - Host entfernen; 4 - zu Hostgruppe hinzufügen; 5 - aus Hostgruppe entfernen; 6 - Vorlage verknüpfen; 7 - Verknüpfung der 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 Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“ oder „bei Aktualisierung des Servicestatus erzeugtes Ereignis“ gesetzt ist: 0 - „Nachricht senden“; 1 - „globales Skript“. Mögliche Werte, wenn `eventsource` des Aktionsobjekts auf „internes Ereignis“ gesetzt ist: 0 - „Nachricht senden“. Eigenschaftsverhalten: - *erforderlich*
esc_period	string	Dauer eines Eskalationsschritts in Sekunden. Muss größer als 60 Sekunden sein. Akzeptiert Sekunden, eine Zeiteinheit mit Suffix oder ein Benutzermakro. Wenn auf 0 oder 0s gesetzt, wird die Standard-Eskalationsperiode der Aktion verwendet. Standard: 0s. Eigenschaftsverhalten: - *unterstützt*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“, „internes Ereignis“ oder „bei Aktualisierung des Servicestatus erzeugtes Ereignis“ gesetzt ist
esc_step_from	integer	Schritt, ab dem die Eskalation beginnt. Standard: 1. Eigenschaftsverhalten: - *unterstützt*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“, „internes Ereignis“ oder „bei Aktualisierung des Servicestatus erzeugtes Ereignis“ gesetzt ist
esc_step_to	integer	Schritt, bei dem die Eskalation endet. Standard: 1. Eigenschaftsverhalten: - *unterstützt*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“, „internes Ereignis“ oder „bei Aktualisierung des Servicestatus erzeugtes Ereignis“ gesetzt ist
evaltype	integer	Auswertungsmethode der Operationsbedingung. Mögliche Werte: 0 - *(Standard)* Und/Oder; 1 - Und; 2 - Oder.

Eigenschaft	Typ	Beschreibung
opcommand	object	Globales Skript, das ausgeführt werden soll. Für das globale Skript muss die Eigenschaft `scriptid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „globales Skript“ gesetzt ist
opcommand_grp	array	Hostgruppen, auf denen globale Skripte ausgeführt werden. Für die Hostgruppen muss die Eigenschaft `groupid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „globales Skript“ gesetzt ist und `opcommand_hst` nicht gesetzt ist
opcommand_hst	array	Host, auf dem globale Skripte ausgeführt werden. Für die Hosts muss die Eigenschaft `hostid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „globales Skript“ gesetzt ist und `opcommand_grp` nicht gesetzt ist
opconditions	array	Operationsbedingungen, die für Auslöser-Aktionen verwendet werden. Das Objekt der Operationsbedingung wird weiter unten ausführlich beschrieben.
opgroup	array	Hostgruppen, zu denen Hosts hinzugefügt werden. Für die Hostgruppen muss die Eigenschaft `groupid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „zu Hostgruppe hinzufügen“ oder „aus Hostgruppe entfernen“ gesetzt ist
opmessage	object	Objekt, das die Daten über die von der Operation gesendete Nachricht enthält. Das Objekt der Operationsnachricht wird weiter unten ausführlich beschrieben. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist
opmessage_grp	array	Benutzergruppen, an die Nachrichten gesendet werden. Für die Benutzergruppen muss die Eigenschaft `usrgrpid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist und `opmessage_usr` nicht gesetzt ist
opmessage_usr	array	Benutzer, an die Nachrichten gesendet werden. Für die Benutzer muss die Eigenschaft `userid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist und `opmessage_grp` nicht gesetzt ist
optemplate	array	Vorlagen, die mit den Hosts verknüpft werden. Für die Vorlagen muss die Eigenschaft `templateid` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „Vorlage verknüpfen“ oder „Verknüpfung der Vorlage aufheben“ gesetzt ist

Eigenschaft	Typ	Beschreibung
opinventory	object	Inventarmodus, auf den der Host gesetzt wird. Für das Inventar muss die Eigenschaft `inventory_mode` definiert sein. Eigenschaftsverhalten: - *erforderlich*, wenn `operationtype` auf „Inventarmodus des Hosts festlegen“ gesetzt ist
optag	array	Host-Tags, die hinzugefügt oder entfernt werden. Für Tags muss die Eigenschaft `tag` definiert sein. Die Eigenschaft `value` ist optional. Eigenschaftsverhalten: - *unterstützt*, 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

Aktionsfilterbedingung

Das Objekt der Aktionsfilterbedingung definiert eine bestimmte Bedingung, die vor der Ausführung der Aktionsoperationen geprüft werden muss.

Property	Type	Beschreibung
conditiontype	integer	Typ der Bedingung. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „durch einen Auslöser erzeugtes Ereignis“ gesetzt ist: 0 - Host-Gruppe; 1 - Host; 2 - Auslöser; 3 - Ereignisname; 4 - Auslöser-Schweregrad; 6 - Zeitperiode; 13 - Host-Vorlage; 16 - Problem ist unterdrückt; 25 - Ereignis-Tag; 26 - Ereignis-Tag-Wert. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „durch eine Discovery-Regel erzeugtes Ereignis“ gesetzt ist: 7 - Host-IP; 8 - Typ des erkannten Dienstes; 9 - Port des erkannten Dienstes; 10 - Discovery-Status; 11 - Dauer der Verfügbarkeit oder Nichtverfügbarkeit; 12 - empfangener Wert; 18 - Discovery-Regel; 19 - Discovery-Prüfung; 20 - Proxy; 21 - Discovery-Objekt. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „durch aktive Agent-Autoregistrierung erzeugtes Ereignis“ gesetzt ist: 20 - Proxy; 22 - Host-Name; 24 - Host-Metadaten. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „internes Ereignis“ gesetzt ist: 0 - Host-Gruppe; 1 - Host; 13 - Host-Vorlage; 23 - Ereignistyp; 25 - Ereignis-Tag; 26 - Ereignis-Tag-Wert. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „bei Aktualisierung des Service-Status erzeugtes Ereignis“ gesetzt ist: 25 - Ereignis-Tag; 26 - Ereignis-Tag-Wert; 27 - Service; 28 - Service-Name. Property behavior: - *required*
value	string	Wert, mit dem verglichen werden soll. Property behavior: - *required*

Property	Type	Beschreibung
value2	string	Sekundärer Wert, mit dem verglichen werden soll. Property behavior: - *required*, wenn `eventsource` des Action-Objekts auf „durch einen Auslöser erzeugtes Ereignis“ gesetzt ist, `conditiontype` auf einen beliebigen möglichen Wert für Auslöser-Aktionen gesetzt ist und der Bedingungstyp (siehe unten) „26“ ist - *required*, wenn `eventsource` des Action-Objekts auf „internes Ereignis“ gesetzt ist, `conditiontype` auf einen beliebigen möglichen Wert für interne Aktionen gesetzt ist und der Bedingungstyp (siehe unten) „26“ ist - *required*, wenn `eventsource` des Action-Objekts auf „bei Aktualisierung des Service-Status erzeugtes Ereignis“ gesetzt ist, `conditiontype` auf einen beliebigen möglichen Wert für Service-Aktionen gesetzt ist und der Bedingungstyp (siehe unten) „26“ 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 beim Ändern von Filterbedingungen definiert werden, wird jedoch bei einer späteren Abfrage erneut generiert.
operator	integer	Bedingungs-Operator. Mögliche Werte: 0 - *(default)* gleich; 1 - ungleich; 2 - enthält; 3 - enthält nicht; 4 - in; 5 - ist größer oder gleich; 6 - ist kleiner oder gleich; 7 - nicht in; 8 - entspricht; 9 - entspricht nicht; 10 - Ja; 11 - Nein.

Note:
Um besser zu verstehen, wie Filter mit verschiedenen Ausdruckstypen verwendet werden, siehe die Beispiele auf den Methodenseiten action.get und action.create.

Die folgenden Operatoren und Werte werden für jeden Bedingungstyp unterstützt.

Condition	Condition name	Supported operators	Expected value
0	Host-Gruppe	gleich, ungleich	ID der Host-Gruppe.
1	Host	gleich, ungleich	Host-ID.
2	Auslöser	gleich, ungleich	Auslöser-ID.
3	Ereignisname	enthält, enthält nicht	Ereignisname.
4	Auslöser-Schweregrad	gleich, ungleich, ist größer oder gleich, ist kleiner oder gleich	Auslöser-Schweregrad. Eine Liste der unterstützten Auslöser-Schweregrade finden Sie in der trigger-Property `severity`.

Condition	Condition name	Supported operators	Expected value
5	Auslöser-Wert	gleich	Auslöser-Wert. Eine Liste der unterstützten Auslöser-Werte finden Sie in der trigger-Property `value`.
6	Zeitperiode	in, nicht in	Zeitpunkt, zu dem das Ereignis ausgelöst wurde, als Zeitperiode.
7	Host-IP	gleich, ungleich	Ein oder mehrere zu prüfende IP-Bereiche, durch Kommas getrennt. Weitere Informationen zu unterstützten Formaten von IP-Bereichen finden Sie im Abschnitt Konfiguration der Netzwerk-Discovery.
8	Typ des erkannten Dienstes	gleich, ungleich	Typ des erkannten Dienstes. Der Diensttyp entspricht dem Typ der Discovery-Prüfung, die zur Erkennung des Dienstes verwendet wurde. Eine Liste der unterstützten Typen finden Sie in der discovery check-Property `type`.
9	Port des erkannten Dienstes	gleich, ungleich	Ein oder mehrere Port-Bereiche, durch Kommas getrennt.
10	Discovery-Status	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 Verfügbarkeit oder Nichtverfügbarkeit	ist größer oder gleich, ist kleiner oder gleich	Zeit in Sekunden, die angibt, wie lange sich das erkannte Objekt bereits im aktuellen Status befindet.
12	Empfangene Werte	gleich, ungleich, ist größer oder gleich, ist kleiner oder gleich, enthält, enthält nicht	Wert, der bei der Durchführung einer Zabbix-Agent-, SNMPv1-, SNMPv2- oder SNMPv3-Discovery-Prüfung zurückgegeben wird.
13	Host-Vorlage	gleich, ungleich	ID der verknüpften Vorlage.
16	Problem ist unterdrückt	Ja, Nein	Kein Wert erforderlich: Die Verwendung des Operators „Ja“ bedeutet, dass das Problem unterdrückt sein muss, „Nein“ bedeutet nicht unterdrückt.
18	Discovery-Regel	gleich, ungleich	ID der Discovery-Regel.
19	Discovery-Prüfung	gleich, ungleich	ID der Discovery-Prüfung.
20	Proxy	gleich, ungleich	ID des Proxy.
21	Discovery-Objekt	gleich	Typ des Objekts, das das Discovery-Ereignis ausgelöst hat. Mögliche Werte: 1 - erkannter Host; 2 - erkannter Dienst.
22	Host-Name	enthält, enthält nicht, entspricht, entspricht nicht	Host-Name. Die Verwendung eines regulären Ausdrucks wird für die Operatoren *entspricht* und *entspricht nicht* in Autoregistrierungsbedingungen unterstützt.
23	Ereignistyp	gleich	Bestimmtes internes Ereignis. Mögliche Werte: 0 - Datenpunkt im Zustand „nicht unterstützt“; 1 - Datenpunkt im Zustand „normal“; 2 - LLD-Regel im Zustand „nicht unterstützt“; 3 - LLD-Regel im Zustand „normal“; 4 - Auslöser im Zustand „unbekannt“; 5 - Auslöser im Zustand „normal“.

Condition	Condition name	Supported operators	Expected value
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	gleich, ungleich, enthält, enthält nicht	Ereignis-Tag.
26	Tag-Wert	gleich, ungleich, enthält, enthält nicht	Ereignis-Tag-Wert.
27	Service	gleich, ungleich	Service-ID.
28	Service-Name	gleich, ungleich	Service-Name.

action.create

Beschreibung

`object action.create(object/array actions)`

Diese Methode ermöglicht das Erstellen neuer Aktionen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Diese Methode ermöglicht das Löschen von Aktionen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super Admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

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 das Abrufen von Aktionen basierend auf den angegebenen Parametern.

Note:
Diese Methode ist für Benutzer aller Art verfügbar. Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen 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 das Aktualisieren vorhandener Aktionen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super Admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen (/manual/web_interface/frontend_sections/users/user_roles).

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 den Abruf von Warnmeldungen entsprechend den angegebenen Parametern.

> **Note:**
> Diese Methode steht Nutzern jeder Art zur Verfügung. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen 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)`

Mit dieser Methode kann die Version der Zabbix-API abgerufen werden.

::: Hinweis: Wichtig 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*.

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)`

Mit dieser Methode können Audit-Log-Einträge entsprechend den angegebenen Parametern abgerufen werden.

> **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.

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 das Erstellen von Aufgaben.

> **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)` 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)`

Mit dieser Methode können Aufgaben entsprechend den angegebenen Parametern abgerufen werden.

> **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.

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)`

Mit dieser Methode können neue Auslöser erstellt werden.

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 entzogen 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 das Löschen von Auslösern.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter User roles.

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)`

Mit dieser Methode können Auslöser entsprechend den angegebenen Parametern abgerufen 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)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
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 Host-Gruppen oder Vorlagen-Gruppen 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.
functions	string/array	Gibt nur Auslöser zurück, die die angegebenen Funktionen verwenden. Eine Liste der unterstützten Funktionen finden Sie auf der Seite supported function.
group	string	Gibt nur Auslöser zurück, die zu Hosts oder Vorlagen aus der Host-Gruppe oder Vorlagen-Gruppe mit dem angegebenen Namen gehören.
host	string	Gibt nur Auslöser zurück, die zu dem 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	Beschreibung
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 nur 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 in Wartung gehören.
withUnacknowledgedEvents	flag	Gibt nur Auslöser zurück, die nicht bestätigte Ereignisse haben.
withAcknowledgedEvents	flag	Gibt nur Auslöser zurück, bei denen alle Ereignisse bestätigt sind.
withLastEventUnacknowledged	flag	Gibt nur Auslöser zurück, bei denen das letzte Ereignis nicht bestätigt ist.
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 deren Datenpunkt-Hosts deaktiviert sind.
lastChangeSince	timestamp	Gibt nur Auslöser zurück, deren Zustand sich nach dem angegebenen Zeitpunkt geändert hat.
lastChangeTill	timestamp	Gibt nur Auslöser zurück, deren Zustand sich vor dem angegebenen Zeitpunkt geändert hat.
only_true	flag	Gibt nur Auslöser zurück, die sich vor Kurzem in einem Problemzustand befanden.
min_severity	integer	Gibt nur Auslöser mit einem Schweregrad zurück, der 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 operator: 0 - *(Standard)* Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht.
expandComment	flag	Erweitert Makros in der Auslöserbeschreibung.
expandDescription	flag	Erweitert Makros im Namen des Auslösers.
expandExpression	flag	Erweitert Funktionen und Makros im Auslöserausdruck.
selectHostGroups	query	Gibt die Host-Gruppen, 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 repräsentieren die im Auslöserausdruck verwendeten Funktionen 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	Beschreibung
selectDependencies	query	Gibt Auslöser, von denen der Auslöser abhängt, in der Eigenschaft `dependencies` zurück.
selectDiscoveryData	query	Gibt eine Eigenschaft `discoveryData` mit den Daten des Auslöser-Erkennungsobjekts zurück. Das Auslöser-Erkennungsobjekt verknüpft einen erkannten Auslöser mit einem Auslöserprototyp, aus dem er erkannt wurde. Es hat die folgenden Eigenschaften: `parent_triggerid` - `(ID)` ID des Auslöserprototyps, aus dem der Auslöser erstellt wurde; `status` - `(int)` Status der Auslösererkennung: 0 - *(Standard)* Auslöser ist 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)` Kennzeichen dafür, 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, die den Auslöser erstellt hat, in der Eigenschaft `discoveryRule` zurück.
selectLastEvent	query	Gibt das letzte signifikante Auslöserereignis in der Eigenschaft `lastEvent` zurück.
selectTags	query	Gibt die Auslöser-Tags in der Eigenschaft `tags` zurück.
inheritedTags	boolean	Gibt Auslöser 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.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit Tags zurück, die von Vorlagen, verknüpften Vorlagen und Hosts geerbt wurden sowie von Datenpunkten, auf die in Auslöser- oder Wiederherstellungsausdrücken verwiesen wird.
selectTemplateGroups	query	Gibt die Vorlagen-Gruppen, zu denen der Auslöser gehört, in der Eigenschaft `templategroups` 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 wird. 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. Mögliche Werte: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.

Parameter	Type	Beschreibung
countOutput	boolean	Diese Parameter sind in der reference commentary 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 Auslöser-Erkennungsobjekt in der Eigenschaft `triggerDiscovery` zurück. Die Auslöser-Erkennungsobjekte verknüpfen den Auslöser mit einem Auslöserprototyp, aus dem er erstellt wurde. Diese Abfrage ist **veraltet**; verwenden Sie stattdessen bitte `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 die Aktualisierung vorhandener Auslöser.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 das Löschen von Auslöser-Prototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen 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 das Erstellen neuer Auslöser-Prototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Auslöser-Prototypen entsprechend den angegebenen Parametern abgerufen 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)` 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)`

Mit dieser Methode können vorhandene Auslöser-Prototypen aktualisiert werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
authentication_type	integer	Standardauthentifizierung. Mögliche Werte: 0 - *(Standard)* Intern; 1 - LDAP.

Eigenschaft	Type	Beschreibung
http_auth_enabled	integer	HTTP-Authentifizierung. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Eigenschaftsverhalten: - *unterstützt*, wenn `$ALLOW_HTTP_AUTH` in der Frontend-Konfigurationsdatei (*zabbix.conf.php*) aktiviert ist.
http_login_form	integer	Standard-Anmeldeformular. Mögliche Werte: 0 - *(Standard)* Zabbix-Anmeldeformular; 1 - HTTP-Anmeldeformular. Eigenschaftsverhalten: - *unterstützt*, wenn `$ALLOW_HTTP_AUTH` in der Frontend-Konfigurationsdatei (*zabbix.conf.php*) aktiviert ist.
http_strip_domains	string	Zu entfernender Domainname. Eigenschaftsverhalten: - *unterstützt*, wenn `$ALLOW_HTTP_AUTH` in der Frontend-Konfigurationsdatei (*zabbix.conf.php*) aktiviert ist.
http_case_sensitive	integer	HTTP-Anmeldung mit Berücksichtigung der Groß-/Kleinschreibung. Mögliche Werte: 0 - Aus; 1 - *(Standard)* Ein. Eigenschaftsverhalten: - *unterstützt*, wenn `$ALLOW_HTTP_AUTH` in der Frontend-Konfigurationsdatei (*zabbix.conf.php*) aktiviert ist.
ldap_auth_enabled	integer	LDAP-Authentifizierung. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
ldap_case_sensitive	integer	LDAP-Anmeldung mit Berücksichtigung der Groß-/Kleinschreibung. Mögliche Werte: 0 - Aus; 1 - *(Standard)* Ein.
ldap_userdirectoryid	ID	ID des Standard-Benutzerverzeichnisses für die LDAP-Authentifizierung. Wird für Benutzergruppen verwendet, bei denen `gui_access` auf LDAP oder Systemstandard gesetzt ist. Eigenschaftsverhalten: - *erforderlich*, wenn `ldap_auth_enabled` auf "Aktiviert" gesetzt ist
saml_auth_enabled	integer	SAML-Authentifizierung. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
saml_case_sensitive	integer	SAML-Anmeldung mit Berücksichtigung der Groß-/Kleinschreibung. Mögliche Werte: 0 - Aus; 1 - *(Standard)* Ein.

Eigenschaft	Type	Beschreibung
passwd_min_length	integer	Anforderung an die minimale Passwortlänge. Mögliche Werte reichen von 1 bis 70. Standard: 8.
passwd_check_rules	integer	Regeln zur Passwortprüfung. Mögliche Bitmap-Werte: 0 - Passwortlänge prüfen; 1 - Prüfen, ob das Passwort lateinische Groß- und Kleinbuchstaben verwendet; 2 - Prüfen, ob das Passwort Ziffern verwendet; 4 - Prüfen, ob das Passwort Sonderzeichen verwendet; 8 - *(Standard)* 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 der möglichen Bitmap-Werte ist zulässig (zum Beispiel 15 für die Prüfung aller Regeln).
ldap_jit_status	integer	Status der LDAP-Bereitstellung. Mögliche Werte: 0 - Deaktiviert für konfigurierte LDAP-IdPs; 1 - Aktiviert für konfigurierte LDAP-IdPs.
saml_jit_status	integer	Status der SAML-Bereitstellung. Mögliche Werte: 0 - Deaktiviert für konfigurierte SAML-IdPs; 1 - Aktiviert für konfigurierte SAML-IdPs.
jit_provision_interval	string	Zeitintervall zwischen JIT-Bereitstellungsanfragen für den angemeldeten Benutzer. Akzeptiert Sekunden und Zeiteinheiten mit Suffix einschließlich Unterstützung für Monate und Jahre (3600s,60m,1h,1d,1M,1y). Mindestwert: 1h. Standard: 1h. Nur für LDAP-Bereitstellung verfügbar.
disabled_usrgrpid	ID	ID der Benutzergruppe, der der deprovisionierte Benutzer zugewiesen wird. Die Benutzergruppe muss deaktiviert sein und kann, solange sie konfiguriert ist, nicht aktiviert oder gelöscht werden. Eigenschaftsverhalten: - *erforderlich*, wenn `ldap_jit_status` auf ”Aktiviert für konfigurierte LDAP-IdPs” gesetzt ist oder `saml_jit_status` auf ”Aktiviert für konfigurierte 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. Eigenschaftsverhalten: - *erforderlich*, wenn `mfa_status` auf ”Aktiviert” gesetzt ist

authentication.get

Beschreibung

`object authentication.get(object parameters)`

Die Methode ermöglicht das Abrufen eines Authentifizierungsobjekts gemäß den angegebenen Parametern.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadministrator* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen for more information.

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)`

Mit dieser Methode können Sie bestehende Authentifizierungseinstellungen aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Prüfen Sie User roles für mehr 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 das Abrufen des Autoregistrierungsobjekts gemäß den angegebenen Parametern.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar.. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Prüfen Sie User roles für mehr Informationen.

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 die Aktualisierung bestehender Autoregistrierungen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Prüfen Sie Benutzer rollen für mehr Informationen.

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 hat 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 gesetzt, der nicht "0" ist), und der Bereitstellungsstatus des Benutzerverzeichnisses aktiviert ist (`provision_status` des Benutzerverzeichnisobjekts ist auf "1" gesetzt), und der Authentifizierungsstatus aller LDAP- oder SAML-Bereitstellungen 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 schreibbar*
roleid	ID	ID der Rolle des Benutzers. Beachten Sie, dass Benutzer ohne Rolle sich nur mit LDAP- oder SAML-Authentifizierung bei Zabbix anmelden können, sofern ihre LDAP-/SAML-Informationen mit den in Zabbix konfigurierten Benutzergruppenzuordnungen übereinstimmen.
attempt_clock	timestamp	Zeit des letzten fehlgeschlagenen Anmeldeversuchs. Verhalten der Eigenschaft: - *schreibgeschützt*
attempt_failed	integer	Anzahl der letzten fehlgeschlagenen Anmeldeversuche. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
attempt_ip	string	IP-Adresse, von der der letzte fehlgeschlagene Anmeldeversuch kam. 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 Sekunden und Zeiteinheiten mit Suffix. Wenn auf 0s gesetzt, läuft die Sitzung nie ab. Standard: 15m.
lang	string	Sprachcode der Benutzersprache, zum Beispiel `en_US`. Standard: `default` - Systemstandard.
name	string	Vorname 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 Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 90s, 1m, 1h). Standard: 30s.
rows_per_page	integer	Anzahl der pro Seite anzuzeigenden Objektzeilen. Standard: 50.
surname	string	Nachname des Benutzers.
theme	string	Theme des Benutzers. Mögliche Werte: `default` - *(Standard)* Systemstandard; `blue-theme` - Blau; `dark-theme` - Dunkel.
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, zu der der Benutzer nach der Anmeldung weitergeleitet wird.
userdirectoryid	ID	ID des Benutzerverzeichnisses, mit dem der Benutzer verknüpft ist. Wird für die Bereitstellung (Erstellen oder Aktualisieren) 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*

Eigenschaft	Typ	Beschreibung
timezone	string	Zeitzone des Benutzers, zum Beispiel `Europe/London`, `UTC`. Standard: `default` - Systemstandard. Eine vollständige Liste der unterstützten Zeitzonen finden Sie in der PHP-Dokumentation.

Medien

Das Medienobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
mediaid	ID	ID des Mediums des Benutzers. Verhalten von Eigenschaften: - *schreibgeschützt*
mediatypeid	ID	ID des Medientyps, der vom Medium des Benutzers verwendet wird. Verhalten von Eigenschaften: - *erforderlich*
sendto	string/array	Adresse, Benutzername oder andere Kennung des Empfängers. Wenn `type` des Medientyps auf „Email“ gesetzt ist, werden Werte als Array dargestellt. Bei anderen Typen von Medientypen wird der Wert als String dargestellt. Verhalten von Eigenschaften: - *erforderlich*
active	integer	Gibt an, ob das Medium aktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
severity	integer	Auslöser-Schweregrade, für die Benachrichtigungen gesendet werden. Mögliche Bitmap-Werte: 1 - Nicht klassifiziert; 2 - Information; 4 - Warnung; 8 - Durchschnittlich; 16 - Hoch; 32 - Katastrophe. Dies ist ein Bitmaskenfeld; jede Summe der möglichen 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 Semikolon 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.

Eigenschaft	Typ	Beschreibung
userdirectory_mediaid	ID	Zuordnungs-ID des Benutzerverzeichnismediums für bereitgestellte Medien. Verhalten von Eigenschaften: - *schreibgeschützt* - *unterstützt* für Benutzer des Typs *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	Gibt an, ob die Benutzersitzung verlängert werden soll. Standardwert: ”true”. Wenn der Wert auf ”false” gesetzt wird, kann die Benutzersitzung geprüft werden, ohne sie zu verlängern.
sessionid	string	Parameterverhalten: - *unterstützt*, wenn `sessionid` gesetzt ist Authentifizierungs-Token des Benutzers.
token	string	Parameterverhalten: - *erforderlich*, wenn `token` nicht gesetzt ist API-Token des Benutzers. Parameterverhalten: - *erforderlich*, 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.

Eigenschaft	Type	Beschreibung
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)`

Diese Methode ermöglicht das Erstellen neuer Benutzer.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

> **Note:**
> Die Stärke des Benutzerpassworts wird gemäß den durch die Authentication API definierten Regeln der Passwortrichtlinie validiert. Siehe Authentication API für weitere Informationen.

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

Einen Benutzer erstellen

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": "Doe123",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

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 das Löschen von Benutzern.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Benutzer entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. 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.

> **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.
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.

Parameter	Typ	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, 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`.
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",
            "refresh": "0s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "3",
            "username": "user",
            "name": "Zabbix",
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
```

```
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
    "id": 1
}
```

Abrufen von Benutzern als *Admin*

Als Benutzer vom Typ *Admin* rufen Sie detaillierte Daten über Ihren eigenen Benutzer sowie eingeschränkte Daten für Benutzer in Ihrer Benutzergruppe ab.

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",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "userid": "3",
            "username": "database-admin",
            "name": "John",
            "surname": "Doe",
            "url": "",
            "autologin": "0",
            "autologout": "0",
```

```
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "2",
            "provisioned": "0",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```


Abrufen von Benutzerdaten

Rufen Sie die Daten eines Benutzers mit der ID „12“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

Antwort:

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "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)`

Mit dieser Methode können Sie sich bei der API anmelden und ein Authentifizierungs-Token erzeugen.

Warning:
Bei Verwendung dieser Methode müssen Sie außerdem user.logout ausführen, um die Erzeugung einer großen Anzahl offener Sitzungseinträge 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 die Abmeldung von der API und macht das aktuelle Authentifizierungs-Token ungültig.

> **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

`(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)`

Mit dieser Methode können LDAP-Benutzer bereitgestellt werden.

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.

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)`

Mit dieser Methode können die TOTP-Geheimnisse von Benutzern zurückgesetzt werden.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen 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, 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 entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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 die Aktualisierung bestehender Benutzer.

Note:
Diese Methode ist für Benutzer jedes Typs verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

Note:
Die Stärke des Benutzerpassworts wird gemäß den durch die Authentication API definierten Regeln der Passwortrichtlinie validiert. Siehe Authentication API für weitere Informationen.

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 hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
usrgrpid	ID	ID der Benutzergruppe. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Benutzergruppe. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
debug_mode	integer	Ob der Debug-Modus aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* deaktiviert; 1 - aktiviert.
gui_access	integer	Frontend-Authentifizierungsmethode der Benutzer in der Gruppe. Mögliche Werte: 0 - *(Standard)* die Standard-Authentifizierungsmethode des Systems verwenden; 1 - interne Authentifizierung verwenden; 2 - LDAP-Authentifizierung verwenden; 3 - Zugriff auf das Frontend deaktivieren.
mfa_status	integer	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).

Eigenschaft	Typ	Beschreibung
mfaid	ID	Für die Benutzer in der Gruppe verwendete MFA-Methode. Verhalten der Eigenschaft: - *unterstützt* für Benutzer vom Typ *Super admin* - *unterstützt*, wenn `mfa_status` des Authentifizierungsobjekts auf "Enabled" gesetzt ist
users_status	integer	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 für die Authentifizierung verwendeten Benutzerverzeichnisses. Verhalten der Eigenschaft: - *unterstützt* für Benutzer vom Typ *Super admin* - *unterstützt*, wenn `gui_access` auf "die Standard-Authentifizierungsmethode des Systems verwenden" oder "LDAP-Authentifizierung verwenden" 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*
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.

usergroup.create

Beschreibung

`object usergroup.create(object/array userGroups)`

Diese Methode ermöglicht das Erstellen neuer Benutzergruppen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* 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 erstellende Benutzergruppen.

Zusätzlich zu den Standard-Benutzergruppeneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
hostgroup_rights	object/array	Berechtigungen für Host-Gruppen, die der Benutzergruppe zugewiesen werden.
templategroup_rights	object/array	Berechtigungen für Vorlagengruppen, die der Benutzergruppe zugewiesen werden.
tag_filters	array	Tag-basierte Berechtigungen, die der Benutzergruppe zugewiesen werden.
users	object/array	Benutzer, die der Benutzergruppe hinzugefügt werden. Für den Benutzer darf nur die Eigenschaft `userid` definiert sein.

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 Benutzergruppen gelöscht werden.

Attention:
Die Benutzergruppe Deprovisioned (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 entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

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)`

Mit dieser Methode können Benutzergruppen entsprechend den angegebenen Parametern abgerufen 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)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
mfaids	ID/array	Gibt nur Benutzergruppen mit den angegebenen MFA-Methoden zurück. Parameterverhalten: - *unterstützt* für Benutzer vom Typ *Super admin*
mfa_status	integer	Gibt nur Benutzergruppen mit dem angegebenen MFA-Status zurück. Eine Liste der unterstützten Status finden Sie auf der Seite Benutzergruppe.
status	integer	Gibt nur Benutzergruppen mit dem angegebenen Status zurück. Eine Liste der unterstützten Status finden Sie auf der Seite Benutzergruppe.
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 tagbasierte Berechtigungen der Benutzergruppe in der Eigenschaft `tag_filters` zurück.
selectUsers	query	Gibt die Benutzer aus der Benutzergruppe in der Eigenschaft `users` zurück. Siehe `user.get` für Einschränkungen basierend auf dem Benutzertyp.
selectHostGroupRights	query	Gibt Berechtigungen der Host-Gruppe der Benutzergruppe in der Eigenschaft `hostgroup_rights` zurück. Eine Liste der Zugriffsebenen auf Host-Gruppen finden Sie auf der Seite Benutzergruppe.
selectTemplateGroupRights	query	Gibt Berechtigungen der Vorlagengruppe der Benutzergruppe in der Eigenschaft `templategroup_rights` zurück. Eine Liste der Zugriffsebenen auf Vorlagengruppen finden Sie auf der Seite Benutzergruppe.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze.
output	query	Eigenschaften des Benutzergruppenobjekts, die zurückgegeben werden sollen. Benutzer vom Typ *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`.

Parameter	Type	Beschreibung
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung 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)`

Mit dieser Methode können vorhandene Benutzergruppen aktualisiert werden.

> **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. Siehe Benutzerrollen für weitere Informationen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften der Benutzergruppe.

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-Benutzergruppeneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
hostgroup_rights	object/array	Berechtigungen für Host-Gruppen, um die aktuell der Benutzergruppe zugewiesenen Berechtigungen zu ersetzen.
templategroup_rights	object/array	Berechtigungen für Vorlagengruppen, um die aktuell der Benutzergruppe zugewiesenen Berechtigungen zu ersetzen.
tag_filters	array	Tag-basierte Berechtigungen, um die aktuell der Benutzergruppe zugewiesenen Berechtigungen zu ersetzen.
users	object/array	Benutzer, um die aktuell der Benutzergruppe zugewiesenen Benutzer zu ersetzen. 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
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

Eigenschaft	Typ	Beschreibung
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.

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode können neue Host-Makros erstellt werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen 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)`

Mit dieser Methode können neue globale Makros erstellt werden.

> **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. Siehe Benutzerrollen für weitere Informationen.

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".

Anfrage:

```
{
    "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)`

Mit dieser Methode können Host-Makros gelöscht werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter User roles.

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 das Löschen globaler Makros.

> **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.

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)`

Mit dieser Methode können Host- und globale Makros entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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	
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 die Aktualisierung vorhandener Host-Makros.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 die Aktualisierung vorhandener globaler Makros.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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 Benutzerverzeichnisobjekt hat die folgenden Eigenschaften.

Property	Type	Description
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 Authentifizierungsprotokolls, das vom Identitätsanbieter für das Benutzerverzeichnis verwendet wird. 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
group_name	string	LDAP-/SAML-Benutzerverzeichnisattribut, das den Gruppennamen enthält, der zum Zuordnen 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	LDAP-/SAML-Benutzerverzeichnisattribut (auch SCIM-Attribut, wenn `scim_status` auf "SCIM provisioning is enabled" gesetzt ist), das den Namen des Benutzers enthält, der bei der Bereitstellung des Benutzers als Wert für die Eigenschaft `name` des Benutzerobjekts verwendet wird. Beispiele: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	LDAP-/SAML-Benutzerverzeichnisattribut (auch SCIM-Attribut, wenn `scim_status` auf "SCIM provisioning is enabled" gesetzt ist), das den Nachnamen des Benutzers enthält, der bei der Bereitstellung des Benutzers als Wert für die Eigenschaft `surname` des Benutzerobjekts verwendet wird. Beispiele: *sn*, *surname*, *lastName*

Property	Type	Description
provision_status	integer	Bereitstellungsstatus des Benutzerverzeichnisses. Mögliche Werte: 0 - *(Standard)* Deaktiviert (die Bereitstellung von Benutzern, die durch dieses Benutzerverzeichnis erstellt wurden, ist deaktiviert); 1 - Aktiviert (die Bereitstellung von Benutzern, die durch dieses Benutzerverzeichnis erstellt wurden, 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 zur Zuordnung von LDAP-/SAML-Benutzergruppenmustern zu Zabbix-Benutzergruppen und Benutzerrollen. Verhalten der Eigenschaft: - *erforderlich*, wenn `provision_status` auf "Enabled" gesetzt ist
provision_media	array	Array von Objekten für Medientypzuordnungen zur Zuordnung von LDAP-/SAML-Medienattributen des Benutzers (z. B. E-Mail) zu Zabbix-Benutzermedien zum Senden von Benachrichtigungen.
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	Host-Name, IP oder URI des LDAP-Servers. Die URI muss das Schema (`ldap://` oder `ldaps://`), den Host und den Port (optional) 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
port	integer	Port des LDAP-Servers. Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
base_dn	string	LDAP-Benutzerverzeichnis-Basispfad zu 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; Platzhalter "*%{user}*" ist obligatorisch) Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist

Property	Type	Description
search_attribute	string	LDAP-Benutzerverzeichnisattribut, anhand dessen das Benutzerkonto anhand der in der Anmeldeanforderung 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	LDAP-Serverkonto für Bindung und Suche auf dem LDAP-Server. Für direkte Benutzerbindung und anonyme 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 für Bindung und Suche auf dem LDAP-Server. Für direkte Benutzerbindung und anonyme 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
group_basedn	string	LDAP-Benutzerverzeichnis-Basispfad zu Gruppen; wird verwendet, um eine Prüfung der Benutzergruppenmitgliedschaft im LDAP-Benutzerverzeichnis zu konfigurieren. Wird 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

Property	Type	Description
group_filter	string	Filterzeichenfolge zum Abrufen von LDAP-Benutzerverzeichnisgruppen, deren Mitglied der Benutzer ist; wird verwendet, um eine Prüfung der Benutzergruppenmitgliedschaft im LDAP-Benutzerverzeichnis zu konfigurieren. Wird bei der Bereitstellung eines Benutzers ignoriert, wenn `group_membership` gesetzt ist. Unterstützte Platzhalter für `group_filter`: *%{attr}* - Suchattribut (ersetzt durch den Eigenschaftswert `search_attribute`); *%{groupattr}* - Gruppenattribut (ersetzt durch den Eigenschaftswert `group_member`); *%{host}* - Host-Name, IP oder URI des LDAP-Servers (ersetzt durch den Eigenschaftswert `host`); *%{user}* - Zabbix-Benutzername. Standard: *(%{groupattr}=%{user})* Beispiele: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* entspricht "User1", wenn ein LDAP-Gruppenobjekt das Attribut "*member*" mit dem Wert "*uid=User1,ou=Users,dc=example,dc=com*" enthält, und gibt die Gruppe zurück, deren Mitglied "User1" ist; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* entspricht "User1", 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, deren Mitglied "User1" ist. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
group_member	string	LDAP-Benutzerverzeichnisattribut, das Informationen über die Gruppenmitglieder enthält; wird verwendet, um eine Prüfung der Benutzergruppenmitgliedschaft im LDAP-Benutzerverzeichnis zu konfigurieren. Wird 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
group_membership	string	LDAP-Benutzerverzeichnisattribut, 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

Property	Type	Description
search_filter	string	Benutzerdefinierte Filterzeichenfolge, die verwendet wird, um einen Benutzer in einem LDAP-Benutzerverzeichnis anhand der in der Anmeldeanforderung bereitgestellten Informationen zu finden und zu authentifizieren. Unterstützte Platzhalter für `search_filter`: *%{attr}* - Name des Suchattributs (z. B. *uid*, *sAMAccountName*); *%{user}* - Zabbix-Benutzername. Standard: *(%{attr}=%{user})* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
start_tls	integer	Konfigurationsoption des LDAP-Servers, die es ermöglicht, die Kommunikation mit dem LDAP-Server mithilfe von Transport Layer Security (TLS) abzusichern. Beachten Sie, dass `start_tls` für Hosts, die das Protokoll `ldaps://` verwenden, auf "Disabled" gesetzt sein muss. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
user_ref_attr	string	LDAP-Benutzerverzeichnisattribut, das verwendet wird, um auf ein Benutzerobjekt zu verweisen. Der Wert von `user_ref_attr` wird verwendet, um Werte aus dem angegebenen Attribut im Benutzerverzeichnis abzurufen und sie anstelle des Platzhalters *%{ref}* in die 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
sp_entityid	string	URL oder eine 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

Property	Type	Description
username_attribute	string	SAML-Benutzerverzeichnisattribut (auch SCIM-Attribut, wenn `scim_status` auf "SCIM provisioning is enabled" gesetzt ist), das den Benutzernamen des Benutzers enthält, der bei der Authentifizierung mit dem Wert der Eigenschaft `username` des Benutzerobjekts verglichen wird. 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 die Zabbix die SAML-Authentifizierungsanforderungen 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 die Zabbix die SAML-Abmeldeanforderungen 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-Name-ID verschlüsselt werden soll. Mögliche Werte: 0 - *(Standard)* Name-ID nicht verschlüsseln; 1 - Name-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
nameid_format	string	Name-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

Property	Type	Description
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 - *(Standard)* Assertions nicht signieren; 1 - 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-Anforderungen mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - *(Standard)* AuthN-Anforderungen nicht signieren; 1 - AuthN-Anforderungen 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 - *(Standard)* Nachrichten nicht signieren; 1 - 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-Abmeldeanforderungen mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - *(Standard)* Abmeldeanforderungen nicht signieren; 1 - Abmeldeanforderungen signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
sign_logout_responses	integer	Gibt an, ob die SAML-Abmeldeantworten mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - *(Standard)* Abmeldeantworten nicht signieren; 1 - Abmeldeantworten signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist

Property	Type	Description
idp_certificate	string	Inhalt des Service-Provider-(SP-)Zertifikats für die Einrichtung des SAML-Single-Sign-on-(SSO-)Dienstes. Beispiele: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in zabbix.conf.php auf `database` gesetzt ist - *nur schreibbar*
sp_private_key	string	Inhalt des privaten Schlüssels des Service Providers (SP) für die Einrichtung des SAML-Single-Sign-on-(SSO-)Dienstes. Ermöglicht sichere Authentifizierung und sicheren Datenaustausch mit dem Identity Provider (IdP). Beispiele: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in zabbix.conf.php auf `database` gesetzt ist - *nur schreibbar*
sp_certificate	string	Inhalt des Service-Provider-(SP-)Zertifikats für die Einrichtung des SAML-Single-Sign-on-(SSO-)Dienstes. Beispiele: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in zabbix.conf.php auf `database` gesetzt ist - *nur schreibbar*
idp_certificate_hash	string	Der md5-Hash des Werts von idp_certificate. Gibt für ein leeres idp_certificate eine leere Zeichenfolge zurück. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in zabbix.conf.php auf `database` gesetzt ist - *schreibgeschützt*
sp_private_key_hash	string	Der md5-Hash des Werts von sp_private_key. Gibt für ein leeres sp_private_key eine leere Zeichenfolge zurück. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in zabbix.conf.php auf `database` gesetzt ist - *schreibgeschützt*

Property	Type	Description
sp_certificate_hash	string	Der md5-Hash des Werts von sp_certificate. Gibt für ein leeres sp_certificate eine leere Zeichenfolge zurück. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in zabbix.conf.php auf `database` gesetzt ist - *schreibgeschützt*

Zuordnungen von Medientypen

Das Objekt für Medientyp-Zuordnungen hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
userdirectory_media	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 das Erstellen neuer Benutzerverzeichnisse.

> **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)`

Mit dieser Methode können Benutzerverzeichnisse gelöscht werden. Ein Benutzerverzeichnis kann nicht gelöscht werden, wenn es direkt von mindestens einer Benutzergruppe verwendet wird.
Das Standard-LDAP-Benutzerverzeichnis kann nicht gelöscht werden, wenn `authentication.ldap_configured` auf 1 gesetzt ist oder wenn weitere Benutzerverzeichnisse 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)`

Mit dieser Methode können Benutzerverzeichnisse entsprechend den angegebenen Parametern abgerufen werden.

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 die Verbindungseinstellungen des Benutzerverzeichnisses getestet werden.

Note:
Mit dieser Methode kann auch getestet werden, welche konfigurierten Daten mit den Einstellungen des Benutzerverzeichnisses für die Benutzerbereitstellung übereinstimmen (z. B. welche Benutzerrolle, Benutzergruppen und Benutzermedien dem Benutzer zugewiesen werden). Für diese Art von Test sollte die API-Anfrage für ein Benutzerverzeichnis gestellt 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)`

Mit dieser Methode können vorhandene Benutzerverzeichnisse aktualisiert werden.

> **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 hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
hk_events_mode	integer	Internes Housekeeping für Ereignisse und Warnungen aktivieren. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
hk_events_trigger	string	Speicherzeitraum für Auslöser-Daten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 365d.
hk_events_service	string	Speicherzeitraum für Service-Daten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 1d.
hk_events_internal	string	Speicherzeitraum für interne Daten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 1d.
hk_events_discovery	string	Speicherzeitraum für Netzwerkdiscovery-Daten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 1d.
hk_events_autoreg	string	Speicherzeitraum für Autoregistrierungsdaten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 1d.
hk_services_mode	integer	Internes Housekeeping für Services aktivieren. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
hk_services	string	Speicherzeitraum für Service-Daten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 365d.
hk_audit_mode	integer	Internes Housekeeping für Audit aktivieren. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.

Eigenschaft	Typ	Beschreibung
hk_audit	string	Speicherzeitraum für Audit-Daten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 31d.
hk_sessions_mode	integer	Internes Housekeeping für Sitzungen aktivieren. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
hk_sessions	string	Speicherzeitraum für Sitzungsdaten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 31d.
hk_history_mode	integer	Internes Housekeeping für Verlauf aktivieren. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
hk_history_global	integer	Verlaufszeitraum des Datenpunkts überschreiben. Mögliche Werte: 0 - Nicht überschreiben; 1 - *(Standard)* Überschreiben.
hk_history	string	Speicherzeitraum für Verlaufsdaten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 31d.
hk_trends_mode	integer	Internes Housekeeping für Trends aktivieren. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
hk_trends_global	integer	Trendzeitraum des Datenpunkts überschreiben. Mögliche Werte: 0 - Nicht überschreiben; 1 - *(Standard)* Überschreiben.
hk_trends	string	Speicherzeitraum für Trenddaten. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 365d.
db_extension	string	Konfigurations-Flag für die DB-Erweiterung. Wenn dieses Flag auf ”timescaledb” gesetzt ist, ändert der Server sein Verhalten für Housekeeping und das Löschen von Datenpunkten. Eigenschaftsverhalten: - *schreibgeschützt*
compression_availability	integer	Gibt an, ob Datenkomprimierung von der Datenbank (oder ihrer Erweiterung) unterstützt wird. Mögliche Werte: 0 - Nicht verfügbar; 1 - Verfügbar. Eigenschaftsverhalten: - *schreibgeschützt*
compression_status	integer	TimescaleDB-Komprimierung für Verlauf und Trends aktivieren. Mögliche Werte: 0 - *(Standard)* Aus; 1 - Ein.

Eigenschaft	Typ	Beschreibung
compress_older	string	Verlaufs- und Trenddatensätze komprimieren, die älter als der angegebene Zeitraum sind. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 7d.

housekeeping.get

Beschreibung

`object housekeeping.get(object parameters)`

Diese Methode ermöglicht es, das Housekeeping-Objekt entsprechend den angegebenen Parametern abzurufen.

> **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)` 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)`

Mit dieser Methode können vorhandene Housekeeping-Einstellungen aktualisiert werden.

> **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.

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 hat 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
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	Uhrzeit des Tages in Sekunden, zu der der Bericht zum Versand vorbereitet wird. Standard: 0.

Eigenschaft	Typ	Beschreibung
weekdays	integer	Wochentage für den Versand des Berichts. 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). Standard: 0. Eigenschaftsverhalten: - *erforderlich*, wenn `cycle` auf ”weekly” gesetzt ist.
active_since	string	Datum, ab dem begonnen wird. Mögliche Werte: leere Zeichenfolge - *(Standard)* nicht angegeben (wird als 0 gespeichert); bestimmtes Datum im Format YYYY-MM-DD (wird als Zeitstempel für den Tagesbeginn (00:00:00) gespeichert).
active_till	string	Datum, an dem beendet wird. Mögliche Werte: leere Zeichenfolge - *(Standard)* nicht angegeben (wird als 0 gespeichert); bestimmtes Datum im Format YYYY-MM-DD (wird als Zeitstempel für das Tagesende (23:59:59) gespeichert).
subject	string	Betreff der Berichtsnachricht.
message	string	Text der Berichtsnachricht.
status	integer	Ob der Bericht aktiviert oder deaktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
description	text	Beschreibung des Berichts.
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 fehlgeschlagen; ”info” enthält Fehlerinformationen; 3 - Bericht wurde erstellt, aber das Senden 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)`

Mit dieser Methode können neue geplante Berichte erstellt werden.

> **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 entzogen 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 das Löschen geplanter Berichte.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen 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 geplante Berichte entsprechend den angegebenen Parametern abgerufen 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)` 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": "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",
            "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": "Monthly report",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Monthly report",
            "message": "Report accompanying text",
            "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)`

Mit dieser Methode können vorhandene geplante Berichte aktualisiert werden.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen 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
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

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode können neue Bilder erstellt werden.

> **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. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Bilder 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 entzogen 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, Bilder entsprechend den angegebenen Parametern abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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	
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 die Aktualisierung vorhandener Bilder.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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*.

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.

Konnektor

Das Konnektor-Objekt hat die folgenden Eigenschaften.

Property	Type	Description
connectorid	ID	ID des Konnektors. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Konnektors. Property behavior: - *erforderlich* für Erstellungsvorgänge
url	string	Endpunkt-URL, d. h. die URL des Empfängers. Benutzermakros werden unterstützt. Property behavior: - *erforderlich* für Erstellungsvorgänge
protocol	integer	Kommunikationsprotokoll. Mögliche Werte: 0 - *(Standard)* Zabbix Streaming Protocol v1.0.

Property	Type	Description
data_type	integer	Datentyp. Mögliche Werte: 0 - *(Standard)* Datenpunkt-Werte; 1 - Ereignisse.
item_value_type	integer	Eine Summe der zu sendenden Datenpunkt-Werttypen. Mögliche Werte: 1 - Numerisch (Gleitkommazahl); 2 - Zeichen; 4 - Log; 8 - Numerisch (vorzeichenlos); 16 - Text; 32 - Binär; 64 - JSON. Standard: 31 - Alle Datenpunkt-Typen (außer Binär und JSON). Property behavior: - *unterstützt*, wenn `data_type` auf ”Datenpunkt-Werte” gesetzt ist.
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 Sender-Prozesse, die für diesen Konnektor ausgeführt werden. Mögliche Werte: 1-100. Standard: 1.
max_attempts	integer	Anzahl der Versuche. Mögliche Werte: 1-5. Standard: 1.
attempt_interval	string	Das Intervall zwischen Wiederholungsversuchen. Akzeptiert Sekunden. Mögliche Werte: 0s-10s. Standard: 5s. Property behavior: - *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.

Property	Type	Description
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. Property behavior: - *unterstützt*, wenn `authtype` auf ”Basic”, ”NTLM”, ”Kerberos” oder ”Digest” gesetzt ist
password	string	Passwort. Benutzermakros werden unterstützt. Property behavior: - *unterstützt*, wenn `authtype` auf ”Basic”, ”NTLM”, ”Kerberos” oder ”Digest” gesetzt ist
token	string	Bearer-Token. Benutzermakros werden unterstützt. Property behavior: - *erforderlich*, wenn `authtype` auf ”Bearer” gesetzt ist
verify_peer	integer	Gibt an, ob validiert werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - Nicht validieren; 1 - *(Standard)* Validieren.
verify_host	integer	Gibt an, ob validiert werden soll, dass der Hostname der Verbindung mit dem im Zertifikat des Hosts angegebenen Namen übereinstimmt. Mögliche Werte: 0 - Nicht validieren; 1 - *(Standard)* Validieren.
ssl_cert_file	string	Dateipfad des öffentlichen SSL-Schlüssels. Benutzermakros werden unterstützt.
ssl_key_file	string	Dateipfad des privaten SSL-Schlüssels. Benutzermakros werden unterstützt.
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Benutzermakros werden unterstützt.
description	text	Beschreibung des Konnektors.
status	integer	Gibt an, ob der Konnektor aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
tags_evaltype	integer	Tag-Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.

Tag-Filter

Mit dem Tag-Filter können nur übereinstimmende Datenpunkt-Werte oder Ereignisse exportiert werden. Wenn er nicht gesetzt ist, wird alles exportiert. Das Tag-Filter-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Tag-Name. Verhalten der Eigenschaft: - *erforderlich*
operator	integer	Bedingungs-Operator. Mögliche Werte: 0 - *(Standard)* 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)`

Mit dieser Methode können Sie neue Konnektor-Objekte erstellen.

Note:
Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Prüfen Sie Benutzerrollen für mehr Informationen.

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 das Löschen von Konnektoreinträgen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Prüfen Sie die Benutzerrollen für mehr Informationen.

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 den Abruf von Konnektorobjekten entsprechend den angegebenen Parametern.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Prüfen Sie die Benutzerrollen für mehr Informationen.

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)`

Mit dieser Methode können vorhandene Connectors aktualisiert werden.

> **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.

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.

Eigenschaft	Typ	Beschreibung
dashboardid	ID	ID des Dashboards. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Dashboards. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
userid	ID	ID des Benutzers, der Eigentümer des Dashboards ist.
private	integer	Typ der Dashboard-Freigabe. Mögliche Werte: 0 - öffentliches Dashboard; 1 - *(Standard)* privates Dashboard.
display_period	integer	Standard-Anzeigezeitraum der Seite (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.

Dashboard-Seite

Das Objekt der Dashboard-Seite hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
dashboard_pageid	ID	ID der Dashboard-Seite. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
name	string	Name der Dashboard-Seite. Standard: leerer String.
display_period	integer	Anzeigedauer der Dashboard-Seite (in Sekunden). Mögliche Werte: 0, 10, 30, 60, 120, 600, 1800, 3600. Standard: 0 (verwendet die Standard-Anzeigedauer der Seite).
widgets	array	Array von Objekten des Typs Dashboard-Widget.

Dashboard-Widget

Das Dashboard-Widget-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
widgetid	ID	ID des Dashboard-Widgets. Eigenschaftsverhalten: - *schreibgeschützt*
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.

Eigenschaft	Typ	Beschreibung
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 das Erstellen neuer Dashboards.

> **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 User roles.

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.

Parameter	Type	Beschreibung
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)`

Mit dieser Methode können Dashboards gelöscht 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

`(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)`

Mit dieser Methode können Dashboards entsprechend den angegebenen Parametern abgerufen 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 User roles.

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)`

Mit dieser Methode können vorhandene Dashboards aktualisiert 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 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Aktionsprotokoll* in den Methoden `dashboard.create` und `dashboard.update`.

> **Attention:**
> Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft 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	Benutzer-ID. Hinweis: Um mehrere Benutzer zu konfigurieren, erstellen Sie für jeden Benutzer ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.
Aktionen	12	actionids.0	Aktions-ID. Hinweis: Um mehrere Aktionen zu konfigurieren, erstellen Sie für jede Aktion ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.
Medientypen	13	mediatypeids.0	Medientyp-ID. Hinweis: Um mehrere Medientypen zu konfigurieren, erstellen Sie für jeden Medientyp ein Dashboard-Widget-Feldobjekt mit einer inkrementierten 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 inkrementierten Nummer im Eigenschaftsnamen.
Suchzeichenfolge	1	message	Beliebiger Zeichenfolgenwert.
Zeitraum	1	time_period._referenceA	`DASHBOARD._timeperiod` - den Dashboard-Zeitraumauswähler als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) 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 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 Zeitangabe 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

Parameter	type	name	value
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).
Anzuzeigende Zeilen	0	show_lines	Mögliche Werte liegen im Bereich von 1 bis 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Clock* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Clock* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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).
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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Discovery status* in den Methoden `dashboard.create` und `dashboard.update`.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Favorite graphs* in den Methoden `dashboard.create` und `dashboard.update`.

Parameter

Die folgenden Parameter werden für das Widget *Favorite graphs* 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 *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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des *Favoriten-Karten* Widget in den `dashboard.create` und `dashboard.update` Methoden.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Gauge* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei 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 fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Gauge* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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
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.

Parameter	type	name	value
Host über-schreiben	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 *Messinstrument*-Widget 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 dem Aktualisieren 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
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 im Bereich von 1-100. 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 im Bereich von 1-10. Standard: 2.
Größe	0	value_size	Mögliche Werte liegen im Bereich von 1-100. 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

Parameter	type	name	value
Größe	0	units_size	Mögliche Werte liegen im Bereich von 1-100. 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
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 er auf eine der folgenden zeitbezogenen Einheiten gesetzt ist: unixtime, uptime, s.
Farbe	1	units_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Wertbogen			
Bogengröße	0	value_arc_size	Mögliche Werte liegen im Bereich von 1-100. 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 ”Value arc” gesetzt 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 ”Value arc” gesetzt ist oder *Bogen anzeigen* auf ”Aktiviert” gesetzt ist
Größe	0	scale_size	Mögliche Werte liegen im Bereich von 1-100. Standard: 15. Parameterverhalten: - *unterstützt*, wenn ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert ”Value arc” gesetzt ist oder *Bogen anzeigen* auf ”Aktiviert” gesetzt ist

Parameter	type	name	value
Dezimalstellen	0	scale_decimal_places	Mögliche Werte liegen im Bereich von 1-10. Standard: 0. Parameterverhalten: - *unterstützt*, wenn ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert ”Value arc” gesetzt 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.
Beschriftungen anzeigen	0	th_show_labels	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Schwellenwerte* gesetzt sind und entweder ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert ”Value arc” gesetzt ist oder *Bogen anzeigen* auf ”Aktiviert” gesetzt ist
Bogen anzeigen	0	th_show_arc	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Schwellenwerte* gesetzt sind
Bogengröße	0	th_arc_size	Mögliche Werte liegen im Bereich von 1-100. 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Geomap* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Geomap* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

Parameter

Die folgenden Parameter werden für das *Geomap*-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.
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 *Reference* 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` - 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 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.
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 *Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

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 der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
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 *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*, *Zoomstufe* (*optional*, mögliche Werte reichen von 0 bis 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 reichen von 0 bis 30. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Clustering-Modus* auf "Zoomstufe" gesetzt ist
Reference	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*

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Graph* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Graph* sicherzustellen, beachten Sie bitte das in den nachstehenden Tabellen beschriebene Verhalten der Parameter.

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) steht für den jeweiligen Datensatz, während die zweite Zahl, falls vorhanden, für den konfigurierten Host oder Datenpunkt steht.

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 in einem Vorlagen-Dashboard sollten nur Datenpunkte gesetzt werden, die in der Vorlage konfiguriert sind. Hinweis: Um mehrere Datenpunkte zu konfigurieren, erstellen Sie für jeden Datenpunkt ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf "Datenpunktliste" gesetzt ist und *Datenpunkte (Widget)* nicht gesetzt ist
Datenpunkte (Widget)	1	ds.0.itemids.0._reference	Anstelle der ID des Datenpunkts: `ABCDE._itemid` - ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Reference*) als Datenquelle für Datenpunkte festlegen. Hinweis: Um mehrere Widgets zu konfigurieren, erstellen Sie für jedes Widget ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf "Datenpunktliste" gesetzt ist und *Datenpunkte* nicht gesetzt ist
Farbe	1	ds.0.color.0	Hexadezimaler Farbcode (z. B. `FF0000`). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf "Datenpunktliste" gesetzt ist

Parameter	type	name	value
Host-Muster	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 in einem Vorlagen-Dashboard konfiguriert wird.
Datenpunktmuster	1	ds.0.items.0	Name oder Muster des Datenpunkts (z. B. ”*: Number of processed *values per second”). Bei der Konfiguration des Widgets in einem Vorlagen-Dashboard sollten nur Muster für Datenpunkte gesetzt werden, die in 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 gesetzt ist
Farbpalette	0	ds.0.color_palette	Index der Farbpalette. Mögliche Werte liegen im Bereich 0-11. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf ”Datenpunktmuster” gesetzt ist und *Farbe* nicht gesetzt ist
Zeichnen	0	ds.0.type	0 - *(Standard)* Linie; 1 - Punkte; 2 - Treppe; 3 - Balken.
Gestapelt	0	ds.0.stacked	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf ”Linie”, ”Treppe” oder ”Balken” gesetzt ist
Breite	0	ds.0.width	Mögliche Werte liegen im Bereich 1-10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf ”Linie” oder ”Treppe” gesetzt ist
Punktgröße	0	ds.0.pointsize	Mögliche Werte liegen im Bereich 1-10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf ”Punkte” gesetzt ist
Transparenz	0	ds.0.transparency	Mögliche Werte liegen im Bereich 1-10. Standard: 5.

Parameter	type	name	value
Füllung	0	ds.0.fill	Mögliche Werte liegen im Bereich 1-10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf ”Linie” oder ”Treppe” gesetzt ist
Fehlende Daten	0	ds.0.missingdatafunc	0 - *(Standard)* Keine; 1 - Verbunden; 2 - Als 0 behandeln; 3 - Letzter bekannter Wert. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf ”Linie” oder ”Treppe” gesetzt ist
Host überschreiben	1	ds.0.override_hostid	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.
Y-Achse	0	ds.0.axisy	0 - *(Standard)* Links; 1 - Rechts.
Werte invertieren	0	ds.0.invert_values	0 - (Standard) Deaktiviert; 1 - Aktiviert.
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 - all.
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.

Parameter	type	name	value
Tag-Name	1	ds.0.item_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Operator	0	ds.0.item_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 Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Tag-Wert	1	ds.0.item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden

Anzeigeoptionen

Die folgenden Parameter werden für die Konfiguration von *Anzeigeoptionen* unterstützt.

Parameter	type	name	value
Verlaufsauswahl	0	source	0 - *(Standard)* Auto; 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 Perzentillinie (links)	0	show_hostnames	0 - *(Standard)* Auto; 1 - Anzeigen; 2 - Ausblenden. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
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 im Bereich von 1–100. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Links“ gesetzt ist

Parameter	type	name	value
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 im Bereich von 1–100. 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.
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

Parameter	type	name	value
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.
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 im Bereich von 1 bis 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Anzahl der Spalten	0	legend_columns	Mögliche Werte liegen im Bereich von 1 bis 4. Standard: 4. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist 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.
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) steht für den jeweiligen Datensatz, während die zweite Zahl, falls vorhanden, den konfigurierten Host oder Datenpunkt angibt.

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 in 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`). Wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird, sollten nur die Muster für Datenpunkte gesetzt werden, die in 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 im Bereich von 1 bis 10.
Zeichnung	0	or.0.type	0 - Linie; 1 - Punkte; 2 - Treppe; 3 - Balken.
Transparenz	0	or.0.transparency	Mögliche Werte liegen im Bereich von 1 bis 10.
Füllung	0	or.0.fill	Mögliche Werte liegen im Bereich von 1 bis 10.
Punktgröße	0	or.0.pointsize	Mögliche Werte liegen im Bereich von 1 bis 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 Zeitzeichenfolge (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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Graph (classic)* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Graph (classic)* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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

Parameter	type	name	value
Zeitperiode	1	time_period._referenceDASHBOARD._timeperiod	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.
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._referenceABCDE._hostid	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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Graph prototype* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Graph prototype* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

Parameter

Die folgenden Parameter werden für das Widget *Graph prototype* 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 prototype; 3 - Einfacher Graph prototype.
Graph prototype	7	graphid.0	ID des Graph prototype. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Graph prototype“ gesetzt ist
Item prototype	5	itemid.0	ID des Item prototype. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Simple graph prototype“ gesetzt ist
Zeitperiode	1	time_period._reference	`DASHBOARD._timeperiod` - setzt den Zeitperiodenwähler des Dashboards als Datenquelle; `ABCDE._timeperiod` - setzt ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Reference*) als Datenquelle. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode nur in den Parametern *From* und *To* festlegen.
From	1	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 *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
To	1	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 *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._reference	`ABCDE._hostid` - setzt ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Reference*) als Datenquelle für Hosts; `DASHBOARD._hostid` - setzt den Host-Wähler des Dashboards als Datenquelle für Hosts. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Spalten	0	columns	Mögliche Werte liegen im Bereich von 1 bis 24. Standard: 2.
Zeilen	0	rows	Mögliche Werte liegen im Bereich von 1 bis 16. Standard: 1.
Reference	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*

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Honeycomb* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Honeycomb* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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.
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 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 sind, werden die Werte nach dem Aktualisieren 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
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 liegen im Bereich 0-6. 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 liegen im Bereich 1-100. 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` von Settings object und User object: `1F2C33` für ”blue-theme” oder ”hc-light”; `EEEEEE` für ”dark-theme” oder ”hc-dark”.
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” gesetzt ist 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” gesetzt ist 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 liegen im Bereich 0-6. 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 liegen im Bereich 1-100. Standard: 30. Parameterverhalten: - *unterstützt*, wenn *Größe* (Typ) auf ”Benutzerdefiniert” gesetzt ist
Fett	0	secondary_label_bold	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.

Parameter	type	name	value
Farbe	1	secondary_label_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: basierend auf theme von Settings object und User object: 1F2C33 für "blue-theme" oder "hc-light"; EEEEEE für "dark-theme" oder "hc-dark".
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" gesetzt ist 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" gesetzt ist 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 von Settings object und User object: D9E7ED für "blue-theme"; 3D5059 für "dark-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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Host availability* in den Methoden `dashboard.create` und `dashboard.update`.

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.

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` - 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

Beschreibung

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Host card* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Host navigator* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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 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 *Reference* 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
Host-Muster	1	hosts.0	Host-Name oder -Muster. Hinweis: Um mehrere Host-Muster zu konfigurieren, erstellen Sie für jedes Host-Muster ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Host-Mustern darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die Hosts gehören, die den konfigurierten Host-Mustern entsprechen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Status	0	status	-1 - *(Standard)* Beliebig; 0 - Aktiviert; 1 - Deaktiviert. 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 Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags 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.
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 des Tags 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 Reihenfolge des Tags 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
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.
Hosts in Wartung anzeigen	0	maintenance	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Probleme anzeigen	0	show_problems	0 - Alle; 1 - *(Standard)* Nicht unterdrückt; 2 - Keine.
Gruppieren nach			
Attribut	0	group_by.0.attribute	0 - Host-Gruppe; 1 - Tag-Wert; 2 - Schweregrad. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Attributs in der Liste der Gruppierungsattribute. Parameterverhalten: - *erforderlich* bei Konfiguration von *Gruppieren nach*
Wert	1	group_by.0.tag_name	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf das im Parameter *Attribut* festgelegte Gruppierungsattribut. Parameterverhalten: - *erforderlich* bei Konfiguration von *Gruppieren nach* und wenn *Attribut* auf ”Tag-Wert” gesetzt ist
Host-Limit	0	show_lines	Mögliche Werte liegen im Bereich von 1 bis 9999. Standard: 100. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Reference	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*

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Datenpunktkarte* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Datenpunktkarte* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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. 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 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 im Bereich von 0–10. Standard: 1.
Füllung	0	sparkline.fill	Mögliche Werte liegen im Bereich von 0–10. Standard: 3.
Farbe	1	sparkline.color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: `42A5F5`.
Zeitperiode	1	sparkline.time_period._reference	`DASHBOARD._timeperiod` – den Zeitperiodenwähler des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` – ein kompatibles Widget (mit dem Parameter `reference` gleich `ABCDE`) als Datenquelle festlegen. Standard: "" (leer) Alternativ können Sie die Zeitperiode nur in den Parametern *Von* und *Bis* festlegen.
Von	1	sparkline.time_period.from	Gültige Zeitangabe in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Standard: now-1h. Parameterverhalten: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `sparkline.time_period.to` gesetzt ist

Parameter	type	name	value
Bis	1	sparkline.time_period.to	Gültige Zeitangabe in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Standard: now. Parameterverhalten: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
Verlaufsdaten	0	sparkline.history	0 – *(Standard)* Auto; 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Datenpunkt-Verlauf* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Datenpunkt-Verlauf* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Datenpunkthistorie* 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)			
Angezeigte Zeilen	0	show_lines	Mögliche Werte liegen im Bereich von 1 bis 100. Standard: 25.
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 auf einem Vorlagen-Dashboard konfiguriert wird.
Erweiterte Konfiguration (siehe unten)			
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*

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 im Bereich von 1 bis 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 Log-Datenpunkt 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Datenpunkt-Navigator* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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 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.

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.hostid` - legen Sie die Dashboard-Host-Auswahl 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 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 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.

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 inkrementierten 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 Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags 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 Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Datenpunkt-Tags*
Tag-Wert	1	item_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 *Datenpunkt-Tags*
Status	0	state	-1 - *(Standard)* Alle; 0 - Normal; 1 - Nicht unterstützt.
Probleme anzeigen	0	show_problems	0 - Alle; 1 - *(Standard)* Nicht unterdrückt; 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 Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Attribute in der Gruppierungsattributliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Gruppieren nach*
Wert	1	group_by.0.tag_name	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf das im Parameter *Attribut* festgelegte Gruppierungsattribut. Parameterverhalten: - *erforderlich* bei Konfiguration von *Gruppieren nach* und wenn *Attribut* auf ”Host-Tag-Wert” oder ”Datenpunkt-Tag-Wert” gesetzt ist

Parameter	type	name	value
Datenpunktlimit	0	show_lines	Mögliche Werte liegen im Bereich von 1-9999. Standard: 100.
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 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Datenpunktwert* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Datenpunktwert* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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) können nicht dieselbe *Horizontale Position* und *Vertikale Position* gemeinsam nutzen.

Parameter	type	name	value
Vertikale Position	0	desc_v_pos	0 - Oben; 1 - Mitte; 2 - *(Standard)* Unten. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) können nicht dieselbe *Horizontale Position* und *Vertikale Position* gemeinsam nutzen.
Größe	0	desc_size	Mögliche Werte liegen im Bereich von 1 bis 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 reichen von 1 bis 10. Standard: 2.
Größe	0	decimal_size	Mögliche Werte reichen von 1 bis 100. Standard: 35.
Position			
Horizontale Position	0	value_h_pos	0 - Links; 1 - *(Standard)* Zentriert; 2 - Rechts. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) können nicht dieselbe *Horizontale Position* und *Vertikale Position* gemeinsam nutzen.
Vertikale Position	0	value_v_pos	0 - Oben; 1 - *(Standard)* Mitte; 2 - Unten. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) können nicht dieselbe *Horizontale Position* und *Vertikale Position* gemeinsam nutzen.
Größe	0	value_size	Mögliche Werte reichen von 1 bis 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.

Parameter	type	name	value
Größe	0	units_size	Mögliche Werte reichen von 1 bis 100. Standard: 35.
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)* Mitte; 2 - Rechts. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) können nicht dieselbe *Horizontale Position* und *Vertikale Position* gemeinsam nutzen.
Vertikale Position	0	time_v_pos	0 - *(Standard)* Oben; 1 - Mitte; 2 - Unten. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) können nicht dieselbe *Horizontale Position* und *Vertikale Position* gemeinsam nutzen.
Größe	0	time_size	Mögliche Werte liegen im Bereich von 1–100. 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 im Bereich von 0–10. Standard: 1.

Parameter	type	name	value
Füllung	0	sparkline.fill	Mögliche Werte liegen im Bereich von 0–10. Standard: 3.
Farbe	1	sparkline.color	Hexadezimaler Farbcode (z. B. FF0000). Standard: 42A5F5.
Zeitraum	1	sparkline.time_period.from	DASHBOARD._timeperiod – legt den Zeitraumauswahl des Dashboards als Datenquelle fest; ABCDE._timeperiod – legt ein kompatibles Widget (mit dem Parameter reference gleich ABCDE) als Datenquelle fest. Standard: DASHBOARD._timeperiod Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen.
Von	1	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: - *erforderlich*, wenn sparkline.time_period.to gesetzt ist
Bis	1	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: - *erforderlich*, wenn sparkline.time_period.from gesetzt ist
Verlaufsdaten	0	sparkline.history	0 – *(Standard)* Auto; 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des dashboard-Widgets ermöglichen die Konfiguration des Widgets *Karte* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Karte* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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

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*

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

Beschreibung

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Map navigation tree* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Map navigation tree* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Kreisdiagramm* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die `fields`-Eigenschaften des Widgets werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Kreisdiagramm* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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) steht für den jeweiligen Datensatz, während die zweite Zahl, falls vorhanden, für den konfigurierten Host oder Datenpunkt steht.

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 in einem Vorlagen-Dashboard sollten nur Datenpunkte gesetzt werden, die in der Vorlage konfiguriert sind. Hinweis: Um mehrere Datenpunkte zu konfigurieren, erstellen Sie für jeden Datenpunkt ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Parameter behavior: - *erforderlich*, wenn *Datensatztyp* auf ”Datenpunktliste” gesetzt ist und *Datenpunkte (Widget)* nicht gesetzt ist
Datenpunkte (Widget)	1	ds.0.itemids.0._reference	Anstelle der ID des Datenpunkts: `ABCDE._itemid` - setzen Sie ein kompatibles Widget (mit dem auf ”ABCDE” gesetzten Parameter *Reference*) als Datenquelle für Datenpunkte. Hinweis: Um mehrere Widgets zu konfigurieren, erstellen Sie für jedes Widget ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Parameter behavior: - *erforderlich*, wenn *Datensatztyp* auf ”Datenpunktliste” gesetzt ist und *Datenpunkte* nicht gesetzt ist
Farbe	1	ds.0.color.0	Hexadezimaler Farbcode (z. B. `FF0000`). Parameter behavior: - *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 gesetzt werden. Parameter behavior: - *unterstützt*, wenn *Datensatztyp* auf ”Datenpunktliste” gesetzt ist

Parameter	type	name	value
Host-Muster	1	ds.0.hosts.0	Name oder Muster des Hosts (z. B. ”Zabbix*”). Parameter behavior: - *erforderlich*, wenn *Datensatztyp* auf ”Datenpunktmuster” gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Datenpunktmuster	1	ds.0.items.0	Name oder Muster des Datenpunkts (z. B. ”*: Number of processed *values per second”). Bei der Konfiguration des Widgets in einem Vorlagen-Dashboard sollten nur Muster für Datenpunkte gesetzt werden, die in der Vorlage konfiguriert sind. Parameter behavior: - *erforderlich*, wenn *Datensatztyp* auf ”Datenpunktmuster” gesetzt ist
Farbe	1	ds.0.color	Hexadezimaler Farbcode (z. B. FF0000). Parameter behavior: - *unterstützt*, wenn *Datensatztyp* auf ”Datenpunktmuster” gesetzt ist und *Farbpalette* nicht gesetzt ist
Farbpalette	0	ds.0.color_palette	Index der Farbpalette. Mögliche Werte liegen im Bereich 0-11. Standard: 0. Parameter behavior: - *unterstützt*, wenn *Datensatztyp* auf ”Datenpunktmuster” gesetzt ist und *Farbe* nicht gesetzt ist
Aggregationsfunktion	0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - *(Standard)* last.
Datensatzaggregation	0	ds.0.dataset_aggregation	0 - *(Standard)* keine; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum. Parameter behavior: - *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 Tag-Reihenfolge in der Tag-Auswertungsliste. Parameter behavior: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden

Parameter	type	name	value
Operator	0	ds.0.item_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 Tag-Reihenfolge in der Tag-Auswertungsliste. Parameter behavior: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Tag-Wert	1	ds.0.item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameter behavior: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden

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.
Zeichnen	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 *Zeichnen* auf „Ringdiagramm“ gesetzt ist
Strichbreite	0	stroke	Mögliche Werte liegen im Bereich von 0-10. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf „Ringdiagramm“ gesetzt ist
Gesamtwert anzeigen	0	total_show	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Zeichnen* auf „Ringdiagramm“ gesetzt ist
Größe	0	value_size_type	0 - *(Standard)* Auto; 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 im Bereich von 1-100. Standard: 20. Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist
Dezimalstellen	0	decimal_places	Mögliche Werte liegen im Bereich von 0-6. 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 im Bereich von 0-10. Standard: 1.
Sektoren kleiner als N % zusammenführen (Kontrollkästchen)	0	merge	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Sektoren kleiner als N % zusammenführen (Wert)	0	merge_percent	Mögliche Werte liegen im Bereich von 1-10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Sektoren kleiner als N % zusammenführen* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist

Parameter	type	name	value
Sektoren kleiner als N % zusammenführen (Farbe)	1	merge_color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Sektoren kleiner als N % zusammenführen* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist

Zeitperiode

Die folgenden Parameter werden für die Konfiguration von *Zeitperiode* unterstützt.

Parameter	type	name	value
Zeitperiode	1	time_period._referenc	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
Zeilen	0	legend_lines_mode	0 - *(Standard)* Fest; 1 - Variabel. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist

Parameter	type	name	value
Anzahl der Zeilen/ Maximale Anzahl der Zeilen	0	legend_lines	Mögliche Werte liegen im Bereich von 1-10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Anzahl der Spalten	0	legend_columns	Mögliche Werte liegen im Bereich von 1-4. Standard: 4. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Problem hosts* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die `fields`-Eigenschaften des Widgets werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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.
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
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*
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*

Parameter	type	name	value
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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Probleme* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei 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 fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Probleme* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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 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.
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.

Parameter	type	name	value
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - den Host-Selektor 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 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 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 Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der 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 der Konfiguration von *Problem-Tags*
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

Parameter	type	name	value
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 - Getrennt; 2 - Mit Problemnamen.
Unterdrückte Probleme anzeigen	0	show_suppressed	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Bestätigungsstatus	0	acknowledgement_status	0 - *(Standard)* alle; 1 - Unbestätigt; 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 in einem Vorlagen-Dashboard konfiguriert wird.
Zeitachse anzeigen	0	show_timeline	0 - Deaktiviert; 1 - *(Standard)* Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Einträge sortieren nach* auf ”Zeit (absteigend)” oder ”Zeit (aufsteigend)” gesetzt ist
Gesamte Zeile hervorheben	0	highlight_row	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Dieser Parameter wird in Designs mit hohem Kontrast nicht unterstützt.
Zeilen anzeigen	0	show_lines	Mögliche Werte liegen im Bereich von 1 bis 100. Standard: 25.
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*

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Probleme nach Schweregrad* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die `fields`-Eigenschaften des Widgets werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Probleme nach Schweregrad* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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.

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 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).
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*

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 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.
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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Streudiagramm* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft 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
Aktualisierungsinterval	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. 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

Parameter	type	name	value
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	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
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: Datenpunktmuster			

Parameter	type	name	value
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.
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.

Parameter	type	name	value
Host über-schreiben	1	ds.0.override_hostid.ABCDE._hostid	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.
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).

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).
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.
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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *SLA report* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *SLA report* sicherzustellen, beachten Sie bitte das in den nachstehenden Tabellen beschriebene Verhalten der Parameter.

Parameter

Die folgenden Parameter werden für das Widget *SLA-Bericht* 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.

Parameter	type	name	value
SLA	10	slaid.0	SLA-ID. Parameterverhalten: - *erforderlich*
Service	9	serviceid.0	Service-ID.
Anzuzeigende Zeiträume	0	show_periods	Mögliche Werte liegen im Bereich von 1 bis 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *System Information* in den Methoden `dashboard.create` und `dashboard.update`.

> **Attention:**
> Die `fields`-Eigenschaften des Widgets werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *System information* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

Parameter

Die folgenden Parameter werden für das Widget *Systeminformationen* 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 - Hochverfügbarkeitsknoten.
Details zur Prüfung auf Software-Updates anzeigen	0	show_software_update_check_details	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn `AllowSoftwareUpdateCheck` in der Zabbix-Server-Konfiguration aktiviert ist und *Anzeigen* auf „Systemstatistiken“ 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Top hosts* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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.

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 *Reference* 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 Dashboard-Host-Auswahl 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 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.
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 *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

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 der Tags 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	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 auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts in Wartung 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)* Top N; 3 - Bottom N.
Host-Limit	0	show_lines	Mögliche Werte liegen im Bereich von 1-1000. 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*

Parameter	type	name	value
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 *Data* auf „Datenpunktwert“ gesetzt ist.

Note:
Die erste Zahl im Eigenschaftsnamen *Thresholds* (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 sind, werden die Werte nach dem Aktualisieren der Widget-Konfiguration im Zabbix Frontend in aufsteigender Reihenfolge 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. Beim Konfigurieren des Widgets in einem Vorlagen-Dashboard sollten nur Datenpunkte gesetzt werden, die in 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)* Unverändert; 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
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 im Bereich von 0 bis 10. Standard: 2. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ gesetzt ist
Sparkline			

Parameter	type	name	value
Breite	0	columns.0.sparkline.width	Mögliche Werte liegen im Bereich von 0 bis 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 im Bereich von 0 bis 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
Zeitperiode	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - legt den Zeitperiodenwähler des Dashboards als Datenquelle fest; `ABCDE._timeperiod` - legt ein kompatibles Widget (mit dem Parameter `reference` gleich `ABCDE`) als Datenquelle fest. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode 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 in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`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` gesetzt 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 *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist - *erforderlich*, wenn `columns.0.sparkline.time_period.from` gesetzt ist
Verlaufsdaten	0	columns.0.sparkline.history	0 - *(Standard)* Auto; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist

Schwellenwerte

Parameter	type	name	value
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.0.color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Text“ gesetzt ist
Muster	1	columns.0.highlights.0.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 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Zeitperiode	1	columns.0.time_period.reference	`DASHBOARD._timeperiode` - legt den Zeitperiodenwähler des Dashboards als Datenquelle fest; `ABCDE._timeperiod` - legt ein kompatibles Widget (mit dem Parameter `reference` gleich `ABCDE`) als Datenquelle fest. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Aggregationsfunktion* auf „min“, „max“, „avg“, „count“, „sum“, „first“, „last“ gesetzt ist
Von	1	columns.0.time_period.from	Eine Zeitangabe 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 und *Aggregationsfunktion* auf „min“, „max“, „avg“, „count“, „sum“, „first“, „last“ gesetzt ist - *erforderlich*, wenn `columns.0.time_period.to` gesetzt ist

Parameter	type	name	value
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 *Zeitperiode* nicht gesetzt ist und *Aggregationsfunktion* auf „min“, „max“, „avg“, „count“, „sum“, „first“, „last“ gesetzt ist - *erforderlich*, wenn `columns.0.time_period.from` gesetzt ist
Verlaufsdaten	0	columns.0.history	0 - *(Standard)* Auto; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ 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*

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Top items* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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.

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 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.
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.

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 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 items* unterstützt.

Parameter	type	name	value
Host-Sortierung			
Sortieren nach	0	host_ordering_order_by	0 - *(Standard)* Host-Name; 3 - Datenpunkt-Wert.
Datenpunkt-Muster	1	host_ordering_item.0	Datenpunkt-Name oder Muster (z. B. "*: Number of processed *values per second"). Hinweis: Um mehrere Datenpunkt-Muster zu konfigurieren, erstellen Sie für jedes Datenpunkt-Muster ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Beim Konfigurieren des Widgets in einem Vorlagen-Dashboard sollten nur die Muster für Datenpunkte festgelegt werden, die in der Vorlage konfiguriert sind. Parameter behavior: - *erforderlich*, wenn *Sortieren nach* auf "Datenpunkt-Wert" gesetzt ist
Reihenfolge	0	host_ordering_order	2 - *(Standard)* Top N; 3 - Bottom N.

Parameter	type	name	value
Limit	0	host_ordering_limit	Mögliche Werte liegen im Bereich von 1 bis 100. Standard: 10.
Datenpunkt-Sortierung			
Sortieren nach	0	item_ordering_order_by	1 - Host; 2 - Datenpunkt-Name; 3 - *(Standard)* Datenpunkt-Wert.
Host-Muster	1	item_ordering_host.0	Host-Name oder Muster. Hinweis: Um mehrere Host-Muster zu konfigurieren, erstellen Sie für jedes Host-Muster ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Parameter behavior: - *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 im Bereich von 1 bis 100. Standard: 10.

Spalten

Note:
Die erste Zahl im Eigenschaftsnamen (z. B. columns.0.items.0, columns.0.item_tags_evaltype) steht für die jeweilige Spalte, während die zweite Zahl, falls vorhanden, für die konfigurierte Entität steht (z. B. Datenpunkt-Muster, Tag).

Parameter	type	name	value
Datenpunkt-Muster	1	columns.0.items.0	Datenpunkt-Name oder -Muster (z. B. ”*: Number of processed *values per second”). Hinweis: Um mehrere Datenpunkt-Muster zu konfigurieren, erstellen Sie für jedes Datenpunkt-Muster ein Dashboard-Widget-Feldobjekt mit einer erhöhten zweiten Zahl im Eigenschaftsnamen. Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard sollten nur die Muster für Datenpunkte gesetzt werden, die in der Vorlage konfiguriert sind. Parameterverhalten: - *erforderlich*
Datenpunkt-Tags			
Auswertungstyp	0	columns.0.item_tags_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 Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Datenpunkt-Tags*

Parameter	type	name	value
Operator	0	columns.0.item_tags.0.operator	**0** - Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die zweite Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Datenpunkt-Tags*
Tag-Wert	1	columns.0.item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die zweite Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Datenpunkt-Tags*
Basisfarbe	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 vorhanden; 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 im Bereich von 0 bis 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf ”Sparkline” gesetzt ist
Füllung	0	columns.0.sparkline.fill	Mögliche Werte liegen im Bereich von 0 bis 10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf ”Sparkline” gesetzt ist
Farbe	1	columns.0.sparkline.color	Hexadezimaler Farbcode (z. B. FF0000). Standard: 42A5F5. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf ”Sparkline” gesetzt ist

Parameter	type	name	value
Zeitperiode	1	columns.0.sparkline.time_period._reference	`DASHBOARD._timeperiod` - den Zeitperiodenwähler 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 die Zeitperiode 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 Zeitzeichenfolge 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` gesetzt ist
Bis	1	columns.0.sparkline.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 *Anzeige* auf "Sparkline" gesetzt ist - *erforderlich*, wenn `columns.0.sparkline.time_period.from` gesetzt ist
Verlaufsdaten	0	columns.0.sparkline.history	0 - *(Standard)* Auto; 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 im Bereich von 0 bis 10. Standard: 2.
Erweiterte Konfiguration			

Parameter	type	name	value
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.
Zeitperiode	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - den Zeitperiodenwähler 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 die Zeitperiode nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Aggregationsfunktion* auf "min", "max", "avg", "count", "sum", "first", "last" gesetzt ist
Von	1	columns.0.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 und *Aggregationsfunktion* auf "min", "max", "avg", "count", "sum", "first", "last" gesetzt ist - *erforderlich*, wenn `columns.0.time_period.to` gesetzt ist
Bis	1	columns.0.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 und *Aggregationsfunktion* auf "min", "max", "avg", "count", "sum", "first", "last" gesetzt ist - *erforderlich*, wenn `columns.0.time_period.from` gesetzt ist
Verlaufsdaten	0	columns.0.history	0 - *(Standard)* Auto; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf "Numerisch" gesetzt ist
Aggregieren	0	columns.0.aggregate_grouping	0 - *(Standard)* Jeder Datenpunkt 1 - Kombiniert. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf "Numerisch" und *Anzeige* auf "Wie vorhanden", "Balken" oder "Indikatoren" gesetzt ist.

Parameter	type	name	value
Kombinierte Aggregationsfunktion	0	columns.0.column_aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum *(Standard)*. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Aggregieren*.
Kombinierter Spaltenname	1	columns.0.combined_column_name	Beliebige Zeichenfolgenwert. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Aggregieren*.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Top triggers* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Top triggers* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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 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.
Problem	1	problem	Problem-Ereignisname (Groß-/Kleinschreibung wird nicht beachtet, vollständiger Name oder ein 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 inkrementierten Nummer im Eigenschaftsnamen.
Problem-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.

Parameter	type	name	value
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge 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 Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Problem-Tags*
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Problem-Tags*
Zeitraum	1	time_period._referenz	`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 Zeitangabe in absoluter Syntax (`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 Zeitangabe in absoluter Syntax (`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
Auslöser-Limit	0	show_lines	Mögliche Werte liegen im Bereich von 1 bis 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 „Disaster" 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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Feldobjekte des Dashboard-Widgets ermöglichen die Konfiguration des Widgets *Auslöserübersicht* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Auslöserübersicht* sicherzustellen, beachten Sie bitte das in den nachstehenden Tabellen beschriebene Verhalten der Parameter.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *URL* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets fehlerhaft zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *URL* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Verhalten der Parameter.

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

Diese Parameter und die möglichen Eigenschaftswerte für die jeweiligen Dashboard-Widget-Feldobjekte ermöglichen die Konfiguration des Widgets *Web monitoring* in den Methoden `dashboard.create` und `dashboard.update`.

Attention:
Die Eigenschaften von Widget-`fields` werden bei der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dadurch können Benutzer integrierte Widgets ändern und benutzerdefinierte Widgets erstellen, es besteht jedoch auch das Risiko, Widgets fehlerhaft 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 Verhalten der Parameter.

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.
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` - 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.

Parameter	type	name	value
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 Datenpunkt-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
itemid	ID	ID des Datenpunkts. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
delay	string	Aktualisierungsintervall des Datenpunkts. Akzeptiert Sekunden oder eine Zeiteinheit mit Suffix (z. B. 30s, 1m, 2h, 1d) und optional ein oder mehrere benutzerdefinierte Intervalle, alle durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen Intervallen und Planungsintervallen sein. Akzeptiert Benutzermakros. Falls verwendet, muss der Wert aus genau einem einzelnen Makro bestehen. Mehrere Makros oder mit Text gemischte Makros 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) 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
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 entdeckte Objekte
key_	string	Datenpunktschlüssel. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder entdeckte Objekte

Eigenschaft	Typ	Beschreibung
name	string	Name des Datenpunkts. Unterstützt Benutzermakros. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder entdeckte 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-Datenpunkt; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Abhängiger Datenpunkt; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder entdeckte 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 entdeckte Objekte
value_type	integer	Informationstyp des Datenpunkts. Mögliche Werte: 0 - numerischer Gleitkommawert; 1 - Zeichen; 2 - Log; 3 - numerisch ohne Vorzeichen; 4 - Text; 5 - binär; 6 - JSON. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder entdeckte Objekte

Eigenschaft	Typ	Beschreibung
allow_traps	integer	Erlaubt das Befüllen des Werts ähnlich wie bei einem Trapper-Datenpunkt. 0 - *(Standard)* Das Annehmen eingehender Daten nicht erlauben; 1 - Das Annehmen eingehender Daten erlauben. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für entdeckte 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 entdeckte Objekte
description	string	Beschreibung des Datenpunkts. Verhalten der Eigenschaft: - *schreibgeschützt* für entdeckte Objekte
error	string	Fehlertext, wenn es Probleme beim Aktualisieren des Datenpunktwerts gibt. Verhalten der Eigenschaft: - *schreibgeschützt*
flags	integer	Herkunft des Datenpunkts. Mögliche Werte: 0 - ein normaler Datenpunkt; 4 - ein aus einem Prototyp konvertierter Datenpunkt. Verhalten der Eigenschaft: - *schreibgeschützt*
follow_redirects	integer	Antwort-Weiterleitungen beim Abrufen von Daten folgen. Mögliche Werte: 0 - Weiterleitungen nicht folgen; 1 - *(Standard)* Weiterleitungen folgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
headers	array	Array von Headern, die beim Ausführen einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte

Eigenschaft	Typ	Beschreibung
history	string	Eine Zeiteinheit dafür, wie lange die Verlaufsdaten gespeichert werden sollen. Akzeptiert auch Benutzermakros. Standard: 31d. Verhalten der Eigenschaft: - *schreibgeschützt* für entdeckte Objekte
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf "HTTP agent" gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
inventory_link	integer	ID des Host-Inventarfelds, das durch den Datenpunkt befüllt wird. Eine Liste der unterstützten Host-Inventarfelder und ihrer IDs finden Sie auf der Seite Host-Inventar. Standard: 0. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf "numeric float", "character", "numeric unsigned" oder "text" gesetzt ist - *schreibgeschützt* für entdeckte 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 entdeckte Objekte
jmx_endpoint	string	Benutzerdefinierte Verbindungszeichenfolge für 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 entdeckte Objekte
lastclock	timestamp	Zeitpunkt, zu dem der Datenpunktwert zuletzt aktualisiert wurde. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum erweitern, indem Sie den Wert des Parameters *Max history display period* im Menüabschnitt *Administration → General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*
lastns	integer	Nanosekunden des Zeitpunkts, zu dem der Datenpunktwert zuletzt aktualisiert wurde. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum erweitern, indem Sie den Wert des Parameters *Max history display period* im Menüabschnitt *Administration → General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
lastvalue	string	Letzter Wert des Datenpunkts. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum erweitern, indem Sie den Wert des Parameters *Max history display period* im Menüabschnitt *Administration → General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*
logtimefmt	string	Format der Zeit in Log-Einträgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf ”log” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
master_itemid	ID	ID des Master-Datenpunkts. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Dependent item” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
output_format	integer	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 entdeckte Objekte
params	string	Zusätzliche Parameter abhängig vom Typ des Datenpunkts: - ausgeführtes Skript für SSH agent- und TELNET agent-Datenpunkte; - SQL-Abfrage für Database monitor-Datenpunkte; - Formel für Calculated-Datenpunkte; - das Skript für Script- und Browser-Datenpunkte. 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 entdeckte 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 entdeckte Objekte
password	string	Passwort für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”JMX agent” gesetzt ist und `username` gesetzt 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 entdeckte Objekte

Eigenschaft	Typ	Beschreibung
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 entdeckte Objekte
posts	string	HTTP(S)-Anfrage-Body-Daten. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist und `post_type` auf ”JSON data” oder ”XML data” gesetzt ist - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist und `post_type` auf ”Raw data” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
prevvalue	string	Vorheriger Wert des Datenpunkts. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum erweitern, indem Sie den Wert des Parameters *Max history display period* im Menüabschnitt *Administration → General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*
privatekey	string	Name der privaten Schlüsseldatei. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” gesetzt ist und `authtype` auf ”public key” gesetzt ist - *schreibgeschützt* für entdeckte Objekte
publickey	string	Name der öffentlichen Schlüsseldatei. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” gesetzt ist und `authtype` auf ”public key” gesetzt ist - *schreibgeschützt* für entdeckte Objekte
query_fields	array	Array von Abfragefeldern, die beim Ausführen einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte 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 oder entdeckte Objekte

Eigenschaft	Typ	Beschreibung
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 - Sowohl Body als auch 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 entdeckte Objekte
snmp_oid	string	SNMP-OID. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SNMP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
ssl_cert_file	string	Pfad zur öffentlichen SSL-Schlüsseldatei. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
ssl_key_file	string	Pfad zur privaten SSL-Schlüsseldatei. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte 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 entdeckte 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.
status_codes	string	Bereiche erforderlicher HTTP-Statuscodes, durch Kommas getrennt. Unterstützt auch Benutzermakros als Teil einer kommagetrennten Liste. Beispiel: 200,200-{$M},{$M},200-400 Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte

Eigenschaft	Typ	Beschreibung
templateid	ID	ID des übergeordneten Vorlagen-Datenpunkts. *Hinweis*: Verwenden Sie die Eigenschaft `hostid`, um die Vorlage anzugeben, zu der der Datenpunkt gehört. Verhalten der Eigenschaft: - *schreibgeschützt*
timeout	string	Timeout der Datenabfrageanforderung des Datenpunkts. Akzeptiert Sekunden oder eine Zeiteinheit mit Suffix (z. B. 30s, 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.`" und "`icmpping`" beginnt, "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) und `snmp_oid` mit "`walk[`" oder "`get[`" beginnt, "Script" (21), "Browser" (22) - *schreibgeschützt* für geerbte und entdeckte Objekte
trapper_hosts	string	Erlaubte Hosts. Verhalten der Eigenschaft: - *schreibgeschützt* für entdeckte Objekte - *unterstützt*, wenn `type` auf "Zabbix trapper" gesetzt ist oder wenn `type` auf "HTTP agent" gesetzt ist und `allow_traps` auf "Allow to accept incoming data" gesetzt ist
trends	string	Eine Zeiteinheit dafür, wie lange die Trenddaten gespeichert werden sollen. Akzeptiert auch Benutzermakros. Standard: 365d. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf "numeric float" oder "numeric unsigned" gesetzt ist - *schreibgeschützt* für entdeckte Objekte
units	string	Werteinheiten. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf "numeric float" oder "numeric unsigned" gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
username	string	Benutzername für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf "SSH agent", "TELNET agent" gesetzt ist oder wenn `type` auf "JMX agent" gesetzt ist 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) oder entdeckte Objekte
uuid	string	Universell eindeutige Kennung, die verwendet wird, um einen importierten Datenpunkt mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn der Datenpunkt zu einer Vorlage gehört

Eigenschaft	Typ	Beschreibung
valuemapid	ID	ID der zugeordneten Wertezuordnung. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf ”numeric float”, ”character” oder ”numeric unsigned” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
verify_host	integer	Ob geprüft werden soll, dass der Hostname für die Verbindung mit dem im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - *(Standard)* Nicht prüfen; 1 - Prüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte Objekte
verify_peer	integer	Ob geprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - *(Standard)* Nicht prüfen; 1 - Prüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder entdeckte 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	Parameterwert. Eigenschaftsverhalten: - *erforderlich*

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*.
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 hat die folgenden Eigenschaften.

Eigenschaft	Type	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 zu Dezimal; 7 - Oktal zu Dezimal; 8 - Hexadezimal zu Dezimal; 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 - Auf Fehler mit regulärem Ausdruck prüfen; 19 - Unveränderte Werte verwerfen; 20 - Unveränderte Werte mit Heartbeat verwerfen; 21 - JavaScript; 22 - Prometheus-Muster; 23 - Prometheus zu JSON; 24 - CSV zu JSON; 25 - Ersetzen; 26 - Nicht unterstützt prüfen; 27 - XML zu JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk zu JSON; 30 - SNMP-Get-Wert. Property behavior: - *required*
params	string	Zusätzliche Parameter, die von der Vorverarbeitungsoption verwendet werden. Mehrere Parameter werden durch das Zeilenumbruchzeichen (\n) getrennt. Wenn `type` auf ”Check unsupported” gesetzt ist, folgen die Parameter der Syntax `<scope>[\n<pattern>]`, wobei *pattern* ein regulärer Ausdruck ist und *scope* einer der folgenden Werte ist: -1 - auf beliebigen Fehler prüfen; 0 - prüfen, ob die Fehlermeldung *pattern* entspricht; 1 - prüfen, ob die Fehlermeldung *pattern* nicht entspricht. Property behavior: - *required* wenn `type` auf ”Custom multiplier” (1), ”Right trim” (2), ”Left trim” (3), ”Trim” (4), ”Regular expression” (5), ”XML XPath” (11), ”JSONPath” (12), ”In range” (13), ”Matches regular expression” (14), ”Does not match regular expression” (15), ”Check for error in JSON” (16), ”Check for error in XML” (17), ”Check for error using regular expression” (18), ”Discard unchanged with heartbeat” (20), ”JavaScript” (21), ”Prometheus pattern” (22), ”Prometheus to JSON” (23), ”CSV to JSON” (24), ”Replace” (25), Check unsupported (26), ”SNMP walk value” (28), ”SNMP walk to JSON” (29) oder ”SNMP get value” (30) gesetzt ist

Eigenschaft	Type	Beschreibung
error_handler	integer	Aktionstyp, der bei einem Fehler im Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix Server gesetzt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert setzen; 3 - Benutzerdefinierte Fehlermeldung setzen. Mögliche Werte, wenn `type` auf ”Check unsupported” gesetzt ist: 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert setzen; 3 - Benutzerdefinierte Fehlermeldung setzen. Property behavior: - *required* wenn `type` auf ”Custom multiplier” (1), ”Regular expression” (5), ”Boolean to decimal” (6), ”Octal to decimal” (7), ”Hexadecimal to decimal” (8), ”Simple change” (9), ”Change per second” (10), ”XML XPath” (11), ”JSONPath” (12), ”In range” (13), ”Matches regular expression” (14), ”Does not match regular expression” (15), ”Check for error in JSON” (16), ”Check for error in XML” (17), ”Check for error using regular expression” (18), ”Prometheus pattern” (22), ”Prometheus to JSON” (23), ”CSV to JSON” (24), ”Check unsupported” (26), ”XML to JSON” (27), ”SNMP walk value” (28), ”SNMP walk to JSON” (29) oder ”SNMP get value” (30) gesetzt ist
error_handler_params	string	Parameter für den Fehler-Handler. Property behavior: - *required* wenn `error_handler` auf ”Set custom value” oder ”Set custom error message” gesetzt ist

Die folgenden Parameter und Fehler-Handler werden für jeden Vorverarbeitungstyp unterstützt.

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehler-Handler
1	Benutzerdefinierter Multiplikator	number[1, 6]			0, 1, 2, 3
2	Rechts trimmen	Zeichenliste[2]			
3	Links trimmen	Zeichenliste[2]			
4	Trimmen	Zeichenliste[2]			
5	Regulärer Ausdruck	pattern[3]	output[2]		0, 1, 2, 3
6	Boolesch zu Dezimal				0, 1, 2, 3
7	Oktal zu Dezimal				0, 1, 2, 3

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehler-Handler
8	Hexadezimal zu Dezimal				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	Auf Fehler mit regulärem Ausdruck prüfen	pattern[3]	output[2]		0, 1, 2, 3
19	Unveränderte Werte verwerfen				
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

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehler-Handler
23	Prometheus zu JSON	pattern[6, 7]			0, 1, 2, 3
24	CSV zu 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 zu 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 zu JSON[10]	Feldname[2]	OID-Präfix[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] Gleitkommazahl (Ganzzahlen werden implizit in Float-Werte umgewandelt)
[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 mit folgender Syntax: `<metric name>{<label name>="<label value>", ...} == <value>`. Jede Komponente des Prometheus-Musters (Metrik, Label-Name, Label-Wert und Metrikwert) kann ein Benutzermakro sein.
[8] Prometheus-Ausgabe mit folgender 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 Datensätze vom Typ "Field name,OID prefix,Format records", die durch ein Zeilenumbruchzeichen getrennt sind.

datenpunkt.create

Beschreibung

```
object item.create(object/array items)
```

Mit dieser Methode können neue Datenpunkte erstellt werden.

Note:
Web-Datenpunkte können nicht über die Zabbix API erstellt werden.

Note:
Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Datenpunkte gelöscht werden.

Note:
Web-Datenpunkte können nicht über die Zabbix-API gelöscht 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 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)`

Mit dieser Methode können Datenpunkte entsprechend den angegebenen Parametern abgerufen 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)` 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.
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.

Parameter	Type	Beschreibung
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`.
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.

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 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	
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)`

Mit dieser Methode können vorhandene Datenpunkte aktualisiert werden.

> **Note:**
> Web-Datenpunkte können nicht über die Zabbix-API aktualisiert werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 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.

Datenpunktprototyp

Das Datenpunktprototyp-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
itemid	ID	ID des Datenpunktprototyps. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
delay	string	Aktualisierungsintervall des Datenpunktprototyps. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 1m, 2h, 1d) und optional ein oder mehrere benutzerdefinierte Intervalle, alle durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen Intervallen und Planungsintervallen sein. Akzeptiert Benutzermakros und LLD-Makros. Falls verwendet, muss der Wert aus genau einem einzelnen Makro bestehen. Mehrere Makros oder mit Text gemischte Makros 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` 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), ”Calculated” (15), ”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 Datenpunktprototyp gehört. Eigenschaftsverhalten: - *konstant* - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
interfaceid	ID	ID der Host-Schnittstelle des Datenpunktprototyps. Eigenschaftsverhalten: - *erforderlich*, wenn der Datenpunktprototyp zu einem Host gehört und `type` auf ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” oder ”SNMP agent” gesetzt ist - *unterstützt*, wenn der Datenpunktprototyp zu einem Host gehört und `type` auf ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” oder ”HTTP agent” gesetzt ist
key_	string	Schlüssel des Datenpunktprototyps. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
name	string	Name des Datenpunktprototyps. Unterstützt Benutzermakros. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
type	integer	Typ des Datenpunktprototyps. 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 - abhängiger Datenpunkt; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. Eigenschaftsverhalten: - *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}. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
value_type	integer	Informationstyp des Datenpunktprototyps. Mögliche Werte: 0 - numerischer Gleitkommawert; 1 - Zeichen; 2 - Protokoll; 3 - numerisch vorzeichenlos; 4 - Text; 5 - binär; 6 - JSON. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
allow_traps	integer	Erlaubt das Befüllen des Werts ähnlich wie bei einem Trapper-Datenpunkt. 0 - *(Standard)* Das Annehmen eingehender Daten nicht erlauben; 1 - Das Annehmen eingehender Daten erlauben. 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; 3 - Kerberos; 4 - Digest. 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 Datenpunktprototyps.
flags	integer	Herkunft des Datenpunktprototyps. Mögliche Werte: 2 - ein Datenpunktprototyp; 6 - ein entdeckter Datenpunktprototyp Eigenschaftsverhalten: - *schreibgeschützt*
follow_redirects	integer	Antwort-Weiterleitungen beim Abrufen von Daten folgen. Mögliche Werte: 0 - Weiterleitungen nicht folgen; 1 - *(Standard)* Weiterleitungen folgen. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
headers	array	Array von Headern, die beim Ausführen einer HTTP-Anfrage gesendet werden. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf "HTTP agent" gesetzt ist - *schreibgeschützt* für geerbte Objekte
history	string	Eine Zeiteinheit, wie lange die Verlaufsdaten gespeichert werden sollen. Akzeptiert auch Benutzermakros und LLD-Makros. Standard: 31d.
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. 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 für JMX agent. Standard: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf "JMX agent" gesetzt ist
logtimefmt	string	Format der Zeit in Protokolleinträgen. Eigenschaftsverhalten: - *unterstützt*, wenn `value_type` auf "log" gesetzt ist - *schreibgeschützt* für geerbte Objekte
master_itemid	ID	ID des Master-Datenpunkts. Eigenschaftsverhalten: - *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. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf "HTTP agent" gesetzt ist - *schreibgeschützt* für geerbte Objekte
params	string	Zusätzliche Parameter abhängig vom Typ des Datenpunktprototyps: - ausgeführtes Skript für Datenpunktprototypen vom Typ SSH agent und TELNET agent; - SQL-Abfrage für Datenpunktprototypen vom Typ Database monitor; - Formel für berechnete Datenpunktprototypen; - das Skript für Datenpunktprototypen vom Typ Script und Browser. Eigenschaftsverhalten: - *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)

Eigenschaft	Typ	Beschreibung
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` gesetzt 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)
post_type	integer	Typ des im Attribut `posts` gespeicherten Post-Datenkörpers. Mögliche Werte: 0 - *(Standard)* Rohdaten. 2 - JSON-Daten. 3 - XML-Daten. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte
posts	string	HTTP(S)-Anfrage-Body-Daten. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist und `post_type` auf ”JSON data” oder ”XML data” gesetzt ist - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist 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” gesetzt ist und `authtype` auf ”public key” gesetzt ist
publickey	string	Name der Datei mit dem öffentlichen Schlüssel. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”SSH agent” gesetzt ist und `authtype` auf ”public key” gesetzt ist
query_fields	array	Array von Abfragefeldern, die beim Ausführen 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

Eigenschaft	Typ	Beschreibung
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 - Sowohl Body 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
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
status	integer	Status des Datenpunktprototyps. Mögliche Werte: 0 - *(Standard)* aktivierter Datenpunktprototyp; 1 - deaktivierter Datenpunktprototyp; 3 - nicht unterstützter Datenpunktprototyp.
status_codes	string	Bereiche der erforderlichen HTTP-Statuscodes, durch Kommas getrennt. Unterstützt auch Benutzermakros oder LLD-Makros als Teil einer kommagetrennten 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 Vorlagen-Datenpunktprototyps. Eigenschaftsverhalten: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
timeout	string	Zeitüberschreitung für die Abfrage von Datenpunktdaten. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 1m). Akzeptiert auch Benutzermakros und LLD-Makros. 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.`" und "`icmpping`" beginnt, "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) und `snmp_oid` mit "`walk[`" oder "`get[`" beginnt, "Script" (21), "Browser" (22) - *schreibgeschützt* für geerbte Objekte
trapper_hosts	string	Erlaubte Hosts. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf "Zabbix trapper" gesetzt ist oder wenn `type` auf "HTTP agent" gesetzt ist und `allow_traps` auf "Allow to accept incoming data" gesetzt ist
trends	string	Eine Zeiteinheit, wie lange die Trenddaten gespeichert werden sollen. Akzeptiert auch Benutzermakros und LLD-Makros. Standard: 365d. Eigenschaftsverhalten: - *unterstützt*, wenn `value_type` auf "numeric float" oder "numeric unsigned" gesetzt ist
units	string	Werteinheiten. Eigenschaftsverhalten: - *unterstützt*, wenn `value_type` auf "numeric float" oder "numeric unsigned" gesetzt ist - *schreibgeschützt* für geerbte Objekte
username	string	Benutzername für die Authentifizierung. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf "SSH agent" oder "TELNET agent" gesetzt ist oder wenn `type` auf "JMX agent" gesetzt ist 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 eindeutige Kennung, die verwendet wird, um importierte Datenpunktprototypen mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Eigenschaftsverhalten: - *unterstützt*, wenn der Datenpunktprototyp zu einer Vorlage gehört
valuemapid	ID	ID der zugeordneten Wertezuordnung. Eigenschaftsverhalten: - *unterstützt*, wenn `value_type` auf "numeric float", "character" oder "numeric unsigned" gesetzt ist - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname für die Verbindung mit dem Namen im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - *(Standard)* Nicht prüfen; 1 - Prüfen. 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 prüfen; 1 - Prüfen. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf "HTTP agent" gesetzt ist - *schreibgeschützt* für geerbte Objekte
discover	integer	Erkennungsstatus des Datenpunktprototyps. Mögliche Werte: 0 - *(Standard)* neue Datenpunkte werden erkannt; 1 - neue Datenpunkte werden nicht erkannt und vorhandene Datenpunkte werden als verloren markiert.

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*

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 von Datenpunkt-Prototypen

Das Vorverarbeitungsobjekt für Datenpunkt-Prototypen hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Der Typ der Vorverarbeitungsoption. Mögliche Werte: 1 - Benutzerdefinierter Multiplikator; 2 - Rechts abschneiden; 3 - Links abschneiden; 4 - Abschneiden; 5 - Regulärer Ausdruck; 6 - Boolesch zu Dezimal; 7 - Oktal zu Dezimal; 8 - Hexadezimal zu Dezimal; 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 - Auf Fehler mit regulärem Ausdruck prüfen; 19 - Unveränderte verwerfen; 20 - Unveränderte mit Heartbeat verwerfen; 21 - JavaScript; 22 - Prometheus-Muster; 23 - Prometheus zu JSON; 24 - CSV zu JSON; 25 - Ersetzen; 26 - Nicht unterstützt prüfen; 27 - XML zu JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk zu JSON; 30 - SNMP-Get-Wert. Eigenschaftsverhalten: - *erforderlich*
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* einer der folgenden Werte ist: -1 - auf beliebigen Fehler prüfen; 0 - prüfen, ob die Fehlermeldung *pattern* entspricht; 1 - prüfen, ob die Fehlermeldung *pattern* nicht entspricht. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”Benutzerdefinierter Multiplikator” (1), ”Rechts abschneiden” (2), ”Links abschneiden” (3), ”Abschneiden” (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), ”Auf Fehler mit regulärem Ausdruck prüfen” (18), ”Unveränderte mit Heartbeat verwerfen” (20), ”JavaScript” (21), ”Prometheus-Muster” (22), ”Prometheus zu JSON” (23), ”CSV zu JSON” (24), ”Ersetzen” (25), ”Nicht unterstützt prüfen” (26), ”SNMP-Walk-Wert” (28), ”SNMP-Walk zu JSON” (29) oder ”SNMP-Get-Wert” (30) gesetzt ist

Eigenschaft	Typ	Beschreibung
error_handler	integer	Aktionstyp, der im Fall eines Fehlers im Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix-Server gesetzt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert setzen; 3 - Benutzerdefinierte Fehlermeldung setzen. Mögliche Werte, wenn `type` auf ”Nicht unterstützt prüfen” gesetzt ist: 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert setzen; 3 - Benutzerdefinierte Fehlermeldung setzen. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”Benutzerdefinierter Multiplikator” (1), ”Regulärer Ausdruck” (5), ”Boolesch zu Dezimal” (6), ”Oktal zu Dezimal” (7), ”Hexadezimal zu Dezimal” (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), ”Auf Fehler mit regulärem Ausdruck prüfen” (18), ”Prometheus-Muster” (22), ”Prometheus zu JSON” (23), ”CSV zu JSON” (24), ”Nicht unterstützt prüfen” (26), ”XML zu JSON” (27), ”SNMP-Walk-Wert” (28), ”SNMP-Walk zu JSON” (29) oder ”SNMP-Get-Wert” (30) gesetzt ist
error_handler_params	string	Parameter des Fehlerhandlers. Eigenschaftsverhalten: - *erforderlich*, wenn `error_handler` auf ”Benutzerdefinierten Wert setzen” oder ”Benutzerdefinierte Fehlermeldung setzen” 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	Zahl[1,6]			0, 1, 2, 3
2	Rechts abschneiden	Zeichenliste[2]			
3	Links abschneiden	Zeichenliste[2]			
4	Abschneiden	Zeichenliste[2]			
5	Regulärer Ausdruck	Muster[3]	Ausgabe[2]		0, 1, 2, 3
6	Boolesch zu Dezimal				0, 1, 2, 3
7	Oktal zu Dezimal				0, 1, 2, 3

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
8	Hexadezimal zu Dezimal				0, 1, 2, 3
9	Einfache Änderung				0, 1, 2, 3
10	Änderung pro Sekunde				0, 1, 2, 3
11	XML XPath	Pfad[4]			0, 1, 2, 3
12	JSONPath	Pfad[4]			0, 1, 2, 3
13	Im Bereich	Min.[1, 6]	Max.[1, 6]		0, 1, 2, 3
14	Entspricht regulärem Ausdruck	Muster[3]			0, 1, 2, 3
15	Entspricht nicht regulärem Ausdruck	Muster[3]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	Pfad[4]			0, 1, 2, 3
17	Auf Fehler in XML prüfen	Pfad[4]			0, 1, 2, 3
18	Auf Fehler mit regulärem Ausdruck prüfen	Muster[3]	Ausgabe[2]		0, 1, 2, 3
19	Unveränderte verwerfen				
20	Unverändert mit Heartbeat verwerfen	Sekunden[5, 6]			
21	JavaScript	Skript[2]			
22	Prometheus Muster	Muster[6, 7]	`value`, `label`, `function`	Ausgabe[8, 9]	0, 1, 2, 3

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
23	Prometheus zu JSON	Muster[6, 7]			0, 1, 2, 3
24	CSV zu JSON	Zeichen[2]	Zeichen[2]	0,1	0, 1, 2, 3
25	Ersetzen	Suchzeichenfolge[2]	Ersetzung[2]		
26	Nicht unterstützt prüfen	scope[1]	Muster[3, 6]		1, 2, 3
27	XML zu 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 zu JSON[10]	Feldname[2]	OID-Präfix[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] Integer- oder Gleitkommazahl
[2] Zeichenfolge
[3] Regulärer Ausdruck
[4] JSONPath oder XML XPath
[5] positiver Integer (mit Unterstützung von Zeitsuffixen, z. B. 30s, 1m, 2h, 1d)
[6] Benutzermakro, LLD-Makro
[7] Prometheus-Muster mit folgender Syntax: `<metric name>{<label name>="<label value>", ...} == <value>`. Jede Komponente des Prometheus-Musters (Metrik, Labelname, Labelwert und Metrikwert) kann ein Benutzermakro oder ein LLD-Makro sein.
[8] Prometheus-Ausgabe mit folgender Syntax: `<label name>` (kann ein Benutzermakro oder ein 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 vom Typ "Feldname,OID-Präfix,Format", die durch ein Zeilenumbruchzeichen getrennt sind.

datenpunktprototyp.create

Beschreibung

```
object itemprototype.create(object/array itemPrototypes)
```

Diese Methode ermöglicht das Erstellen neuer Datenpunkt-Prototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 das Löschen von Datenpunkt-Prototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen 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

Mehrere Datenpunkt-Prototypen löschen

Löschen Sie zwei Datenpunkt-Prototypen.
Abhängige Datenpunkt-Prototypen werden automatisch entfernt, wenn der Master-Datenpunkt oder der 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)`

Mit dieser Methode können Datenpunkt-Prototypen entsprechend den angegebenen Parametern abgerufen 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)` 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.

Parameter	Type	Beschreibung
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.
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)`

Mit dieser Methode können vorhandene Datenpunkt-Prototypen aktualisiert werden.

> **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 entzogen 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)`

Mit dieser Methode können Discovery-Prüfungen entsprechend den angegebenen Parametern abgerufen werden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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 Regel zur Netzwerkdiscovery. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
druleid	ID	ID der Discovery-Regel. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
iprange	string	Ein oder mehrere zu prüfende IP-Bereiche, durch Kommas getrennt. Weitere Informationen zu unterstützten Formaten von IP-Bereichen finden Sie im Abschnitt Konfiguration der Netzwerkdiscovery. 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 Sekunden oder eine Zeiteinheit mit 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, falls 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 das Erstellen neuer Discovery-Regeln.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 das Löschen von Discovery-Regeln.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 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)`

Diese Methode ermöglicht es, Discovery-Regeln entsprechend den angegebenen Parametern abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können bestehende Discovery-Regeln aktualisiert werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen 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 hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
default_lang	string	Standardsprache des Systems. Standard: `en_US`.

Eigenschaft	Typ	Beschreibung
default_timezone	string	Standardzeitzone des Systems. Standard: `system` - Systemstandard. Die vollständige Liste der unterstützten Zeitzonen finden Sie in der PHP-Dokumentation.
default_theme	string	Standard-Theme. Mögliche Werte: `blue-theme` - *(Standard)* Blau; `dark-theme` - Dunkel; `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 anzeigen, die keine Super-Admins sind und nicht zu Benutzergruppen mit aktiviertem Debug-Modus gehören. Mögliche Werte: 0 - *(Standard)* Technische Fehler nicht anzeigen; 1 - Technische Fehler anzeigen.
history_period	string	Maximaler Zeitraum für die Anzeige von Verlaufsdaten in den Dashboard-Widgets „Letzte Daten“, „Web“ und „Datenübersicht“. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 24h.
period_default	string	Standardzeitraum des Zeitfilters. Akzeptiert Sekunden und Zeiteinheiten mit Suffix sowie Unterstützung für Monate und Jahre (30s, 1m, 2h, 1d, 1M, 1y). Standard: 1h.
max_period	string	Maximaler Zeitraum für den Zeitfilter. Akzeptiert Sekunden und Zeiteinheiten mit Suffix sowie Unterstützung für Monate und Jahre (30s, 1m, 2h, 1d, 1M, 1y). 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.

Eigenschaft	Typ	Beschreibung
severity_color_2	string	Farbe für den Schweregrad „Warnung“ als hexadezimaler Farbcode. Standard: FFC859.
severity_color_3	string	Farbe für den Schweregrad „Durchschnitt“ 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 „Durchschnitt“. Standard: Durchschnitt.
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	Anzeigezeitraum für OK-Auslöser. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Standard: 5m.
blink_period	string	Blinkzeitraum bei Änderung des Auslöserstatus. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. 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.

Eigenschaft	Typ	Beschreibung
problem_unack_style	integer	Blinken für nicht bestätigte PROBLEM-Ereignisse. Mögliche Werte: 0 - Blinken nicht anzeigen; 1 - *(Standard)* Blinken anzeigen.
problem_ack_style	integer	Blinken für bestätigte PROBLEM-Ereignisse. Mögliche Werte: 0 - Blinken nicht anzeigen; 1 - *(Standard)* Blinken anzeigen.
ok_unack_style	integer	Blinken für nicht bestätigte RESOLVED-Ereignisse. Mögliche Werte: 0 - Blinken nicht anzeigen; 1 - *(Standard)* Blinken anzeigen.
ok_ack_style	integer	Blinken für bestätigte RESOLVED-Ereignisse. Mögliche Werte: 0 - Blinken nicht anzeigen; 1 - *(Standard)* Blinken anzeigen.
url	string	Frontend-URL.
discovery_groupid	ID	ID der Hostgruppe, in die entdeckte Hosts automatisch aufgenommen werden.
default_inventory_mode	integer	Standardmodus für Hostinventar. Mögliche Werte: -1 - *(Standard)* Deaktiviert; 0 - Manuell; 1 - Automatisch.
alert_usrgrpid	ID	ID der Benutzergruppe, an die die Alarmmeldung bei Datenbankausfall gesendet wird. Wenn auf „0“ gesetzt, 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	Zeitintervall, während dessen das Anmeldeformular gesperrt wird, wenn die Anzahl fehlgeschlagener Anmeldeversuche den im Feld login_attempts definierten Wert überschreitet. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. 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	HTTP-Header X-Frame-Options. Standard: SAMEORIGIN.

Eigenschaft	Typ	Beschreibung
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 zum Zabbix Server. Möglicher Wertebereich: 1-30s. Standard: 3s. Eigenschaftsverhalten: - *erforderlich*
socket_timeout	string	Standard-Timeout für das Netzwerk. Möglicher Wertebereich: 1-300s. Standard: 3s. Eigenschaftsverhalten: - *erforderlich*
media_type_test_timeout	string	Netzwerk-Timeout für Medientyp-Tests. Möglicher Wertebereich: 1-300s. Standard: 65s. Eigenschaftsverhalten: - *erforderlich*
item_test_timeout	string	Netzwerk-Timeout für Datenpunkt-Tests. Möglicher Wertebereich: 1-600s. Standard: 60s. Eigenschaftsverhalten: - *erforderlich*
script_timeout	string	Netzwerk-Timeout für die Skriptausführung. Möglicher Wertebereich: 1-300s. Standard: 60s. Eigenschaftsverhalten: - *erforderlich*
report_test_timeout	string	Netzwerk-Timeout für Tests geplanter Berichte. Möglicher Wertebereich: 1-300s. Standard: 60s. Eigenschaftsverhalten: - *erforderlich*
auditlog_enabled	integer	Gibt an, ob die Audit-Protokollierung aktiviert werden soll. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.

Eigenschaft	Typ	Beschreibung
auditlog_mode	integer	Gibt an, ob die Audit-Protokollierung für Low-Level-Discovery-, Netzwerk-Discovery- und Autoregistrierungsaktivitäten aktiviert werden soll, die vom Server (Systembenutzer) durchgeführt werden. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
ha_failover_delay	string	Failover-Verzögerung in Sekunden. Standard: 1m. Eigenschaftsverhalten: - *schreibgeschützt*
geomaps_tile_provider	string	Kachelanbieter für Geomap. 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, um benutzerdefinierte Werte für `geomaps_tile_url`, `geomaps_max_zoom` und `geomaps_attribution` anzugeben.
geomaps_tile_url	string	Kachel-URL für Geomap. Eigenschaftsverhalten: - *unterstützt*, wenn `geomaps_tile_provider` auf eine leere Zeichenfolge gesetzt ist
geomaps_max_zoom	integer	Maximale Zoomstufe für Geomap. Mögliche Werte: 0-30. Eigenschaftsverhalten: - *unterstützt*, wenn `geomaps_tile_provider` auf eine leere Zeichenfolge gesetzt ist
geomaps_attribution	string	Attributionstext für Geomap. Eigenschaftsverhalten: - *unterstützt*, wenn `geomaps_tile_provider` auf eine leere Zeichenfolge gesetzt ist
vault_provider	integer	Vault-Anbieter. Mögliche Werte: 0 - *(Standard)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Werte geheimer Makros auflösen durch: Mögliche Werte: 0 - *(Standard)* nur Server; 1 - Server und Proxys unabhängig voneinander.

Eigenschaft	Typ	Beschreibung
timeout_zabbix_agent	string	Nicht mehr als `timeout_*` Sekunden für die Verarbeitung aufwenden. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 1m). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-600s. Standard: 3s. Standard für `timeout_browser`: 60s. Eigenschaftsverhalten: - *erforderlich*
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		

settings.get

Beschreibung

`object settings.get(object parameters)`

Mit dieser Methode kann das Einstellungsobjekt entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung, die Methode aufzurufen, 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 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": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "geomaps_max_zoom": "0",
```

```
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```

Quelle

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

Beschreibung

`object settings.update(object settings)`

Mit dieser Methode können vorhandene allgemeine Einstellungen aktualisiert werden.

> **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. Siehe Benutzerrollen für weitere Informationen.

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.
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)`

Mit dieser Methode können entdeckte Hosts entsprechend den angegebenen Parametern abgerufen 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)` 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.

Parameter	Type	Beschreibung
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	
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.

Eigenschaft	Typ	Beschreibung
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.
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

Eigenschaft	Typ	Beschreibung
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.
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)`

Mit dieser Methode können Ereignisse aktualisiert werden. Die folgenden Aktualisierungsaktionen können durchgeführt werden:

- Ereignis schließen. Wenn das Ereignis bereits behoben ist, wird diese Aktion übersprungen.
- Ereignis quittieren. Wenn das Ereignis bereits quittiert ist, wird diese Aktion übersprungen.

- Quittierung des Ereignisses aufheben. Wenn das Ereignis nicht quittiert 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.
- Ereignisrang auf Ursache ändern. Wenn das Ereignis bereits als Ursache eingestuft ist, wird diese Aktion übersprungen.
- Ereignisrang auf Symptom ändern. Wenn das Ereignis bereits als Symptom eingestuft ist, wird diese Aktion übersprungen.

Attention:
Nur Auslöser-Ereignisse können aktualisiert werden.
Nur Problemereignisse können 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. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Parameter, die die IDs der Ereignisse und die Aktualisierungsvorgänge enthalten, die durchgeführt werden sollen.

Parameter	Type	Beschreibung
eventids	ID/array	IDs der Ereignisse, die bestätigt werden sollen. Parameter behavior: - *erforderlich*
action	integer	Aktion(en) zur Ereignisaktualisierung. Mögliche Bitmap-Werte: 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 - Ereignisrang in Ursache ändern; 256 - Ereignisrang in Symptom ändern. Dies ist ein Bitmaskenfeld; jede Summe der möglichen Bitmap-Werte ist zulässig (zum Beispiel 34 für das Bestätigen und Unterdrücken eines Ereignisses). Parameter behavior: - *erforderlich*
cause_eventid	ID	ID des Ursache-Ereignisses. Parameter behavior: - *erforderlich*, wenn `action` das Bit „Ereignisrang in Symptom ändern“ enthält
message	string	Text der Nachricht. Parameter behavior: - *erforderlich*, wenn `action` das Bit „Nachricht hinzufügen“ enthält

Parameter	Type	Beschreibung
severity	integer	Neuer Schweregrad für Ereignisse. Mögliche Werte: 0 - nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe. Parameter behavior: - *erforderlich*, wenn `action` das Bit „Schweregrad ändern“ enthält
suppress_until	integer	Unix-Zeitstempel, bis zu dem das Ereignis unterdrückt werden muss. Wenn auf „0“ gesetzt, ist die Unterdrückung unbegrenzt. Parameter behavior: - *erforderlich*, wenn `action` das Bit „Ereignis unterdrücken“ enthält

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Ereignisse unter 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
}
```

Quelle

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

Beschreibung

`integer/array event.get(object parameters)`

Mit dieser Methode können Ereignisse entsprechend den angegebenen Parametern abgerufen werden.

Attention:
Diese Methode kann Ereignisse einer gelöschten Entität zurückgeben, wenn diese Ereignisse noch nicht vom Housekeeper entfernt wurden.

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 User roles.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
eventids	ID/array	Nur Ereignisse mit den angegebenen IDs zurückgeben.
groupids	ID/array	Nur Ereignisse zurückgeben, die von Objekten erstellt wurden, die zu den angegebenen Host-Gruppen gehören.
hostids	ID/array	Nur Ereignisse zurückgeben, die von Objekten erstellt wurden, die zu den angegebenen Hosts gehören.
objectids	ID/array	Nur Ereignisse zurückgeben, die von den angegebenen Objekten erstellt wurden.
source	integer	Nur Ereignisse mit dem angegebenen Typ zurückgeben. Eine Liste der unterstützten Ereignistypen finden Sie auf der Seite zum Ereignisobjekt. Standard: 0 - Auslöser-Ereignisse.

Parameter	Type	Beschreibung
object	integer	Nur Ereignisse zurückgeben, die von Objekten des angegebenen Typs erstellt wurden. Eine Liste der unterstützten Objekttypen finden Sie auf der Seite zum Ereignisobjekt. Standard: 0 - Auslöser.
acknowledged	boolean	Wenn auf `true` gesetzt, werden nur bestätigte Ereignisse zurückgegeben.
action	integer	Nur Ereignisse zurückgeben, für die die angegebenen Aktionen zur Ereignisaktualisierung durchgeführt wurden. Für mehrere Aktionen verwenden Sie die Summe beliebiger zulässiger Bitmap-Werte als Bitmaske (zum Beispiel 34 für das Bestätigen und Unterdrücken eines Ereignisses).
action_userids	ID/array	Nur Ereignisse mit den angegebenen IDs von Benutzern zurückgeben, die die Aktionen zur Ereignisaktualisierung durchgeführt haben.
suppressed	boolean	Wenn auf `true` gesetzt, werden nur unterdrückte Ereignisse zurückgegeben.
symptom	boolean	Wenn auf `true` gesetzt, werden nur Symptom-Ereignisse zurückgegeben.
severities	integer/array	Nur Ereignisse mit den angegebenen Ereignisschweregraden zurückgeben. Gilt nur, wenn object ein Auslöser ist.
trigger_severities	integer/array	Nur Ereignisse mit den angegebenen Auslöser-Schweregraden zurückgeben. Gilt nur, wenn object ein Auslöser ist.
evaltype	integer	Methode zur Tag-Auswertung. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Nur Ereignisse mit den angegebenen Tags zurückgeben. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Ereignisse zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht.
eventid_from	string	Nur Ereignisse mit IDs zurückgeben, die größer oder gleich der angegebenen ID sind.
eventid_till	string	Nur Ereignisse mit IDs zurückgeben, die kleiner oder gleich der angegebenen ID sind.
time_from	timestamp	Nur Ereignisse zurückgeben, die zum angegebenen Zeitpunkt oder später erstellt wurden.
time_till	timestamp	Nur Ereignisse zurückgeben, die zum angegebenen Zeitpunkt oder früher erstellt wurden.
problem_time_from	timestamp	Gibt nur Ereignisse zurück, die sich ab `problem_time_from` im Problemzustand befanden, unabhängig von ihrem aktuellen Status. Gilt nur, wenn die Quelle ein Auslöser-Ereignis und das Objekt ein Auslöser ist. Dieser Parameter ist erforderlich, wenn `problem_time_till` angegeben ist.
problem_time_till	timestamp	Gibt nur Ereignisse zurück, die sich bis `problem_time_till` im Problemzustand befanden, unabhängig von ihrem aktuellen Status. Gilt nur, wenn die Quelle ein Auslöser-Ereignis und das Objekt ein Auslöser ist. Dieser Parameter ist erforderlich, wenn `problem_time_from` angegeben ist.
value	integer/array	Nur Ereignisse mit den angegebenen Werten zurückgeben.

Parameter	Type	Beschreibung
selectAcknowledges	query	Eine Eigenschaft `acknowledges` mit Ereignisaktualisierungen zurückgeben. Ereignisaktualisierungen werden in umgekehrter chronologischer Reihenfolge sortiert. Das Objekt der Ereignisaktualisierung hat die folgenden Eigenschaften: `acknowledgeid` - `(ID)` ID der Bestätigung; `userid` - `(ID)` ID des Benutzers, der das Ereignis aktualisiert hat; `clock` - `(timestamp)` Zeitpunkt, zu dem das Ereignis aktualisiert wurde; `message` - `(string)` Text der Nachricht; `action` - `(integer)` durchgeführte 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, falls für das aktuelle Ereignis gerade eine Rangänderung durchgeführt wird; `username` - `(string)` Benutzername des Benutzers, der das Ereignis aktualisiert hat; `name` - `(string)` Vorname des Benutzers, der das Ereignis aktualisiert hat; `surname` - `(string)` Nachname des Benutzers, der das Ereignis aktualisiert hat. Unterstützt `count`.
selectAlerts	query	Eine Eigenschaft `alerts` mit den durch das Ereignis erzeugten Warnungen zurückgeben. Warnungen werden in umgekehrter chronologischer Reihenfolge sortiert.
selectHosts	query	Eine Eigenschaft `hosts` mit Hosts zurückgeben, die das Objekt enthalten, das das Ereignis erstellt hat. Wird nur für durch Auslöser, Datenpunkte oder LLD-Regeln erzeugte Ereignisse unterstützt.
selectRelatedObject	query	Eine Eigenschaft `relatedObject` mit dem Objekt zurückgeben, das das Ereignis erstellt hat. Der Typ des zurückgegebenen Objekts hängt vom Ereignistyp 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 Ereignis unterdrückt hat; `suppress_until` - `(integer)` Zeitpunkt, bis zu dem das Ereignis unterdrückt wird.
selectTags	query	Eine Eigenschaft `tags` mit Ereignis-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 abgeglichen werden soll. Unterstützt keine Eigenschaften vom `text`-Datentyp.
sortfield	string/array	Das Ergebnis nach den angegebenen Eigenschaften sortieren. Mögliche Werte: `eventid`, `objectid`, `clock`. Mögliche Werte bei gemeinsamer Verwendung mit `groupBy`: `objectid`. Mögliche Werte bei gemeinsamer Verwendung mit `countOutput` und `groupBy`: `objectid`, `rowscount`.

Parameter	Type	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 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:

- eine Reihe von Objekten zurück;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde, aber der Parameter `groupBy` nicht verwendet wurde zurück;
- eine Reihe von Objekten mit Aggregationsergebnissen, wenn der Parameter `groupBy` verwendet wurde zurück.

Beispiele

Abrufen von Auslöser-Ereignissen

Rufen Sie die neuesten Ereignisse des Auslösers „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",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "suppressed": "0",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
```

```
                    "surname": "Administrator"
                }
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


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 einem 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",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "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)`

Mit dieser Methode können entdeckte Services entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. 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
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*.

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

Eigenschaft	Typ	Beschreibung
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.
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.

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode können neue Graphen erstellt werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Graphen gelöscht werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter User roles.

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)`

Mit dieser Methode können Graphen entsprechend den angegebenen Parametern abgerufen 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)` 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.
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.

Parameter	Type	Beschreibung
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	
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)`

Mit dieser Methode können vorhandene Diagramme aktualisiert werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter User roles.

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.
itemid	ID	Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen ID des Datenpunkts.
calc_fnc	integer	Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen 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.
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.

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode können Graph-Datenpunkte entsprechend den angegebenen Parametern abgerufen 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)` 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.
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.

Eigenschaft	Typ	Beschreibung
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 das Erstellen neuer Graphen-Prototypen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 das Löschen von Graph-Prototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 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)`

Mit dieser Methode können Graph-Prototypen entsprechend den angegebenen Parametern abgerufen 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)` 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)`

Mit dieser Methode können vorhandene Graphprototypen aktualisiert werden.

> **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 entzogen 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.

Hochverfügbarkeitsknoten

> **Note:**
> Knoten werden vom Zabbix Server erstellt und können nicht über die API geändert werden.

Das Objekt „Hochverfügbarkeitsknoten“ hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
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 sich der Knoten verbindet.
port	integer	Port, auf dem der Knoten läuft.
lastaccess	integer	Heartbeat-Zeit, d. h. Zeitpunkt der letzten Aktualisierung vom Knoten. UTC-Zeitstempel.

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode kann anhand der angegebenen Parameter eine Liste von Knoten eines Hochverfügbarkeitsclusters abgerufen werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Weitere Informationen finden Sie unter Benutzerrollen.

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 hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
hostid	ID	ID des Hosts. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
host	string	Technischer Name des Hosts. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
description	text	Beschreibung des Hosts.
flags	integer	Herkunft des Hosts. Mögliche Werte: 0 - ein normaler Host; 4 - ein aus einem Prototyp konvertierter Host. Verhalten der Eigenschaft: - *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 - straight; 5 - OEM; 6 - RMCP+.
ipmi_password	string	IPMI-Passwort.
ipmi_privilege	integer	IPMI-Berechtigungsstufe. Mögliche Werte: 1 - callback; 2 - *(Standard)* Benutzer; 3 - Operator; 4 - Admin; 5 - OEM.
ipmi_username	string	IPMI-Benutzername.
maintenance_from	timestamp	Startzeit der aktuell wirksamen Wartung. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
maintenance_status	integer	Status der aktuell wirksamen Wartung. Mögliche Werte: 0 - *(Standard)* keine Wartung; 1 - Wartung ist aktiv. Verhalten der Eigenschaft: - *schreibgeschützt*
maintenance_type	integer	Typ der aktuell wirksamen Wartung. Mögliche Werte: 0 - *(Standard)* Wartung mit Datenerfassung; 1 - Wartung ohne Datenerfassung. Verhalten der Eigenschaft: - *schreibgeschützt*
maintenanceid	ID	ID der Wartung, die derzeit für den Host wirksam ist. Verhalten der Eigenschaft: - *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.
proxyid	ID	ID des Proxys, der zur Überwachung des Hosts verwendet wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `monitored_by` auf ”Proxy” gesetzt ist
proxy_groupid	ID	ID der Proxy-Gruppe, die zur Überwachung des Hosts verwendet wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `monitored_by` auf ”Proxy group” 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.
tls_accept	integer	Verbindungen vom Host. Mögliche Bitmap-Werte: 1 - *(Standard)* Keine Verschlüsselung; 2 - PSK; 4 - Zertifikat. Dies ist ein Bitmaskenfeld; jede Summe der möglichen Bitmap-Werte ist zulässig (zum Beispiel 6 für PSK und Zertifikat).
tls_issuer	string	Zertifikatsaussteller.
tls_subject	string	Zertifikatssubjekt.

Eigenschaft	Typ	Beschreibung
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, welches PSK verwendet werden soll. Verhalten der Eigenschaft: - *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 aus mindestens 32 hexadezimalen Ziffern bestehen. Verhalten der Eigenschaft: - *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. Verhalten der Eigenschaft: - *schreibgeschützt*
assigned_proxyid	ID	ID des vom Zabbix Server zugewiesenen Proxys, wenn der Host von einer Proxy-Gruppe überwacht wird. Verhalten der Eigenschaft: - *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
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

ID	Eigenschaft	Type	Beschreibung	Maximale Länge
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
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 das Erstellen neuer Hosts.

> **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 entzogen 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“ und konfigurieren Sie PSK-Verschlüsselung. Beachten Sie, dass der Host für die Verwendung von PSK vorkonfiguriert 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)`

Mit dieser Methode können Hosts gelöscht werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 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)`

Diese Methode ermöglicht es, Hosts entsprechend den angegebenen Parametern abzurufen.

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)` 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.

Parameter	Type	Beschreibung
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.
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.

Parameter	Type	Beschreibung
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`.
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.

Parameter	Type	Beschreibung
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`.
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.

Parameter	Type	Beschreibung
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.
sortfield	string/array	Unterstützt nur Eigenschaften vom Datentyp `string` und `text`. 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`. Diese Abfrage ist **veraltet**, bitte verwenden Sie stattdessen `selectDiscoveryRules`.
selectHostDiscovery	query	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)`

Mit dieser Methode können mehreren angegebenen Hosts gleichzeitig mehrere zugehörige Objekte hinzugefügt werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 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, verknüpfte Objekte von mehreren Hosts zu entfernen.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können vorhandene Hosts aktualisiert werden.

> **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 entzogen 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.
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
}
```

Aktualisieren der Host-Verschlüsselung

Aktualisieren Sie den Host „10590“ so, dass für Verbindungen vom Host zum Zabbix-Server nur PSK-Verschlüsselung verwendet wird, und ändern Sie die PSK-Identität sowie den PSK-Schlüssel. Beachten Sie, dass der Host für die Verwendung von PSK vorkonfiguriert 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
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 das Erstellen neuer Host-Gruppen.

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. Siehe Benutzerrollen für weitere Informationen.

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 Host-Gruppen.

Eine Host-Gruppe kann nicht gelöscht werden, wenn:

- sie Hosts enthält, die nur zu dieser Gruppe gehören;
- sie als intern markiert ist;
- sie von einem Host-Prototyp 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 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 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)`

Mit dieser Methode können Host-Gruppen entsprechend den angegebenen Parametern abgerufen 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 User roles.

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.

Parameter	Type	Beschreibung
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.
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.

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	
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)`

Mit dieser Methode können mehrere verknüpfte Objekte gleichzeitig zu allen angegebenen Host-Gruppen hinzugefügt werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter User roles.

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, verknüpfte Objekte aus mehreren Host-Gruppen zu entfernen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. 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 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)`

Mit dieser Methode können Berechtigungen und Tag-Filter auf alle Untergruppen einer Host-Gruppe angewendet werden.

Note:
Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar.
Die Berechtigung zum Aufrufen dieser 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
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

Parameter	Type	Beschreibung
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 die Aktualisierung bestehender Host-Gruppen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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.

Eigenschaft	Type	Beschreibung
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*

Eigenschaft	Typ	Beschreibung
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*

Eigenschaft	Type	Beschreibung
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

Eigenschaft	Typ	Beschreibung
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
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 das Erstellen neuer Host-Prototypen.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Host-Prototypen gelöscht werden.

> **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 entzogen 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)`

Mit dieser Methode können Host-Prototypen entsprechend den angegebenen Parametern abgerufen 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)` 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.

Parameter	Type	Beschreibung
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	
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 die Aktualisierung vorhandener Host-Prototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen 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

Parameter	Typ	Beschreibung
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. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
interfaces	object/array	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 wurde für die Arbeit mit Host-Schnittstellen entwickelt.

Objektreferenzen:

- Host-Schnittstelle
  - Details

Verfügbare Methoden:

- hostinterface.create - neue Host-Schnittstellen erstellen
- hostinterface.delete - Host-Schnittstellen löschen
- hostinterface.get - Host-Schnittstellen abrufen
- hostinterface.massadd - Host-Schnittstellen zu Hosts hinzufügen
- hostinterface.massremove - Host-Schnittstellen von Hosts entfernen
- hostinterface.update - Host-Schnittstellen aktualisieren

Host-Interface-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `hostinterface` API.

Host-Schnittstelle

Das Objekt der Host-Schnittstelle hat die folgenden Eigenschaften.

> **Attention:**
> Beachten Sie, dass sowohl die Eigenschaften `ip` als auch `dns` für Erstellungsoperationen *erforderlich* sind. Wenn Sie DNS nicht verwenden möchten, setzen Sie es auf eine leere Zeichenfolge.

Eigenschaft	Typ	Beschreibung
interfaceid	ID	ID der Schnittstelle. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsoperationen

Eigenschaft	Typ	Beschreibung
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 Erstellungsoperationen
type	integer	Schnittstellentyp. Mögliche Werte: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
ip	string	Von der Schnittstelle verwendete IP-Adresse. Kann leer sein, wenn die Verbindung über DNS hergestellt wird. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
dns	string	Von der Schnittstelle verwendeter DNS-Name. Kann leer sein, wenn die Verbindung über IP hergestellt wird. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
port	string	Von der Schnittstelle verwendete Portnummer. Kann Benutzermakros enthalten. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
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 Erstellungsoperationen
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. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen

Eigenschaft	Typ	Beschreibung
details	object	Zusätzliches Objekt mit Details 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.create

Beschreibung

`object hostinterface.create(object/array hostInterfaces)`

Mit dieser Methode können neue Host-Schnittstellen erstellt werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Host-Schnittstellen.

Die Methode akzeptiert Host-Schnittstellen mit den Standard-Host-Schnittstelleneigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Host-Schnittstellen unter der Eigenschaft `interfaceids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Host-Schnittstellen.

Beispiele

Eine neue Schnittstelle erstellen

Erstellen Sie eine sekundäre IP-Agent-Schnittstelle auf dem Host „30052“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
```

```
        "port": "10050"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Eine Schnittstelle mit SNMP-Details erstellen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```

Siehe auch

- hostinterface.massadd
- host.massadd

Quelle

CHostInterface::create() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

Beschreibung

`object hostinterface.delete(array hostInterfaceIds)`

Diese Methode ermöglicht das Löschen von Host-Schnittstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 Host-Schnittstellen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Host-Schnittstellen unter der Eigenschaft `interfaceids` enthält.

Beispiele

Eine Host-Schnittstelle löschen

Löschen Sie die Host-Schnittstelle mit der ID 30062.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Siehe auch

- hostinterface.massremove
- host.massremove

Quelle

CHostInterface::delete() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

Beschreibung

`integer/array hostinterface.get(object parameters)`

Diese Methode ermöglicht es, Host-Schnittstellen entsprechend den angegebenen Parametern abzurufen.

> **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)` 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`.
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*.

hostinterface.massadd

Beschreibung

`object hostinterface.massadd(object parameters)`

Mit dieser Methode können Host-Schnittstellen gleichzeitig zu mehreren Hosts hinzugefügt werden.

> **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 entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die auf den angegebenen Hosts zu erstellenden Host-Schnittstellen enthalten.

Die Methode akzeptiert die folgenden Parameter.

Parameter	Typ	Beschreibung
interfaces	object/array	Host-Schnittstellen, die auf den angegebenen Hosts erstellt werden sollen. Parameterverhalten: - *erforderlich*
hosts	object/array	Hosts, die aktualisiert werden sollen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Host-Schnittstellen unter der Eigenschaft `interfaceids` enthält.

Beispiele

Schnittstellen erstellen

Erstellen Sie eine Schnittstelle auf zwei Hosts.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Siehe auch

- hostinterface.create
- host.massadd

- Host

Quelle

CHostInterface::massAdd() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

Beschreibung

`object hostinterface.massremove(object parameters)`

Mit dieser Methode können Host-Schnittstellen von den angegebenen Hosts entfernt werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. 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 IDs der zu aktualisierenden Hosts und die zu entfernenden Schnittstellen enthalten.

Parameter	Typ	Beschreibung
interfaces	object/array	Zu entfernende Host-Schnittstellen der angegebenen Hosts. Im Host-Schnittstellenobjekt dürfen nur die Eigenschaften `ip`, `dns` und `port` definiert sein. Parameterverhalten: - *erforderlich*
hostids	ID/array	IDs der zu aktualisierenden Hosts. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Host-Schnittstellen unter der Eigenschaft `interfaceids` enthält.

Beispiele

Schnittstellen entfernen

Entfernen Sie die SNMP-Schnittstelle "127.0.0.1" von zwei Hosts.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Siehe auch

- hostinterface.delete
- host.massremove

Quelle

CHostInterface::massRemove() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

Beschreibung

`object hostinterface.update(object/array hostInterfaces)`

Diese Methode ermöglicht die Aktualisierung bestehender Host-Schnittstellen.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 aktualisierende Eigenschaften der Host-Schnittstelle.

Die Eigenschaft `interfaceid` muss für jede Host-Schnittstelle definiert werden, 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-Schnittstellen unter der Eigenschaft `interfaceids` enthält.

Beispiele

Ändern eines Ports einer Host-Schnittstelle

Ändern Sie den Port einer Host-Schnittstelle.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "10055"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
```

```
    "id": 1
}
```

Ändern mehrerer Ports von Host-Schnittstellen

Ändern Sie den Port mehrerer Host-Schnittstellen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": [
        {
            "interfaceid": "30049",
            "port": "10055"
        },
        {
            "interfaceid": "30050",
            "port": "10055"
        },
        {
            "interfaceid": "30051",
            "port": "10055"
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30049",
            "30050",
            "30051"
        ]
    },
    "id": 1
}
```

Quelle

CHostInterface::update() 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.

Eigenschaft	Typ	Beschreibung
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.
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.

Eigenschaft	Typ	Beschreibung
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.
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.

Eigenschaft	Typ	Beschreibung
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.
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.

Eigenschaft	Typ	Beschreibung
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.
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.

Eigenschaft	Typ	Beschreibung
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.
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 Kartenform-Objekt 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 zur Anordnung 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)`

Mit dieser Methode können neue Karten erstellt 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 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)`

Mit dieser Methode können Karten gelöscht 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

`(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)`

Mit dieser Methode können Karten entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Diese Methode steht Benutzern aller Typen 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
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)`

Mit dieser Methode können vorhandene Karten aktualisiert 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 User roles.

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 den Export von Konfigurationsdaten als serialisierte Zeichenkette.

> **Note:**
> Diese Methode steht allen Benutzern zur Verfügung. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die zu exportierenden Objekte und das zu verwendende Format definieren.

Parameter	Type	Beschreibung
format	string	Format, in dem die Daten exportiert werden müssen. Mögliche Werte: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - unverarbeitetes PHP-Array. Parameter behavior: - *erforderlich*
prettyprint	boolean	Macht die Ausgabe durch Hinzufügen von Einrückungen besser lesbar. Mögliche Werte: `true` - Einrückung hinzufügen; `false` - *(Standard)* keine Einrückung hinzufügen.
options	object	Zu exportierende Objekte. Das Objekt `options` hat die folgenden Parameter: `host_groups` - `(array)` IDs der zu exportierenden Host-Gruppen; `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 Vorlagen-Gruppen; `templates` - `(array)` IDs der zu exportierenden Vorlagen. Benutzer des Typs *Admin* und *User* dürfen nur die Objekte exportieren, für die sie eine *read-only*- oder *read-write*-Berechtigung haben, sowie Bilder, jedoch keine Medientypen. Parameter behavior: - *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 den Import von Konfigurationsdaten aus einer serialisierten Zeichenkette.

Note:
Diese Methode steht Nutzern jeder Art zur Verfügung. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Prüfen Sie Benutzerrollen für mehr Informationen.

Parameter

`(object)` Parameter, die die zu importierenden Daten und Regeln dazu enthalten, wie die Daten verarbeitet werden sollen.

Parameter	Type	Beschreibung
format	string	Format der serialisierten Zeichenfolge. Mögliche Werte: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *erforderlich*
source	string	Serialisierte Zeichenfolge mit den Konfigurationsdaten. Parameter behavior: - *erforderlich*

Parameter	Type	Beschreibung
rules	object	Regeln dazu, wie neue und bestehende Objekte importiert werden sollen. Benutzer vom Typ *Admin* dürfen nur die Objekte importieren, für die sie *Lese-/Schreib-* Berechtigung haben, sowie Karten. Beispielsweise dürfen ein Host und seine Entitäten (Datenpunkte, Auslöser, Diagramme usw.) nur importiert werden, wenn die Benutzergruppe des Benutzers eine Berechtigung für die Hostgruppe hat, 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. Parameter behavior: *- erforderlich*

Attention:
Wenn keine Regeln angegeben werden, wird die Konfiguration nicht aktualisiert.

Das Objekt `rules` unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
discoveryRules	object	Regeln dazu, wie LLD-Regeln importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue LLD-Regeln erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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`.
graphs	object	Regeln dazu, wie Diagramme importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Diagramme erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 dazu, wie Hostgruppen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Hostgruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Hostgruppen aktualisiert; Standard: `false`.
template_groups	object	Regeln dazu, wie Vorlagengruppen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagengruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Vorlagengruppen aktualisiert; Standard: `false`.

Parameter	Type	Beschreibung
hosts	object	Regeln dazu, wie Hosts importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Hosts erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Hosts aktualisiert; Standard: `false`.
httptests	object	Regeln dazu, wie Webszenarien importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Webszenarien erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Webszenarien aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Webszenarien, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
images	object	Regeln dazu, wie Bilder importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Bilder erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Bilder aktualisiert; Standard: `false`.
items	object	Regeln dazu, wie Datenpunkte importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Datenpunkte erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 dazu, wie Karten importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Karten erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Karten aktualisiert; Standard: `false`.
mediaTypes	object	Regeln dazu, wie Medientypen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Medientypen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Medientypen aktualisiert; Standard: `false`.
templateLinkage	object	Regeln dazu, wie Vorlagenverknüpfungen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die nicht mit dem importierten Host oder der importierten Vorlage verknüpft sind, aber in den importierten Daten vorhanden sind, verknüpft; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die mit dem importierten Host oder der importierten Vorlage verknüpft sind, aber in den importierten Daten nicht vorhanden sind, getrennt, ohne dabei von den getrennten Vorlagen geerbte Entitäten (Datenpunkte, Auslöser usw.) zu entfernen; Standard: `false`.

Parameter	Type	Beschreibung
templates	object	Regeln dazu, wie Vorlagen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Vorlagen aktualisiert; Standard: `false`.
templateDashboards	object	Regeln dazu, wie Vorlagen-Dashboards importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen-Dashboards erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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`.
triggers	object	Regeln dazu, wie Auslöser importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Auslöser erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 dazu, wie Wertzuordnungen von Hosts oder Vorlagen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Wertzuordnungen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 erlaubt es, die Importdatei mit den aktuellen Systemelementen zu vergleichen und zeigt, was geändert wird, wenn diese Importdatei importiert wird.

> **Note:**
> Diese Methode ist für Benutzer jeden Typs verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Pürfen Sie die Benutzerrollen für weitere Informationen.

Parameter

`(object)` Parameter, die die möglichen zu importierenden Daten und Regeln dazu enthalten, wie die Daten verarbeitet werden sollen.

Parameter	Type	Beschreibung
format	string	Format der serialisierten Zeichenfolge. Mögliche Werte: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *required*
source	string	Serialisierte Zeichenfolge, die die Konfigurationsdaten enthält. Parameter behavior: - *required*

Parameter	Type	Beschreibung
rules	object	Regeln dazu, wie neue und bestehende Objekte verglichen werden sollen. Benutzer des Typs *Admin* und *User* können neue Objekte nur mit bestehenden Objekten vergleichen, für die sie über *read-only*- oder *read-write*- Berechtigung verfügen. Der Parameter `rules` wird in der folgenden Tabelle ausführlich beschrieben. Parameter behavior: *- required*

Attention:
Wenn keine Regeln angegeben werden, gibt es nichts zu aktualisieren und das Ergebnis bleibt leer.

Note:
Der Vergleich wird nur für Host-Gruppen 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
discoveryRules	object	Regeln dazu, wie LLD-Regeln importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue LLD-Regeln erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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`.
graphs	object	Regeln dazu, wie Diagramme importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Diagramme erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 dazu, wie Host-Gruppen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Host-Gruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Host-Gruppen aktualisiert; Standard: `false`.
template_groups	object	Regeln dazu, wie Vorlagengruppen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagengruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Vorlagengruppen aktualisiert; Standard: `false`.

Parameter	Type	Beschreibung
hosts	object	Regeln dazu, wie Hosts importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Hosts erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Hosts aktualisiert; Standard: `false`. Dieser Parameter hat keinen Einfluss auf die Ausgabe. Er ist nur aus Konsistenzgründen mit `configuration.import` zulässig.
httptests	object	Regeln dazu, wie Webszenarien importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Webszenarien erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Webszenarien aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Webszenarien, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
images	object	Regeln dazu, wie Bilder importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Bilder erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Bilder aktualisiert; Standard: `false`. Dieser Parameter hat keinen Einfluss auf die Ausgabe. Er ist nur aus Konsistenzgründen mit `configuration.import` zulässig.
items	object	Regeln dazu, wie Datenpunkte importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Datenpunkte erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 dazu, wie Karten importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Karten erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Karten aktualisiert; Standard: `false`. Dieser Parameter hat keinen Einfluss auf die Ausgabe. Er ist nur aus Konsistenzgründen mit `configuration.import` zulässig.
mediaTypes	object	Regeln dazu, wie Medientypen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Medientypen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Medientypen aktualisiert; Standard: `false`. Dieser Parameter hat keinen Einfluss auf die Ausgabe. Er ist nur aus Konsistenzgründen mit `configuration.import` zulässig.

Parameter	Type	Beschreibung
templateLinkage	object	Regeln dazu, wie Vorlagenverknüpfungen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die nicht mit dem importierten Host oder der importierten Vorlage verknüpft sind, aber in den importierten Daten vorhanden sind, verknüpft; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die mit dem importierten Host oder der importierten Vorlage verknüpft sind, aber in den importierten Daten nicht vorhanden sind, getrennt, ohne dabei von den getrennten Vorlagen geerbte Entitäten (Datenpunkte, Auslöser usw.) zu entfernen; Standard: `false`.
templates	object	Regeln dazu, wie Vorlagen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende Vorlagen aktualisiert; Standard: `false`.
templateDashboards	object	Regeln dazu, wie Vorlagen-Dashboards importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen-Dashboards erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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`.
triggers	object	Regeln dazu, wie Auslöser importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Auslöser erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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 dazu, wie Wertzuordnungen von Hosts oder Vorlagen importiert werden. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Wertzuordnungen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden bestehende 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

`(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 Korrelationsfilter-Objekt definiert eine Reihe von Bedingungen, die erfüllt sein müssen, um die konfigurierten Korrelationsoperationen auszuführen. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
conditions	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 Filterbedingungen. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
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. Verhalten der Eigenschaft: - *schreibgeschützt*
formula	string	Benutzerdefinierter Ausdruck, der zur Auswertung von Bedingungen in 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. Verhalten der Eigenschaft: - *erforderlich*, 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

Eigenschaft	Typ	Beschreibung
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 die Erstellung neuer Korrelationen.

Note:
Diese Methode steht nur dem Benutzertyp *Superadmin* zur Verfügung. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzer rollen.

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)`

Mit dieser Methode können Korrelationen 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 entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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, Korrelationen entsprechend den angegebenen Parametern abzurufen.

> **Note:**
> Diese Methode ist für Benutzer jeden Typs verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Prüfen Sie Benutzer- Rollen für mehr Informationen.

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)`

Mit dieser Methode lassen sich bestehende Korrelationen aktualisieren.

Note:
Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar.. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Prüfen Sie Bentuzer- Rollen für mehr Informationen.

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 Objekt der Low-Level-Discovery-Regel hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
itemid	ID	ID der LLD-Regel. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
delay	string	Aktualisierungsintervall der LLD-Regel. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 1m, 2h, 1d) und optional ein oder mehrere benutzerdefinierte Intervalle, alle durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen und Planungsintervallen sein. Akzeptiert Benutzermakros. Falls verwendet, muss der Wert aus genau einem einzelnen Makro bestehen. Mehrere Makros oder mit Text gemischte Makros 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), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”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 die LLD-Regel gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
flags	integer	Herkunft der Discovery-Regel. Mögliche Werte: 1 - eine Low-Level-Discovery-Regel; 5 - eine aus einem Prototyp konvertierte Low-Level-Discovery-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 vererbte Objekte
name	string	Name der LLD-Regel. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für vererbte Objekte

Eigenschaft	Typ	Beschreibung
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 ist) oder ein entdeckter Host (der eine LLD-Regel an seiner Wurzel hat). Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für vererbte 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 vererbte Objekte
allow_traps	integer	Erlaubt das Befüllen des Werts ähnlich wie beim Trapper-Datenpunkt. Mögliche Werte: 0 - *(Standard)* Das Annehmen eingehender Daten nicht erlauben; 1 - Das Annehmen eingehender Daten erlauben. 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 vererbte Objekte (wenn `type` auf ”HTTP agent” gesetzt ist)
description	string	Beschreibung der LLD-Regel.

Eigenschaft	Typ	Beschreibung
error	string	Fehlertext, wenn es Probleme beim Aktualisieren des Werts der LLD-Regel gibt. Verhalten der Eigenschaft: - *schreibgeschützt*
follow_redirects	integer	Antwort-Weiterleitungen beim Abrufen von Daten folgen. Mögliche Werte: 0 - Weiterleitungen nicht folgen; 1 - *(Standard)* Weiterleitungen folgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte
headers	array	Array von Headern, die beim Ausführen einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte 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 vererbte 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 Datenpunkte, die nicht mehr entdeckt werden, gelöscht werden. Akzeptiert Sekunden, Zeiteinheiten mit Suffix oder ein Benutzermakro. Standard: `7d`.
lifetime_type	integer	Szenario zum Löschen verlorener LLD-Ressourcen. Mögliche Werte: 0 - *(Standard)* Löschen, nachdem der Schwellenwert für die Lebensdauer erreicht wurde; 1 - Nicht löschen; 2 - Sofort löschen.
enabled_lifetime	string	Zeitraum, nach dem Datenpunkte, die nicht mehr entdeckt werden, deaktiviert werden. Akzeptiert Sekunden, Zeiteinheiten mit Suffix oder ein Benutzermakro. Standard: `0`.

Eigenschaft	Typ	Beschreibung
enabled_lifetime_type	integer	Szenario zum Deaktivieren verlorener LLD-Ressourcen. Mögliche Werte: 0 - Deaktivieren, nachdem der Schwellenwert für die Lebensdauer erreicht wurde; 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 vererbte Objekte
output_format	integer	Gibt an, ob die Antwort in JSON konvertiert werden soll. Mögliche Werte: 0 - *(Standard)* Roh speichern; 1 - In JSON konvertieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte
params	string	Zusätzliche Parameter abhängig vom Typ der LLD-Regel: - ausgeführtes Skript für SSH- und Telnet-LLD-Regeln; - SQL-Abfrage für LLD-Regeln vom Typ Datenbankmonitor; - Formel für berechnete LLD-Regeln; - das Skript für LLD-Regeln vom Typ Script und Browser. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” oder ”Browser” gesetzt ist - *schreibgeschützt* für vererbte 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 vererbte Objekte
password	string	Passwort für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”JMX agent” gesetzt ist und `username` gesetzt ist - *unterstützt*, wenn `type` auf ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” oder ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte (wenn `type` auf ”HTTP agent” gesetzt ist)
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 vererbte Objekte

Eigenschaft	Typ	Beschreibung
posts	string	HTTP(S)-Anfrage-Body-Daten. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist und `post_type` auf ”JSON data” oder ”XML data” gesetzt ist - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist und `post_type` auf ”Raw data” gesetzt ist - *schreibgeschützt* für vererbte Objekte
privatekey	string	Name der privaten Schlüsseldatei. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” gesetzt ist und `authtype` auf ”public key” gesetzt ist
publickey	string	Name der öffentlichen Schlüsseldatei. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” gesetzt ist und `authtype` auf ”public key” gesetzt ist
query_fields	array	Array von Abfragefeldern, die beim Ausführen einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte 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 vererbte 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 - Sowohl Body als auch 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 vererbte Objekte
snmp_oid	string	SNMP-OID. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SNMP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte
ssl_cert_file	string	Dateipfad des öffentlichen SSL-Schlüssels. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte

Eigenschaft	Typ	Beschreibung
ssl_key_file	string	Dateipfad des privaten SSL-Schlüssels. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte 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.
status_codes	string	Bereiche erforderlicher HTTP-Statuscodes, durch Kommas getrennt. Unterstützt auch Benutzermakros als Teil einer kommagetrennten Liste. Beispiel: 200,200-{$M},{$M},200-400 Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte
templateid	ID	ID der übergeordneten Vorlagen-LLD-Regel. Verhalten der Eigenschaft: - *schreibgeschützt*
timeout	string	Zeitüberschreitung für die Abfrageanforderung von Datenpunktdaten. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 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.`” und ”`icmpping`” beginnt, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) und `snmp_oid` mit ”`walk[`” oder ”`get[`” beginnt, ”Script” (21), ”Browser” (22) - *schreibgeschützt* für vererbte Objekte
trapper_hosts	string	Zulässige Hosts. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”Zabbix trapper” gesetzt ist oder wenn `type` auf ”HTTP agent” gesetzt ist und `allow_traps` auf ”Allow to accept incoming data” gesetzt ist

Eigenschaft	Typ	Beschreibung
username	string	Benutzername für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent”, ”TELNET agent” gesetzt ist oder wenn `type` auf ”JMX agent” gesetzt ist und `password` gesetzt ist - *unterstützt*, wenn `type` auf ”Simple check”, ”Database monitor” oder ”HTTP agent” gesetzt ist - *schreibgeschützt* für vererbte Objekte (wenn `type` auf ”HTTP agent” gesetzt ist)
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte LLD-Regeln mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn die LLD-Regel zu einer Vorlage gehört
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname für die Verbindung mit dem 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 vererbte 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 vererbte 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”
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 von LLD-Regeln

Das Objekt zur Vorverarbeitung von LLD-Regeln hat 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 zu JSON; 24 - CSV zu JSON; 25 - Ersetzen; 27 - XML zu JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk zu 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 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 zu JSON” (23), ”CSV zu JSON” (24), ”Ersetzen” (25), ”SNMP-Walk-Wert” (28), ”SNMP-Walk zu JSON” (29) oder ”SNMP-Get-Wert” (30) gesetzt ist
error_handler	integer	Aktionstyp, der bei einem Fehler im Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix-Server gesetzt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert setzen; 3 - Benutzerdefinierte Fehlermeldung setzen. 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 zu JSON” (23), ”CSV zu JSON” (24), ”XML zu JSON” (27), ”SNMP-Walk-Wert” (28), ”SNMP-Walk zu JSON” (29) oder ”SNMP-Get-Wert” (30) gesetzt ist
error_handler_params	string	Parameter für den Fehlerbehandler. Verhalten der Eigenschaft: - *erforderlich*, wenn `error_handler` auf ”Benutzerdefinierten Wert setzen” oder ”Benutzerdefinierte Fehlermeldung setzen” gesetzt ist

Die folgenden Parameter und Fehlerbehandler werden für jeden Vorverarbeitungstyp unterstützt.

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerbehandler
5	Regulärer Ausdruck	Muster[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	Muster[1]			0, 1, 2, 3
15	Entspricht nicht regulärem Ausdruck	Muster[1]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	Pfad[3]			0, 1, 2, 3
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 zu JSON	Muster[5, 6]			0, 1, 2, 3
24	CSV zu JSON	Zeichen[2]	Zeichen[2]	0,1	0, 1, 2, 3
25	Ersetzen	Suchzeichenfolge[2]	Ersetzung[2]		
27	XML zu 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

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerbehandler
29	SNMP-Walk zu JSON[7]	Feldname[2]	OID-Präfix[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, Label-Name, Label-Wert und Metrikwert) kann ein Benutzermakro sein.
[7] Unterstützt mehrere Datensätze vom Typ "Feldname,OID-Präfix,Format", die durch ein Zeilenumbruchzeichen getrennt sind.

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”

Eigenschaft	Type	Beschreibung
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.
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.

Eigenschaft	Type	Beschreibung
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*

Zeitraum für die Überschreibungsoperation einer LLD-Regel

Der Zeitraum für die Überschreibungsoperation einer LLD-Regel ist ein Aktualisierungsintervallwert, der für einen erkannten Datenpunkt festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
delay	string	Überschreibt das Aktualisierungsintervall des Datenpunktprototyps. Akzeptiert Sekunden oder eine Zeiteinheit mit Suffix (z. B. 30s, 1m, 2h, 1d) und optional ein oder mehrere benutzerdefinierte Intervalle, jeweils durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen und Planungsintervallen sein. Akzeptiert Benutzermakros oder LLD-Makros. Falls verwendet, muss der Wert aus genau einem einzelnen Makro bestehen. Mehrere Makros oder mit Text gemischte Makros 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` Property behavior: - *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*

Tag einer Überschreibungsoperation für LLD-Regeln

Das Tag-Objekt einer Überschreibungsoperation für LLD-Regeln enthält den Tag-Namen und den Wert, die für das entdeckte Objekt gesetzt werden. Es hat die folgenden Eigenschaften:

Property	Type	Description
tag	string	Neuer Tag-Name. Verhalten der Eigenschaft: - *erforderlich*
value	string	Neuer Tag-Wert.

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)`

Mit dieser Methode können neue LLD-Regeln erstellt werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Diese Methode ermöglicht das Löschen von LLD-Regeln.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

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)`

Mit dieser Methode können LLD-Regeln entsprechend den angegebenen Parametern abgerufen 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)` 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.
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.

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, mit denen abgeglichen wird. Unterstützt keine Eigenschaften des Datentyps `text`. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem die LLD-Regel 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 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)` 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)`

Mit dieser Methode können vorhandene LLD-Regeln aktualisiert werden.

Note:
Diese Methode ist nur für Benutzertypen vom Typ *Admin* und *Super admin* 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 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.
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, sodass der Inhalt des Makros *{#FSTYPE}* mit dem regulären Ausdruck *@File systems for discovery* übereinstimmt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
```

```
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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`.

LLD-Regelprototyp

Das Objekt des Low-Level-Discovery-Regelprototyps hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
itemid	ID	ID des LLD-Regelprototyps. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
ruleid	ID	ID der übergeordneten LLD-Regel/des übergeordneten LLD-Regelprototyps. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Erstellungsvorgänge
delay	string	Aktualisierungsintervall des LLD-Regelprototyps. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 1m, 2h, 1d) und optional ein oder mehrere benutzerdefinierte Intervalle, alle durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen Intervallen und Planungsintervallen sein. Akzeptiert Benutzermakros. Falls verwendet, muss der Wert aus genau einem einzelnen Makro bestehen. Mehrere Makros oder mit Text gemischte Makros 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), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”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 LLD-Regelprototyp gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
flags	integer	Herkunft des LLD-Regelprototyps. Mögliche Werte: 3 - ein Low-Level-Discovery-Regelprototyp; 7 - ein entdeckter Low-Level-Discovery-Regelprototyp. Verhalten der Eigenschaft: - *schreibgeschützt*
interfaceid	ID	ID der Host-Schnittstelle des LLD-Regelprototyps. Verhalten der Eigenschaft: - *erforderlich*, wenn der LLD-Regelprototyp zu einem Host gehört und `type` auf ”Zabbix agent”, ”IPMI agent”, ”JMX agent” oder ”SNMP agent” gesetzt ist - *unterstützt*, wenn der LLD-Regelprototyp zu einem Host gehört und `type` auf ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” oder ”HTTP agent” gesetzt ist
key_	string	Schlüssel des LLD-Regelprototyps. Mindestens ein LLD-Makro ist erforderlich. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
name	string	Name des LLD-Regelprototyps. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
type	integer	Typ des LLD-Regelprototyps. 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. Verhalten der Eigenschaft: - *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}. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
allow_traps	integer	Erlaubt das Befüllen des Werts ähnlich wie beim Trapper-Datenpunkt. Mögliche Werte: 0 - *(Standard)* Das Annehmen eingehender Daten nicht erlauben; 1 - Das Annehmen eingehender Daten erlauben. 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. 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 LLD-Regelprototyps.
error	string	Fehlertext, wenn es Probleme bei der Aktualisierung des Werts des LLD-Regelprototyps gibt. Verhalten der Eigenschaft: - *schreibgeschützt*
follow_redirects	integer	Antwort-Weiterleitungen beim Abrufen von Daten folgen. Mögliche Werte: 0 - Weiterleitungen nicht folgen; 1 - *(Standard)* Weiterleitungen folgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte
headers	array	Array von Headern, die beim Ausführen 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
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

Eigenschaft	Typ	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
lifetime	string	Zeitraum, nach dem Datenpunkte, die nicht mehr entdeckt werden, gelöscht werden. Akzeptiert Sekunden, Zeiteinheiten mit Suffix oder ein Benutzermakro. Standard: `7d`.
lifetime_type	integer	Szenario zum Löschen verlorener LLD-Ressourcen. Mögliche Werte: 0 - *(Standard)* Löschen, nachdem der Schwellenwert für die Lebensdauer erreicht wurde; 1 - Nicht löschen; 2 - Sofort löschen.
enabled_lifetime	string	Zeitraum, nach dem Datenpunkte, die nicht mehr entdeckt werden, deaktiviert werden. Akzeptiert Sekunden, Zeiteinheiten mit Suffix oder ein Benutzermakro. Standard: `0`.
enabled_lifetime_type	integer	Szenario zum Deaktivieren verlorener LLD-Ressourcen. Mögliche Werte: 0 - Deaktivieren, nachdem der Schwellenwert für die Lebensdauer erreicht wurde; 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	Ob die Antwort in JSON konvertiert werden soll. Mögliche Werte: 0 - *(Standard)* Roh 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 LLD-Regelprototyps: - ausgeführtes Skript für SSH- und Telnet-LLD-Regeln; - SQL-Abfrage für LLD-Regeln des Datenbankmonitors; - Formel für berechnete LLD-Regeln; - das Skript für Script- 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)

Eigenschaft	Typ	Beschreibung
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` gesetzt 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 Post-Daten-Bodys. 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	HTTP(S)-Anfrage-Body-Daten. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist und `post_type` auf ”JSON data” oder ”XML data” gesetzt ist - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist 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” gesetzt ist 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” gesetzt ist und `authtype` auf ”public key” gesetzt ist
query_fields	array	Array von Abfragefeldern, die beim Ausführen 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

Eigenschaft	Typ	Beschreibung
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 - Sowohl Body als auch 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	Pfad zur Datei mit dem öffentlichen SSL-Schlüssel. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_key_file	string	Pfad zur Datei mit dem privaten SSL-Schlüssel. 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 des LLD-Regelprototyps. Mögliche Werte: 0 - *(Standard)* normal; 1 - nicht unterstützt. Verhalten der Eigenschaft: - *schreibgeschützt*
status	integer	Status des LLD-Regelprototyps. Mögliche Werte: 0 - *(Standard)* aktivierter LLD-Regelprototyp; 1 - deaktivierter LLD-Regelprototyp.
status_codes	string	Bereiche erforderlicher HTTP-Statuscodes, durch Kommas getrennt. Unterstützt auch Benutzermakros als Teil einer kommagetrennten 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 übergeordneten LLD-Regelprototyps der Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
timeout	string	Zeitüberschreitung für die Abfrage von Datenpunktdaten. Akzeptiert Sekunden oder Zeiteinheiten mit Suffix (z. B. 30s, 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.`" und "`icmpping`" beginnt, "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) und `snmp_oid` mit "`walk[`" oder "`get[`" beginnt, "Script" (21), "Browser" (22) - *schreibgeschützt* für geerbte Objekte
trapper_hosts	string	Erlaubte Hosts. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf "Zabbix trapper" gesetzt ist oder wenn `type` auf "HTTP agent" gesetzt ist und `allow_traps` auf "Allow to accept incoming data" gesetzt ist
username	string	Benutzername für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf "SSH agent", "TELNET agent" gesetzt ist oder wenn `type` auf "JMX agent" gesetzt ist 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 eindeutige Kennung, die verwendet wird, um importierte LLD-Regelprototypen mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn der LLD-Regelprototyp zu einer Vorlage gehört
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname für die Verbindung mit dem 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. Verhalten der Eigenschaft: - *erforderlich*
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*

Eigenschaft	Typ	Beschreibung
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*
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*
path	string	Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird. Verhalten der Eigenschaft: - *erforderlich*

Vorverarbeitung von LLD-Regelprototypen

Das Objekt für die Vorverarbeitung von LLD-Regelprototypen hat 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 nach JSON; 24 - CSV nach JSON; 25 - Ersetzen; 27 - XML nach JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk nach 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 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 nach JSON” (23), ”CSV nach JSON” (24), ”Ersetzen” (25), ”SNMP-Walk-Wert” (28), ”SNMP-Walk nach JSON” (29) oder ”SNMP-Get-Wert” (30) gesetzt ist

Eigenschaft	Typ	Beschreibung
error_handler	integer	Aktionstyp, der bei einem Fehler im Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix Server gesetzt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert setzen; 3 - Benutzerdefinierte Fehlermeldung setzen. 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 nach JSON” (23), ”CSV nach JSON” (24), ”XML nach JSON” (27), ”SNMP-Walk-Wert” (28), ”SNMP-Walk nach JSON” (29) oder ”SNMP-Get-Wert” (30) gesetzt ist
error_handler_params	string	Parameter für den Fehler-Handler. Verhalten der Eigenschaft: - *erforderlich*, wenn `error_handler` auf ”Benutzerdefinierten Wert setzen” oder ”Benutzerdefinierte Fehlermeldung setzen” gesetzt ist

Die folgenden Parameter und Fehler-Handler werden für jeden Vorverarbeitungstyp unterstützt.

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehler-Handler
5	Regulärer Ausdruck	Muster[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	Muster[1]			0, 1, 2, 3
15	Entspricht nicht regulärem Ausdruck	Muster[1]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	Pfad[3]			0, 1, 2, 3
17	Auf Fehler in XML prüfen	Pfad[3]			0, 1, 2, 3
20	Unveränderte Werte mit Heartbeat verwerfen	Sekunden[4, 5]			

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehler-Handler
21	JavaScript	Skript[2]			
23	Prometheus-Muster nach JSON	Muster[5, 6]			0, 1, 2, 3
24	CSV nach JSON	Zeichen[2]	Zeichen[2]	0,1	0, 1, 2, 3
25	Ersetzen	Suchzeichenfolge[2]	Ersetzung[2]		
27	XML nach 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 nach JSON[7]	Feldname[2]	OID-Präfix[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 Prometheus-Musterkomponente (Metrik, Label-Name, Label-Wert und Metrikwert) kann ein Benutzermakro sein.
[7] Unterstützt mehrere Datensätze vom Typ "Feldname,OID-Präfix,Format", die durch ein Zeilenumbruchzeichen getrennt sind.

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*
step	integer	Eindeutige Reihenfolgenummer der Überschreibung. Property behavior: - *required*

Property	Type	Description
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*

Zeitraum für die Überschreibungsoperation des LLD-Regelprototyps

Der Zeitraum für die Überschreibungsoperation des LLD-Regelprototyps ist ein Aktualisierungsintervallwert, der für den erkannten Datenpunkt festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
delay	string	Überschreibt das Aktualisierungsintervall des Datenpunktprototyps. Akzeptiert Sekunden oder eine Zeiteinheit mit Suffix (z. B. 30s, 1m, 2h, 1d) und optional ein oder mehrere benutzerdefinierte Intervalle, jeweils durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen und Planungsintervallen sein. Akzeptiert Benutzermakros oder LLD-Makros. Falls verwendet, muss der Wert aus genau einem einzelnen Makro bestehen. Mehrere Makros oder mit Text gemischte Makros 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)`

Mit dieser Methode können neue LLD-Regelprototypen erstellt werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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 das Löschen von LLD-Regelprototypen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen 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)`

Mit dieser Methode können LLD-Regelprototypen entsprechend den angegebenen Parametern abgerufen werden.

> **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
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 die Aktualisierung vorhandener LLD-Regelprototypen.

Beachten Sie, dass die Aktualisierung bereits entdeckter Prototypen eingeschränkt ist.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen 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 Medientyp-Objekt hat die folgenden Eigenschaften.

Property	Type	Description
mediatypeid	ID	ID des Medientyps. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Medientyps. Property behavior: - *erforderlich* für Erstellungsvorgänge

Property	Type	Description
type	integer	Vom Medientyp verwendeter Transport. Mögliche Werte: 0 - E-Mail; 1 - Skript; 2 - SMS; 4 - webhook. Property behavior: - *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. Property behavior: - *erforderlich*, wenn `type` auf ”Script” gesetzt ist
gsm_modem	string	Name des seriellen Geräts des GSM-Modems. Property behavior: - *erforderlich*, wenn `type` auf ”SMS” gesetzt ist
passwd	string	Authentifizierungspasswort. Property behavior: - *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. Property behavior: - *erforderlich*, wenn `type` auf ”Email” gesetzt ist
smtp_helo	string	SMTP-HELO. Property behavior: - *unterstützt*, wenn `type` auf ”Email” gesetzt ist
smtp_server	string	SMTP-Server. Property behavior: - *erforderlich*, wenn `type` auf ”Email” gesetzt ist
smtp_port	integer	Port des SMTP-Servers, zu dem eine Verbindung hergestellt wird. Standard: 25. Property behavior: - *unterstützt*, wenn `type` auf ”Email” gesetzt ist
smtp_security	integer	Zu verwendende Sicherheitsstufe für die SMTP-Verbindung. Mögliche Werte: 0 - *(Standard)* Keine; 1 - STARTTLS; 2 - SSL/TLS. Property behavior: - *unterstützt*, wenn `type` auf ”Email” gesetzt ist

Property	Type	Description
smtp_verify_host	integer	SSL-Hostprüfung für SMTP. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja. Property behavior: - *unterstützt*, wenn `smtp_security` auf ”STARTTLS” oder ”SSL/TLS” gesetzt ist
smtp_verify_peer	integer	SSL-Peer-Prüfung für SMTP. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja. Property behavior: - *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 Passwort; 2 - OAuth-Token. OAuth-Authentifizierung ist für den E-Mail-Anbieter *Office365 relay* nicht zulässig. Property behavior: - *unterstützt*, wenn `type` auf ”Email” gesetzt ist
redirection_url	string	URL des Zabbix Frontend, an die nach der OAuth-Autorisierung zurückgeleitet wird. Standard: Wert der API-Einstellungseigenschaft `url` mit dem Teil `zabbix.php?action=oauth.authorize` Property behavior: - *erforderlich*, wenn `smtp_authentication` auf ”OAuth token” gesetzt ist
client_id	string	Die im OAuth-Autorisierungsserver registrierte Client-ID. Property behavior: - *erforderlich*, wenn `smtp_authentication` auf ”OAuth token” gesetzt ist
client_secret	string	Das im OAuth-Autorisierungsserver registrierte Client-Secret. Nur für Benutzer vom Typ Super Admin zugänglich. Property behavior: - *erforderlich*, wenn `smtp_authentication` auf ”OAuth token” gesetzt ist
authorization_url	string	OAuth-URL mit Parametern zum Abrufen von Zugriffs- und Aktualisierungstoken. Property behavior: - *erforderlich*, wenn `smtp_authentication` auf ”OAuth token” gesetzt ist

Property	Type	Description
token_url	string	OAuth-URL zum Austauschen des Autorisierungstokens gegen Zugriffs- und Aktualisierungstoken. Diese URL wird vom Server auch verwendet, um ein ungültiges Zugriffstoken zu aktualisieren. Property behavior: - *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 - Zugriffstoken enthält einen gültigen Wert 2 - Aktualisierungstoken enthält einen gültigen Wert 3 - Beide Token enthalten einen gültigen Wert. Property behavior: - *unterstützt*, wenn `smtp_authentication` auf "OAuth token" gesetzt ist
access_token	string	Wert des OAuth-Zugriffstokens. Property behavior: - *erforderlich*, wenn `smtp_authentication` auf "OAuth token" gesetzt ist
access_token_updated	timestamp	Zeitstempel der letzten Änderung von `access_token`, die vom Server beim Aktualisieren mit `refresh_token` oder von der API bei Token-Änderungen durchgeführt wurde. Property behavior: - *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 Token-Änderungen gesetzt. Der Zeitstempel wird durch Addition des Werts von `access_token_updated` berechnet. Property behavior: - *unterstützt*, wenn `smtp_authentication` auf "OAuth token" gesetzt ist
refresh_token	string	Wert des OAuth-Aktualisierungstokens. Property behavior: - *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. Property behavior: - *unterstützt*, wenn `smtp_authentication` auf "Normal password" gesetzt ist

Property	Type	Description
maxsessions	integer	Die maximale Anzahl von Alarmen, die parallel verarbeitet werden können. Mögliche Werte, wenn `type` auf ”SMS” gesetzt ist: 1. Mögliche Werte, wenn `type` auf ”Email”, ”Script” oder ”Webhook” gesetzt ist: 0-100. Standard: 1.
maxattempts	integer	Die maximale Anzahl von Versuchen, einen Alarm zu senden. Mögliche Werte: 1-100. Standard: 3.
attempt_interval	string	Das Intervall zwischen Wiederholungsversuchen. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Mögliche Werte: 0-1h. Standard: 10s.
message_format	integer	Nachrichtenformat. Mögliche Werte: 0 - Klartext; 1 - *(Standard)* HTML.
script	text	Property behavior: - *unterstützt*, wenn `type` auf ”Email” gesetzt ist Inhalt des webhook-Skripts (JavaScript).
timeout	string	Property behavior: - *erforderlich*, wenn `type` auf ”Webhook” gesetzt ist Zeitüberschreitung des webhook-Skripts. Akzeptiert Sekunden und Zeiteinheiten mit Suffix. Mögliche Werte: 1-60s. Standard: 30s.
process_tags	integer	Property behavior: - *unterstützt*, wenn `type` auf ”Webhook” gesetzt ist JSON-Eigenschaftswerte in der Antwort des Webhook-Skripts als Tags verarbeiten. 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.
show_event_menu	integer	Property behavior: - *unterstützt*, wenn `type` auf ”Webhook” gesetzt ist Einen Eintrag im Ereignismenü einschließen, 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 Ereignismenüeintrag und keine Eigenschaft `urls` einschließen; 1 - Ereignismenüeintrag und Eigenschaft `urls` einschließen. Property behavior: - *unterstützt*, wenn `type` auf ”Webhook” gesetzt ist

Property	Type	Description
event_menu_url	string	URL, die im Eintrag des Ereignismenüs und in der von problem.get und event.get zurückgegebenen Eigenschaft `urls` verwendet wird. Property behavior: - *unterstützt*, wenn `show_event_menu` auf "Include event menu entry and `urls` property" gesetzt ist
event_menu_name	string	Name, der für den Eintrag des Ereignismenüs und in der von problem.get und event.get zurückgegebenen Eigenschaft `urls` verwendet wird. Property behavior: - *unterstützt*, wenn `show_event_menu` auf "Include event menu entry and `urls` property" gesetzt ist
parameters	array	Parameter für Webhook oder Skript. Property behavior: - *unterstützt*, wenn `type` auf "Webhook" oder "Script" gesetzt ist
description	text	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 neue Medientypen erstellt werden.

> **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. Siehe Benutzerrollen für weitere Informationen.

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 Medientypen 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 entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

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 Medientypen entsprechend den angegebenen Parametern abgerufen 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)` 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`.
output	query	Mögliche Eigenschaften des Media type object für Benutzer des Typs *Admin* und *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`. 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`.

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 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)`

Diese Methode ermöglicht die Aktualisierung vorhandener Medientypen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* 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 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. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
type	integer	Typ der MFA-Methode. Mögliche Werte: 1 - TOTP (zeitbasierte Einmalpasswörter); 2 - Duo Universal Prompt.
name	string	Eindeutiger Name der MFA-Methode. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
hash_function	integer	Typ der Hash-Funktion zur Generierung von TOTP-Codes. Mögliche Werte: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf "TOTP" gesetzt ist
code_length	integer	Länge des Verifizierungscodes. Mögliche Werte: 6 - 6-stellig; 8 - 8-stellig. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf "TOTP" gesetzt ist
api_hostname	string	Vom Duo-Authentifizierungsdienst bereitgestellter API-Hostname. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf "Duo Universal Prompt" gesetzt ist
clientid	string	Vom Duo-Authentifizierungsdienst bereitgestellte Client-ID. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf "Duo Universal Prompt" gesetzt ist
client_secret	string	Vom Duo-Authentifizierungsdienst bereitgestelltes Client-Secret. Eigenschaftsverhalten: - *nur schreibbar* - *erforderlich*, wenn `type` auf "Duo Universal Prompt" gesetzt ist

mfa.create

Beschreibung

`object mfa.create(object/array MFA methods)`

Mit dieser Methode können neue MFA-Methoden erstellt werden.

> **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.

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)`

Mit dieser Methode können MFA-Methoden 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 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)`

Diese Methode ermöglicht es, MFA-Methoden entsprechend den angegebenen Parametern abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. 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
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`.
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	

Parameter	Type	Beschreibung
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 die Aktualisierung vorhandener MFA-Methoden.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* 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 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.

Eigenschaft	Typ	Beschreibung
moduleid	ID	ID des Moduls, wie sie in der Datenbank gespeichert ist. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
id	string	Eindeutige Modul-ID, wie sie von einem Entwickler in der Datei manifest.json des Moduls definiert wird. Mögliche Werte für integrierte Module: siehe Beschreibung der Eigenschaft „type“ in Dashboard widget. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
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 Erstellungsvorgänge
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 neue Frontend-Module installiert werden.

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.

Attention:
Moduldateien müssen manuell in die richtigen Unterverzeichnisse entpackt werden, entsprechend der Eigenschaft `relative_path` der Module.

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 Module deinstalliert werden.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

> **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)`

Die Methode ermöglicht es, Module entsprechend den angegebenen Parametern abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. 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
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 vorhandene Module aktualisiert werden.

> **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.

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.
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.

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode können Probleme entsprechend den angegebenen Parametern abgerufen werden.

Diese Methode dient zum Abrufen ungelöster Probleme. Optional können, falls angegeben, zusätzlich kürzlich gelöste Probleme abgerufen werden. Der Zeitraum, der bestimmt, wie alt „kürzlich“ ist, wird unter *Administration* → *General* definiert. Probleme, die vor diesem Zeitraum gelöst wurden, werden nicht in der Problemtabelle gespeichert. 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 noch nicht vom Housekeeper entfernt wurden.

Note:
Diese Methode steht Benutzern aller Typen 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
eventids	ID/array	Gibt nur Probleme mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur Probleme zurück, die von Objekten erstellt wurden, die zu den angegebenen Host-Gruppen gehören.
hostids	ID/array	Gibt nur Probleme zurück, die von Objekten erstellt wurden, die zu den angegebenen Hosts gehören.
objectids	ID/array	Gibt nur Probleme zurück, die von den angegebenen Objekten erstellt wurden.
source	integer	Gibt nur Probleme mit dem angegebenen Typ zurück. Eine Liste der unterstützten Ereignistypen finden Sie auf der Seite zum Problemereignisobjekt. Standard: 0 - Problem, das von einem Auslöser erstellt wurde.
object	integer	Gibt nur Probleme zurück, die von Objekten des angegebenen Typs erstellt wurden. Eine Liste der unterstützten Objekttypen finden Sie auf der Seite zum Problemereignisobjekt. Standard: 0 - Auslöser.
acknowledged	boolean	Wenn auf `true` gesetzt, werden nur bestätigte Probleme zurückgegeben.
action	integer	Gibt nur Probleme zurück, für die die angegebenen Aktionen zur Ereignisaktualisierung ausgeführt wurden. Verwenden Sie bei mehreren Aktionen die Summe beliebiger zulässiger Bitmap-Werte als Bitmaske (zum Beispiel 34 für Ereignis bestätigen und unterdrücken).
action_userids	ID/array	Gibt nur Probleme mit den angegebenen IDs von Benutzern zurück, die die Aktualisierungsaktionen für Problemereignisse ausgeführt haben.
suppressed	boolean	Wenn auf `true` gesetzt, werden nur unterdrückte Probleme zurückgegeben.
symptom	boolean	Wenn auf `true` gesetzt, werden nur Symptom-Problemereignisse zurückgegeben.
severities	integer/array	Gibt nur Probleme mit den angegebenen Ereignisschweregraden zurück. Gilt nur, wenn object ein Auslöser ist.
evaltype	integer	Auswertungsmethode für Tags. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Gibt nur Probleme mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Probleme zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht.
recent	boolean	Wenn auf `true` gesetzt, werden nur kürzlich behobene Probleme zurückgegeben (abhängig von `ok_period`).
eventid_from	string	Gibt nur Probleme mit IDs zurück, die größer oder gleich der angegebenen ID sind.
eventid_till	string	Gibt nur Probleme mit IDs zurück, die kleiner oder gleich der angegebenen ID sind.

Parameter	Type	Beschreibung
time_from	timestamp	Gibt nur Probleme zurück, die nach oder zum angegebenen Zeitpunkt erstellt wurden.
time_till	timestamp	Gibt nur Probleme zurück, die vor oder zum angegebenen Zeitpunkt erstellt wurden.
selectAcknowledges	query	Gibt eine Eigenschaft `acknowledges` mit den Problemaktualisierungen zurück. Problemaktualisierungen werden in umgekehrter chronologischer Reihenfolge sortiert. Das Objekt der Problemaktualisierung 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 für das aktuelle Ereignis eine Rangänderung durchgeführt wird. Unterstützt `count`.
selectTags	query	Gibt eine Eigenschaft `tags` mit den Problem-Tags zurück. Ausgabeformat: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Gibt eine Eigenschaft `suppression_data` mit der Liste aktiver Wartungen und manueller Unterdrückungen zurück: `maintenanceid` - `(ID)` ID der Wartung; `userid` - `(ID)` ID des Benutzers, der das Problem unterdrückt hat; `suppress_until` - `(integer)` Zeitpunkt, bis zu dem das Problem unterdrückt wird.
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`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `eventid`.
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) Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter countOutput verwendet wurde.

Beispiele

Abrufen von Problemereignissen eines Auslösers

Rufen Sie aktuelle Ereignisse des Auslösers „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",
            "opdata": "",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "Problem behoben",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0"
                }
            ],
            "suppression_data": [
                {
```

```
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "id": 1
}
```

Abrufen von Problemen, die von einem angegebenen Benutzer bestätigt wurden

Abrufen von Problemen, die von einem 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 auf localhost ist seit 5 Minuten nicht erreichbar",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
```

```
            "suppressed": "0"
        }
    ],
    "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 hat die folgenden Eigenschaften.

Property	Type	Description
proxyid	ID	ID des Proxy. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Proxy. Property behavior: - *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 für die Verbindung. Property behavior: - *erforderlich*, wenn `proxy_groupid` nicht 0 ist

Property	Type	Description
local_port	string	Lokale Proxy-Portnummer für die Verbindung. Benutzermakros werden unterstützt. Standard: 10051. Property behavior: - *unterstützt*, wenn `proxy_groupid` nicht 0 ist
operating_mode	integer	Typ des Proxy. Mögliche Werte: 0 - aktiver Proxy; 1 - passiver Proxy. Property behavior: - *erforderlich* für Erstellungsvorgänge
description	text	Beschreibung des Proxy.
lastaccess	timestamp	Zeitpunkt, zu dem sich der Proxy zuletzt mit dem Server verbunden hat. Property behavior: - *schreibgeschützt*
address	string	IP-Adresse oder DNS-Name für die Verbindung. Benutzermakros werden unterstützt. Property behavior: - *erforderlich*, wenn der Betriebsmodus des Zabbix Proxy passiv ist
port	string	Portnummer für die Verbindung. Benutzermakros werden unterstützt. Standard: 10051. Property behavior: - *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 Host. Mögliche Werte: 1 - *(Standard)* Keine Verschlüsselung; 2 - PSK; 4 - Zertifikat.
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	Zertifikatsaussteller.
tls_subject	string	Zertifikatssubjekt.

Property	Type	Description
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, welches PSK verwendet werden soll. Property behavior: - *nur schreibbar* - *erforderlich*, wenn `tls_connect` auf ”PSK” gesetzt ist oder `tls_accept` das Bit ”PSK” enthält
tls_psk	string	Vorab geteilter Schlüssel (PSK); muss aus mindestens 32 Hexadezimalstellen bestehen. Property behavior: - *nur schreibbar* - *erforderlich*, wenn `tls_connect` auf ”PSK” gesetzt ist oder `tls_accept` das Bit ”PSK” enthält
custom_timeouts	integer	Gibt an, ob globale Datenpunkt-Timeouts auf Proxy-Ebene überschrieben werden sollen. Mögliche Werte: 0 - *(Standard)* globale Einstellungen verwenden; 1 - Timeouts überschreiben.
timeout_zabbix_agent	string	Nicht mehr als `timeout_*` Sekunden für die Verarbeitung aufwenden. Akzeptiert Sekunden oder eine Zeiteinheit mit Suffix (z. B. 30s, 1m). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-600s. Standard: ””. Property behavior: - *erforderlich*, wenn `custom_timeouts` auf 1 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		
version	integer	Version des Proxy. Dreiteilige Zabbix-Versionsnummer, wobei für jeden Teil zwei Dezimalstellen verwendet werden, z. B. 60401 für Version 6.4.1, 70002 für Version 7.0.2 usw. 0 - Unbekannte Proxy-Version. Property behavior: - *schreibgeschützt*

Property	Type	Description
compatibility	integer	Version des Proxy im Vergleich zur Zabbix-Server-Version. Mögliche Werte: 0 - Nicht definiert; 1 - Aktuelle Version (Proxy und Server haben dieselbe Hauptversion); 2 - Veraltete Version (die Proxy-Version ist älter als die Server-Version, wird aber teilweise unterstützt); 3 - Nicht unterstützte Version (die Proxy-Version ist älter als die vorherige LTS-Release-Version des Servers oder die Hauptversion des Servers ist älter als die Hauptversion des Proxy). Property behavior: - *schreibgeschützt*
state	integer	Status des Proxy. Mögliche Werte: 0 - Unbekannt; 1 - Offline; 2 - Online. Property behavior: - *schreibgeschützt*

proxy.create

Beschreibung

`object proxy.create(object/array proxies)`

Mit dieser Methode können neue Proxys erstellt werden.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* 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 erstellende Proxys.

Zusätzlich zu den Standard-Proxy-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	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.

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
}
```

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 das Löschen von Proxys.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter User roles.

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)`

Die Methode ermöglicht es, Proxys entsprechend den angegebenen Parametern 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 User roles.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	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.
selectAssignedHosts	query	Gibt eine Eigenschaft `assignedHosts` mit den dem Proxy zugewiesenen Hosts zurück. Unterstützt `count`.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit den vom Proxy überwachten Hosts zurück. Unterstützt `count`.
selectProxyGroup	query	Gibt eine Eigenschaft `proxyGroup` 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 sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parameter	Type	Beschreibung
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)`

Mit dieser Methode können vorhandene Proxys aktualisiert werden.

> **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.

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	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.

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 hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
proxy_groupid	ID	ID der Proxy-Gruppe. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Proxy-Gruppe. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
description	text	Beschreibung der Proxy-Gruppe.
failover_delay	string	Zeitraum, in dem 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 an 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 - Wiederherstellung; 3 - Online; 4 - Degradiert. Verhalten der Eigenschaft: - *schreibgeschützt*

proxygroup.create

Beschreibung

`object proxygroup.create(object/array proxyGroups)`

Diese Methode ermöglicht das Erstellen neuer Proxy-Gruppen.

> **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.

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 das Löschen von Proxy-Gruppen.

> **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 User roles.

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 Proxy-Gruppen entsprechend den angegebenen Parametern abgerufen werden.

> **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
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 die Aktualisierung bestehender Proxy-Gruppen.

> **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.

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.

Eigenschaft	Typ	Beschreibung
regexpid	ID	ID des regulären Ausdrucks. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des regulären Ausdrucks. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
test_string	string	Testzeichenfolge.

Ausdrücke

Das Objekt „expressions“ hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
expression	string	Regulärer Ausdruck. Verhalten der Eigenschaft: - *erforderlich*
expression_type	integer	Typ des regulären Ausdrucks. Mögliche Werte: 0 - Zeichenfolge enthalten; 1 - Beliebige Zeichenfolge enthalten; 2 - Zeichenfolge nicht enthalten; 3 - Ergebnis ist TRUE; 4 - Ergebnis ist FALSE. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
exp_delimiter	string	Ausdruckstrennzeichen. Standardwert: ",". Mögliche Werte: "," oder "." oder "/". Verhalten der Eigenschaft: - *unterstützt*, wenn `expression_type` auf „Beliebige Zeichenfolge enthalten“ gesetzt ist
case_sensitive	integer	Groß-/Kleinschreibung beachten. Standardwert: 0. Mögliche Werte: 0 - Groß-/Kleinschreibung nicht beachten; 1 - Groß-/Kleinschreibung beachten.

regexp.create

Beschreibung

`object regexp.create(object/array regularExpressions)`

Mit dieser Methode können neue globale reguläre Ausdrücke erstellt werden.

Note:
Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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",
      "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": [
            "16"
        ]
    },
    "id": 1
}
```

Quelle

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Beschreibung

`object regexp.delete(array regexpids)`

Diese Methode ermöglicht das Löschen globaler regulärer Ausdrücke.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können globale reguläre Ausdrücke entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Diese Methode ist nur für *Super Admin* verfügbar. 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
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"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

Antwort:


```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "Dateisysteme für die Erkennung",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Netzwerkschnittstellen für die Erkennung",
      "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)`

Mit dieser Methode können vorhandene globale reguläre Ausdrücke aktualisiert werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* 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 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

Aktualisieren des globalen regulären Ausdrucks für die Dateisystemerkennung.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "Dateisysteme für die Erkennung",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        }
      ]
    },
    "id": 1
}
```

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 hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
ui	array	Array der Objekte vom Typ UI-Element.
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
services.read.mode	integer	Schreibgeschützter Zugriff auf Services. Mögliche Werte: 0 - Schreibgeschützter Zugriff auf die Services, die durch die Eigenschaften `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. Den angegebenen Services, einschließlich untergeordneter Services, wird für die Benutzerrolle schreibgeschützter Zugriff gewährt. Schreibgeschützter 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. Den über Tags abgeglichenen Services, einschließlich untergeordneter Services, wird für die Benutzerrolle schreibgeschützter Zugriff gewährt. Schreibgeschützter 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 Eigenschaften `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. Den angegebenen Services, einschließlich untergeordneter Services, wird für die Benutzerrolle Lese-/Schreibzugriff gewährt. Lese-/Schreibzugriff überschreibt 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. Den über Tags abgeglichenen Services, einschließlich untergeordneter Services, wird für die Benutzerrolle Lese-/Schreibzugriff gewährt. Lese-/Schreibzugriff überschreibt schreibgeschützten Zugriff auf die Services. Verhalten der Eigenschaft: - *unterstützt*, wenn `services.write.mode` auf "0" gesetzt ist
modules	array	Array der Modul-Objekte.
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 - Deaktiviert; 1 - *(Standard)* Aktiviert.

Eigenschaft	Typ	Beschreibung
api.mode	integer	Modus für die Behandlung der in der Eigenschaft `api` aufgeführten API-Methoden. Mögliche Werte: 0 - *(Standard)* Sperrliste; 1 - Zulassungsliste.
api	array	Array von API-Methoden.
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.

UI-Element

Das UI-Element-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
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 → Probleme*; `monitoring.hosts` - *Monitoring → Hosts*; `monitoring.latest_data` - *Monitoring → Letzte Daten*; `monitoring.maps` - *Monitoring → Karten*; `services.services` - *Services → Services*; `services.sla_report` - *Services → SLA-Bericht*; `inventory.overview` - *Inventar → Übersicht*; `inventory.hosts` - *Inventar → Hosts*; `reports.availability_report` - *Berichte → Verfügbarkeitsbericht*; `reports.top_triggers` - *Berichte → Top 100 Auslöser*. 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` - *Berichte → Geplante Berichte*; `reports.notifications` - *Berichte → Benachrichtigungen*; `configuration.template_groups` - *Datenerfassung → Vorlagengruppen*; `configuration.host_groups` - *Datenerfassung → Host-Gruppen*; `configuration.templates` - *Datenerfassung → Vorlagen*; `configuration.hosts` - *Datenerfassung → Hosts*; `configuration.maintenance` - *Datenerfassung → Wartung*; `configuration.discovery` - *Datenerfassung → Discovery*; `configuration.trigger_actions` - *Warnmeldungen → Aktionen → Auslöser-Aktionen*; `configuration.service_actions` - *Warnmeldungen → Aktionen → Service-Aktionen*; `configuration.discovery_actions` - *Warnmeldungen → Aktionen → Discovery-Aktionen*; `configuration.autoregistration_actions` - *Warnmeldungen → Aktionen → Aktionen zur automatischen Registrierung*; `configuration.internal_actions` - *Warnmeldungen → Aktionen → Interne Aktionen*. Mögliche Werte, wenn `type` des Role-Objekts auf "Super admin" gesetzt ist: `reports.system_info` - *Berichte → Systeminformationen*; `reports.audit` - *Berichte → Auditprotokoll*; `reports.action_log` - *Berichte → Aktionsprotokoll*; `configuration.event_correlation` - *Datenerfassung → Ereigniskorrelation*; `administration.media_types` - *Warnmeldungen → Medientypen*; `administration.scripts` - *Warnmeldungen → Skripte*; `administration.user_groups` - *Benutzer → Benutzergruppen*; `administration.user_roles` - *Benutzer → Benutzerrollen*; `administration.users` - *Benutzer → Benutzer*; `administration.api_tokens` - *Benutzer → API-Tokens*; `administration.authentication` - *Benutzer → Authentifizierung*; `administration.general` - *Administration → Allgemein*; `administration.audit_log` - *Administration → Auditprotokoll*; `administration.housekeeping` - *Administration → Bereinigung*; `administration.proxy_groups` - *Administration → Proxy-Gruppen*; `administration.proxies` - *Administration → Proxys*; `administration.macros` - *Administration → Makros*; `administration.queue` - *Administration → Warteschlange*. Verhalten von Eigenschaften: - *erforderlich*

Eigenschaft	Typ	Beschreibung
status	integer	Gibt an, ob der Zugriff auf das UI-Element aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* 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 hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
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-Tokens verwalten; `change_problem_ranking` - Die Problemrangfolge von Ursache zu Symptom und umgekehrt ändern; `edit_own_media` - Erstellen/Bearbeiten eigener Medien erlauben. 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. Mögliche Werte, wenn `type` des Role-Objekts auf ”User” oder ”Admin” gesetzt ist: `invoke_execute_now` - erlaubt das Ausführen von Datenpunkt-Prüfungen für Benutzer, die nur Leseberechtigungen auf dem Host haben. Mögliche Werte, wenn `type` des Role-Objekts auf ”Super admin” gesetzt ist: `edit_user_media` - Erstellen/Bearbeiten von Medien für Benutzer erlauben. Eigenschaftsverhalten: - *erforderlich*
status	integer	Ob der Zugriff zum Ausführen der Aktion aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.

role.create

Beschreibung

`object role.create(object/array roles)`

Mit dieser Methode können neue Rollen erstellt werden.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* 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 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 das Löschen von Rollen.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Rollen entsprechend den angegebenen Parametern abgerufen 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)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
roleids	ID/array	Gibt nur Rollen mit den angegebenen IDs zurück.
selectRules	query	Gibt eine Eigenschaft `rules` mit den Rollenregeln zurück.

Parameter	Typ	Beschreibung
selectUsers	query	Gibt eine Eigenschaft `users` mit den Benutzern zurück, denen die Rolle zugewiesen ist. Siehe `user.get` für Einschränkungen basierend auf dem Benutzertyp.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `roleid`, `name`.
countOutput	boolean	Diese Parameter sind 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)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Rollendaten

Rufen Sie die Daten der Rolle "Super admin role" und ihre 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",
        "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 die Aktualisierung bestehender Rollen.

> **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. Siehe Benutzerrollen für weitere Informationen.

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 hat 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 „Skript“, „IPMI“, „SSH“, „TELNET“ oder „Webhook“ gesetzt ist

Eigenschaft	Typ	Beschreibung
scope	integer	Geltungsbereich des Skripts. Mögliche Werte: 1 - action operation; 2 - manual host action; 4 - manual event action. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
execute_on	integer	Wo das Skript ausgeführt werden soll. Mögliche Werte: 0 - auf Zabbix Agent ausführen; 1 - auf Zabbix Server ausführen. Dies wird nur *unterstützt*, wenn die Ausführung globaler Skripte auf dem Zabbix Server aktiviert ist; 2 - *(Standard)* auf Zabbix Server oder Proxy ausführen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Skript“ gesetzt ist
menu_path	string	Durch Schrägstriche getrennte Ordner, die beim Klicken auf einen Host oder ein Ereignis eine menüartige Navigation im Frontend 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	Passwort für SSH-Skripte mit Passwortauthentifizierung und TELNET-Skripte. 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 für SSH-Skripte mit Authentifizierung per öffentlichem Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH“ und `authtype` auf „public key“ gesetzt ist
privatekey	string	Name der Datei mit dem privaten Schlüssel für SSH-Skripte mit Authentifizierung per öffentlichem Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH“ und `authtype` auf „public key“ gesetzt ist
port	string	Portnummer für SSH- und TELNET-Skripte. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „SSH“ oder „TELNET“ gesetzt ist

Eigenschaft	Typ	Beschreibung
groupid	ID	ID der Hostgruppe, auf der das Skript ausgeführt werden kann. Wenn auf „0“ gesetzt, ist das Skript in allen Hostgruppen verfügbar. Standard: 0.
usrgrpid	ID	ID der Benutzergruppe, die das Skript ausführen darf. Wenn auf „0“ gesetzt, ist das Skript für alle Benutzergruppen verfügbar. Standard: 0.
host_access	integer	Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist Zum Ausführen des Skripts erforderliche Host-Berechtigungen. Mögliche Werte: 2 - *(Standard)* Lesen; 3 - Schreiben.
confirmation	string	Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist Text des Bestätigungs-Pop-ups. Das Pop-up wird angezeigt, wenn versucht wird, das Skript aus dem Zabbix Frontend auszuführen.
timeout	string	Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist Zeitüberschreitung für die Ausführung des webhook-Skripts in Sekunden. Zeitsuffixe werden unterstützt (z. B. 30s, 1m). Mögliche Werte: 1-60s. Standard: 30s.
parameters	array	Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Webhook“ gesetzt ist Array von webhook-Eingabeparametern.
description	string	Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Webhook“ gesetzt ist 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

Eigenschaft	Typ	Beschreibung
manualinput	integer	Gibt an, ob das Skript benutzerseitig 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 Aufforderung für die manuelle Eingabe. Verhalten der Eigenschaft: - *erforderlich*, wenn `manualinput` auf „Enabled“ gesetzt ist
manualinput_validator	string	Ein Zeichenkettenfeld zur Validierung der vom Benutzer bereitgestellten Eingabe. Die Zeichenkette besteht entweder aus einem regulären Ausdruck oder aus einer durch Kommas getrennten Menge von Werten. Verhalten der Eigenschaft: - *erforderlich*, wenn `manualinput` auf „Enabled“ gesetzt ist
manualinput_validator_type	integer	Bestimmt den erwarteten Typ der Benutzereingabe. Mögliche Werte: 0 - *(Standard)* Zeichenkette. 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 das Erstellen neuer Skripte.

> **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. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Skripte gelöscht werden.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode kann ein Skript auf einem Host oder Ereignis ausgeführt werden. Ausgenommen sind Skripte vom Typ URL. Diese sind nicht ausführbar.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung, die Methode aufzurufen, kann in den Einstellungen der Benutzerrolle entzogen 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)`

Mit dieser Methode können Skripte entsprechend den angegebenen Parametern abgerufen 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)` 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	

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

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)`

Mit dieser Methode können alle verfügbaren Skripte für das angegebene Ereignis oder ein bestimmtes Skript abgerufen werden, wenn eine Skript-ID angegeben ist. Wenn manualinput angegeben wird, ersetzt sie das Makro {MANUALINPUT} durch den angegebenen Wert.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können alle verfügbaren Skripte auf dem angegebenen Host oder ein bestimmtes Skript abgerufen werden, wenn eine Skript-ID angegeben ist. Wenn manualinput angegeben wird, ersetzt es das Makro {MANUALINPUT} durch den angegebenen Wert.

> **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)` 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 die Aktualisierung vorhandener Skripte.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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*.

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.

Eigenschaft	Typ	Beschreibung
serviceid	ID	ID des Service. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
algorithm	integer	Regel zur Statusberechnung. Nur anwendbar, wenn untergeordnete Services vorhanden sind. Mögliche Werte: 0 - Status auf OK setzen; 1 - kritischster Status, wenn alle untergeordneten Services Probleme haben; 2 - kritischster Status der untergeordneten Services. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
name	string	Name des Service. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
sortorder	integer	Position des Service, die für die Sortierung verwendet wird. Mögliche Werte: 0-999. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
weight	integer	Gewichtung des Service. Mögliche Werte: 0-1000000. Standard: 0.
propagation_rule	integer	Regel für die Statusweitergabe. Mögliche Werte: 0 - *(Standard)* Service-Status unverändert weitergeben - ohne Änderungen; 1 - den weitergegebenen Status um den angegebenen `propagation_value` erhöhen (um 1 bis 5 Schweregrade); 2 - den weitergegebenen Status um den 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 dem angegebenen `propagation_value` setzen. Verhalten der Eigenschaft: - *erforderlich*, wenn `propagation_value` gesetzt ist

Eigenschaft	Typ	Beschreibung
propagation_value	integer	Wert für die 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. Verhalten der Eigenschaft: - *erforderlich*, wenn `propagation_rule` gesetzt ist
status	integer	Gibt an, ob sich der Service im Zustand OK oder Problem befindet. Wenn sich der Service im Problemzustand befindet, ist `status` gleich: - dem Schweregrad des kritischsten Problems; - dem höchsten Status eines untergeordneten Service im Problemzustand. Wenn sich der Service im Zustand OK befindet, ist `status` gleich: -1. Verhalten der Eigenschaft: - *schreibgeschützt*
description	string	Beschreibung des Service.
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Services mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben wird.
created_at	integer	Unix-Zeitstempel, wann der Service erstellt wurde.
readonly	integer	Zugriff auf den Service. Mögliche Werte: 0 - Lesen und Schreiben; 1 - Schreibgeschützt. Verhalten der Eigenschaft: - *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 stellen eine Zustandsänderung eines Service dar. Sie verfügen über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitpunkt, zu dem die Zustandsänderung des Service eingetreten ist.
value	integer	Status des Service. Eine Liste der möglichen Werte finden Sie in der `status`-Eigenschaft des Service.

Problem-Tag

Problem-Tags ermöglichen die Verknüpfung von Services mit Problemereignissen. Das Problem-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Problem-Tags. Verhalten der Eigenschaft: - *erforderlich*
operator	integer	Zuordnungsbedingung operator. Mögliche Werte: 0 - *(Standard)* Gleich; 2 - Enthält.
value	string	Wert des Problem-Tags.

service.create

Beschreibung

`object service.create(object/array services)`

Mit dieser Methode können neue Services erstellt 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 User roles.

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 das Löschen von Services.

> **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

(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)

Diese Methode ermöglicht es, Services entsprechend den angegebenen Parametern abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. 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
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)`

Mit dieser Methode können vorhandene Services aktualisiert 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 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)`

Mit dieser Methode können neue SLA-Objekte erstellt werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können SLA-Einträge gelöscht werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 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)`

Mit dieser Methode können SLA-Objekte entsprechend den angegebenen Parametern abgerufen 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 User roles.

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)`

Mit dieser Methode können die Daten des Service Level Indicator (SLI) für ein Service Level Agreement (SLA) berechnet 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)` 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 die Aktualisierung bestehender SLA-Einträge.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter User roles.

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 das Erstellen neuer Symbolzuordnungen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Symbolzuordnungen 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 entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Diese Methode ermöglicht das Abrufen von Symbolzuordnungen entsprechend den angegebenen Parametern.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können vorhandene Icon-Maps aktualisiert werden.

> **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. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können neue Token erstellt werden.

Note:
Die Berechtigung *API-Token verwalten* permission ist erforderlich, damit die Benutzerrolle Token für andere Benutzer verwalten kann.

Attention:
Ein mit dieser Methode erstellter Token muss außerdem generiert werden, bevor er 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 das Löschen von Tokens.

> **Note:**
> Die Berechtigung *Manage API tokens* permission ist erforderlich, damit die Benutzerrolle Tokens für andere Benutzer verwalten kann.

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)`

Mit dieser Methode können Token generiert werden.

Note:
Die Berechtigung *Manage API tokens* permission ist erforderlich, damit die Benutzerrolle Token für andere Benutzer verwalten kann.

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)`

Mit dieser Methode können Token entsprechend den angegebenen Parametern abgerufen werden.

> **Note:**
> Nur der Benutzertyp *Super admin* darf Token anderer 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 die Aktualisierung vorhandener Token.

> **Note:**
> Die Berechtigung *Manage API tokens* permission ist erforderlich, damit die Benutzerrolle Token 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.

Eigenschaft	Typ	Beschreibung
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)`

Mit dieser Methode können Trenddaten entsprechend den angegebenen Parametern abgerufen werden.

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	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.
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.

Eigenschaft	Typ	Beschreibung
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.
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, den Verlauf von Datenpunkten zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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

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)`

Diese Methode ermöglicht es, Verlaufsdaten entsprechend den angegebenen Parametern abzurufen.

Attention:
Diese Methode kann historische Daten einer gelöschten Entität zurückgeben, wenn diese Daten noch nicht vom Housekeeper entfernt wurden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
history	integer	Zurückzugebende History-Objekttypen. Mögliche Werte: 0 - numerischer Gleitkommawert; 1 - Zeichen; 2 - Log; 3 - *(Standard)* numerisch vorzeichenlos; 4 - Text; 5 - binär; 6 - JSON.
hostids	ID/array	Nur History der angegebenen Hosts zurückgeben.
itemids	ID/array	Nur History der angegebenen Datenpunkte zurückgeben.
maxValueSize	integer	Maximale Anzahl von Bytes, die in der Eigenschaft `value` zurückgegeben werden. Möglicher Wertebereich: 1-134217728 (1 Byte bis 128MiB). Wenn auf `null` gesetzt, wird keine Begrenzung angewendet. Standard: 65536 (64KiB). Parameter behavior: - *unterstützt*, wenn `history` auf ”binary” oder ”JSON” gesetzt ist
time_from	timestamp	Nur Werte zurückgeben, die nach oder zum angegebenen Zeitpunkt empfangen wurden.

Parameter	Type	Beschreibung
time_till	timestamp	Nur Werte zurückgeben, die vor oder zum angegebenen Zeitpunkt empfangen wurden.
sortfield	string/array	Das Ergebnis nach den angegebenen Eigenschaften sortieren. Mögliche Werte: `itemid`, `clock`, `ns`.
search	object	Ergebnisse zurückgeben, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben sind, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützt keine Suche nach `value`, wenn `history` auf 6 (JSON) gesetzt ist.
countOutput	boolean	Diese Parameter sind im Referenzkommentar 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*.

Eigenschaft	Typ	Beschreibung
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*.

template.create

Beschreibung

`object template.create(object/array templates)`

Diese Methode ermöglicht das Erstellen neuer Vorlagen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Diese Methode ermöglicht es, Vorlagen entsprechend den angegebenen Parametern abzurufen.

> **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)` 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.

Parameter	Type	Beschreibung
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.
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`.

Parameter	Type	Beschreibung
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.
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)`

Mit dieser Methode können mehreren angegebenen Vorlagen gleichzeitig mehrere zugehörige Objekte hinzugefügt werden.

> **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 entzogen 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)`

Mit dieser Methode können verknüpfte Objekte aus mehreren Vorlagen entfernt werden.

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 entzogen 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)`

Mit dieser Methode können vorhandene Vorlagen aktualisiert werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Vorlagen gelöscht werden.

Beim Löschen einer Vorlage werden auch alle zugehörigen Entitäten (Datenpunkte, Auslöser, Diagramme usw.) von allen Hosts oder Vorlagen entfernt, mit denen sie verknüpft ist. Wenn diese Entitäten beibehalten werden sollen, heben Sie zunächst die Verknüpfung der Vorlage mit den entsprechenden Hosts oder Vorlagen auf, indem Sie die Methoden `host.update`, `host.massremove`, `template.update` oder `template.massremove` verwenden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 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.

Vorlagen-Dashboard

Das Vorlagen-Dashboard-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
dashboardid	ID	ID des Vorlagen-Dashboards. Property behavior: - *read-only* - *required* für Aktualisierungsvorgänge
name	string	Name des Vorlagen-Dashboards. Property behavior: - *required* für Erstellungsvorgänge
templateid	ID	ID der Vorlage, zu der das Dashboard gehört. Property behavior: - *constant* - *required* für Erstellungsvorgänge
display_period	integer	Standard-Anzeigezeitraum der Seite (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 verwendet wird, um importierte Vorlagen-Dashboards mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn nicht angegeben.

Seite des Vorlagen-Dashboards

Das Objekt der Seite des Vorlagen-Dashboards hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
dashboard_pageid	ID	ID der Dashboard-Seite. Verhalten der Eigenschaft: - *schreibgeschützt*
name	string	Name der Dashboard-Seite. Standard: leerer String.
display_period	integer	Anzeigedauer der Dashboard-Seite (in Sekunden). Mögliche Werte: 0, 10, 30, 60, 120, 600, 1800, 3600. Standard: 0 (verwendet die Standard-Anzeigedauer der Seite).
widgets	array	Array von Objekten des Typs Widget des Vorlagen-Dashboards.

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)`

Mit dieser Methode können neue Vorlagen-Dashboards erstellt werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Vorlagen-Dashboards gelöscht werden.

> **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 entzogen 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)`

Mit dieser Methode können Vorlagen-Dashboards entsprechend den angegebenen Parametern abgerufen 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)` 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)`

Mit dieser Methode können vorhandene Vorlagen-Dashboards aktualisiert werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter User roles.

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 das Erstellen neuer Vorlagengruppen.

> **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. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Vorlagengruppen gelöscht werden.

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. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen 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)`

Mit dieser Methode können Vorlagengruppen entsprechend den angegebenen Parametern abgerufen 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)` 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, mehrere zugehörige Objekte gleichzeitig zu allen angegebenen Vorlagengruppen hinzuzufügen.

> **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 entzogen 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)`

Mit dieser Methode können verknüpfte Objekte aus mehreren Vorlagengruppen entfernt werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen *Admin* und *Super admin* 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 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)`

Mit dieser Methode können Berechtigungen auf alle Untergruppen von Vorlagengruppen angewendet werden.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* 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	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 die Aktualisierung bestehender Vorlagengruppen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 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.

Eigenschaft	Typ	Beschreibung
maintenanceid	ID	ID der Wartung. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Wartung. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
active_since	timestamp	Zeitpunkt, zu dem die Wartung aktiv wird (einschließlich). Der angegebene Wert wird auf Minuten abgerundet. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
active_till	timestamp	Zeitpunkt, zu dem die Wartung nicht mehr aktiv ist (ausschließlich). Der angegebene Wert wird auf Minuten abgerundet. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
description	string	Beschreibung der Wartung.
maintenance_type	integer	Typ der Wartung. Mögliche Werte: 0 - *(Standard)* mit Datenerfassung; 1 - ohne Datenerfassung.
tags_evaltype	integer	Auswertungsmethode für Problem-Tags. 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

Problem-Tag

Das Problem-Tag-Objekt wird verwendet, um festzulegen, welche Probleme unterdrückt werden müssen, wenn die Wartung wirksam wird. Tags können nur angegeben werden, wenn `maintenance_type` des Wartungsobjekts auf „mit Datensammlung“ gesetzt ist. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Name des Problem-Tags. Verhalten der Eigenschaft: - *erforderlich*
operator	integer	Bedingungs-Operator. Mögliche Werte: 0 - Gleich; 2 - *(Standard)* Enthält.
value	string	Wert des Problem-Tags.

maintenance.create

Beschreibung

`object maintenance.create(object/array maintenances)`

Diese Methode ermöglicht das Erstellen neuer Wartungen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* 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 erstellende Wartungen.

Zusätzlich zu den Standard-Wartungseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Host-Gruppen, die gewartet werden. Für die Host-Gruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `hosts` nicht gesetzt ist
hosts	object/array	Hosts, die gewartet werden. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `groups` nicht gesetzt ist
timeperiods	object/array	Wartungs-Zeitperioden. Parameterverhalten: - *erforderlich*
tags	object/array	Problem-Tags. Definieren, welche Probleme unterdrückt werden müssen. Wenn keine Tags angegeben werden, werden alle Probleme aktiver Wartungs-Hosts unterdrückt. Parameterverhalten: - *unterstützt*, wenn `maintenance_type` des Wartungsobjekts auf „with data collection“ gesetzt ist

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 Host-Gruppe mit der ID „2“ und 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 wirksam werden 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": "2"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
```

```
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Zeitperiode

Quelle

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Beschreibung

`object maintenance.delete(array maintenanceIds)`

Mit dieser Methode können Wartungszeiträume gelöscht werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die 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 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)`

Mit dieser Methode können Wartungen entsprechend den angegebenen Parametern abgerufen werden.

> **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 Wartungen zurück, die den angegebenen Host-Gruppen zugewiesen sind.
hostids	ID/array	Gibt nur Wartungen zurück, die den angegebenen Hosts zugewiesen sind.
maintenanceids	ID/array	Gibt nur Wartungen mit den angegebenen IDs zurück.
selectHostGroups	query	Gibt eine Eigenschaft `hostgroups` mit den der Wartung zugewiesenen Host-Gruppen zurück.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit den der Wartung zugewiesenen Hosts zurück.
selectTags	query	Gibt eine Eigenschaft `tags` mit Problem-Tags der Wartung zurück.
selectTimeperiods	query	Gibt eine Eigenschaft `timeperiods` mit 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 sind im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parameter	Type	Beschreibung
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 Wartungen

Rufen Sie alle konfigurierten Wartungen sowie die Daten zu den zugewiesenen Host-Gruppen, definierten Zeiträumen und Problem-Tags ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
            "tags": [
```

```
                {
                    "tag": "service",
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Siehe auch

- Host
- Host-Gruppe
- Zeitperiode

Quelle

CMaintenance::get() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Beschreibung

`object maintenance.update(object/array maintenances)`

Diese Methode ermöglicht die Aktualisierung bestehender Wartungen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Benutzerrolleneinstellungen entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Wartungseigenschaften.

Die Eigenschaft `maintenanceid` muss für jede Wartung definiert sein, 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	Typ	Beschreibung
groups	object/array	Host-Gruppen, die die aktuellen Gruppen ersetzen. Für die Host-Gruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `hosts` nicht gesetzt 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 gesetzt ist
timeperiods	object/array	Wartungs-Zeitperioden, die die aktuellen Zeiträume ersetzen.

Parameter	Typ	Beschreibung
tags	object/array	Problem-Tags, die die aktuellen Tags ersetzen. Parameterverhalten: - *unterstützt*, wenn `maintenance_type` des Wartungsobjekts auf „with data collection“ gesetzt ist

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 hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
httptestid	ID	ID des Webszenarios. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
hostid	ID	ID des Hosts, zu dem das Webszenario gehört. Eigenschaftsverhalten: - *konstant* - *erforderlich* für Erstellungsvorgänge
name	string	Name des Webszenarios. Eigenschaftsverhalten: - *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 Sekunden oder eine Zeiteinheit mit Suffix (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, das für die einfache HTTP-, NTLM-, Kerberos- oder Digest-Authentifizierung verwendet wird.
http_proxy	string	Proxy, der vom Webszenario verwendet wird, angegeben als *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Benutzername, der für die einfache HTTP-, NTLM-, Kerberos- oder Digest-Authentifizierung verwendet wird.
retries	integer	Anzahl der Versuche, mit denen ein Webszenario jeden Schritt ausführt, bevor es fehlschlägt. Standard: 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-Datei mit dem privaten Schlüssel, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).
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.

Eigenschaft	Typ	Beschreibung
templateid	ID	ID des Webszenarios der übergeordneten Vorlage. Eigenschaftsverhalten: - *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 übereinstimmt. Mögliche Werte: 0 - *(Standard)* Host-Überprüfung 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 zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Eigenschaftsverhalten: - *unterstützt*, wenn das Webszenario zu einer Vorlage gehört

Szenarioschritt

Das Objekt „Szenarioschritt“ definiert eine bestimmte Prüfung eines Webszenarios. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
name	string	Name des Szenarioschritts. Property behavior: - *required*
no	integer	Sequenznummer des Schritts in einem Webszenario. Property behavior: - *required*
url	string	Zu prüfende URL. Property behavior: - *required*
follow_redirects	integer	Gibt an, ob HTTP-Weiterleitungen gefolgt werden soll. Mögliche Werte: 0 - Weiterleitungen nicht folgen; 1 - *(Standard)* Weiterleitungen folgen.
headers	array	HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Header des Szenarioschritts überschreiben die für das Webszenario angegebenen Header.
posts	string/array	HTTP-POST-Variablen als Zeichenfolge (rohe POST-Daten) oder als Array von HTTP-Feldern (Formulardaten).
required	string	Text, der in der Antwort vorhanden sein muss.
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 erforderlicher HTTP-Statuscodes, durch Kommas getrennt.

Eigenschaft	Type	Beschreibung
timeout	string	Zeitüberschreitung der Anfrage in Sekunden. Akzeptiert Sekunden, eine Zeiteinheit mit Suffix 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 das Erstellen neuer Webszenarien.

> **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 Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen 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 das Löschen von Webszenarien.

> **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 entzogen 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)`

Mit dieser Methode können Webszenarien entsprechend den angegebenen Parametern abgerufen werden.

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 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.
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	

Parameter	Type	Beschreibung
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 die Aktualisierung bestehender Webszenarien.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Benutzerrolleneinstellungen entzogen 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	Type	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. Property behavior: - *erforderlich*, wenn `type` auf ”1”, ”2”, ”3”, ”4” gesetzt ist - *unterstützt*, wenn `type` auf ”5” gesetzt ist
newvalue	string	Wert, auf den der ursprüngliche Wert abgebildet wird. Property behavior: - *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 das Erstellen neuer Wertezuordnungen.

> **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. Siehe Benutzerrollen für weitere Informationen.

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 das Löschen von Wertezuordnungen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Siehe Benutzerrollen für weitere Informationen.

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)`

Mit dieser Methode können Wertzuordnungen entsprechend den angegebenen Parametern abgerufen 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)` 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)`

Mit dieser Methode können vorhandene Wertezuordnungen aktualisiert werden.

> **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.

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 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.
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`.

Parameter	Type	Beschreibung
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).
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 Verlauf

ZBXNEXT-10069 Dem Verfahren history.get wurde der neue Parameter `maxValueSize` hinzugefügt. Wenn Sie Daten für Datenpunkte mit einem Binärwert (`history=5`) anfordern, setzen Sie `maxValueSize` auf `null`; andernfalls gibt `history.get` den Wert des Datenpunkts auf 64 KiB gekürzt zurück.

Host

ZBXNEXT-10387 `host.massupdate`: Methode entfernt. Bitte verwenden Sie stattdessen `host.update`.

Hostgruppe

ZBXNEXT-10387 `hostgroup.massupdate`: Methode entfernt. Bitte verwenden Sie stattdessen `hostgroup.update`, `hostgroup.massadd` oder `hostgroup.massremove`.

hostinterface

ZBXNEXT-10387 `hostinterface.replacehostinterfaces`: Methode entfernt. Bitte verwenden Sie stattdessen `hostinterface.updat` `hostinterface.massadd` oder `hostinterface.massremove`.

Vorlage

ZBXNEXT-10387 `template.massupdate`: Methode entfernt. Bitte verwenden Sie stattdessen `template.update`, `template.massadd` oder `template_massremove`.

Vorlagengruppe

ZBXNEXT-10387 `templategroup.massupdate`: Methode entfernt. Bitte verwenden Sie stattdessen `templategroup.update`, `templategroup.massadd` oder `templategroup.massremove`.

Andere Änderungen und Fehlerbehebungen Konnektor

ZBXNEXT-10069 Neuer Wert `item_value_type` (64 - JSON) zum Objekt connector hinzugefügt.

Dashboard

ZBXNEXT-4769 Neues Dashboard-Widget-Feld `show_hostnames` zum Widget graph hinzugefügt.
ZBXNEXT-9546 Neue Dashboard-Widget-Felder `ds.0.item_tags_evaltype`, `ds.0.item_tags.0.tag`, `ds.0.item_tags.0.operator` und `ds.0.item_tags.0.value` zu den Widgets svggraph und piechart hinzugefügt.
ZBXNEXT-10158 Neue Dashboard-Widget-Felder `columns.0.aggregate_columns`, `columns.0.combined_column_name` und `columns.0.column_aggregate_function` zum Widget topitems hinzugefügt.
ZBXNEXT-9424, ZBXNEXT-4769 Neuer Dashboard-Widget-Typ scatterplot hinzugefügt.
ZBXNEXT-8160 Neue Dashboard-Widget-Felder `clustering_mode` und `clustering_zoom_level` zum Widget geomap hinzugefügt.
ZBXNEXT-943 Neues Dashboard-Widget-Feld `ds.0.invert_values` zum Widget graph hinzugefügt.

templatedashboard

ZBXNEXT-9424 Neuer Widget-Typ scatterplot für Vorlagen-Dashboards hinzugefügt.

Verlauf

ZBXNEXT-10069 Neuer `history`-Wert (6 - JSON) zur Methode history.get hinzugefügt.
ZBXNEXT-10069 Neues `JSON history`-Objekt hinzugefügt.

Hostprototyp

ZBXNEXT-9546 Neuer Parameter `selectInheritedTags` zur Methode hostprototype.get hinzugefügt.

httptest

ZBXNEXT-9546 Neue Parameter `inheritedTags` und `selectInheritedTags` zur Methode httptest.get hinzugefügt.

Datenpunkt

ZBXNEXT-10069 Neuen `value_type`-Wert (6 - JSON) zum Objekt Datenpunkt hinzugefügt.
ZBXNEXT-9546 Neue Parameter `inheritedTags` und `selectInheritedTags` zur Methode item.get hinzugefügt.

Datenpunktprototyp

ZBXNEXT-10069 Dem Objekt itemprototype wurde ein neuer `value_type`-Wert (6 - JSON) hinzugefügt.
ZBXNEXT-9546 Der Methode itemprototype.get wurde der neue Parameter `selectInheritedTags` hinzugefügt.

Vorlage

ZBXNEXT-9546 Neue Parameter `inheritedTags` und `selectInheritedTags` zur Methode template.get hinzugefügt.

Auslöser

ZBXNEXT-9546 Neue Parameter `inheritedTags` und `selectInheritedTags` zur Methode trigger.get hinzugefügt.

Auslöserprototyp

ZBXNEXT-9546 Neuer Parameter `selectInheritedTags` zur Methode triggerprototype.get hinzugefügt.

triggertag

ZBXNEXT-5467 Eigenschaft `automatic` zum Objekt Auslöser-Tag hinzugefügt.

Benutzerverzeichnis

ZBXNEXT-9834 Eigenschaften `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_ha` und `sp_certificate_hash` zum Objekt Benutzerverzeichnis hinzugefügt.

Anhang 3. Änderungen in 8.0

8.0.1 Diese Version ist noch nicht veröffentlicht.

19 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 im Fall von lang laufenden Benutzerskripten. Das Senden von Daten an den Zabbix Server oder Proxy ist mit dem Dienstprogramm Zabbix sender oder dem Protokoll von Zabbix sender möglich. Das Senden von Daten an den Zabbix Server ist auch 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[]` Zabbix-Agent-Datenpunkte

Der Zabbix-Agent-Datenpunkt `system.run[]` ist ein Datenpunkt für einen benutzerdefinierten Befehl (verknüpft mit einem vordefinierten Schlüssel `system.run[]`, zum Beispiel `system.run[myscript.sh]`), der auf dem Host ausgeführt werden kann, auf dem der Zabbix Agent läuft.

Hinweis: `system.run[]`-Datenpunkte sind standardmäßig deaktiviert und müssen bei Verwendung aktiviert (erlaubt) und in der Konfigurationsdatei des Zabbix Agent oder Agent 2 definiert werden (Konfigurationsparameter `AllowKey`).

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

Der HTTP-Agent Datenpunkt ist ein Datenpunkt zum Ausführen von Datenanfragen über HTTP/HTTPS.
HTTP-Agent-Datenpunkte sind nützlich, um Anfragen an HTTP-Endpunkte zu senden und Daten von Diensten wie *Elasticsearch* und *OpenWeatherMap* abzurufen, den Status der Zabbix-API oder den Status des Apache- oder Nginx-Webservers zu prüfen usw.
HTTP-Agent-Datenpunkte (mit aktivierter Trapper-Funktion) 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 Überwachungsfunktionen von Zabbix Agent 2. Sie sind in der Programmiersprache Go geschrieben und bieten eine Alternative zu ladbaren Modulen (geschrieben in C).

Ein Plugin ist ein Go-Paket, das die Struktur definiert und eine oder mehrere Plugin-Schnittstellen implementiert (*Exporter*, *Collector*, *Configurator*, *Runner*, *Watcher*). Es werden zwei Typen von Zabbix Agent 2-Plugins unterstützt:

- Integrierte Plugins (unterstützt seit Zabbix 4.4.0)
- Ladbare Plugins (unterstützt seit Zabbix 6.0.0)

Siehe die Liste der von Zabbix bereitgestellten Plugins.

Anleitungen und Tutorials zum Erstellen eigener Plugins finden Sie im 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.

1 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. 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. Daher sollte diese Funktion 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 Quellkompatibilität 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_**
und **zbx_module_uninit()**:

```
int zbx_module_init(void);
```

Diese Funktion sollte die erforderliche Initialisierung für das Modul durchführen, falls nötig. 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 wird in einer ZBX_METRIC-Struktur definiert; siehe den folgenden Abschnitt für Details. Die Liste wird durch eine ZBX_METRIC-Struktur beendet, deren Feld ”key” den Wert NULL hat.

```
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 anzugeben, 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. Callback-Funktionen werden als Felder der Struktur ZBX_HISTORY_WRITE_CBS bereitgestellt; Felder können NULL sein, wenn das Modul nicht am Verlauf eines bestimmten Typs interessiert ist.

```
int zbx_module_uninit(void);
```

Diese Funktion sollte die erforderliche Deinitialisierung durchführen, falls nötig, z. B. das Freigeben zugewiesener Ressourcen, das Schließen von Dateideskriptoren 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 wird, wenn der Zabbix Agent mit dem Flag ”-p” gestartet wird (z. B. ”1,1000”, kann NULL sein). Eine Beispieldefinition kann wie folgt 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 Datenpunktwert erfolgreich abgerufen wurde. Andernfalls sollten sie SYSINFO_RET_FAIL zurückgeben. Siehe das Beispielmodul ”dummy” unten für Details dazu, wie Informationen aus AGENT_REQUEST abgerufen und Informationen in AGENT_RESULT gesetzt werden.

Bereitstellen von Callbacks für den Verlaufsexport

> **Attention:**
> Der Verlaufsexport 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 Array "history" mit "history_num" Elementen als Argumente annehmen. Abhängig vom Typ der zu exportierenden Verlaufsdaten 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;
```

Die Callbacks werden von den Verlaufssynchronisierungsprozessen des Zabbix Server am Ende des Verlaufssynchronisierungsvorgangs verwendet, nachdem die Daten in die Zabbix-Datenbank geschrieben und im Wertespeicher gespeichert wurden.

Attention:
Im Fall eines internen Fehlers im Modul für den Verlaufsexport wird empfohlen, das Modul so zu schreiben, dass es nicht die gesamte Überwachung blockiert, bis es sich erholt, sondern stattdessen Daten verwirft und dem Zabbix Server erlaubt, weiterzulaufen.

Module erstellen

Module sind derzeit dafür vorgesehen, innerhalb des Zabbix-Quellcodebaums erstellt zu werden, da die Modul-API von einigen Datenstrukturen abhängt, die in den Zabbix-Headern definiert sind.

Der wichtigste Header für ladbare Module ist **include/module.h**, der diese Datenstrukturen definiert. Weitere notwendige System-Header, die dafür sorgen, dass **include/module.h** korrekt funktioniert, sind **stdlib.h** und **stdint.h**.

Mit diesen Informationen ist alles für die Erstellung des Moduls vorbereitet. 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. Ein Beispiel finden Sie im untenstehenden „dummy“-Modul.

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 – Modul(e), die beim Start geladen werden. Die Module müssen sich in einem durch LoadModulePath angegebenen Verzeichnis befinden, oder dem Modulnamen muss ein Pfad vorangestellt sein. Wenn der vorangestellte Pfad absolut ist (mit '/' beginnt), wird LoadModulePath ignoriert. Es ist zulässig, mehrere LoadModule-Parameter anzugeben.

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 Agent werden die Module mariadb.so, apache.so und kernel.so aus dem Verzeichnis /usr/local/lib/zabbix/agent geladen, während dummy.so aus /usr/local/lib/zabbix geladen wird. Der Agent kann nicht gestartet werden, wenn ein Modul fehlt, bei fehlerhaften Berechtigungen oder wenn eine Shared Library kein Zabbix-Modul ist.

Frontend-Konfiguration

Ladbare Module werden von Zabbix Agent, Server und Proxy unterstützt. Daher hängt der Datenpunkttyp im Zabbix Frontend davon ab, wo das Modul geladen wird. Wenn das Modul in den Agent geladen wird, sollte der Datenpunkttyp „Zabbix agent“ oder „Zabbix agent (active)“ sein. Wenn das Modul in den Server oder Proxy geladen wird, sollte der Datenpunkttyp „Simple check“ sein.

Der Verlaufsexport über Zabbix-Module benötigt keine Frontend-Konfiguration. Wenn das Modul erfolgreich vom Server geladen wird und die Funktion **zbx_module_history_write_cbs()** bereitstellt, die mindestens eine Callback-Funktion ungleich NULL zurückgibt, wird der Verlaufsexport 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"

/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
```

```
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
```

```
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                 as not supported by zabbix                 *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}
```

```c
/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
```

```
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

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 im Bereich von 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 nicht für Windows-Agenten 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 soll, sollten Sie wahrscheinlich das Parsen einer modulspezifischen Konfigurationsdatei implementieren.

3 Frontend-Module

Überblick

Es ist möglich, die Funktionalität des Zabbix Frontend zu erweitern, indem Module von Drittanbietern hinzugefügt oder eigene Module entwickelt werden, ohne dass der Quellcode von Zabbix geändert werden muss.

Beachten Sie, dass der Modulcode mit denselben Berechtigungen ausgeführt wird wie der Quellcode von Zabbix. Das bedeutet:

- Module von Drittanbietern können schädlich sein. Sie müssen den Modulen vertrauen, die Sie installieren;
- Fehler im Code eines Drittanbieter-Moduls können das Frontend zum Absturz bringen. Falls dies geschieht, entfernen Sie einfach den Modulcode aus dem Frontend. Sobald Sie das Zabbix Frontend neu laden, sehen Sie einen Hinweis, dass einige Module fehlen. Gehen Sie zu Modulverwaltung (unter *Administration* → *Allgemein* → *Module*) und klicken Sie erneut auf *Verzeichnis durchsuchen*, 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 leicht erkennen zu können.

Unmittelbar bevor Sie ein Modul installieren:

- Stellen Sie sicher, dass Sie das Modul aus einer vertrauenswürdigen Quelle heruntergeladen haben. Die Installation von schädlichem Code kann zu Folgen wie Datenverlust führen
- Verschiedene Versionen desselben Moduls (dieselbe ID) können parallel installiert werden, aber jeweils kann nur eine einzelne Version aktiviert sein

Schritte zur Installation eines Moduls:

- Entpacken Sie Ihr Modul in seinem eigenen Ordner im Ordner `modules` des Zabbix Frontends
- Stellen Sie sicher, dass Ihr Modulordner mindestens die Datei manifest.json enthält
- Navigieren Sie zu Modulverwaltung und klicken Sie auf die Schaltfläche *Verzeichnis durchsuchen*
- Das neue Modul wird zusammen mit seiner Version, seinem Autor, seiner Beschreibung und seinem 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, dass der Benutzer des Webservers mindestens Lese- und Suchzugriff (`r-x`) auf den Modulordner und alle Unterverzeichnisse sowie Lesezugriff (`r--`) auf alle darin enthaltenen Dateien hat.
Frontend ist abgestürzt	Der Modulcode ist nicht mit der aktuellen Zabbix-Version oder der Server-Konfiguration kompatibel. Bitte löschen Sie die Moduldateien und laden Sie das Frontend neu. Sie sehen einen Hinweis, dass einige Module fehlen. Gehen Sie zu Modulverwaltung und klicken Sie erneut auf *Verzeichnis durchsuchen*, um nicht vorhandene Module aus der Datenbank zu entfernen.
Es erscheint eine Fehlermeldung über identischen Namespace, identische ID oder identische Aktionen	Das neue Modul hat versucht, einen Namespace, eine ID oder Aktionen zu registrieren, die bereits von anderen aktivierten Modulen registriert wurden. Deaktivieren Sie das in der Fehlermeldung genannte konfliktverursachende Modul, bevor Sie das neue aktivieren.
Technische Fehlermeldungen werden angezeigt	Melden Sie Fehler dem Entwickler des Moduls.

Entwicklung von Modulen

Informationen zur Entwicklung benutzerdefinierter Module finden Sie im Developer center.

20 Anhänge

Bitte verwenden Sie die Seitenleiste, um auf Inhalte im Abschnitt „Anhänge“ zuzugreifen.

1 Installation und Einrichtung

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte in diesem Abschnitt zuzugreifen.

1 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 eigener 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 zur Datenbankerstellung für jede 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 Sicherheitslücken funktioniert. Benutzer sollten sich bewusst sein, dass bei Verwendung einiger anderer Kodierungen bekannte Sicherheitsprobleme bestehen. Informationen zum Wechsel zu UTF-8 finden Sie unter Reparieren von Zeichensatz und Sortierung der Zabbix-Datenbank. Siehe auch Einschränkungen beim Filtern 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

Die Zeichensätze utf8 (auch bekannt als utf8mb3) und utf8mb4 werden unterstützt (jeweils mit der Sortierung utf8_bin bzw. utf8mb4_bin), damit Zabbix Server/Proxy ordnungsgemäß mit der MySQL-Datenbank funktioniert. Für neue Installationen wird die Verwendung von utf8mb4 empfohlen.

Deterministische Auslöser müssen beim Import des Schemas erstellt werden. Unter MySQL und MariaDB erfordert dies, dass `GLOBAL log_bin_trust_function_creators = 1` gesetzt wird, 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 aus Zabbix-**Paketen** installieren, folgen Sie den Anweisungen für Ihre Plattform.

Wenn Sie Zabbix aus den Quellen installieren:

- Erstellen und konfigurieren Sie eine Datenbank und einen Benutzer.

```
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 setzen Sie utf8mb4 als Standardzeichensatz (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 stoppen, 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 die Installation aus Zabbix-**Paketen** durchführen, folgen Sie den Anweisungen für Ihre Plattform.

Wenn Sie Zabbix aus den Quellen installieren:

- Erstellen Sie einen Datenbankbenutzer.

Der folgende Shell-Befehl erstellt den Benutzer `zabbix`. Geben Sie bei Aufforderung ein Passwort an 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), wobei der zuvor erstellte Benutzer als Eigentümer verwendet wird (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importieren Sie das anfängliche Schema und die Daten (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/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### hier stoppen, wenn Sie eine Datenbank für den Zabbix-Proxy erstellen
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
Die obigen Befehle dienen als Beispiel und funktionieren in den meisten GNU/Linux-Installationen. Je nachdem, wie Ihr System/Ihre Datenbank konfiguriert ist, können Sie auch andere Befehle verwenden, zum Beispiel:

 `psql -U <username>`

 Falls Sie Probleme beim Einrichten der Datenbank haben, wenden Sie sich bitte an Ihren Datenbankadministrator.

TimescaleDB

Anweisungen zum Erstellen und Konfigurieren von TimescaleDB finden Sie in einem separaten Abschnitt.

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.

2 Reparatur des Zeichensatzes und der Sortierung der Zabbix-Datenbank

MySQL/MariaDB

In der Vergangenheit verwendeten MySQL und Derivate 'utf8' als Alias für utf8mb3 - MySQLs eigene 3-Byte-Implementierung des standardmäßigen UTF8, das 4 Byte verwendet. Ab MySQL 8.0.28 und MariaDB 10.6.1 ist der Zeichensatz 'utf8mb3' veraltet, und seine Unterstützung wird irgendwann eingestellt, während 'utf8' zu einer Referenz 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 erkennen, stellen Sie sicher, dass Sie zuerst den Zabbix Server und das DB-Schema auf 6.0.x oder höher aktualisieren, bevor Sie die Konvertierung zu utf8mb4 ausführen.

1. Prüfen Sie den Zeichensatz und die Kollation 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 Kollation ist nicht 'utf8mb4_bin', daher müssen wir dies korrigieren.

2. Stoppen Sie Zabbix.

3. Erstellen Sie eine Sicherungskopie der Datenbank!

4. Korrigieren Sie Zeichensatz und Kollation 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 Kollation 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 dem Datenträger benötigt.

7. Wenn keine Fehler auftreten, sollten Sie eventuell eine Sicherungskopie der korrigierten Datenbank erstellen.

8. Starten Sie Zabbix.

3 Datenbank-Upgrade auf Primärschlüssel

Übersicht

Dieser Abschnitt enthält Anweisungen zum manuellen Upgrade von Tabellen in bestehenden Installationen auf Primärschlüssel.

Das Upgrade auf Primärschlüssel optimiert die Art und Weise, wie Daten indiziert und abgerufen werden, was Abfragen beschleunigen und Speicherplatz sparen kann. Außerdem verbessert es die Datenverwaltung und Synchronisierung in Cluster-Umgebungen, 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 ausreichend freier Festplattenspeicher verfügbar ist; abhängig von der Größe Ihrer Datenbank und den gespeicherten Daten kann der Prozess bis zum 2,5-Fachen des derzeit von Verlaufstabellen verwendeten Speicherplatzes benötigen.

Primärschlüssel werden seit Zabbix 6.0 in neuen Installationen für alle Tabellen verwendet.

Es gibt kein automatisches Datenbank-Upgrade auf Primärschlüssel; bestehende Installationen können jedoch manuell **nach** dem Upgrade des Zabbix Server auf 6.0 oder neuer aktualisiert werden.

> **Attention:**
> Seit Zabbix 7.0 werden beim Upgrade von Tabellen auf Primärschlüssel Tabellen auch für die Verwendung von double-precision-Datentypen aktualisiert.

 Wenn Sie Zabbix 7.0 (oder neuer) verwenden, nutzen die Tabellen bereits double precision. Die Anweisungen auf dieser Seite können jedoch weiterhin befolgt 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 in Erwägung ziehen, 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:

- 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.

MySQL

Export und Import müssen in tmux/screen durchgeführt 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 (empfohlen) oder als beliebiger Benutzer mit FILE-Berechtigungen an.
- Starten Sie MySQL mit aktivierter Variablen 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 Daten.

Stellen Sie die Verbindung über mysqlsh her. Bei Verwendung einer Socket-Verbindung muss möglicherweise der Pfad angegeben werden.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Wechseln Sie mit Folgendem 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 Sie die Meldung „JavaScript is not supported“ erhalten, unterstützt Ihre MySQL-Shell-Installation kein JS. Installieren Sie in diesem Fall das offizielle MySQL Shell package 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 benötigt mehr Zeit und sollte nur verwendet werden, wenn ein Upgrade mit *mysqlsh* nicht möglich ist.

Tabellen-Upgrade

- Melden Sie sich an der MySQL-Konsole als root (empfohlen) oder als beliebiger Benutzer mit FILE-Berechtigungen an.
- 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 bei gestopptem Server

`max_execution_time` (in MySQL) oder `max_statement_time` (in MariaDB) muss vor der Datenmigration deaktiviert werden, um einen 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;
```

Folgen Sie den 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. Bearbeiten Sie in diesem Fall die Dateipfade in den Abfragen entsprechend 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 Datenexport deaktiviert werden, um einen 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
```

Folgen Sie den 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 unterbrochen wird. Bei Installationen mit TimescaleDB überspringen Sie diesen Abschnitt und fahren Sie mit PostgreSQL + TimescaleDB fort.

Siehe auch: Wichtige Hinweise

Tabellen-Upgrade

- 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 bei gestopptem Server

- Exportieren Sie den aktuellen Verlauf, importieren Sie ihn in die temporäre Tabelle und fügen Sie dann die Daten in die neuen Tabellen ein, wobei Duplikate ignoriert werden:

```
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 auch 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

- Aktuelle Verlaufsdaten exportieren, in die temporäre Tabelle importieren und dann die Daten in neue Tabellen einfügen, wobei Duplikate ignoriert werden:

```
\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;
```

- Befolgen Sie die 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 unterbrochen 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/timescal` aus:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- Führen Sie TimescaleDB-Hypertable-Migrationsskripte entsprechend den Komprimierungseinstellungen aus:
  - Wenn die Komprimierung aktiviert ist (bei der Standardinstallation), führen Sie Skripte aus `/usr/share/zabbix/sql-scripts/` 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 Skripte aus `/usr/share/zabbix/sql-scripts/postgresql/timescaled` 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

4 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 Neuaufbau von Indizes

MySQL baut Indizes für den Primärschlüssel während des Vorgangs `ALTER TABLE` automatisch neu auf. Es wird jedoch dringend empfohlen, Indizes zusätzlich manuell mit der Anweisung `OPTIMIZE TABLE` neu aufzubauen, um eine optimale Datenbankleistung sicherzustellen.

Für den Neuaufbau von Indizes kann vorübergehend zusätzlicher Speicherplatz in der Größe erforderlich sein, die auch die Tabelle selbst belegt. 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 stehen auch weitere Optionen zur Verfügung:

- Eine Erhöhung des MySQL-Parameters `sort_buffer_size` kann helfen, die Speicherplatznutzung beim manuellen Neuaufbau von Indizes zu verringern. Das Ändern dieser Variablen kann sich jedoch auf die gesamte Speichernutzung der Datenbank auswirken.
- Ziehen Sie in Betracht, Speicherplatz freizugeben, indem Sie möglicherweise nicht benötigte Daten löschen.
- Ziehen Sie in Betracht, den housekeeper-Parameter *Data storage period* vor der Ausführung des housekeepers 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 dringend empfohlen, siehe Wichtige Hinweise zum Neuaufbau von Indizes).

```
OPTIMIZE TABLE auditlog;
```

Primärschlüssel ohne Ausfallzeit ändern

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 zugleich den für die Änderung der Tabelle `auditlog` verwendeten Speicherplatz.

1. Eine neue Tabelle mit dem neuen Primärschlüssel erstellen und Indizes anlegen.

```
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. Tabellen austauschen.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Daten aus der alten Tabelle in die neue Tabelle kopieren.

```
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. Die alte Tabelle löschen.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Wichtige Hinweise zum Neuaufbau von Indizes

PostgreSQL baut Indizes für den Primärschlüssel während des Vorgangs `ALTER TABLE` automatisch neu auf. Es wird jedoch dringend empfohlen, Indizes zusätzlich manuell mit der Anweisung `REINDEX TABLE CONCURRENTLY` neu aufzubauen, um eine optimale Datenbankleistung sicherzustellen.

Für den Neuaufbau von Indizes kann vorübergehend bis zum Dreifachen des derzeit von Indizes belegten Speicherplatzes erforderlich sein. Um die aktuelle Größe der Indizes zu ermitteln, können Sie die folgende Abfrage ausführen:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Falls der verfügbare Speicherplatz ein Problem darstellt, folgen Sie den Anweisungen unter Primärschlüssel ohne Ausfallzeit ändern. Es stehen auch weitere Optionen zur Verfügung:

- Eine Erhöhung des PostgreSQL-Parameters `maintenance_work_mem` kann helfen, die Speicherplatznutzung beim manuellen Neuaufbau von Indizes zu verringern. Eine Änderung dieser Variablen kann sich jedoch auf die gesamte Speichernutzung der Datenbank auswirken.
- Wenn Sie über eine weitere Festplatte oder einen weiteren Tablespace mit mehr verfügbarem Speicherplatz verfügen, können Sie in Erwägung ziehen, den temporären Speicherort für den Neuaufbau der Indizes zu ändern. Sie können den PostgreSQL-Parameter `temp_tablespaces` festlegen, um einen anderen Tablespace für temporäre Objekte anzugeben.
- Ziehen Sie in Betracht, Speicherplatz freizugeben, indem Sie möglicherweise nicht benötigte Daten löschen.
- Ziehen Sie in Betracht, den Parameter *Datenspeicherungszeitraum* des housekeeper vor der Ausführung des housekeeper zu verringern.

Primärschlüssel mit Ausfallzeit ändern

1. Den aktuellen Primärschlüssel der Tabelle `auditlog` löschen und den neuen Primärschlüssel hinzufügen.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Indizes neu erstellen (optional, aber dringend empfohlen, siehe Wichtige Hinweise zum Neuerstellen von Indizes).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Primärschlüssel ohne Ausfallzeit ändern

Die manuelle Methode zum Ändern des Primärschlüssels wird hier beschrieben. Alternativ kann die Erweiterung `pg_repack` in Betracht gezogen werden, um eine neue Tabelle zu erstellen, Daten zu kopieren und die Tabellen auszutauschen.

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                                    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 aus 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

5 Sichere Verbindung zur Datenbank

Übersicht

Dieser Abschnitt enthält Schritte zur Einrichtung von Zabbix sowie Konfigurationsbeispiele für sichere TLS-Verbindungen zwischen:

Datenbank	Zabbix-Komponenten
MySQL	Zabbix Frontend, Zabbix Server, Zabbix Proxy
PostgreSQL	Zabbix Frontend, Zabbix Server, Zabbix Proxy

Um die Verbindungsverschlüsselung innerhalb des DBMS einzurichten, finden Sie Einzelheiten in der offiziellen Dokumentation des jeweiligen Anbieters:

- MySQL: Quell- und Replikat-Replikationsdatenbankserver.
- MySQL: Datenbankserver für Gruppenreplikation usw.
- PostgreSQL Verschlüsselungsoptionen.

Alle Beispiele basieren auf den GA-Releases von MySQL CE (8.0) und PgSQL (13), die über offizielle Repositories für CentOS 8 verfügbar sind.

Voraussetzungen

Für die Einrichtung der Verschlüsselung ist Folgendes erforderlich:

- Ein 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.

- Eine Datenbank-Engine (RDBMS), die aus dem offiziellen Repository des Entwicklers installiert und gepflegt wird. Betriebssysteme werden häufig mit veralteten Versionen von Datenbanksoftware ausgeliefert, für die keine Unterstützung für Verschlüsselung implementiert ist, zum Beispiel RHEL-7-basierte Systeme und PostgreSQL 9.2, MariaDB 5.5 ohne Unterstützung für Verschlüsselung.

Terminologie

Das Setzen dieser Option erzwingt die Verwendung einer TLS-Verbindung zur Datenbank vom Zabbix Server/Proxy und Frontend zur Datenbank:

- `required` - Verbindung über TLS als Transportmodus ohne Identitätsprüfungen herstellen
- `verify_ca` - Verbindung über TLS herstellen und das Zertifikat verifizieren
- `verify_full` - Verbindung über TLS herstellen, das Zertifikat verifizieren und prüfen, dass die durch 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 Configure DB connection das Kontrollkästchen *Database TLS encryption*, um die Transportverschlüsselung zu aktivieren.
- Aktivieren Sie das Kontrollkästchen *Verify database certificate*, das erscheint, wenn das Feld *TLS encryption* aktiviert ist, um die Verschlüsselung mit Zertifikaten zu aktivieren.

Note:
Für MySQL ist das Kontrollkästchen *Database TLS encryption* deaktiviert, wenn *Database host* auf localhost gesetzt ist, da eine Verbindung, die eine Socket-Datei (unter Unix) oder Shared Memory (unter Windows) verwendet, nicht verschlüsselt werden kann.
Für PostgreSQL ist das Kontrollkästchen *TLS encryption* deaktiviert, wenn der Wert des Feldes *Database 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):

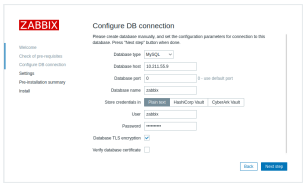
Parameter	Beschreibung
Database TLS CA file	Geben Sie den vollständigen Pfad zu einer gültigen TLS-Zertifizierungsstellen-Datei (CA) an.
Database TLS key file	Geben Sie den vollständigen Pfad zu einer gültigen TLS-Schlüsseldatei an.
Database TLS certificate file	Geben Sie den vollständigen Pfad zu einer gültigen TLS-Zertifikatsdatei an.
Database host verification	Aktivieren Sie dieses Kontrollkästchen, um die Host-Verifizierung zu aktivieren. Für MYSQL deaktiviert, da die PHP-MySQL-Bibliothek das Überspringen der Validierung des Peer-Zertifikats nicht erlaubt.
Database TLS cipher list	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 schreibbar sind, erzeugt das Frontend im Bericht System information eine Warnung mit dem Hinweis „TLS certificate files must be read-only.“ (wird nur angezeigt, wenn der PHP-Benutzer Eigentümer des Zertifikats ist).

Passwortgeschützte Zertifikate werden nicht unterstützt.

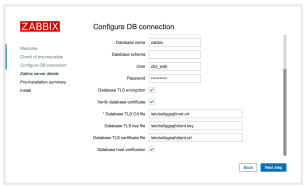
Anwendungsfälle

Das Zabbix Frontend verwendet eine GUI-Oberfläche, um mögliche Optionen zu definieren: `required`, `verify_ca`, `verify_full`. Geben Sie die erforderlichen Optionen im Schritt *Configure DB connections* des Installationsassistenten an. Diese Optionen werden wie folgt der Konfigurationsdatei (zabbix.conf.php) zugeordnet:

GUI-Einstellungen	Konfigurationsdatei	Beschreibung	Ergebnis
	... // Used for TLS connection. $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	*Database TLS encryption* aktivieren *Verify database certificate* nicht aktivieren	Modus `required` 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. *Database TLS encryption* und *Verify database certificate* aktivieren 2. Pfad zur *Database TLS CA file* angeben	Modus `verify_ca` aktivieren.

GUI-Einstellungen	Konfigurationsdatei	Beschreibung	Ergebnis

	...
	// Used for TLS connection with strictly defined Cipher list.
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '<cipher_list>';
	...
	
	Oder:
	
	...
	// Used for TLS connection without Cipher list defined - selected by MySQL server
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '';
	...
		1. *Database TLS encryption* und *Verify database certificate* aktivieren
		2. Pfad zur *Database TLS key file* angeben
		3. Pfad zur *Database TLS CA file* angeben
		4. Pfad zur *Database TLS certificate file* angeben
		5. *Database TLS cipher list* angeben (optional)
			Modus `verify_full` für MySQL aktivieren.

	... $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. *Database TLS encryption* und *Verify database certificate* aktivieren 2. Pfad zur *Database TLS key file* angeben 3. Pfad zur *Database TLS CA file* angeben 4. Pfad zur *Database TLS certificate file* angeben 5. *Database host verification* aktivieren	Modus `verify_full` für PostgreSQL aktivieren.

Siehe auch: Beispiele für die Verschlüsselungskonfiguration für MySQL, Beispiele für die Verschlüsselungskonfiguration für PostgreSQL.

Zabbix Server/Proxy-Konfiguration

Sichere Verbindungen zur Datenbank können mit den entsprechenden Parametern in der Zabbix-server- und/oder proxy-Konfigurationsdatei eingerichtet werden.

Konfiguration	Ergebnis
Keine	Verbindung zur Datenbank ohne Verschlüsselung.
1. `DBTLSConnect=required` setzen	Server/Proxy stellen eine TLS-Verbindung zur Datenbank her. Eine unverschlüsselte Verbindung ist nicht zulässig.
1. `DBTLSConnect=verify_ca` setzen 2. `DBTLSCAFile` setzen - die Datei der TLS-Zertifizierungsstelle angeben	Server/Proxy stellen nach der Überprüfung des Datenbankzertifikats eine TLS-Verbindung zur Datenbank her.
1. `DBTLSConnect=verify_full` setzen 2. `DBTLSCAFile` setzen - die Datei der TLS-Zertifizierungsstelle angeben	Server/Proxy stellen nach der Überprüfung des Datenbankzertifikats und der Identität des Datenbank-Host eine TLS-Verbindung zur Datenbank her.
1. `DBTLSCAFile` setzen - die Datei der TLS-Zertifizierungsstelle angeben 2. `DBTLSCertFile` setzen - die Datei des öffentlichen Client-Schlüsselzertifikats angeben 3. `DBTLSKeyFile` setzen - die Datei des privaten Client-Schlüssels angeben	Server/Proxy stellen beim Verbinden mit der Datenbank ein Client-Zertifikat bereit.
1. `DBTLSCipher` setzen - die Liste der Verschlüsselungsverfahren, die der Client für Verbindungen mit TLS-Protokollen bis TLS 1.2 zulässt oder `DBTLSCipher13` - die Liste der Verschlüsselungsverfahren, die der Client für Verbindungen mit dem TLS-1.3-Protokoll zulässt	(MySQL) Die TLS-Verbindung wird mit einem Verschlüsselungsverfahren aus der angegebenen Liste hergestellt. (PostgreSQL) Das Setzen dieser Option wird als Fehler betrachtet.

1 MySQL-Verschlüsselungskonfiguration

Überblick

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 keine Verschlüsselungsoptionen verfügbar. In diesem Fall verwendet eine Verbindung zwischen dem 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 noch viele weitere Kombinationen verfügbar.

Voraussetzungen

Installieren Sie die MySQL-Datenbank aus dem offiziellen Repository.

Weitere Informationen zur Verwendung des MySQL-Repositorys 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 aktiviert 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 sofort einsatzbereit für den `required`-Verschlüsselungsmodus. Ein serverseitiges Zertifikat wird nach der Ersteinrichtung und dem Start erstellt.

Erstellen Sie Benutzer und Rollen für die Hauptkomponenten:

```
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 Identitätsprüfung verwendet wird, sondern der Benutzer so konfiguriert ist, dass er nur verschlüsselte Verbindungen verwendet. Weitere Details zur Konfiguration von Benutzern finden Sie in der MySQL-Dokumentation.

Führen Sie Folgendes aus, um die Verbindung zu prüfen (Socket-Verbindungen können nicht zum Testen sicherer Verbindungen verwendet werden):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

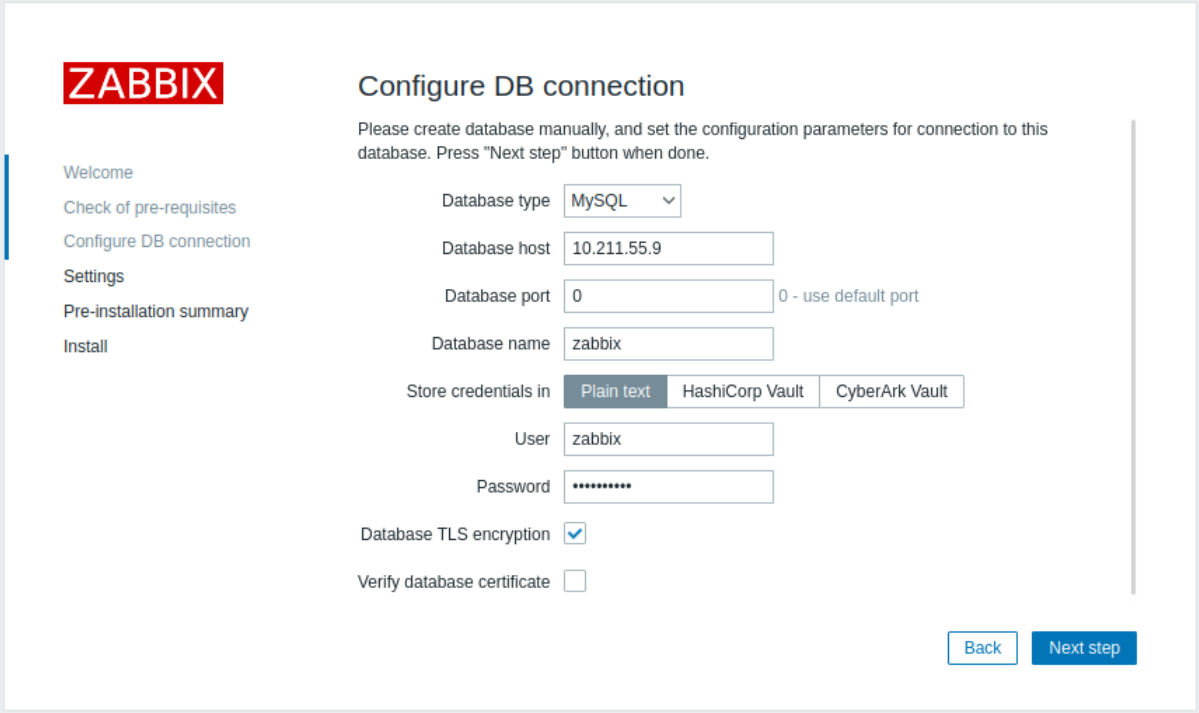
Prüfen Sie den aktuellen Status und die verfügbaren Cipher Suites:

```
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 eine reine 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 eine reine Transportverschlüsselung für Verbindungen zwischen Server und der Datenbank zu aktivieren, konfigurieren Sie */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Verschlüsselung mit Verifizierung der Zertifizierungsstelle

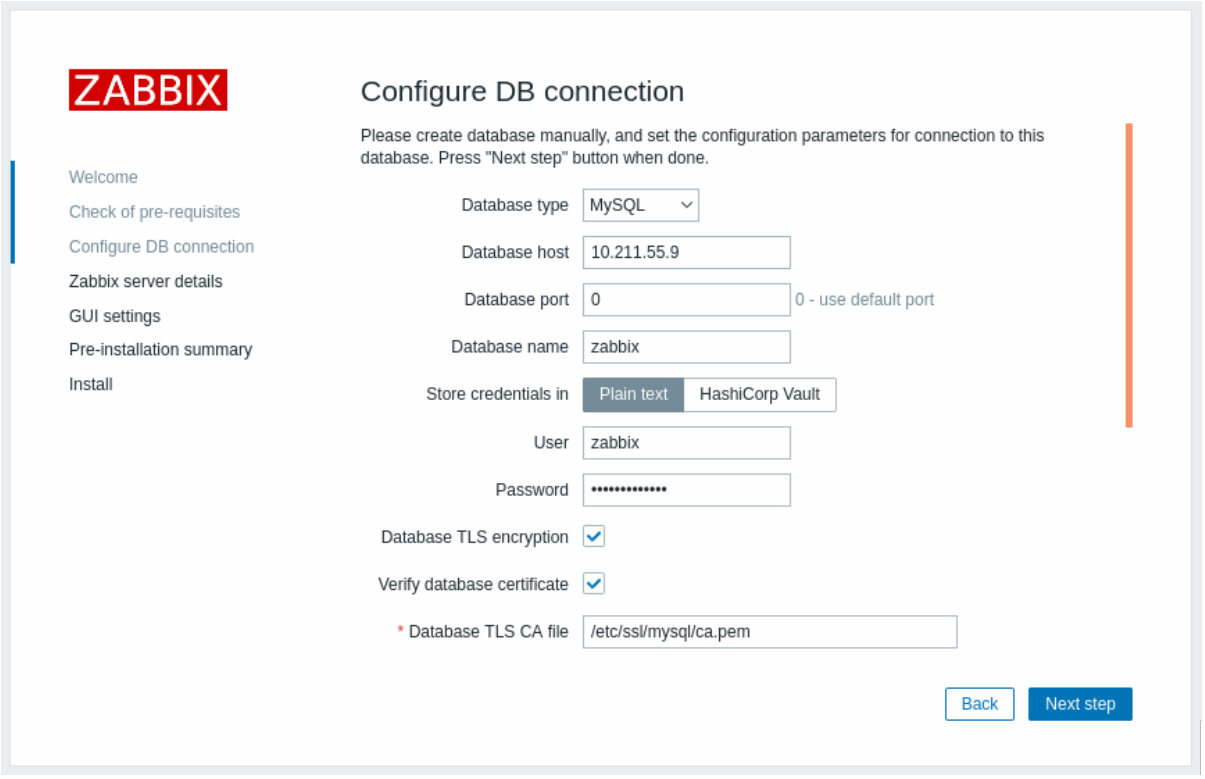
Kopieren Sie die erforderliche MySQL-CA auf den Zabbix-Frontend-Server und weisen Sie die entsprechenden Berechtigungen zu, damit der Webserver diese Datei lesen kann.

Note:
Dieser Modus funktioniert unter RHEL 7 aufgrund älterer MySQL-Bibliotheken nicht.

Frontend

Um die Verschlüsselung mit Zertifikatsprüfung für Verbindungen zwischen dem Zabbix Frontend und der Datenbank zu aktivieren:

- Aktivieren Sie *Database TLS encryption* und *Verify database certificate*
- Geben Sie den Pfad zur Database-TLS-CA-Datei an



Alternativ kann dies in */etc/zabbix/web/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'] = '';
...
```

Zur Fehlerbehebung kann der Benutzer mit einem Befehlszeilenwerkzeug prüfen, 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 Zertifikatsprüfung für Verbindungen zwischen dem Zabbix Server und der Datenbank zu aktivieren, konfigurieren Sie */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

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
...
```

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 MySQL-Serverzertifikat sollte das Feld Common Name enthalten, das auf den FQDN gesetzt ist, da das Zabbix Frontend den DNS-Namen zur Kommunikation mit der Datenbank oder die IP-Adresse des Datenbank-Hosts verwendet.

Erstellen Sie einen MySQL-Benutzer:

```
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
```

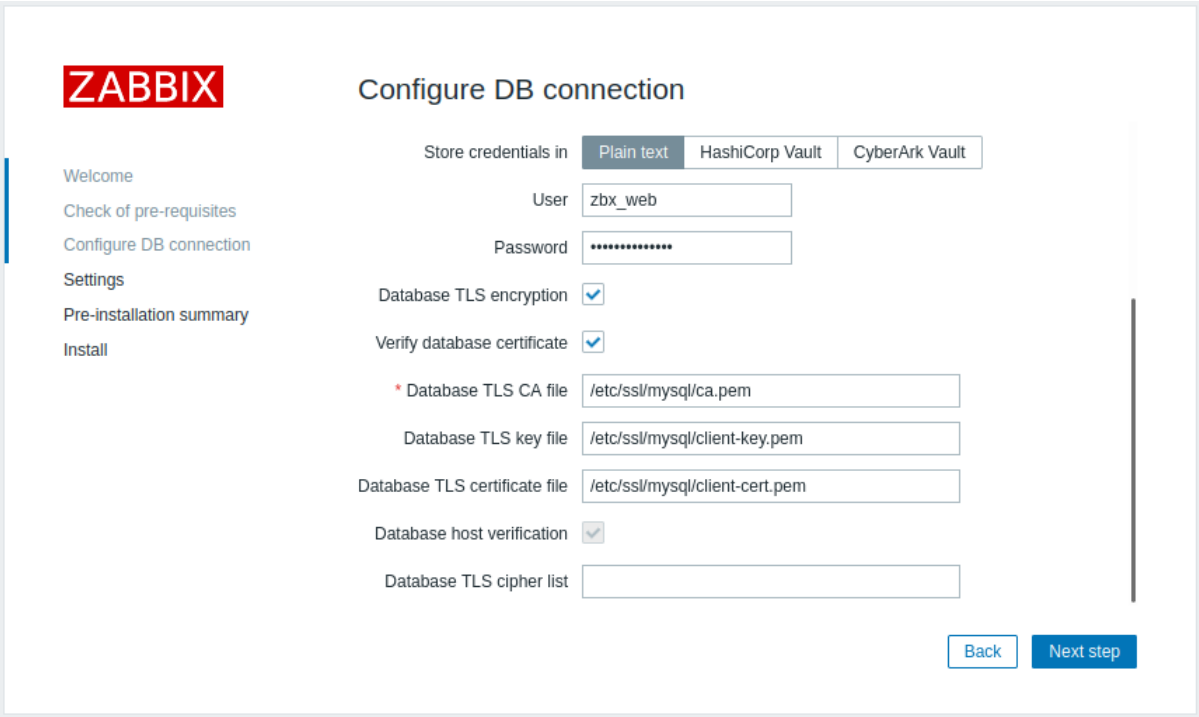
Frontend

So aktivieren Sie Verschlüsselung mit vollständiger Verifizierung für Verbindungen zwischen dem Zabbix Frontend und der Datenbank:

- Aktivieren Sie *Database TLS encryption* und *Verify database certificate*
- Geben Sie den Pfad zu *Database TLS key file* an
- Geben Sie den Pfad zu *Database TLS CA file* an
- Geben Sie den Pfad zu *Database TLS certificate file* an

Beachten Sie, dass *Database host verification* aktiviert und ausgegraut ist – dieser Schritt kann bei 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 unterstützt werden. Alternativ können die Chiffren explizit festgelegt werden, entsprechend den Anforderungen an die Chiffrenkonfiguration.



Alternativ kann dies in */etc/zabbix/web/zabbix.conf.php* festgelegt werden:

```
...
// Wird für eine TLS-Verbindung mit streng definierter Chiffrenliste verwendet.
$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
...
// oder
...
// Wird für eine TLS-Verbindung ohne definierte Chiffrenliste verwendet - Auswahl durch den 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 Verifizierung für Verbindungen zwischen dem Zabbix Server und der Datenbank zu aktivieren, konfigurieren Sie */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

2 PostgreSQL-Verschlüsselungskonfiguration

Übersicht

Dieser Abschnitt enthält mehrere Beispiele für die Konfiguration der Verschlüsselung für CentOS 8.2 und PostgreSQL 13.

> **Note:**
> Die Verbindung zwischen dem Zabbix Frontend und PostgreSQL kann nicht verschlüsselt werden (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 Remote-Verbindungen über das Netzwerk möglich sind, muss das Lauschen auf der tatsächlichen Netzwerkschnittstelle erlaubt werden.

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'
...
```

Für die Zugriffskontrolle passen Sie */var/lib/pgsql/13/data/pg_hba.conf* an:

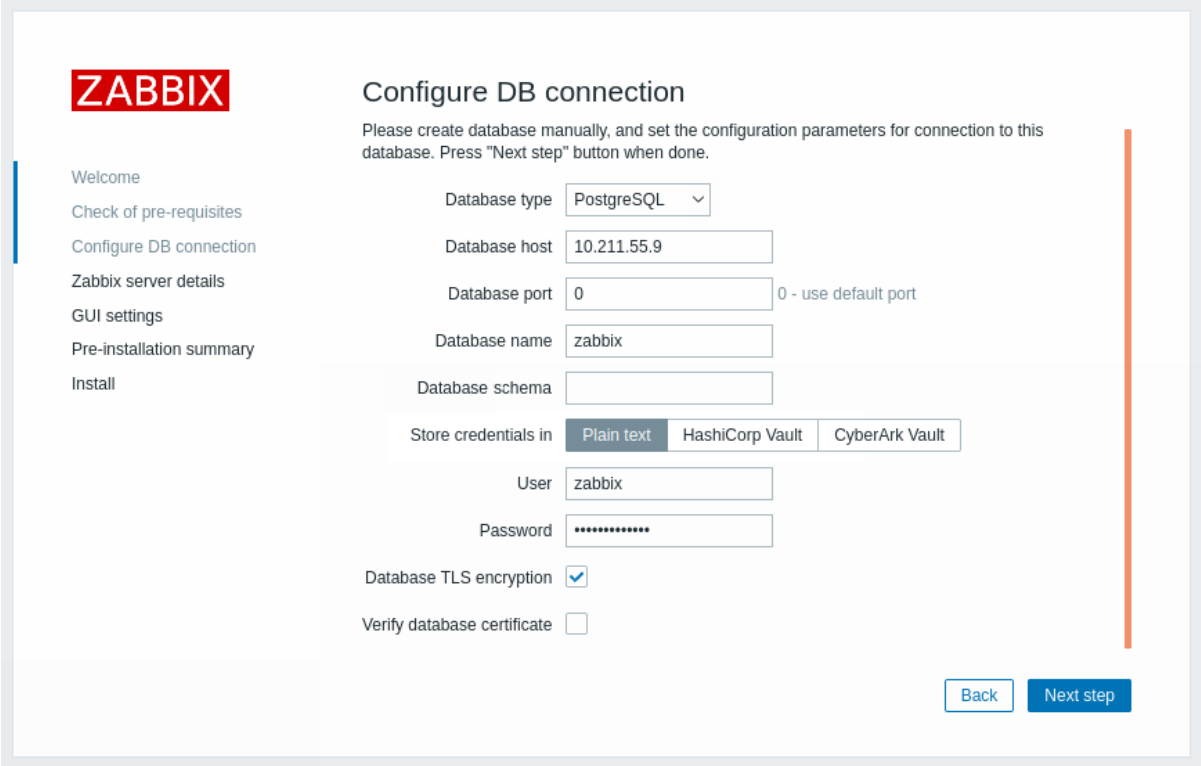
```
...
### require
hostssl all all 0.0.0.0/0 md5

### 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
...
```

Ausschließliche Transportverschlüsselung

Frontend

Um eine reine 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 für Verbindungen zwischen Server und Datenbank eine reine Transportverschlüsselung zu aktivieren, konfigurieren Sie */etc/zabbix/zabbix_server.conf*:

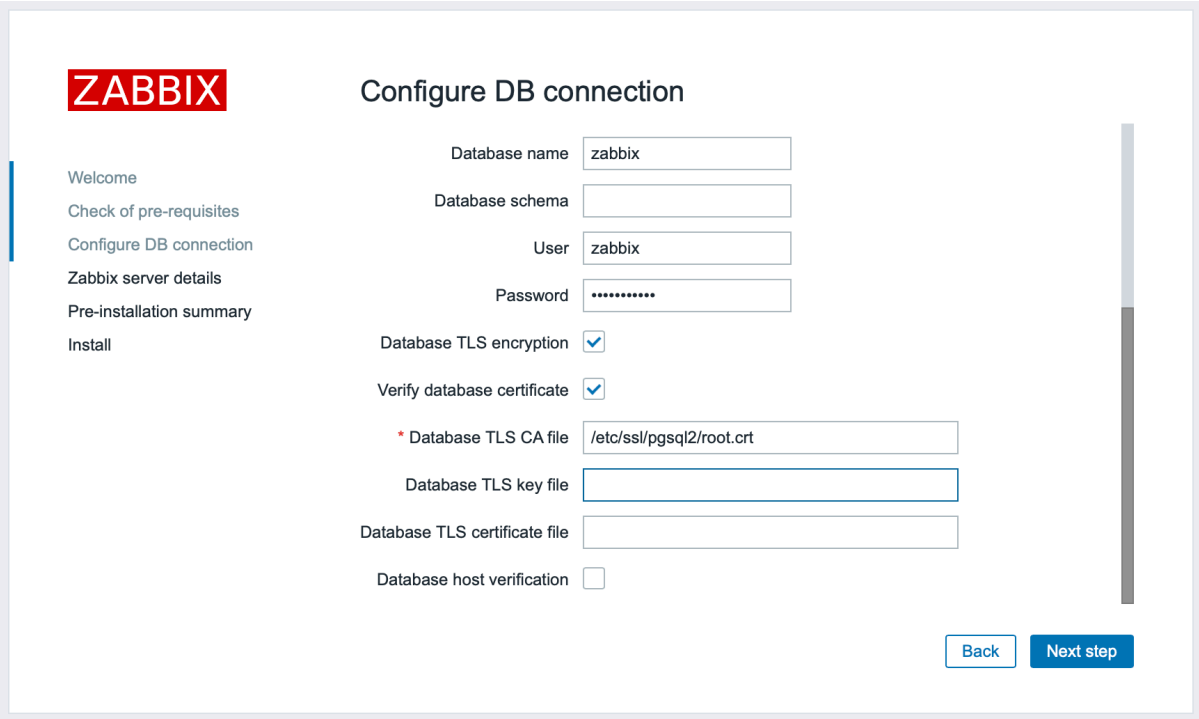
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Verschlüsselung mit Verifizierung der Zertifizierungsstelle

Frontend

So aktivieren Sie die Verschlüsselung mit Überprüfung der Zertifizierungsstelle 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 CA file* an



Alternativ kann dies in */etc/zabbix/web/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 Zertifikatsprüfung für Verbindungen zwischen dem Zabbix Server und der Datenbank zu aktivieren, konfigurieren Sie */etc/zabbix/zabbix_server.conf:*

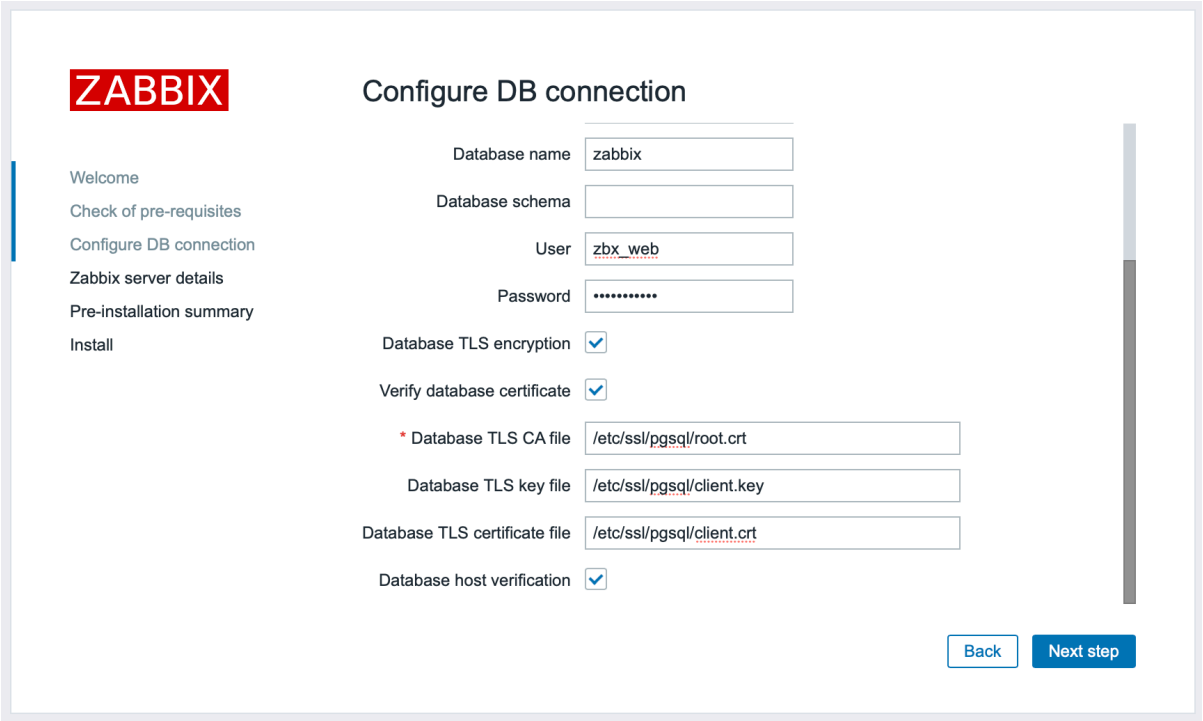
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Verschlüsselung mit vollständiger Verifizierung

Frontend

Um Verschlüsselung mit Zertifikat und Überprüfung der Identität des Datenbank-Hosts für Verbindungen zwischen Zabbix Frontend und der Datenbank zu aktivieren:

- 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
- Aktivieren Sie *Database host verification*



Alternativ kann dies in */etc/zabbix/web/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 Zabbix-Server und der Datenbank zu aktivieren, konfigurieren Sie */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

6 Sichere Verbindung zum Frontend

Überblick

Dieser Abschnitt enthält Schritte zur Einrichtung von Zabbix sowie Konfigurationsbeispiele für sichere TLS-Verbindungen zwischen dem Zabbix Frontend und dem Zabbix Server.

Konfiguration

Standardmäßig ist die Kommunikation zwischen Zabbix Frontend und Zabbix Server unverschlüsselt. Für mehr Sicherheit aktivieren Sie TLS auf beiden Seiten. Nachfolgend finden Sie ein Beispiel für die einfachste Möglichkeit, dies zu tun.

1. Zertifikate und Schlüssel erzeugen.

Erstellen Sie ein Arbeitsverzeichnis:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Erstellen Sie ein CA-Zertifikat (passen Sie den Wert `MyZabbixCA` so an, dass er dem tatsächlichen Common Name entspricht):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Erzeugen Sie einen privaten Schlüssel und ein Zertifikat für den Zabbix Server (passen Sie den Wert `zabbix-server.example.com` so an, dass er dem tatsächlichen Common Name entspricht):

```
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
```

Erzeugen Sie einen privaten Schlüssel und ein Zertifikat für das Zabbix Frontend (passen Sie den Wert `zabbix-frontend-node` so an, dass er dem tatsächlichen Common Name entspricht):

```
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. Korrekte Berechtigungen festlegen.

Für den Zabbix Server (passen Sie Eigentümer/Gruppe entsprechend dem Zabbix-Server-Daemon-Benutzer 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/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. Zabbix Server konfigurieren.

Fügen Sie in `zabbix_server.conf` Folgendes hinzu:

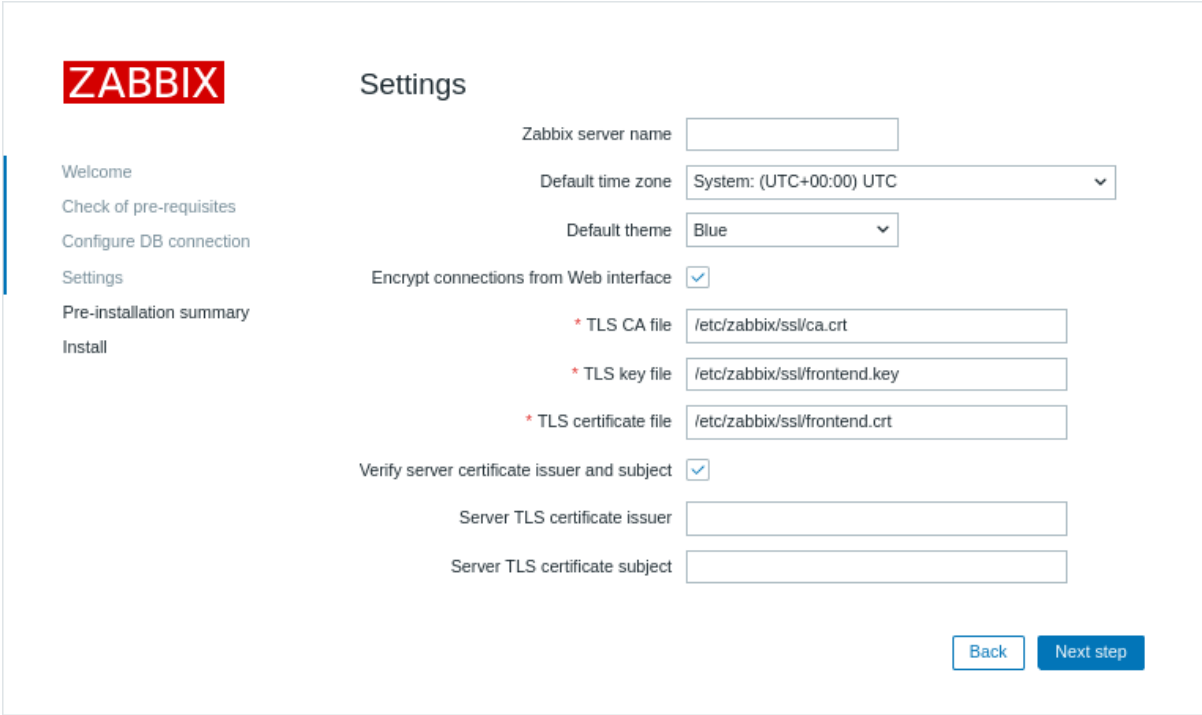
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Optional:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

Starten Sie dann den Server neu:

```
sudo systemctl restart zabbix-server
```

4. Zabbix Frontend konfigurieren.

Aktivieren Sie während der Installation der Weboberfläche die Option *Encrypt connections from Web interface* (und bei Bedarf die Option *Verify server certificate issuer and subject*) und füllen Sie die Felder *TLS CA file*, *TLS key file*, *TLS certificate file* (sowie bei Bedarf *Server TLS certificate issuer* und *Server TLS certificate subject*) aus:



Parameter	Beschreibung
TLS CA file	Geben Sie den vollständigen Pfad zur Zertifizierungsstellen-(CA-)Zertifikatsdatei an, die zur Überprüfung des Serverzertifikats verwendet wird.
TLS key file	Geben Sie den vollständigen Pfad zur privaten Schlüsseldatei des Clients an, die dem Clientzertifikat entspricht.
TLS certificate file	Geben Sie den vollständigen Pfad zur Clientzertifikatsdatei an, wenn eine gegenseitige TLS-Authentifizierung erforderlich ist.
Server TLS certificate issuer	Geben Sie einen Distinguished Name (DN) des Ausstellers an, der mit dem Serverzertifikat abgeglichen werden soll.
Server TLS certificate subject	Geben Sie einen Distinguished Name (DN) des Subjekts an, der mit dem Serverzertifikat abgeglichen werden soll.

Bearbeiten Sie bei bestehenden Installationen die folgenden Felder in `zabbix.conf.php`:

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// 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 Zabbix-Server-Protokolldatei keine Fehlermeldungen vorhanden sind:

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 Einrichtung von TimescaleDB

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 offiziellen PostgreSQL-Repository.

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 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 Datenpunkt-Verlaufszeitraums
- Überschreiben des Datenpunkt-Trendzeitraums

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-Hypertabellen 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 nicht im Frontend aktivieren.

Note:
Benutzern wird empfohlen, sich vor der Verwendung der Komprimierung mit der Komprimierung in der TimescaleDB-Dokumentation vertraut zu machen.

Beachten Sie, dass es bestimmte durch die Komprimierung bedingte Einschränkungen gibt, insbesondere:

- Änderungen an komprimierten Chunks (Inserts, Deletes, Updates) sind nicht zulässig
- Schemaänderungen für komprimierte Tabellen sind nicht zulässig.

Die Komprimierungseinstellungen können im Block *Komprimierung von Verlauf, Trends und Audit-Log* im Abschnitt *Administration > Housekeeping* des Zabbix-Frontend 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 × `HousekeepingFrequency` Stunden wirksam (festgelegt in zabbix_server.conf) Nach dem Deaktivieren der Komprimierung werden neue Chunks, die in den Komprimierungszeitraum fallen, nicht komprimiert. Bereits zuvor 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 wird 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 Performance bei Trendaktualisierungen sollten Sie möglicherweise das „chunk_time_interval“ für die Tabellen `trends` und `trends_uint` von 30 Tagen auf 7 Tage oder weniger reduzieren, abhängig davon, wie viele Datenpunkte Trends

verwenden. Der Zweck dieser Einstellung besteht darin, die Best Practices von TimescaleDB einzuhalten und sicherzustellen, dass die Chunk-Größe innerhalb der im System verfügbaren Ressourcen bleibt.

8 Elasticsearch-Einrichtung

Zabbix kann Verlaufsdaten alternativ zu einer relationalen Datenbank in Elasticsearch speichern.

> **Attention:**
> Die Unterstützung von Elasticsearch ist derzeit experimentell.

Diese Anleitung behandelt die Einrichtung für Elasticsearch 7.X. Wenn Sie eine andere Version verwenden, funktioniert ein Teil der Funktionalität möglicherweise nicht wie vorgesehen.

Die Einrichtung umfasst das Erstellen eines Elasticsearch-Speicherorts für jeden Werttyp, das Einrichten der Vorverarbeitung (falls erforderlich) und das Verbinden von Zabbix mit Ihrer Elasticsearch-Instanz.

Elasticsearch kann die folgenden Werttypen speichern:

Werttyp des Datenpunkts	Datenbanktabelle	Elasticsearch-Typ
Numerisch (unsigned)	history_uint	uint
Numerisch (float)	history	dbl
Zeichen	history_str	str
Log	history_log	log
Text	history_text	text
Binär	history_bin	von Zabbix nicht unterstützt
JSON	history_json	json

Wichtige Hinweise

- Elasticsearch erfordert libcurl. Siehe Anforderungen für Details.
- Der housekeeper löscht keine Daten aus Elasticsearch.
- Wenn alle Verlaufsdaten in Elasticsearch gespeichert werden, werden Trends **nicht** berechnet oder in der Datenbank gespeichert. Ziehen Sie eine Verlängerung der Speicherdauer für Verlaufsdaten in Betracht.
- Wenn Elasticsearch verwendet wird, sind Bereichsabfragen zum Abrufen von Werten aus der Datenbank durch den Zeitstempel des Datenspeicherzeitraums begrenzt.
- Elasticsearch wird für Zabbix Proxy nicht unterstützt; bitte verwenden Sie stattdessen SQLite.

Falls 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}
```

Verlauf in datumsbasierten Indizes speichern

Anstatt alle Verlaufsdaten in einen einzelnen Index (z. B. `uint`) zu schreiben, kann Elasticsearch diese Daten auf mehrere datumsbasierte Indizes verteilen (z. B. `uint-2026-01-01`, `uint-2026-01-02`). Dadurch wird es einfacher, das Datenvolumen und die Aufbewahrung im Zeitverlauf zu verwalten.

Um dies zu aktivieren, müssen Sie:

- eine Index-Vorlage für jeden Werttyp erstellen, den Sie speichern möchten — diese teilt Elasticsearch mit, welches Mapping angewendet werden soll, wenn automatisch ein neuer datumsbasierter Index erstellt wird.
- eine Ingest-Pipeline für jeden Werttyp erstellen — sie verarbeitet jeden eingehenden Wert und leitet ihn an den richtigen datumsbasierten Index weiter.
- den Parameter `HistoryStorageDateIndex` in der Zabbix-Server-Konfigurationsdatei konfigurieren — dadurch wird das Speichern von Werten in mehreren datumsbasierten Indizes aktiviert.

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"
         }
      }
   ]
}'
```

Konfiguration des Zabbix Server

Legen Sie in Ihrer Zabbix-Server-Konfigurationsdatei (`zabbix_server.conf`) die folgenden Parameter fest:

- `HistoryStorageURL` - die URL Ihrer Elasticsearch-Instanz.
- `HistoryStorageTypes` - kommagetrennte Liste der Werttypen, die in Elasticsearch gespeichert werden sollen.

Zum Beispiel, um Werte der Typen *Character*, *Log*, *Text* und *JSON* in Elasticsearch zu speichern (während *Numeric*-Werte in einer Datenbank verbleiben):

```
HistoryStorageURL=http://localhost:9200
HistoryStorageTypes=str,log,text,json
```

Wenn Sie datumsbasierte Indizes für **alle** in Elasticsearch gespeicherten Werte verwenden, setzen Sie zusätzlich den Parameter `HistoryStorageDateIndex`:

```
HistoryStorageDateIndex=1
```

Starten Sie nach den Änderungen den Zabbix Server neu:

```
systemctl restart zabbix-server
```

Konfigurieren des Zabbix Frontend

Deklarieren Sie in Ihrer Zabbix-Frontend-Konfigurationsdatei (`zabbix.conf.php`) `$HISTORY` als globale Variable und setzen Sie

die Werte `url` und `types` so, dass sie mit der Server-Konfiguration übereinstimmen:

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;

$HISTORY['url']   = 'http://localhost:9200';
$HISTORY['types'] = ['str', 'log', 'text', 'json'];
```

Fehlerbehebung

Die folgenden Schritte können Ihnen bei der Fehlerbehebung von Problemen mit Ihrer Elasticsearch-Einrichtung helfen:

1. 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 dies 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": {} }
```

2. 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.

3. Prüfen Sie, ob sich Shards in einem Fehlerzustand befinden; ein Neustart von Elasticsearch kann dies möglicherweise beheben.
4. Vergewissern Sie sich, dass Ihre Elasticsearch-Konfiguration den Zugriff vom Zabbix Server und Zabbix Frontend erlaubt.
5. Verwenden Sie den Zabbix-Server-Konfigurationsparameter `LogSlowQueries`, um langsame Abfragen zu identifizieren.
6. Prüfen Sie die Elasticsearch-Protokolle auf Fehler.
7. Wenn Sie Ihre Elasticsearch-Einrichtung zurücksetzen und von vorn 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/*"
```

9 Distributionsspezifische Hinweise zur Einrichtung von Nginx für Zabbix

RHEL

Unter RHEL 8, 9 und 10 ist Nginx in AppStream enthalten – keine zusätzlichen Repositories erforderlich. Aktivieren und installieren Sie es einfach mit:

```
dnf module list nginx
dnf -y install nginx
```

Unter RHEL 7 kann Nginx über das CentOS 7 Extras-Repository (bei Verwendung von CentOS) installiert werden mit:

```
yum -y install nginx
```

Wenn Sie die allerneuesten Upstream-Builds bevorzugen, konfigurieren Sie das offizielle NGINX-Repository unter `/etc/yum.repos.d/nginx.`

```
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
```

10 Agent als root ausführen

Seit Zabbix **5.0.0** enthält die systemd-Service-Datei für den Zabbix Agent in den offiziellen Paketen ausdrücklich die Direktiven `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-Service-Datei 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 dann den folgenden Inhalt hinzu:

```
[Service]
User=root
Group=root
```

Laden Sie die Daemons neu und starten Sie den Dienst zabbix-agent neu:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Für den **Zabbix Agent** wird dadurch die Funktionalität zur Konfiguration des Benutzers in der Datei `zabbix_agentd.conf` wieder aktiviert. Nun müssen Sie die Konfigurationsparameter `User=root` und `AllowRoot=1` in der Konfigurationsdatei des Agent setzen.

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** legt dies den Benutzer, unter dem er ausgeführt wird, vollständig fest. Es sind keine zusätzlichen Änderungen erforderlich.

11 Zabbix Agent unter Microsoft Windows

Konfiguration des Agent

Beide Generationen von Zabbix-Agenten werden als Windows-Dienst ausgeführt. Für Zabbix Agent 2 ersetzen Sie 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 Standard-Konfigurationsdatei, die sich im selben Verzeichnis wie die Agent-Binärdatei befindet;
- eine in der Befehlszeile angegebene Konfigurationsdatei.

Bei mehreren Instanzen muss jede Agent-Instanz ihre eigene Konfigurationsdatei haben (eine der Instanzen kann die Standard-Konfigurationsdatei verwenden).

Eine Beispiel-Konfigurationsdatei ist im Zabbix-Quellarchiv verfügbar als:

- `conf/zabbix_agentd.conf` für Zabbix-Agent;
- `conf/zabbix_agent2.conf` für Zabbix agent2.

Wenn Sie Zabbix-Agent/Agent 2 für Windows aus einem Archiv als Dienst installieren möchten, ohne die Konfigurationsdatei ausdrücklich anzugeben, dann sollten vor der Installation des Agent:

- `conf/zabbix_agentd.conf` manuell in das Verzeichnis kopiert werden, in dem zabbix_agentd.exe installiert wird;
- `conf/zabbix_agent2.conf` und das Verzeichnis `conf/zabbix_agent2.d` manuell in das Verzeichnis kopiert werden, in dem zabbix_agent2.exe installiert wird.

Weitere Informationen zur Konfiguration des Zabbix-Windows-Agent finden Sie in den Optionen der Konfigurationsdatei.

Parameter Hostname

Um aktive Prüfungen auf einem Host durchzuführen, muss für den Zabbix Agent der Hostname definiert sein. Außerdem muss der auf der Agent-Seite gesetzte Hostname-Wert exakt mit dem im Frontend für den Host konfigurierten ”Host- Namen” übereinstimmen.

Der Hostname-Wert auf der Agent-Seite kann entweder über den Parameter **Hostname** oder **HostnameItem** in der Konfigurationsdatei des Agent definiert werden - oder es werden die Standardwerte verwendet, wenn keiner dieser Parameter angegeben ist.

Der Standardwert für den Parameter **HostnameItem** ist der vom Agent-Schlüssel ”system.hostname” zurückgegebene Wert. Unter Windows gibt dieser das Ergebnis der Funktion gethostname() zurück, die Namespace-Provider abfragt, um den lokalen Hostnamen zu ermitteln. Wenn kein Namespace-Provider antwortet, wird der NetBIOS-Name zurückgegeben.

Der Standardwert für **Hostname** ist der vom Parameter HostnameItem zurückgegebene Wert. Das bedeutet, dass, wenn beide Parameter nicht angegeben sind, der tatsächliche Hostname der NetBIOS-Name des Hosts ist; der Zabbix Agent verwendet den NetBIOS-Hostnamen, um die Liste der aktiven Prüfungen vom Zabbix Server abzurufen und 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 nur in GROSSBUCHSTABEN vorliegt;
- *host* - Groß-/Kleinschreibung wird beachtet, gibt den vollständigen, tatsächlichen Windows-Hostnamen zurück (ohne Domain);
- *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 zurück (ohne den abschließenden Punkt).

Transform ermöglicht die Angabe einer zusätzlichen Transformationsregel für den Hostnamen:

- *none* (Standard) - die ursprüngliche Groß-/Kleinschreibung verwenden;
- *lower* - den Text in Kleinbuchstaben umwandeln.

Um also die Konfiguration der Datei zabbix_agentd.conf zu vereinfachen und zu vereinheitlichen, können drei verschiedene Ansätze verwendet werden:

1. Die Parameter **Hostname** oder **HostnameItem** nicht definieren, dann verwendet der Zabbix Agent den NetBIOS-Hostnamen als Hostnamen.
2. Den Parameter **Hostname** nicht definieren und **HostnameItem** wie folgt festlegen:
   HostnameItem=system.hostname[host] - damit der Zabbix Agent den vollständigen, tatsächlichen Windows-Hostnamen (unter Beachtung der Groß-/Kleinschreibung) als Hostnamen verwendet

HostnameItem=system.hostname[shorthost,lower] - damit der Zabbix Agent nur den Teil des Hostnamens vor dem ersten Punkt verwendet, in Kleinbuchstaben umgewandelt.
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, Stoppen und Deinstallieren des Windows-Dienstes verwendet wird. Wenn beispielsweise in der Zabbix-Agent-Konfigurationsdatei `Hostname=Windows_db_server` angegeben ist, wird der Agent als Windows-Dienst "`Zabbix Agent [Windows_db_server]`" installiert. Um daher für jede Zabbix-Agent-Instanz einen anderen Windows-Dienstnamen zu haben, muss jede Instanz einen anderen Hostnamen verwenden.

Installation des Agent als Windows-Dienst

Bevor Sie den Agent installieren, kopieren Sie `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 für alle Prüfungen im Zusammenhang mit laufenden 64-Bit-Prozessen eine 64-Bit-Version des Zabbix Agent erforderlich, damit sie 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 wie folgt als Dienste 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.

Agent starten

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 Konfigurations- datei 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
```

Agent stoppen

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 des Zabbix Agent mit der Standard- Konfigurationsdatei zu deinstallieren:

```
zabbix_agentd.exe --uninstall
```

Um eine einzelne Instanz des Zabbix Agent mit einer nicht standardmäßigen Konfigurationsdatei zu deinstallieren:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Um mehrere Instanzen des Zabbix Agent aus den 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 auf NUMA-Knoten verteilt sind. Wenn logische CPUs ungleichmäßig verteilt sind, sind CPU-Leistungsmetriken für einige CPUs möglicherweise nicht verfügbar. Wenn es beispielsweise 72 logische CPUs mit 2 NUMA-Knoten gibt, müssen beide Knoten jeweils 36 CPUs haben.

12 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.

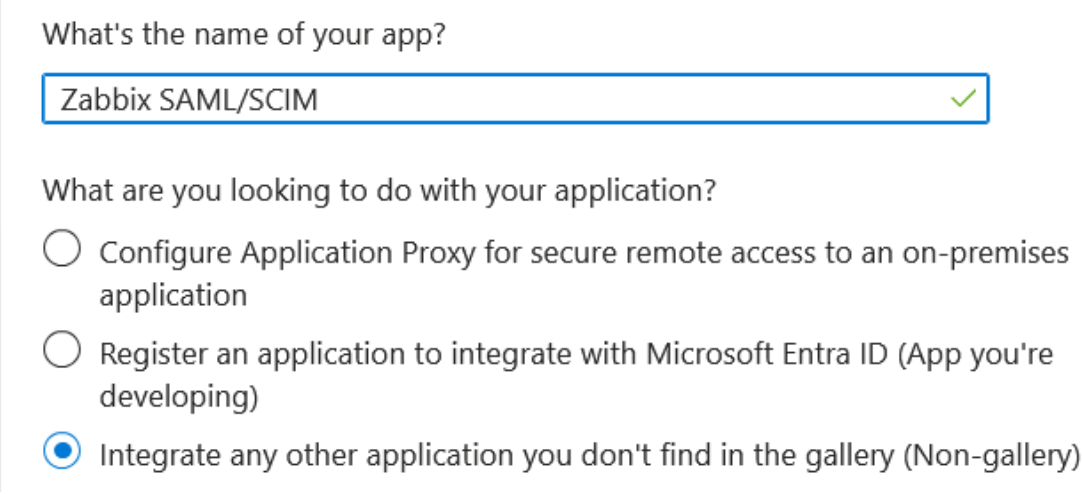
Microsoft Entra ID-Konfiguration

Anwendung erstellen

1. Melden Sie sich im Microsoft Entra Admin Center unter Microsoft Entra ID an. Zu Testzwecken können Sie in Microsoft Entra ID ein kostenloses Testkonto erstellen.

2. Wählen Sie im Microsoft Entra Admin Center *Applications -> Enterprise applications -> New application -> Create your own application* aus.

3. Geben Sie den Namen Ihrer App ein und wählen Sie die Option *Integrate any other application...* aus. Klicken Sie danach auf *Create*.

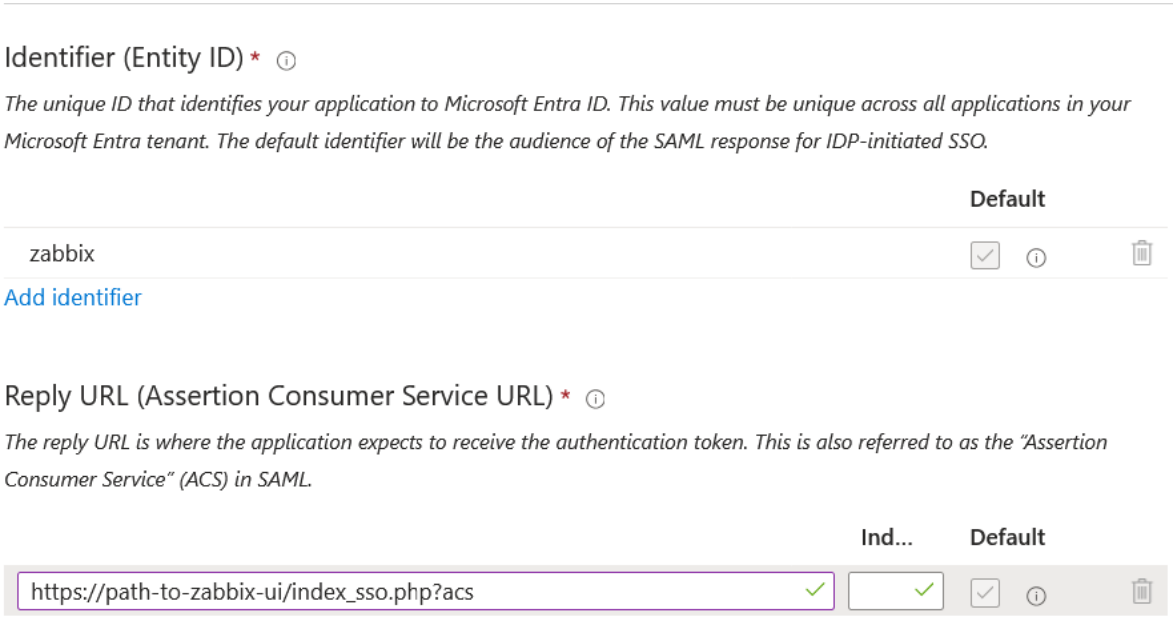


Einrichten von Single Sign-on

1. Gehen Sie auf der Anwendungsseite zu *Set up single sign on* und klicken Sie auf *Get started*. Wählen Sie dann *SAML* aus.

2. Bearbeiten Sie *Basic SAML Configuration*:

- Legen Sie in *Identifier (Entity ID)* einen eindeutigen Namen fest, um Ihre App gegenüber Microsoft Entra ID zu identifizieren, zum Beispiel `zabbix`;
- Legen Sie in *Reply URL (Assertion Consumer Service URL)* den Zabbix-Endpunkt für Single Sign-on fest: `https://<path-to-zabbix-u`



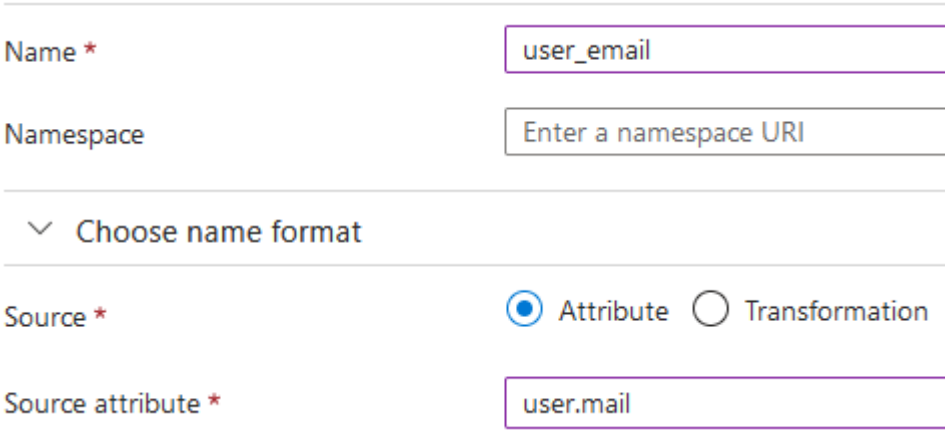
Beachten Sie, dass "https" erforderlich ist. Damit dies mit Zabbix funktioniert, muss in `conf/zabbix.conf.php` die folgende Zeile hinzugefügt werden:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

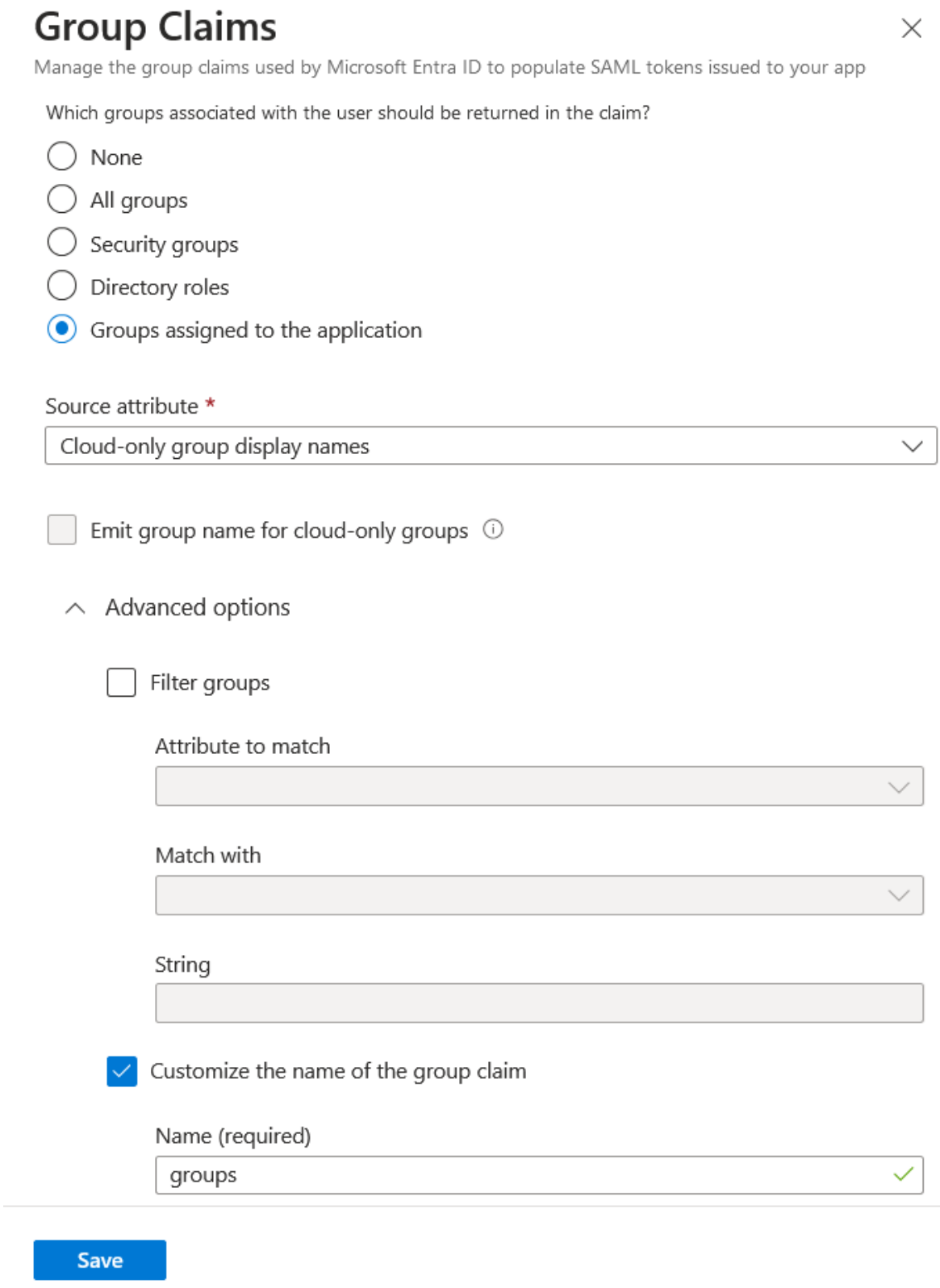
3. Bearbeiten Sie *Attributes & Claims*. Sie müssen alle Attribute hinzufügen, die Sie an Zabbix übergeben möchten (user_name, user_lastname, user_email, user_mobile, groups).

Die Attributnamen sind frei wählbar. Es können unterschiedliche Attributnamen verwendet werden, jedoch ist erforderlich, dass sie mit dem jeweiligen Feldwert in den Zabbix-SAML-Einstellungen übereinstimmen.

- Klicken Sie auf *Add new claim*, um ein Attribut hinzuzufügen:



- Klicken Sie auf *Add a group claim*, um ein Attribut zum Übergeben von Gruppen an Zabbix hinzuzufügen:



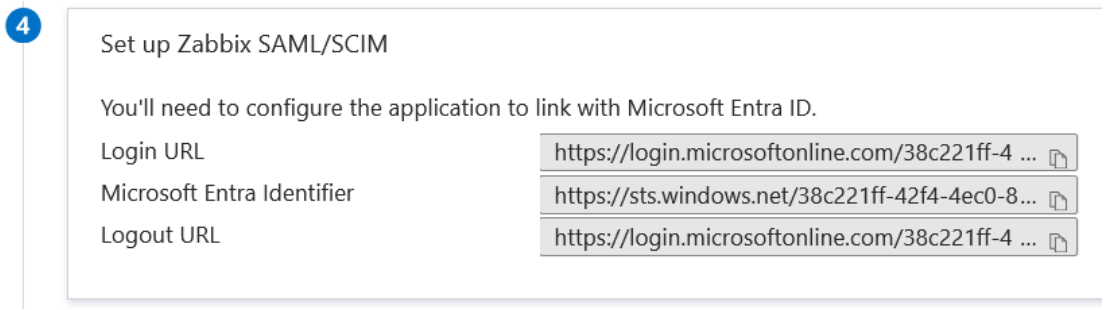
Bei diesem Claim ist es wichtig, dass die Gruppennamen (und nicht die Gruppen-IDs) über das ausgewählte *Source attribute* an Zabbix übergeben werden. Andernfalls funktioniert die JIT-Benutzerbereitstellung nicht korrekt.

4. Laden Sie unter *SAML Certificates* das von Entra ID bereitgestellte Base64-Zertifikat herunter und legen Sie es in `conf/certs` der Zabbix-Frontend-Installation ab.

Setzen Sie die Berechtigung 644 darauf, indem Sie Folgendes ausführen:

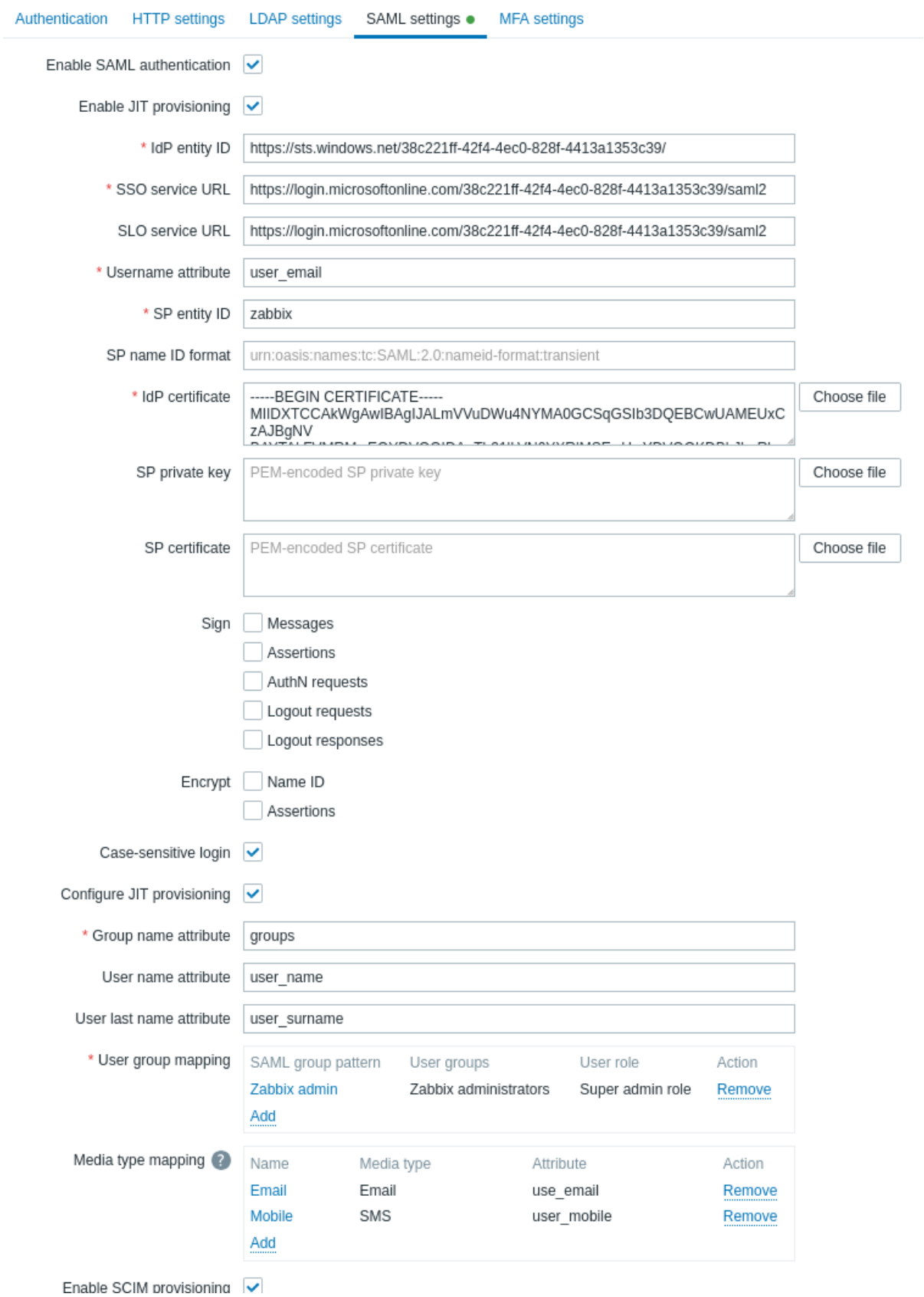
```
chmod 644 entra.cer
```

5. Verwenden Sie die Werte aus *Set up <your app name>* in Entra ID, um die Zabbix-SAML-Authentifizierung zu konfigurieren (siehe nächsten Abschnitt):



Zabbix-Konfiguration

1. Gehen Sie in Zabbix zu den SAML-Einstellungen und füllen Sie die Konfigurationsoptionen basierend auf der Entra ID-Konfiguration aus:



Zabbix-Feld	Einrichtungsfeld in Entra ID	Beispielwert
IdP entity ID	Microsoft Entra-Bezeichner	
SSO service URL	Anmelde-URL	
SLO service URL	Abmelde-URL	
SP entity ID	Bezeichner (Entity ID)	
Username attribute	Benutzerdefiniertes Attribut (Claim)	`user_email`
Group name attribute	Benutzerdefiniertes Attribut (Claim)	`groups`
User name attribute	Benutzerdefiniertes Attribut (Claim)	`user_name`

Zabbix-Feld	Einrichtungsfeld in Entra ID	Beispielwert
User last name attribute	Benutzerdefiniertes Attribut (Claim)	`user_lastname`

2. Fügen Sie das von Entra ID bereitgestellte Base64-Zertifikat hinzu.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in zabbix.conf.php gesetzt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen — es sind keine Dateien im Dateisystem erforderlich.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in zabbix.conf.php gesetzt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in ui/conf/certs oder in dem in zabbix.conf.php konfigurierten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank. Beachten Sie, dass, wenn `$SSO['CERT_STORAGE']` nicht gesetzt oder auskommentiert ist, Dateispeicherung angenommen wird und Zertifikate aus ui/conf/certs gelesen werden.

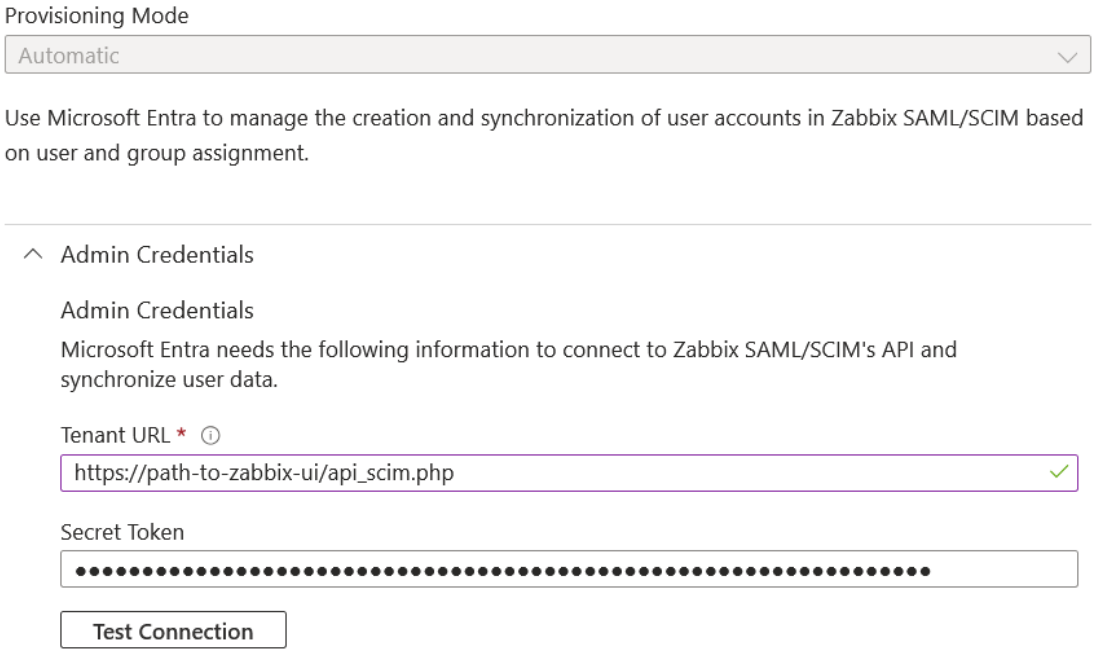
Es ist außerdem erforderlich, die Benutzergruppen-Zuordnung zu konfigurieren. Die Medien-Zuordnung ist optional.

3. Klicken Sie auf die Schaltfläche *Update*, um diese Einstellungen zu speichern.

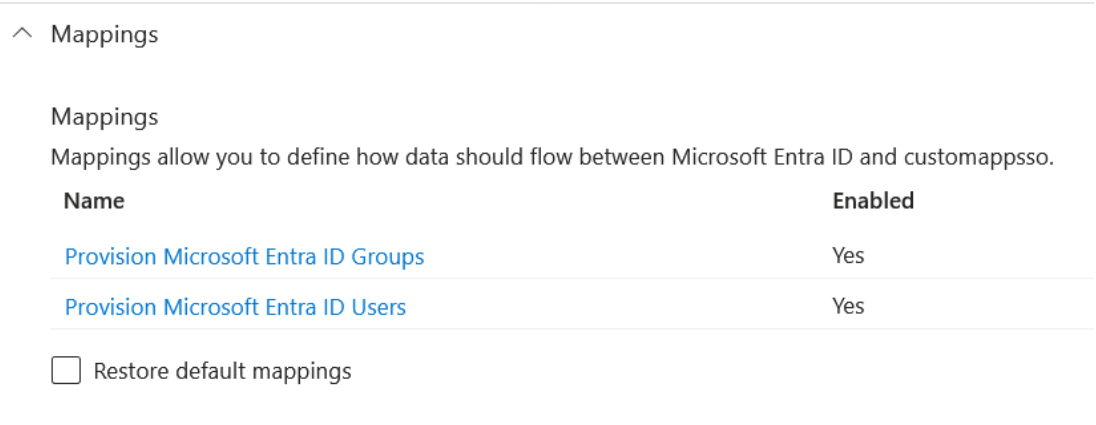
SCIM-Benutzerbereitstellung

1. Öffnen Sie auf der Anwendungsseite in Entra ID im Hauptmenü die Seite *Provisioning*. Klicken Sie auf *Get started* und wählen Sie dann den Modus für die automatische Bereitstellung aus:

- Setzen Sie in *Tenant URL* den folgenden Wert: `https://<path-to-zabbix-ui>/api_scim.php`
- Geben Sie in *Secret token* ein Zabbix-API-Token mit Super-Admin-Berechtigungen ein.
- Klicken Sie auf *Test connection*, um zu prüfen, ob die Verbindung hergestellt wurde.

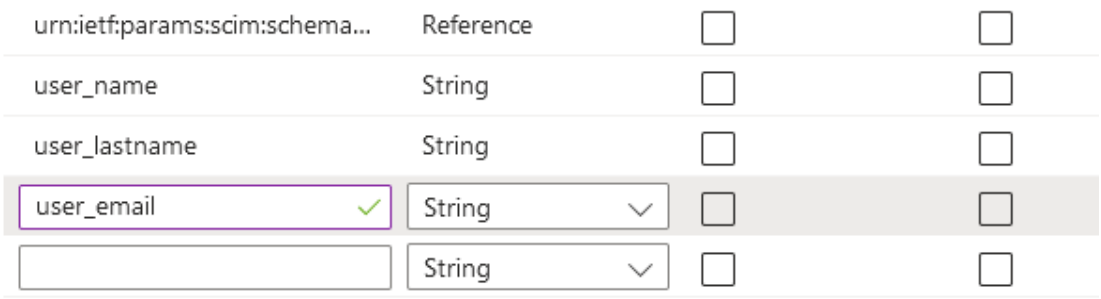


2. Nun können Sie alle Attribute hinzufügen, die per SCIM an Zabbix übergeben werden. Klicken Sie dazu auf *Mappings* und dann auf *Provision Microsoft Entra ID Users*.



Aktivieren Sie unten in der Liste *Attribute Mapping* die Option *Show advanced options* und klicken Sie dann auf *Edit attribute list for customappsso*.

Fügen Sie unten in der Attributliste eigene Attribute vom Typ 'String' hinzu:



Speichern Sie die Liste.

3. Nun können Sie Zuordnungen für die hinzugefügten Attribute erstellen. Klicken Sie unten in der Liste *Attribute Mapping* 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.

Gehen Sie dazu zum *Microsoft Entra admin center* und fügen Sie dann Benutzer/Gruppen auf den entsprechenden Seiten *Users* und *Groups* hinzu.

5. Wenn Benutzer und Gruppen in Entra ID erstellt wurden, können Sie zum Menü *Users and groups* Ihrer Anwendung wechseln und sie zur App hinzufügen.

6. Gehen Sie zum Menü *Provisioning* Ihrer Anwendung und klicken Sie auf *Start provisioning*, damit Benutzer in Zabbix bereitgestellt werden.

Beachten Sie, dass die Users-PATCH-Anfrage in Entra ID keine Änderungen an Medien unterstützt.

Signierung von Authentifizierungsanfragen

Es ist möglich, Entra ID so zu konfigurieren, dass die Signatur signierter Authentifizierungsanfragen validiert 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 Konfiguration des Zabbix Frontend, indem Sie Folgendes hinzufügen:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Fehlerbehebung

Bei MicroSoft Edge-Browsern können Authentifizierungsprobleme auftreten, wenn ein Benutzer, der versucht, sich über SAML bei Zabbix anzumelden, bereits mit dem MicroSoft Edge-Profil angemeldet ist. Ein Hinweis auf ein solches Problem ist, dass sich der Benutzer bei Verwendung von MicroSoft Edge im privaten Modus möglicherweise bei Zabbix anmelden kann.

Um Authentifizierungsprobleme in diesem Fall zu vermeiden, kann es erforderlich sein, `requestedAuthnContext` in der Zabbix Frontend-Konfigurationsdatei (*zabbix.conf.php*) auf "false" zu setzen.

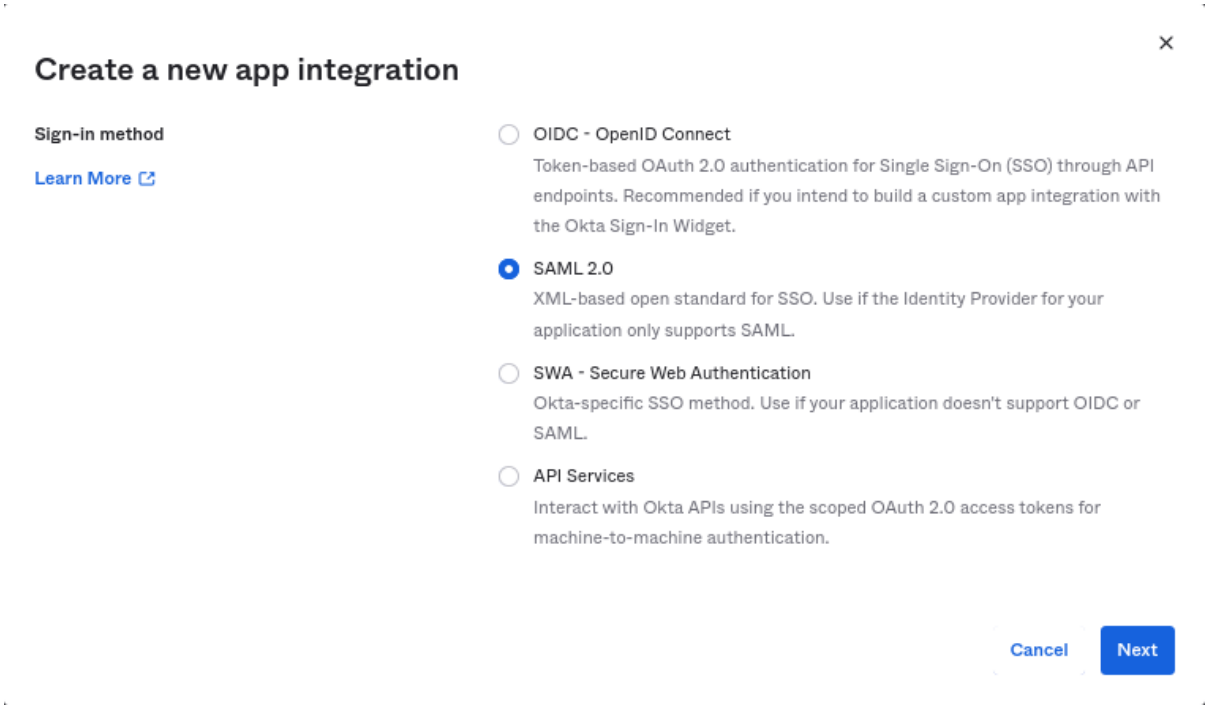
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 SAML-Einrichtung mit Okta

Dieser Abschnitt enthält Richtlinien für die Konfiguration von Okta, um die SAML-2.0-Authentifizierung und die Benutzerbereitstellung für Zabbix zu aktivieren.

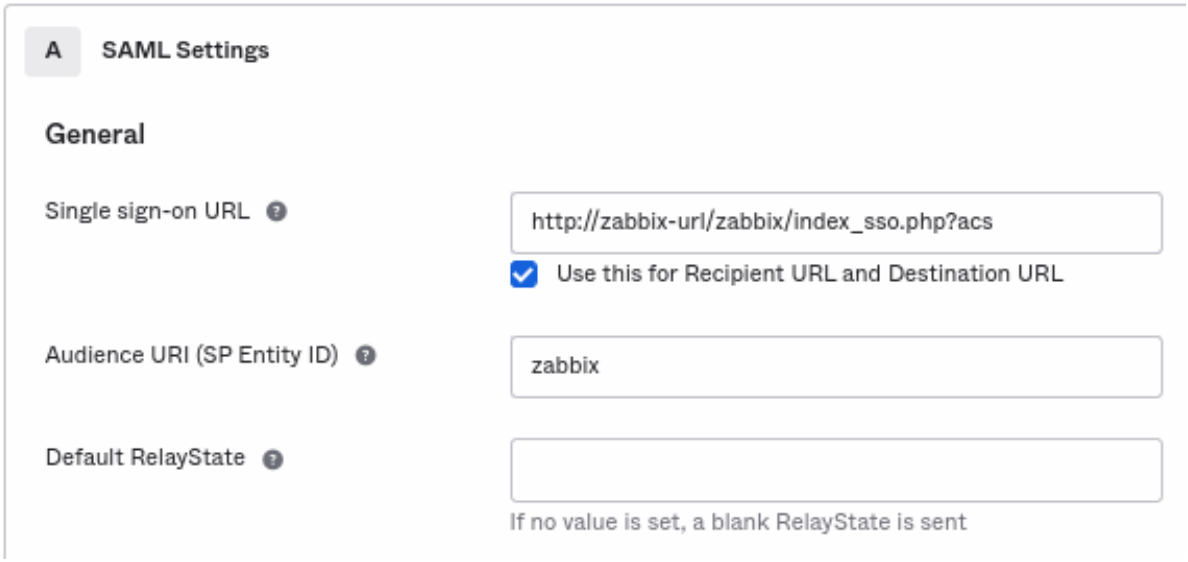
Okta-Konfiguration

1. Gehen Sie zu https://developer.okta.com/signup/ und registrieren/melden Sie sich bei Ihrem Konto an.
2. Navigieren Sie in der Okta-Weboberfläche zu *Applications* → *Applications*.
3. Klicken Sie auf *Create App Integration*.

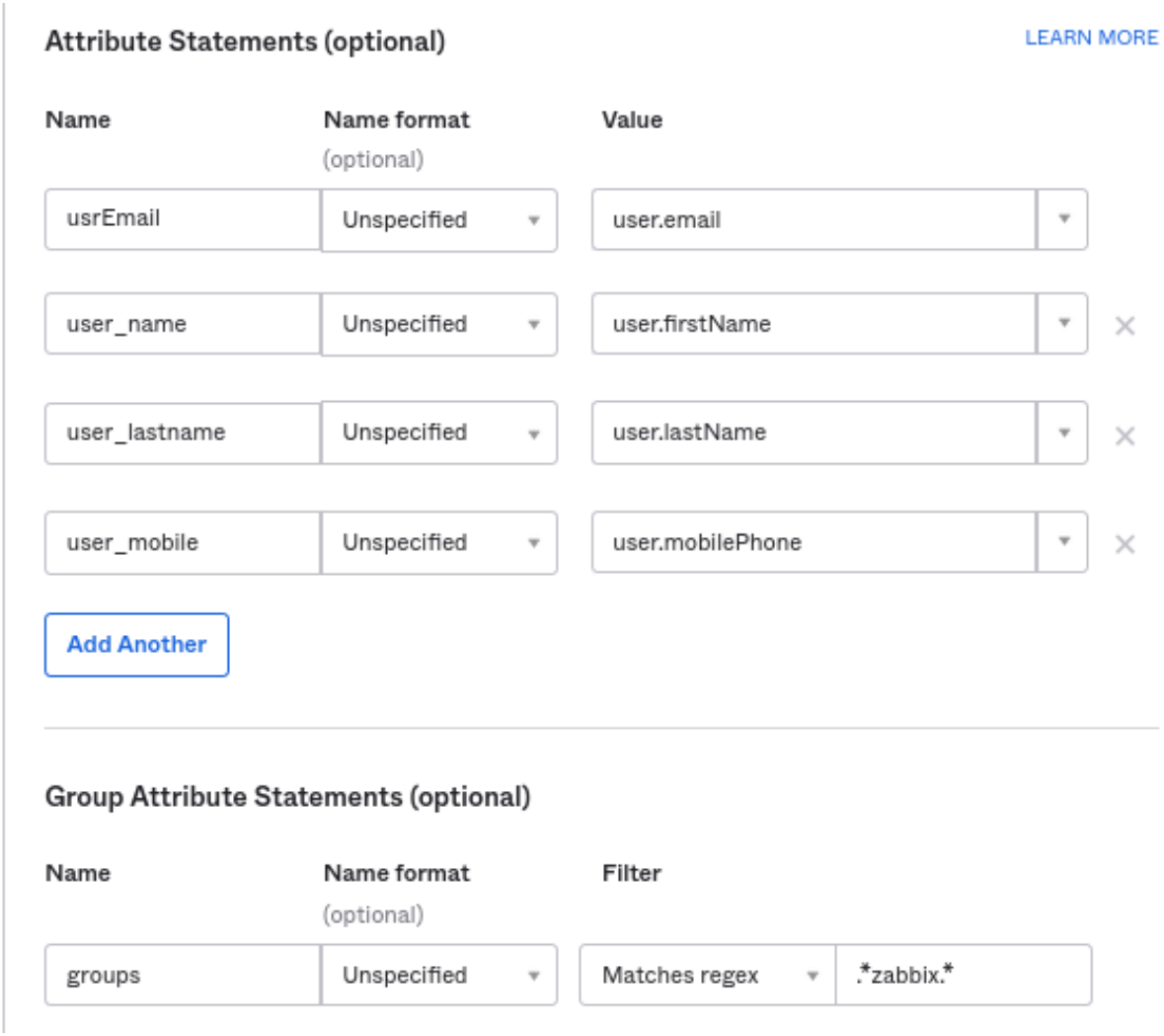


Wählen Sie „SAML 2.0" als Anmeldemethode aus und klicken Sie auf *Next*.

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*.



- Fügen Sie unter **General** Folgendes hinzu:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs`
    Beachten Sie die Verwendung von „http“ und nicht von „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`
    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 entsprechend Ihren Anforderungen aus.
- Fügen Sie unter **Attribute Statements/Group Attribute Statements** Folgendes hinzu:



Diese Attributanweisungen werden in die mit Zabbix geteilten SAML-Assertions eingefügt.

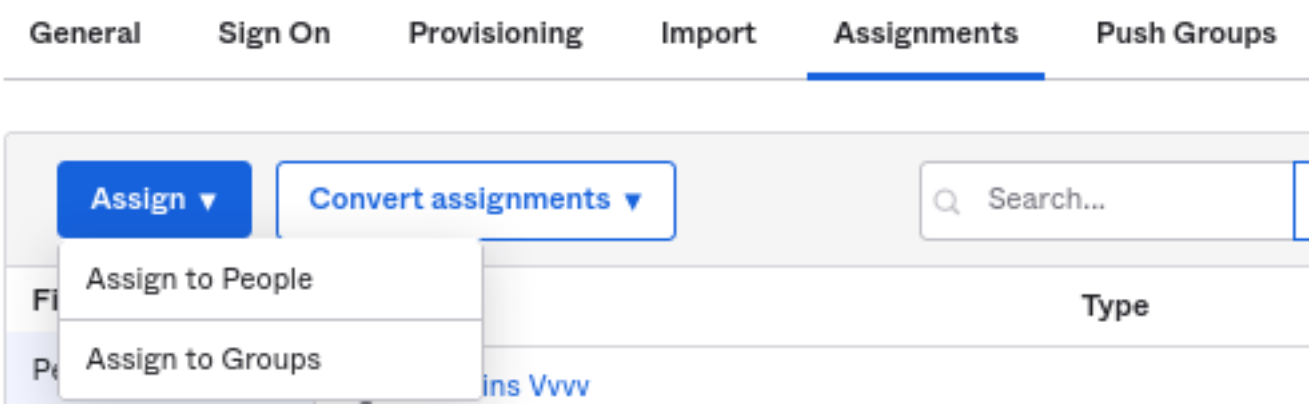
Die hier verwendeten Attributnamen sind beliebige Beispiele. Sie können andere Attributnamen verwenden, diese müssen jedoch mit dem jeweiligen Feldwert in den Zabbix-SAML-Einstellungen übereinstimmen.

Wenn Sie die SAML-Anmeldung bei Zabbix *ohne* JIT-Benutzerbereitstellung 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 dann 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 auf der nächsten Registerkarte „I'm a software vendor. I'd like to integrate my app with Okta“ aus und klicken Sie auf *Finish*.

7. Navigieren Sie zur Registerkarte „Assignments“ der neu erstellten Anwendung und klicken Sie auf die Schaltfläche *Assign*; wählen Sie dann „Assign to People“ aus der Dropdown-Liste 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 644 dafür, indem Sie Folgendes ausführen:

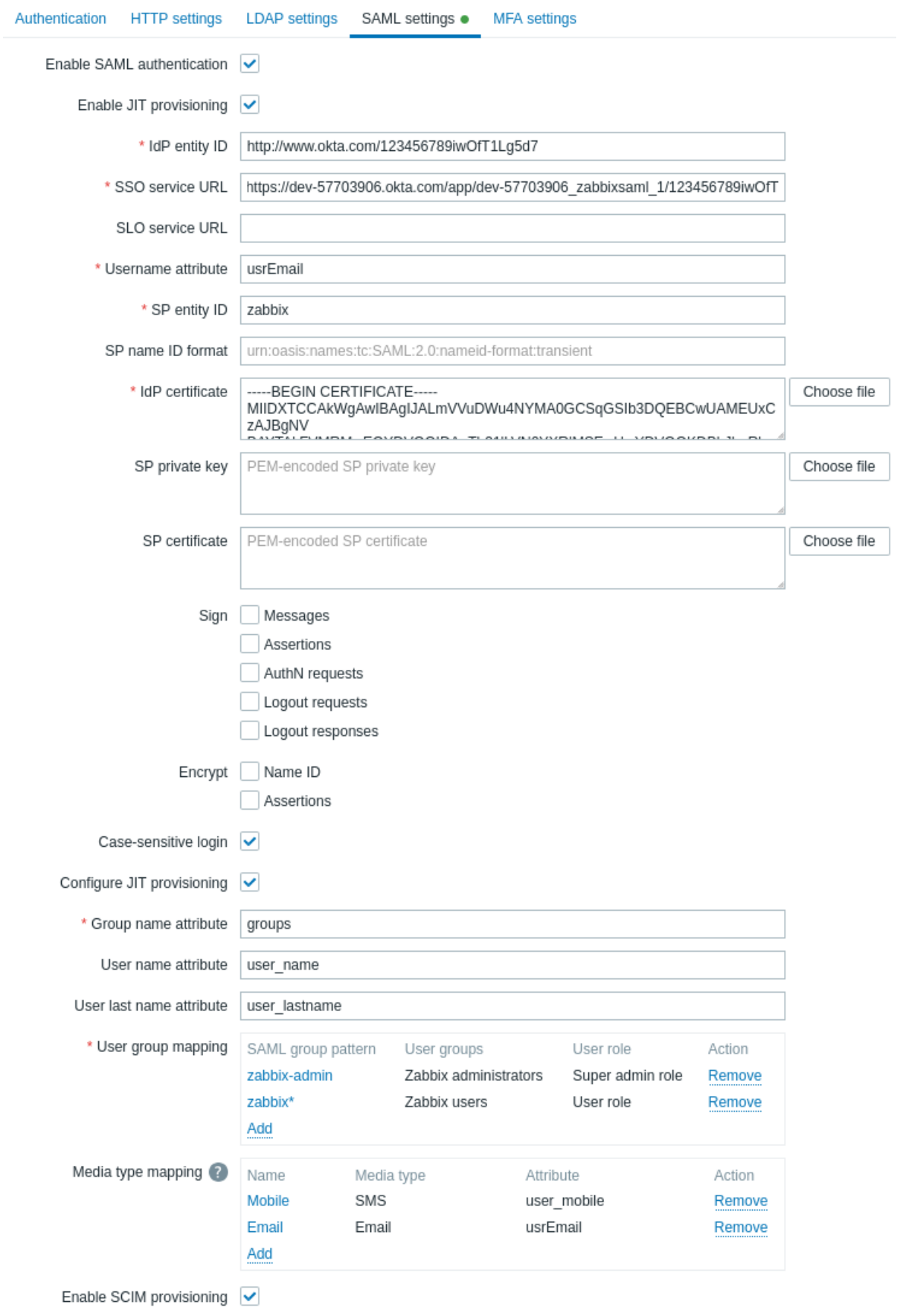
```
chmod 644 idp.crt
```

10. Navigieren Sie zur Registerkarte „Sign On“ und klicken Sie auf die Schaltfläche *View Setup Instructions*.

Die **Anweisungen** zur Einrichtung werden in einer neuen Registerkarte geöffnet; lassen Sie diese Registerkarte geöffnet, während Sie Zabbix konfigurieren.

Zabbix-Konfiguration

1. Gehen Sie in Zabbix zu den SAML-Einstellungen und füllen Sie die Konfigurationsoptionen anhand der Einrichtungsanweisungen von Okta aus:



Zabbix-Feld	Einrichtungsfeld in Okta	Beispielwert
IdP-Entitäts-ID	Identity Provider Issuer	
SSO-Service-URL	Identity Provider Single Sign-On URL	
Benutzernamenattribut	Attributname	`usrEmail`
SP-Entitäts-ID	Audience URI	`zabbix`
Gruppennamenattribut	Attributname	`groups`

Zabbix-Feld	Einrichtungsfeld in Okta	Beispielwert
Vornamenattribut des Benutzers	Attributname	`user_name`
Nachnamenattribut des Benutzers	Attributname	`user_lastname`

Außerdem müssen die Zuordnung von Benutzergruppen und Medien konfiguriert werden.

2. Fügen Sie das Base64-Zertifikat hinzu, das in den Okta-SAML-Einrichtungsanweisungen bereitgestellt wird.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in zabbix.conf.php gesetzt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen — es sind keine Dateien im Dateisystem erforderlich.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in zabbix.conf.php gesetzt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in ui/conf/certs oder in dem in zabbix.conf.php konfigurierten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank.
Beachten Sie, dass, wenn `$SSO['CERT_STORAGE']` nicht gesetzt oder auskommentiert ist, Dateispeicherung angenommen wird und Zertifikate aus ui/conf/certs gelesen werden.

3. Wenn *Assertion Encryption* in Okta auf ”Encrypted” gesetzt wurde, sollte in Zabbix auch das Kontrollkästchen ”Assertions” des Parameters *Encrypt* aktiviert werden.

4. Klicken Sie auf die Schaltfläche *Update*, um diese Einstellungen zu speichern.

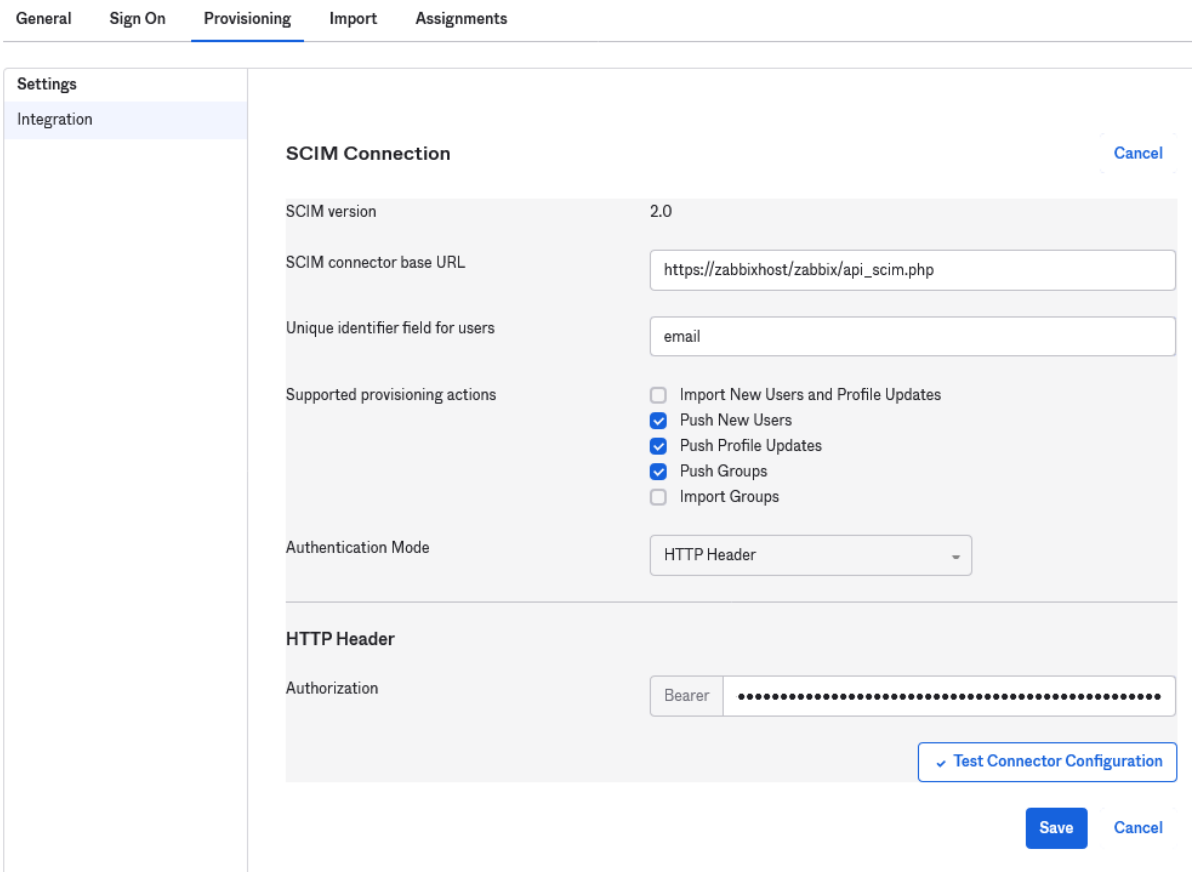
SCIM-Provisionierung

1. Um die SCIM-Provisionierung zu aktivieren, gehen Sie in Okta zu ”General” -> ”App Settings” der Anwendung.

Aktivieren Sie das Kontrollkästchen *Enable SCIM provisioning*. Dadurch erscheint ein neuer Reiter *Provisioning*.

2. Gehen Sie zum Reiter ”Provisioning”, um eine SCIM-Verbindung einzurichten:

- Geben Sie in *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_scim.php`
- *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 bei Ihnen Authentifizierungsprobleme auftreten, siehe Weiterleitung des Authorization-Headers.

3. Klicken Sie auf *Test Connector Configuration*, um die Verbindung zu testen. Wenn alles korrekt ist, wird eine Erfolgsmeldung angezeigt.

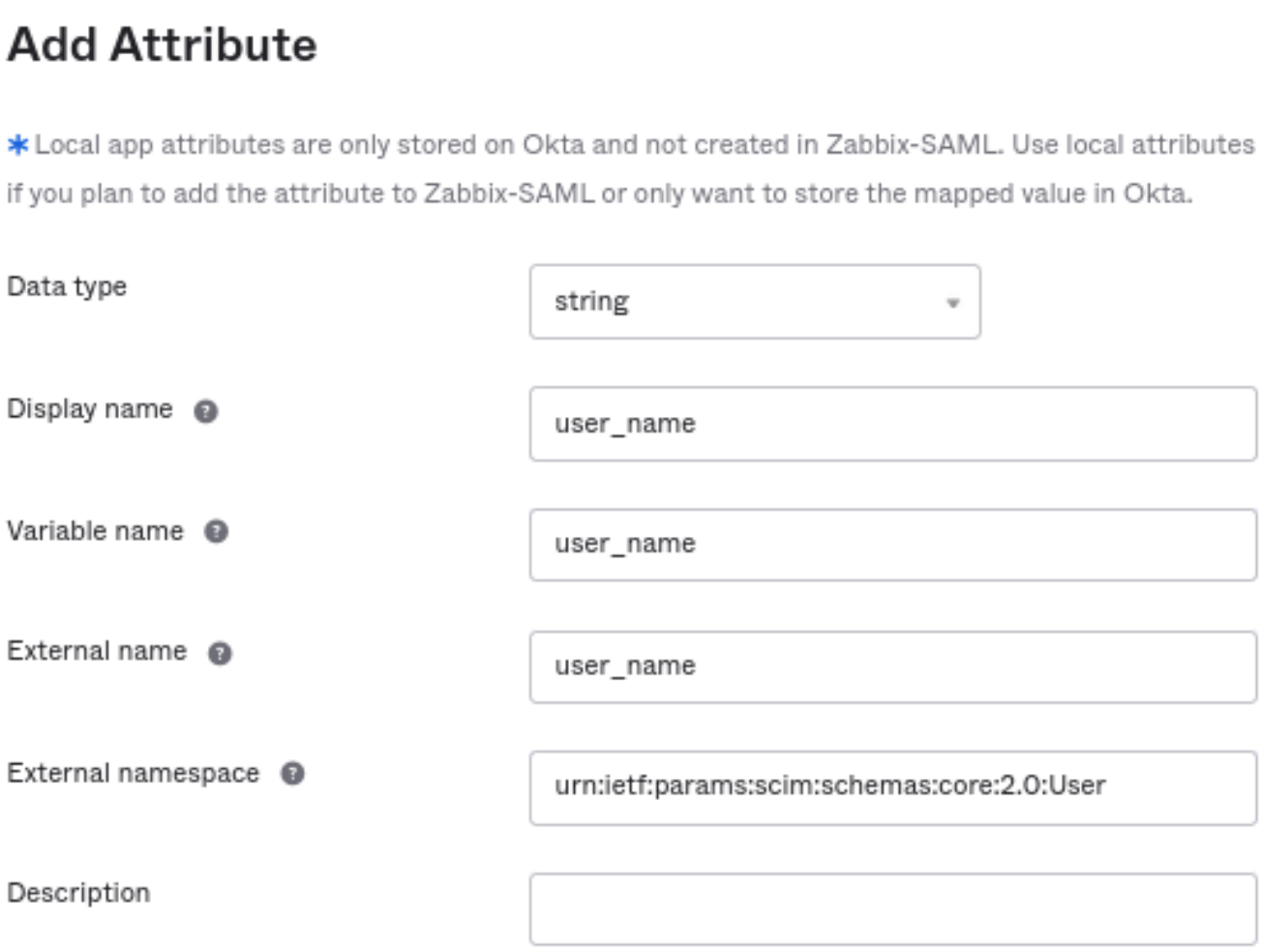
4. Stellen Sie unter ”Provisioning” -> ”To App” sicher, dass die folgenden Kontrollkästchen aktiviert sind:

- Create Users
- Update User Attributes
- Deactivate Users

Dadurch wird sichergestellt, dass diese Anfragetypen an Zabbix gesendet werden.

5. Stellen Sie sicher, dass alle in SAML definierten Attribute auch in SCIM definiert sind. Sie können den Profile Editor für Ihre App unter ”Provisioning” -> ”To App” aufrufen, indem Sie auf *Go to Profile Editor* klicken.

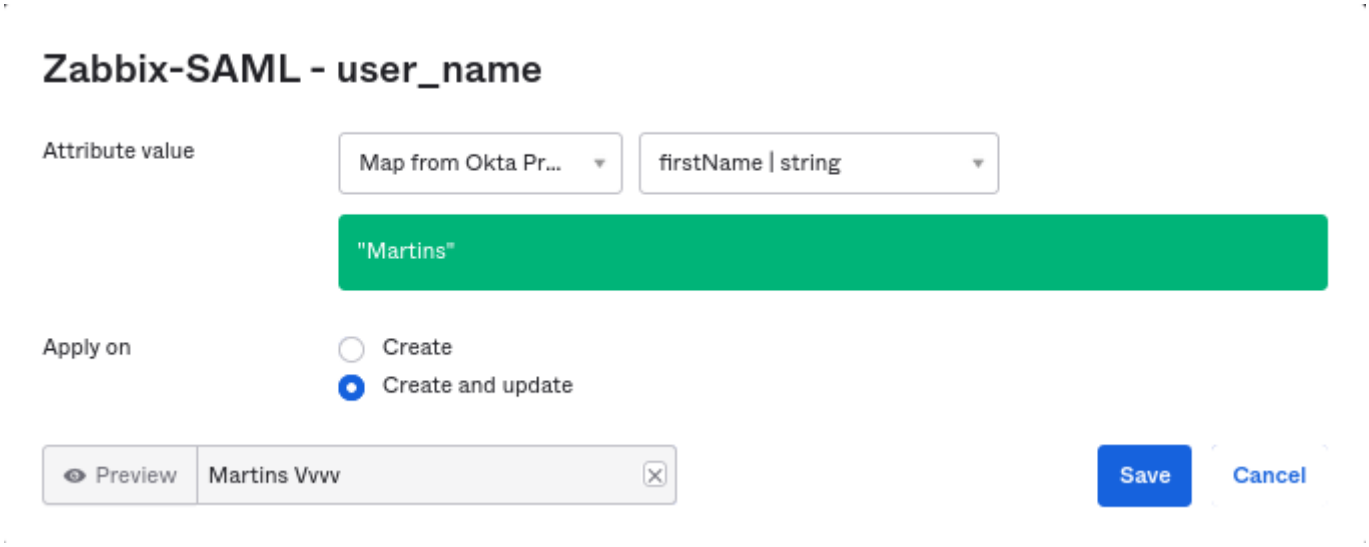
Klicken Sie auf *Add Attribute*. Füllen Sie die Werte für *Display name*, *Variable name* und *External name* mit dem SAML-Attributnamen aus, zum Beispiel `user_name`.



External namespace sollte mit dem Benutzerschema übereinstimmen: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Gehen Sie in Ihrer Anwendung zu ”Provisioning” -> ”To App” -> ”Attribute Mappings”. Klicken Sie unten auf *Show Unmapped Attributes*. Neu hinzugefügte Attribute werden angezeigt.

7. Ordnen Sie jedes hinzugefügte Attribut zu.



8. Fügen Sie Benutzer im Reiter ”Assignments” hinzu. Die Benutzer müssen zuvor unter *Directory -> People* hinzugefügt werden. Alle diese Zuweisungen werden als Anfragen an Zabbix gesendet.

9. Fügen Sie Gruppen im Reiter ”Push Groups” hinzu. Das Zuordnungsmuster für Benutzergruppen in den Zabbix-SAML-Einstellungen muss mit einer hier angegebenen Gruppe übereinstimmen. Wenn es keine Übereinstimmung gibt, kann der Benutzer in Zabbix nicht erstellt werden.

Informationen über Gruppenmitglieder werden jedes Mal gesendet, wenn eine Änderung vorgenommen wird.

14 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

Anwendung erstellen

1. Melden Sie sich bei Ihrem Konto in 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 entsprechenden Anwendung. Die Anweisungen auf dieser Seite basieren auf dem Beispiel der Anwendung *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

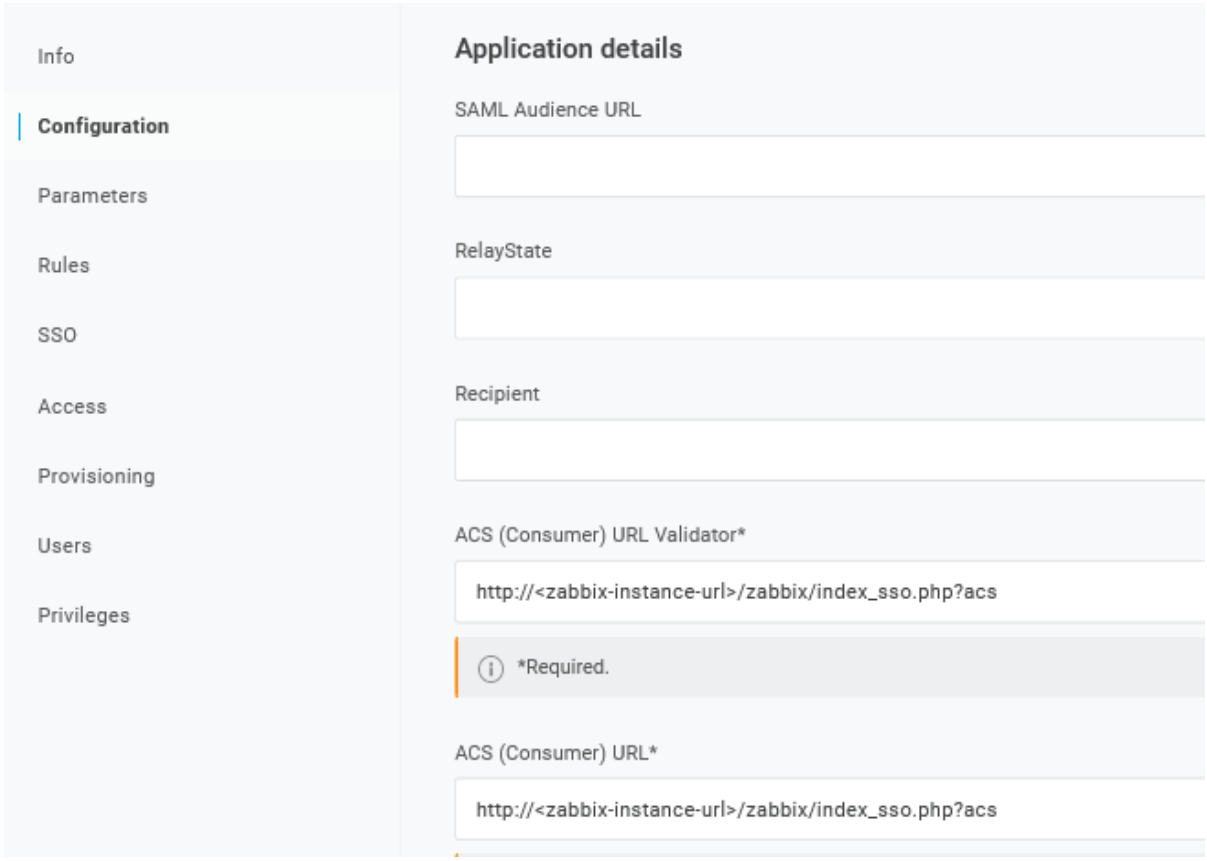
4. Zu Beginn können Sie den Anzeigenamen Ihrer Anwendung anpassen. Sie können außerdem das Symbol und die Anwendungsdetails hinzufügen. Klicken Sie danach auf *Save*.

Einrichten der SSO-/SCIM-Bereitstellung

1. Legen Sie in *Configuration -> Application details* den Zabbix-Endpunkt für Single Sign-on `http://<zabbix-instance-url>/zabbix/in` als Wert für diese Felder fest:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Beachten Sie die Verwendung von „http“ und nicht „https“, damit der Parameter `acs` in der Anfrage nicht abgeschnitten wird.



Es ist auch möglich, „https“ zu verwenden. Damit dies mit Zabbix funktioniert, muss in `conf/zabbix.conf.php` die folgende Zeile hinzugefügt werden:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Lassen Sie die anderen Optionen auf ihren Standardwerten.

2. Legen Sie in *Configuration* -> *API connection* die folgenden Werte fest:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: sollte alle benutzerdefinierten Attribute enthalten, die Sie über SCIM an Zabbix übergeben möchten, z. B. `user_name`, `user_lastname`, `user_email` und `user_mobile`:

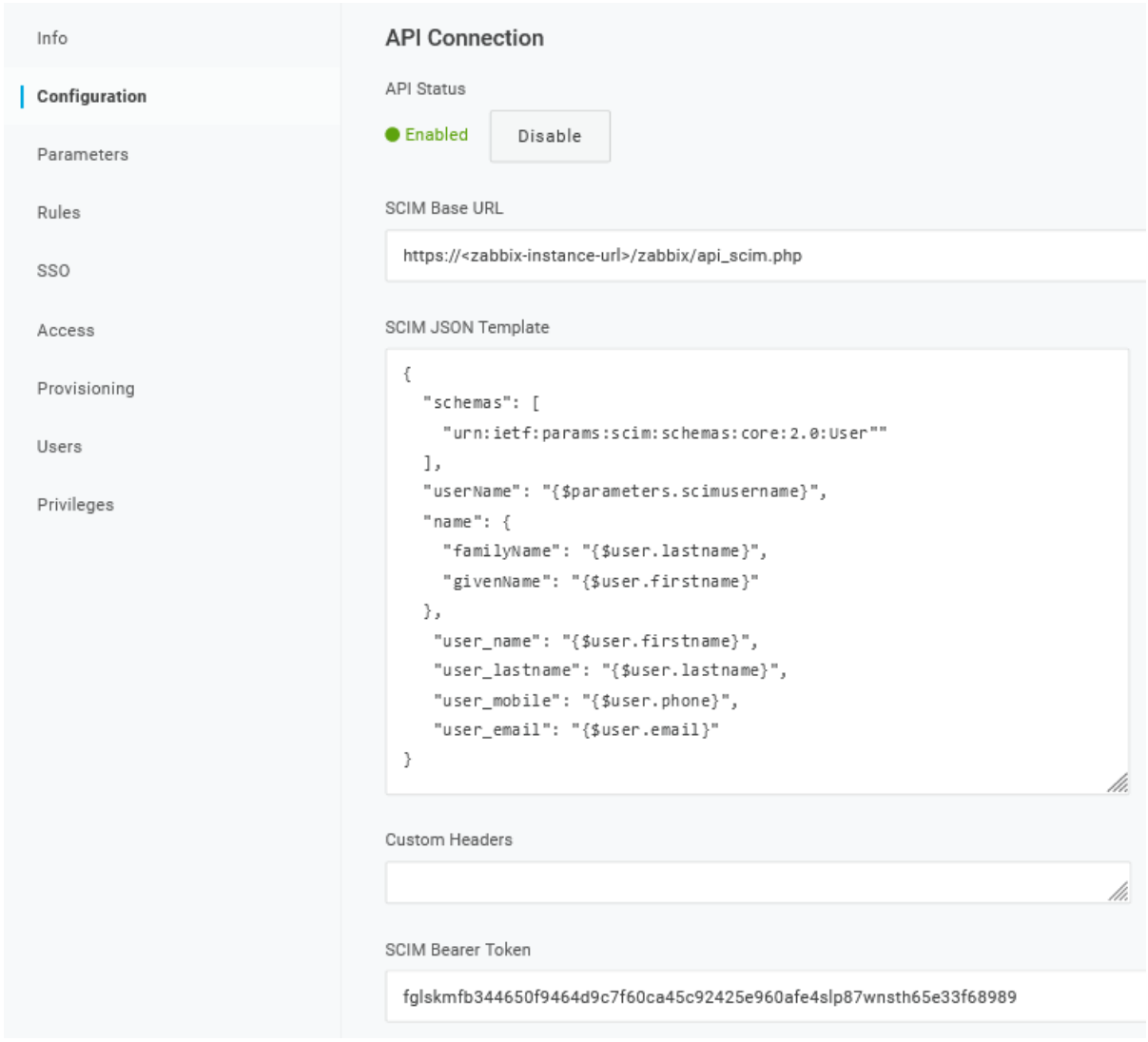
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Die Attributnamen sind frei wählbar. Es können andere Attributnamen verwendet werden, jedoch ist erforderlich, dass sie mit dem jeweiligen Feldwert in den Zabbix-SAML-Einstellungen übereinstimmen.

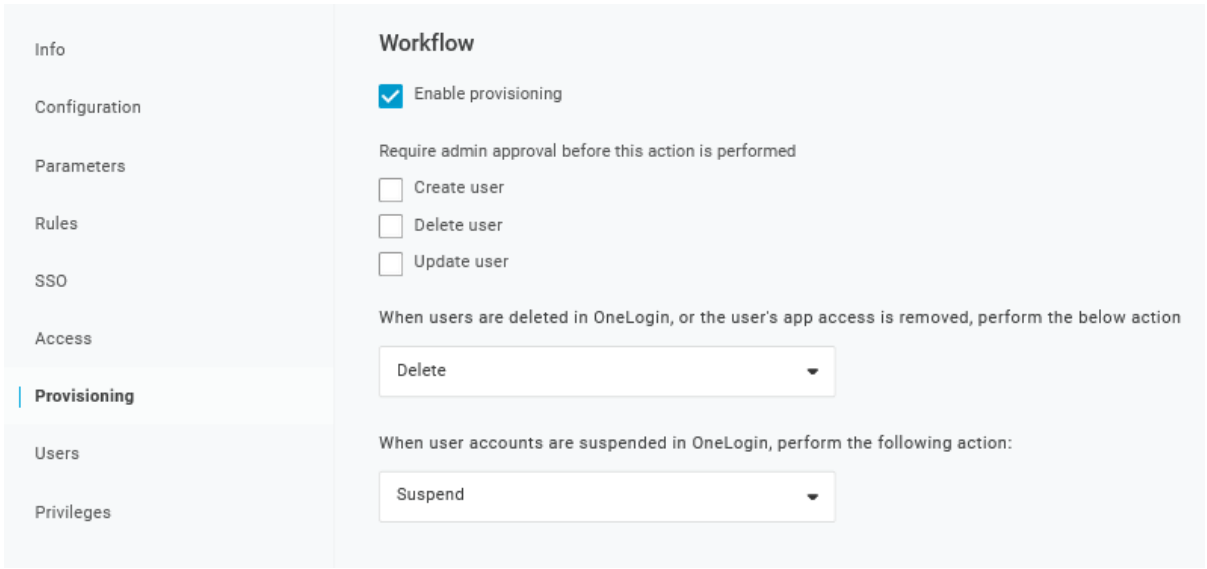
Beachten Sie, dass OneLogin für die Benutzerbereitstellung in der Antwort ein Attribut „name“ mit „givenName“ und „familyName“ erhalten muss, auch wenn dies vom Dienstanbieter nicht verlangt wurde. Daher ist es notwendig, dies im Schema im Konfigurationsteil der Anwendung anzugeben.

- *SCIM Bearer Token*: Geben Sie ein Zabbix-API-Token mit Super-Admin-Berechtigungen ein.

Klicken Sie auf *Enable*, um die Verbindung zu aktivieren.

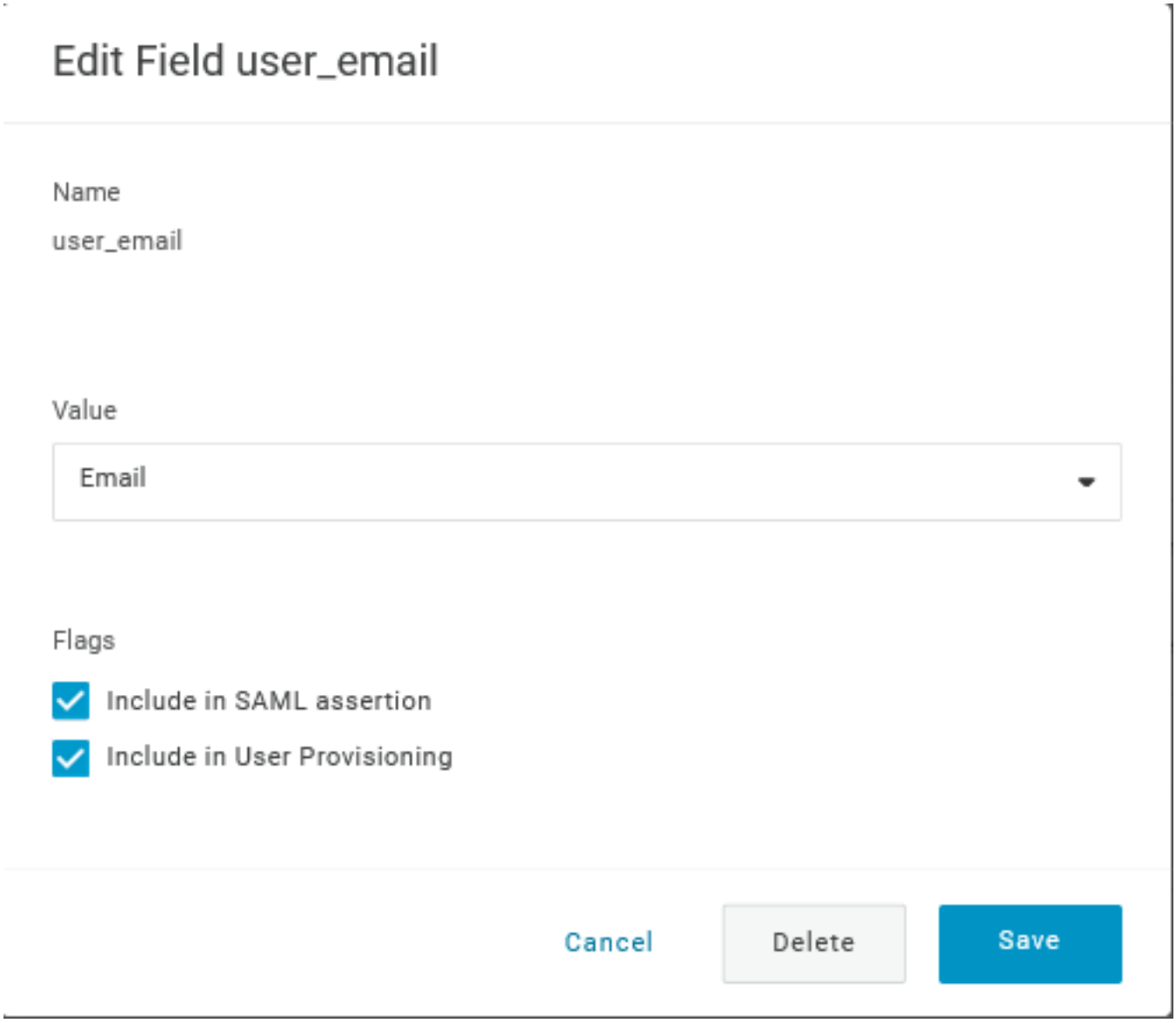


3. Aktivieren Sie auf der Seite *Provisioning* die Option Provisioning:



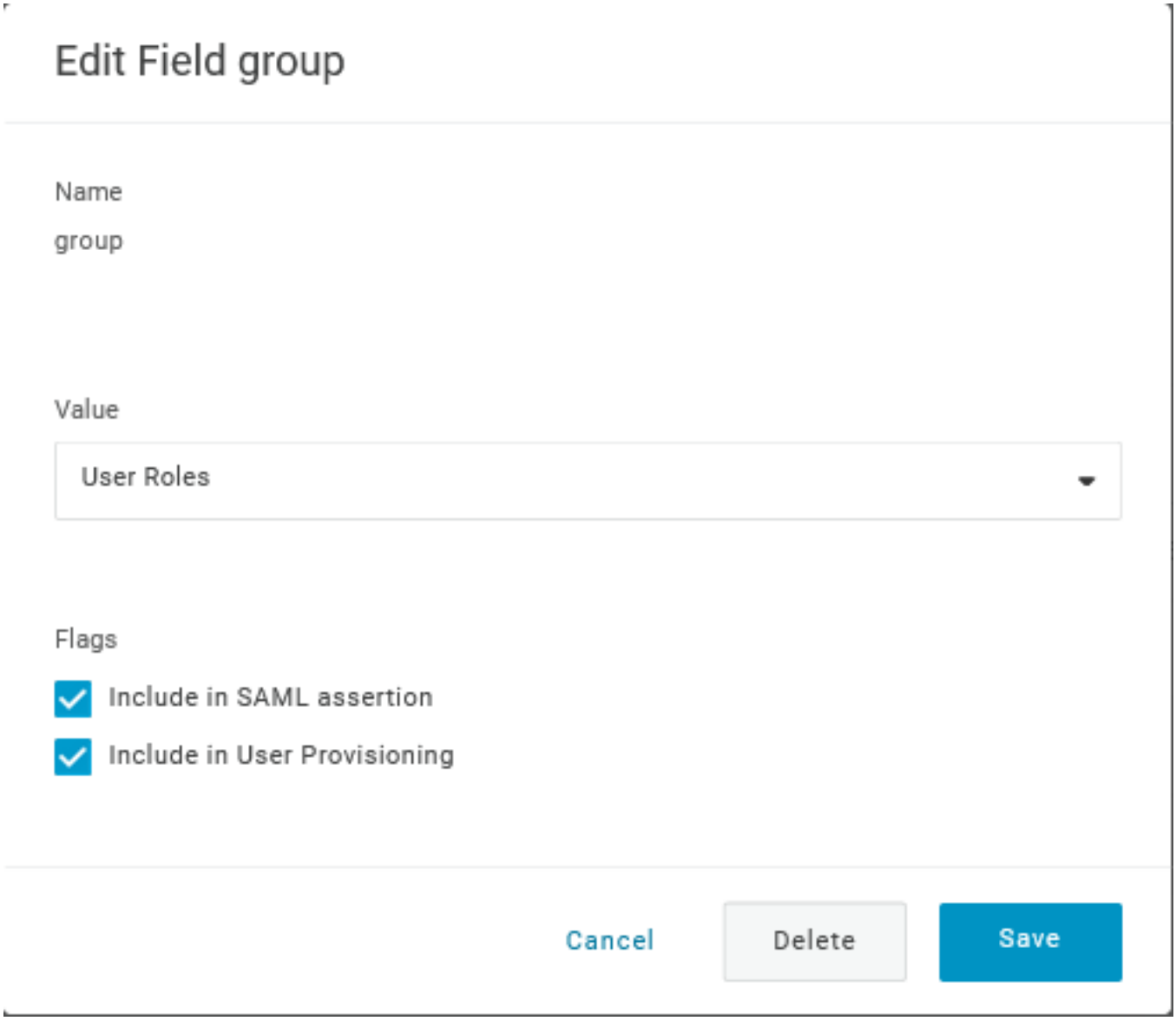
4. Die Seite *Parameters* enthält eine Liste der Standardparameter:

- Stellen Sie sicher, dass „scimusername“ mit dem Benutzeranmeldewert in OneLogin übereinstimmt (z. B. E-Mail);
- Aktivieren Sie für den Parameter „Groups“ die Option *Include in User Provisioning*;
- Klicken Sie auf „+“, um die benutzerdefinierten Parameter zu erstellen, die für SAML-Assertions und die Benutzerbereitstellung erforderlich sind, z. B. `user_name`, `user_lastname`, `user_email` und `user_mobile`:

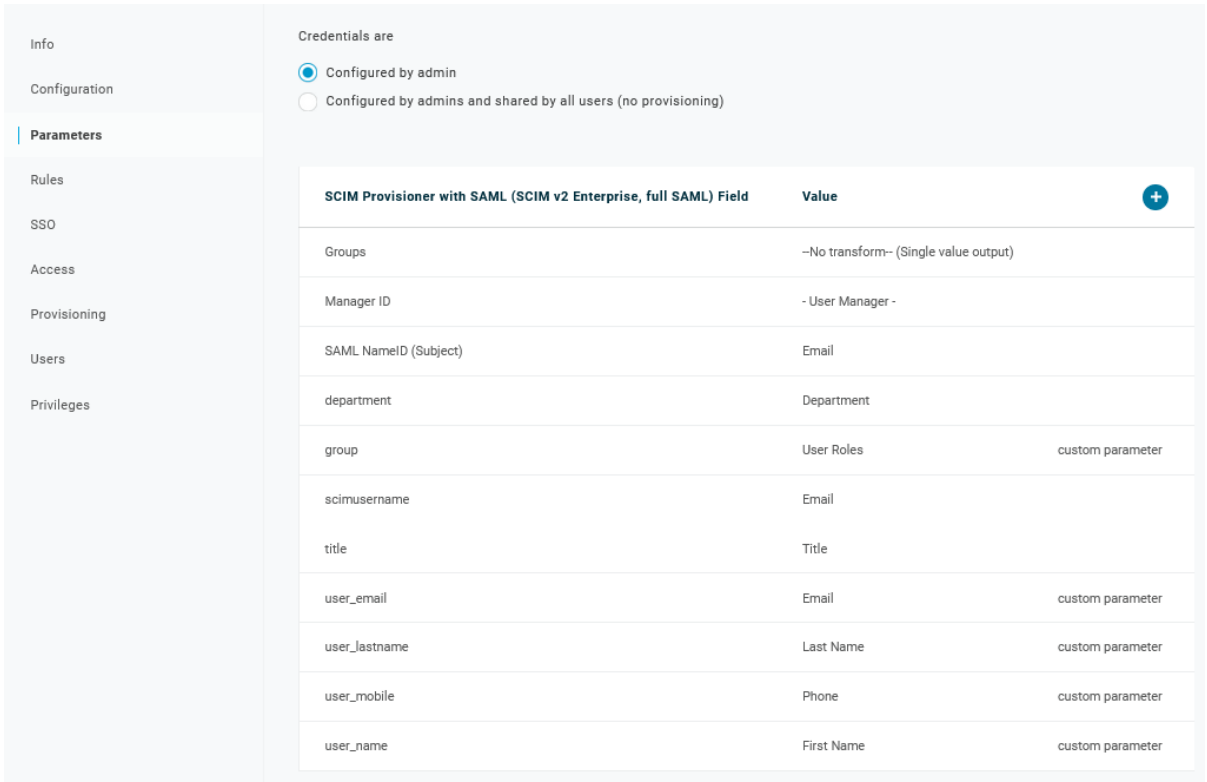


Achten Sie beim Hinzufügen eines Parameters darauf, sowohl die Option *Include in SAML assertion* als auch *Include in User Provisioning* zu aktivieren.

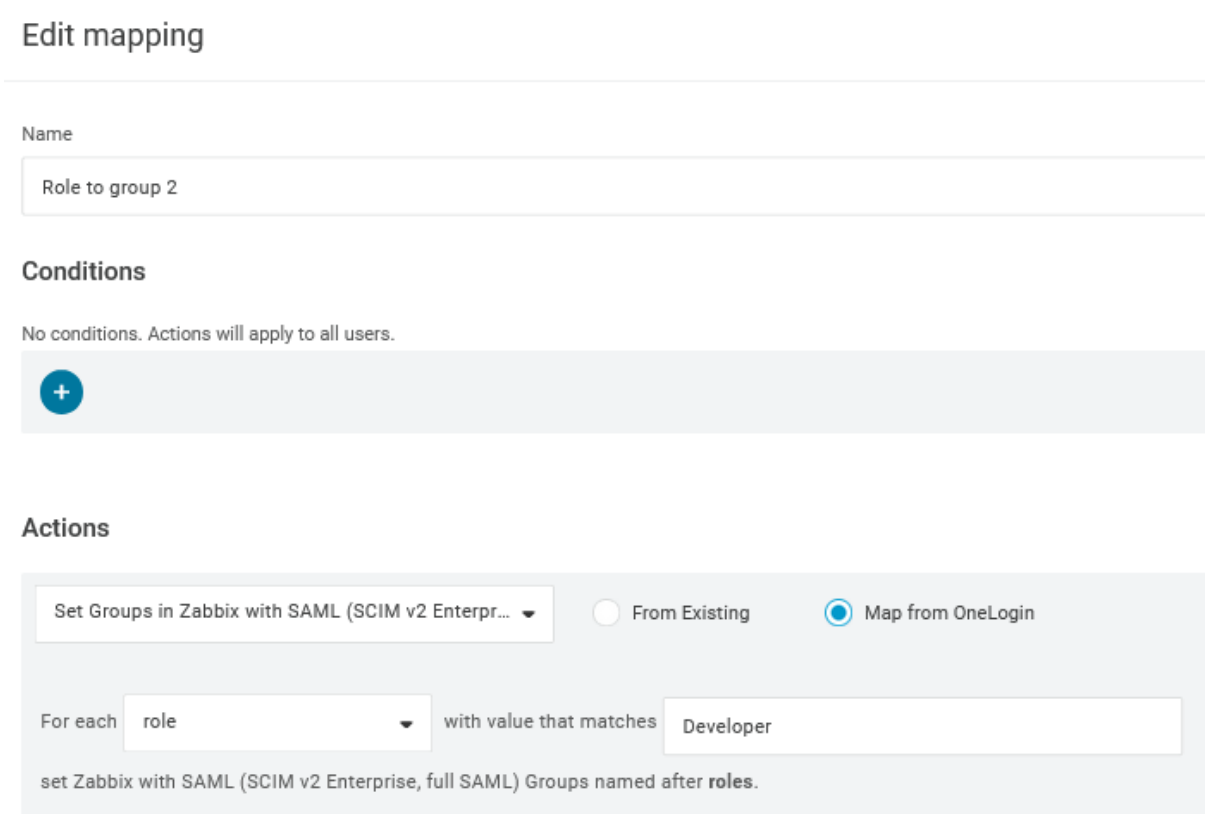
- Fügen Sie einen Parameter „group“ hinzu, der den Benutzerrollen in OneLogin entspricht. Benutzerrollen werden als durch ein Semikolon ; getrennte Zeichenfolge übergeben. Die OneLogin-Benutzerrollen werden zum Erstellen von Benutzergruppen in Zabbix verwendet:



Überprüfen Sie die Parameterliste:



5. Erstellen Sie auf der Seite *Rules* Zuordnungen von Benutzerrollen zum Standardparameter Groups.



Sie können einen regulären Ausdruck verwenden, um bestimmte Rollen als Gruppen zu übergeben. Die Rollennamen sollten kein ; enthalten, da OneLogin dieses als Trennzeichen verwendet, wenn ein Attribut mit mehreren Rollen gesendet wird.

6. Laden Sie das IdP-Zertifikat herunter. Setzen Sie die Berechtigung 644 dafür, indem Sie Folgendes ausführen:

```
chmod 644 idp.crt
```

Zabbix-Konfiguration

1. Gehen Sie in Zabbix zu den SAML-Einstellungen und füllen Sie die Konfigurationsoptionen basierend auf der OneLogin-Konfiguration aus:

Zabbix-Feld	Einrichtungsfeld in OneLogin	Beispielwert
IdP-Entity-ID	Issuer URL (siehe Registerkarte SSO Ihrer Anwendung in OneLogin)	

Zabbix-Feld	Einrichtungsfeld 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)	
Attribut für Benutzernamen	Benutzerdefinierter Parameter	`user_email`
Attribut für Gruppennamen	Benutzerdefinierter Parameter	`group`
Attribut für Vornamen	Benutzerdefinierter Parameter	`user_name`
Attribut für Nachnamen	Benutzerdefinierter Parameter	`user_lastname`

Es ist außerdem erforderlich, die Benutzergruppen-Zuordnung zu konfigurieren. Die Medien-Zuordnung ist optional. Klicken Sie auf *Aktualisieren*, um diese Einstellungen zu speichern.

2. Fügen Sie das von OneLogin bereitgestellte Base64-Zertifikat hinzu.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in zabbix.conf.php gesetzt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen — es sind keine Dateien im Dateisystem erforderlich.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in zabbix.conf.php gesetzt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in ui/conf/certs oder in dem in zabbix.conf.php konfigurierten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank. Beachten Sie, dass, wenn `$SSO['CERT_STORAGE']` nicht gesetzt ist oder auskommentiert wurde, Dateispeicherung angenommen wird und Zertifikate aus ui/conf/certs gelesen werden.

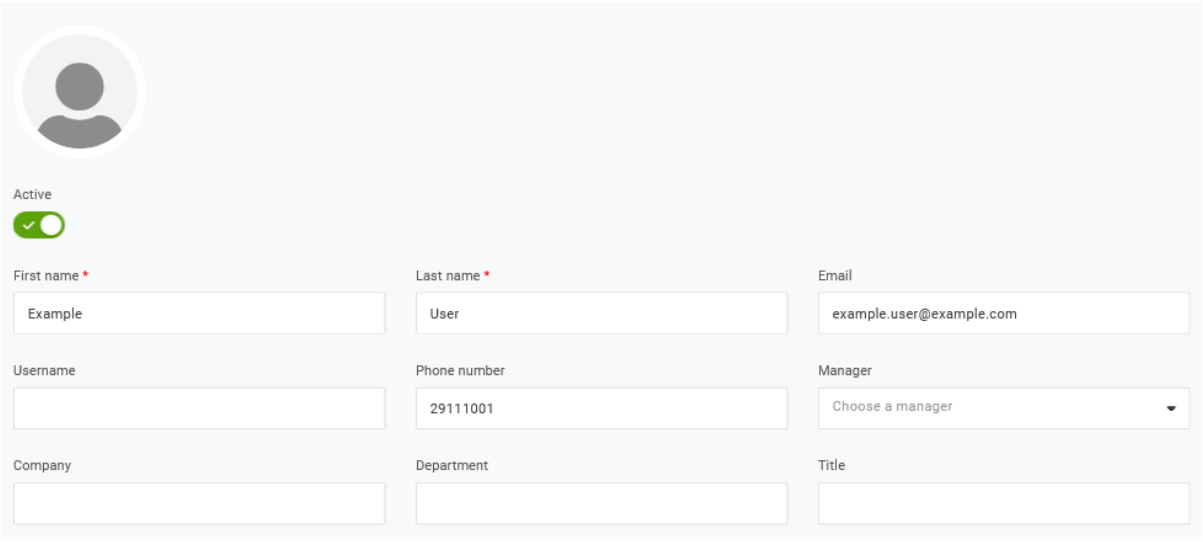
Sie können den Zertifikat-Download in OneLogin unter *Applications* -> *SSO* aufrufen und dort unter dem aktuellen Zertifikat auf *View details* klicken.

3. Klicken Sie auf die Schaltfläche *Aktualisieren*, um diese Einstellungen zu speichern.

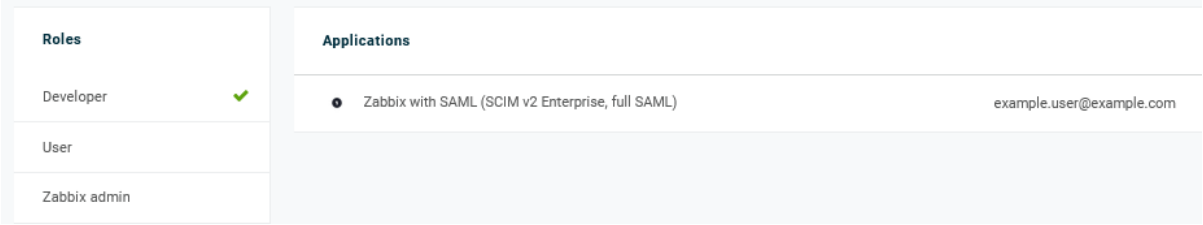
SCIM-Benutzerbereitstellung

Wenn die Benutzerbereitstellung aktiviert ist, können Benutzer und ihre Rollen jetzt in OneLogin hinzugefügt/aktualisiert werden und werden sofort in Zabbix bereitgestellt.

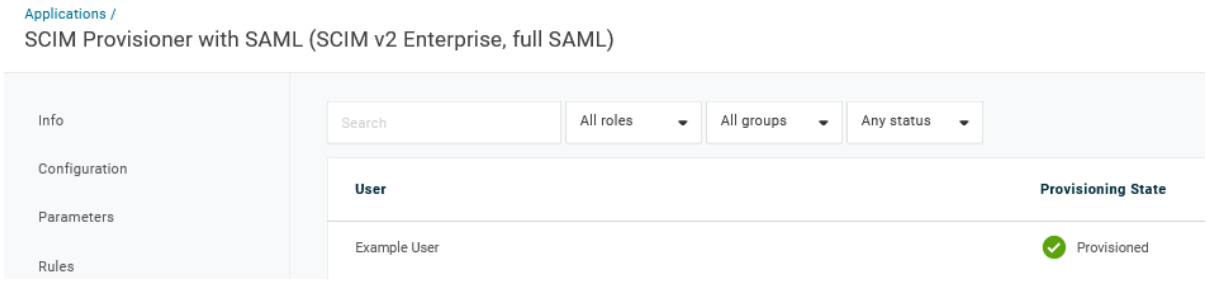
Sie können zum Beispiel einen neuen Benutzer erstellen:



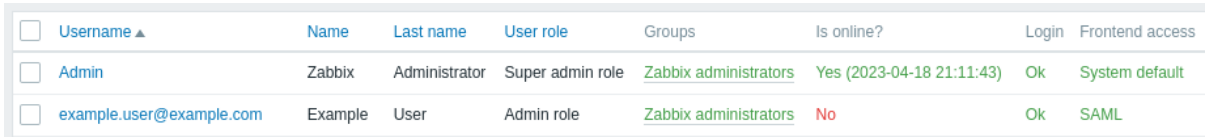
Fügen Sie ihn einer Benutzerrolle und der Anwendung hinzu, die den Benutzer bereitstellt:



Beim Speichern des Benutzers wird er in Zabbix bereitgestellt. Unter Application -> Users können Sie den Bereitstellungsstatus der aktuellen Anwendungsbenutzer prüfen:



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

15 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.

Um den Zabbix-Webservice aus den Quellen zu kompilieren, siehe Installing Zabbix web service.

Führen Sie nach der Installation zabbix_web_service auf dem Rechner aus, auf dem der Webservice installiert ist:

```
zabbix_web_service
```

Konfiguration

Um eine ordnungsgemäße Kommunikation zwischen allen beteiligten Elementen sicherzustellen, vergewissern Sie sich, dass die Konfigurationsdatei des Servers und die Konfigurationsparameter des Frontends korrekt konfiguriert sind.

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 Web-Service zu aktivieren. Die URL sollte das Format `http[s]://host:port/report` haben.

- Standardmäßig lauscht der Web-Service auf Port 10053. Ein anderer Port kann in der Konfigurationsdatei des Web-Service angegeben werden.
- Die Angabe des Pfads `/report` ist zwingend erforderlich (der Pfad ist fest im Code hinterlegt und kann nicht geändert werden).

Beispiel:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
Es wird dringend empfohlen, die Verschlüsselung zwischen Zabbix-Server und Zabbix-Web-Service mithilfe von Zertifikaten einzurichten. Standardmäßig werden die zwischen Zabbix-Server und Zabbix-Web-Service übertragenen Daten nicht verschlüsselt, was zu unbefugtem Zugriff führen kann.

StartReportWriters

Dieser Parameter legt fest, wie viele Report-Writer-Prozesse gestartet werden sollen. Wenn er nicht gesetzt ist oder den Wert 0 hat, ist die Berichtserstellung deaktiviert. Abhängig von der Anzahl und Häufigkeit der benötigten Berichte können 1 bis 100 Report-Writer-Prozesse aktiviert werden.

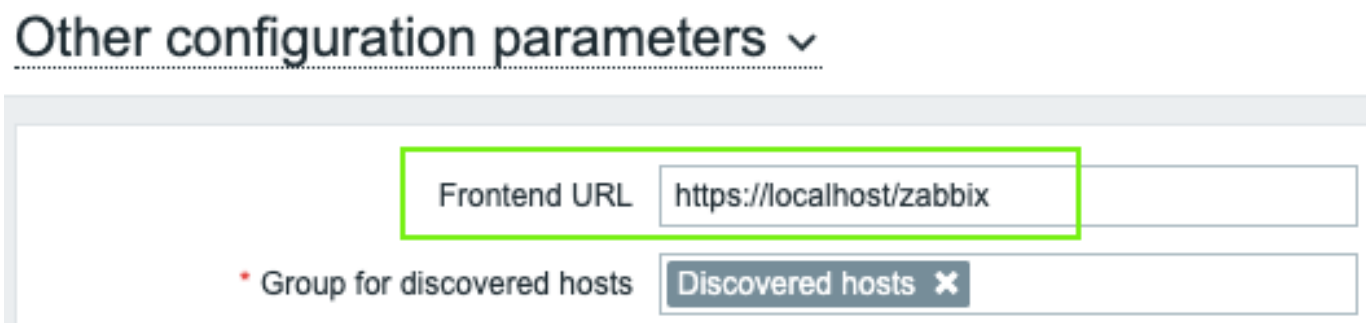
Beispiel:

```
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.

16 Zusätzliche Frontend-Sprachen

Übersicht

Um in der Zabbix-Weboberfläche eine andere Sprache als Englisch zu verwenden, muss die entsprechende Locale 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 Folgendes aus:

```
locale -a
```

Wenn einige benötigte Sprachen nicht aufgeführt sind, öffnen Sie die Datei */etc/locale.gen* und entfernen Sie die Auskommentierung der erforderlichen Locales. Da Zabbix die UTF-8-Kodierung verwendet, müssen Sie Locales mit UTF-8-Zeichensatz auswählen.

Führen Sie nun Folgendes aus:

```
locale-gen
```

Starten Sie den Webserver neu.

Die Locales sollten nun installiert sein. Möglicherweise ist es erforderlich, die Zabbix Frontend-Seite im Browser mit Strg + F5 neu zu laden, damit die neuen Sprachen angezeigt werden.

Zabbix installieren

Wenn Zabbix direkt aus dem Zabbix git repository installiert wird, müssen Übersetzungsdateien manuell erzeugt werden. Führen Sie zum Erzeugen der Übersetzungsdateien Folgendes aus:

```
make gettext
locale/make_mo.sh
```

Dieser Schritt ist nicht erforderlich, wenn Zabbix aus Paketen oder aus source-tar.gz-Dateien installiert wird.

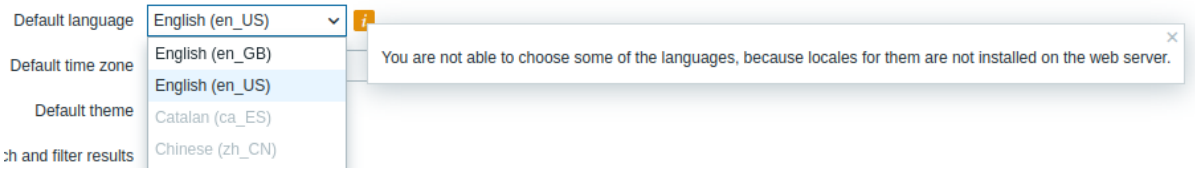
Auswählen einer Sprache

Es gibt mehrere Möglichkeiten, in der Zabbix-Weboberfläche eine Sprache auszuwählen:

- Bei der Installation der Weboberfläche – im Frontend-Installationsassistenten. Die ausgewählte Sprache wird als Systemstandard festgelegt.

- Nach der Installation kann die Standardsprache des Systems im Menüabschnitt *Administration→General→GUI* geändert werden.
- Die Sprache für einen bestimmten Benutzer kann im Benutzerprofil geändert werden.

Wenn ein Locale für eine Sprache auf dem Rechner nicht installiert ist, wird diese Sprache in der Zabbix-Sprachauswahl ausgegraut angezeigt.
Neben der Sprachauswahl wird ein oranges Symbol angezeigt, wenn mindestens ein Locale fehlt.
Beim Klicken auf dieses Symbol wird folgende Meldung angezeigt:
„Sie können einige der Sprachen nicht auswählen, da die entsprechenden Locales auf dem Webserver nicht installiert sind.“



2 Prozesskonfiguration

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte in diesem Abschnitt zuzugreifen.

1 Zabbix Server

Übersicht

Die von der Zabbix-Server-Konfigurationsdatei (zabbix_server.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
AlertScriptsPath	Der Speicherort benutzerdefinierter Benachrichtigungsskripte.
AllowRoot	Erlaubt, dass der Server als „root“ ausgeführt wird.
AllowSoftwareUpdateCheck	Erlaubt der Zabbix-Benutzeroberfläche, Informationen über Software-Updates von zabbix.com zu empfangen.
AllowUnsupportedDBVersions	Erlaubt dem Server, mit nicht unterstützten Datenbankversionen zu arbeiten.
CacheSize	Die Größe des Konfigurations-Caches.
CacheUpdateFrequency	Dieser Parameter legt fest, wie oft Zabbix die Aktualisierung des Konfigurations-Caches in Sekunden durchführt.
DBHost	Der Name des Datenbank-Hosts.
DBName	Der Name der Datenbank.
DBPassword	Das Datenbankpasswort.
DBPort	Der Datenbank-Port, wenn kein lokaler Socket verwendet wird.
DBSchema	Der Name des Datenbankschemas. Wird für PostgreSQL verwendet.
DBSocket	Der Pfad zur MySQL-Socket-Datei.
DBUser	Der Datenbankbenutzer.
DBTLSConnect	Wenn diese Option auf den angegebenen Wert gesetzt wird, wird die Verwendung einer TLS-Verbindung zur Datenbank erzwungen.
DBTLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Überprüfung des Datenbankzertifikats enthält.
DBTLSCertFile	Der vollständige Pfadname einer Datei, die das Zabbix-Serverzertifikat zur Authentifizierung bei der Datenbank enthält.
DBTLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel zur Authentifizierung bei der Datenbank enthält.
DBTLSCipher	Die Liste der Verschlüsselungs-Chiffren, die der Zabbix-Server für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Wird nur für MySQL unterstützt.
DBTLSCipher13	Die Liste der Verschlüsselungs-Chiffresuites, die der Zabbix-Server für das TLS-v1.3-Protokoll zulässt. Wird nur für MySQL ab Version 8.0.16 unterstützt.
DebugLevel	Gibt die Debug-Stufe an.
EnableGlobalScripts	Aktiviert globale Skripte auf dem Zabbix-Server.
ExportDir	Das Verzeichnis für den Echtzeit-Export von Ereignissen, Verlauf und Trends im durch Zeilenumbrüche getrennten JSON-Format. Wenn gesetzt, wird der Echtzeit-Export aktiviert.

Parameter	Beschreibung
ExportFileSize	Die maximale Größe pro Exportdatei in Byte.
ExportType	Die durch Kommas getrennte Liste von Entitätstypen (Ereignisse, Verlauf, Trends) für den Echtzeit-Export (standardmäßig alle Typen).
ExternalScripts	Der Speicherort externer Skripte.
Fping6Location	Der Speicherort von fping6.
FpingLocation	Der Speicherort von fping.
FrontendAllowedIP	Eine durch Kommas getrennte Liste von IP-Adressen oder CIDR-Bereichen, die Verbindungen vom Frontend herstellen dürfen.
HANodeName	Der Knotenname des Hochverfügbarkeitsclusters.
HistoryCacheSize	Die Größe des Verlaufs-Caches.
HistoryIndexCacheSize	Die Größe des Verlaufsindex-Caches.
HistoryStorageDateIndex	Aktiviert die Vorverarbeitung von Verlaufswerten im Verlaufsspeicher, um Werte basierend auf dem Datum in verschiedenen Indizes zu speichern.
HistoryStorageURL	Die HTTP[S]-URL des Verlaufsspeichers.
HistoryStorageTypes	Eine durch Kommas getrennte Liste von Werttypen, die an den Verlaufsspeicher gesendet werden.
HousekeepingFrequency	Dieser Parameter legt fest, wie oft Zabbix das Housekeeping-Verfahren in Stunden durchführt.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
JavaGateway	Die IP-Adresse (oder der Hostname) des Zabbix Java gateway.
JavaGatewayPort	Der Port, auf dem das Zabbix Java gateway lauscht.
ListenBacklog	Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
ListenIP	Eine durch Kommas getrennte Liste von IP-Adressen, auf denen der Trapper lauschen soll.
ListenPort	Der Port, auf dem der Trapper lauscht.
LoadModule	Das Modul, das beim Start des Servers geladen werden soll.
LoadModulePath	Der vollständige Pfad zum Speicherort der Servermodule.
LogFile	Der Name der Protokolldatei.
LogFileSize	Die maximale Größe der Protokolldatei.
LogSlowQueries	Legt fest, wie lange eine Datenbankabfrage dauern darf, bevor sie in Millisekunden protokolliert wird.
LogType	Der Typ der Protokollausgabe.
MaxConcurrentChecksPerPoller	Die maximale Anzahl asynchroner Prüfungen, die gleichzeitig von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller ausgeführt werden können.
MaxHousekeeperDelete	Pro Aufgabe in einem Housekeeping-Zyklus werden nicht mehr als „MaxHousekeeperDelete“ Zeilen (entsprechend [tablename], [field], [value]) gelöscht.
NodeAddress	Die IP-Adresse oder der Hostname mit optionalem Port, um zu überschreiben, wie sich das Frontend mit dem Server verbinden soll.
PidFile	Der Name der PID-Datei.
ProblemHousekeepingFrequency	Legt fest, wie oft Zabbix Probleme für gelöschte Auslöser löscht.
ProxyConfigFrequency	Legt fest, wie oft der Zabbix-Server Konfigurationsdaten an einen Zabbix-Proxy sendet.
ProxyDataFrequency	Legt fest, wie oft der Zabbix-Server Verlaufsdaten von einem Zabbix-Proxy anfordert.
ServiceManagerSyncFrequency	Legt fest, wie oft Zabbix die Konfiguration eines Service-Managers synchronisiert.
SMSDevices	Eine durch Kommas getrennte Liste von Modemdateien, die vom Zabbix-Server verwendet werden dürfen.
SNMPTrapperFile	Die temporäre Datei, die zur Übergabe von Daten vom SNMP-Trap-Daemon an den Server verwendet wird.
SocketDir	Das Verzeichnis zum Speichern der von internen Zabbix-Diensten verwendeten IPC-Sockets.
SourceIP	Die Quell-IP-Adresse.
SSHKeyLocation	Der Speicherort öffentlicher und privater Schlüssel für SSH-Prüfungen und Aktionen.
SSLCertLocation	Der Speicherort der SSL-Clientzertifikatdateien für die Client-Authentifizierung.
SSLKeyLocation	Der Speicherort der SSL-Dateien mit privaten Schlüsseln für die Client-Authentifizierung.
SSLCALocation	Überschreibt den Speicherort der Dateien der Zertifizierungsstelle (CA) für die Überprüfung von SSL-Serverzertifikaten.
StartAgentPollers	Die Anzahl vorab geforkter Instanzen asynchroner Zabbix-Agent-Poller.
StartAlerters	Die Anzahl vorab geforkter Instanzen von Alertern.
StartBrowserPollers	Die Anzahl vorab geforkter Instanzen von Browser-Datenpunkt-Pollern.
StartConnectors	Die Anzahl vorab geforkter Instanzen von Connector-Workern.
StartDBSyncers	Die Anzahl vorab geforkter Instanzen von Verlaufssynchronisierern.
StartDiscoverers	Die Anzahl vorab geforkter Instanzen von Discovery-Workern.
StartEscalators	Die Anzahl vorab geforkter Instanzen von Eskalatoren.
StartHistoryPollers	Die Anzahl vorab geforkter Instanzen von Verlaufs-Pollern.

Parameter	Beschreibung
StartHTTPAgentPollers	Die Anzahl vorab geforkter Instanzen asynchroner HTTP-Agent-Poller.
StartHTTPPollers	Die Anzahl vorab geforkter Instanzen von HTTP-Pollern.
StartIPMIPollers	Die Anzahl vorab geforkter Instanzen von IPMI-Pollern.
StartJavaPollers	Die Anzahl vorab geforkter Instanzen von Java-Pollern.
StartLLDProcessors	Die Anzahl vorab geforkter Instanzen von Workern für Low-Level-Discovery (LLD).
StartODBCPollers	Die Anzahl vorab geforkter Instanzen von ODBC-Pollern.
StartPingers	Die Anzahl vorab geforkter Instanzen von ICMP-Pingern.
StartPollersUnreachable	Die Anzahl vorab geforkter Instanzen von Pollern für nicht erreichbare Hosts (einschließlich IPMI und Java).
StartPollers	Die Anzahl vorab geforkter Instanzen von Pollern.
StartPreprocessors	Die Anzahl vorab gestarteter Instanzen von Vorverarbeitungs-Workern.
StartProxyPollers	Die Anzahl vorab geforkter Instanzen von Pollern für passive Proxys.
StartReportWriters	Die Anzahl vorab geforkter Instanzen von Berichtsschreibern.
StartSNMPPollers	Die Anzahl vorab geforkter Instanzen asynchroner SNMP-Poller.
StartSNMPTrapper	Wenn auf 1 gesetzt, wird ein SNMP-Trapper-Prozess gestartet.
StartTimers	Die Anzahl vorab geforkter Instanzen von Timern.
StartTrappers	Die Anzahl vorab geforkter Instanzen von Trappern.
StartVMwareCollectors	Die Anzahl vorab geforkter VMware-Collector-Instanzen.
StatsAllowedIP	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen externer Zabbix-Instanzen. Die Statistikabfrage wird nur von den hier aufgeführten Adressen akzeptiert.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix-Proxy, Agent, Webservice sowie bei SNMP-Prüfungen gewartet wird (außer bei SNMP-`walk[OID]`- und `get[OID]`-Datenpunkten).
TLSCAFile	Der vollständige Pfadname einer 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 einer Datei, die das Serverzertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherAll13	Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikatsbasierte Verschlüsselung.
TLSCipherCert13	Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für PSK-basierte Verschlüsselung.
TLSCipherPSK13	Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für PSK-basierte Verschlüsselung.
TLSCRLFile	Der vollständige Pfadname einer Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSFrontendAccept	Welche eingehenden Verbindungen vom Frontend akzeptiert werden sollen.
TLSFrontendCertIssuer	Zulässiger Aussteller des Frontend-Zertifikats.
TLSFrontendCertSubject	Zulässiger Betreff des Frontend-Zertifikats.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Servers enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSListen	Steuert TLS auf dem Trapper-Socket.
TmpDir	Das temporäre Verzeichnis.
TrapperTimeout	Gibt das Timeout in Sekunden an für: - das Abrufen historischer Daten von einem Zabbix-Proxy; - das Senden von Konfigurationsdaten an einen Zabbix-Proxy; - die Ausführung globaler Skripte oder entfernter Befehle auf dem Zabbix-Server.
TrendCacheSize	Die Größe des Trend-Caches.
TrendFunctionCacheSize	Die Größe des Trendfunktions-Caches.
UnavailableDelay	Legt fest, wie oft ein Host während des Nichtverfügbarkeitszeitraums auf Verfügbarkeit geprüft wird.
UnreachableDelay	Legt fest, wie oft ein Host während des Nichterreichbarkeitszeitraums auf Verfügbarkeit geprüft wird.

Parameter	Beschreibung
UnreachablePeriod	Legt fest, nach wie vielen Sekunden der Nichterreichbarkeit ein Host als nicht verfügbar behandelt wird.
User	Legt die Berechtigungen auf einen bestimmten, vorhandenen Benutzer im System ab.
ValueCacheSize	Die Größe des Verlaufswert-Caches.
Vault	Gibt den Vault-Anbieter an.
VaultDBPath	Gibt einen Speicherort an, von dem Datenbankzugangsdaten anhand von Schlüsseln abgerufen werden sollen.
VaultPrefix	Benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage.
VaultTLSCertFile	Der Name der SSL-Zertifikatdatei, die für die Client-Authentifizierung verwendet wird.
VaultTLSKeyFile	Der Name der SSL-Datei mit privatem Schlüssel, die für die Client-Authentifizierung verwendet wird.
VaultToken	Das HashiCorp-Vault-Authentifizierungstoken.
VaultURL	Die HTTP[S]-URL des Vault-Servers.
VMwareCacheSize	Die Shared-Memory-Größe zum Speichern von VMware-Daten.
VMwareFrequency	Die Verzögerung in Sekunden zwischen der Datenerfassung von einem einzelnen VMware-Dienst.
VMwarePerfFrequency	Die Verzögerung in Sekunden zwischen dem Abruf von Leistungsindikatorstatistiken von einem einzelnen VMware-Dienst.
VMwareTimeout	Die maximale Anzahl von Sekunden, die ein VMware-Collector auf eine Antwort von einem VMware-Dienst wartet.
WebDriverURL	Die HTTP[S]-URL der WebDriver-Schnittstelle.
WebServiceURL	HTTP[S]-URL zum Zabbix-Webservice im Format <host:port>/report.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass ein Parameter verpflichtend ist.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standardwerten, 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

AlertScriptsPath

Der Speicherort von benutzerdefinierten Warnskripten (hängt von der Installationsvariablen *datadir* zur Kompilierzeit ab).

Standard: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Erlaubt dem Server, als „root“ zu laufen. Wenn deaktiviert und der Server von „root“ gestartet wird, versucht der Server stattdessen, zum Benutzer „zabbix“ zu wechseln. Hat keine Auswirkung, wenn er unter einem normalen Benutzer gestartet wird.

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

CacheSize

Die Größe des Konfigurations-Caches in Byte. Die Größe des Shared Memory zum Speichern von Host-, Datenpunkt- und Auslöser-Daten.

Standard: 32M
 Bereich: 128K-64G

CacheUpdateFrequency

Dieser Parameter bestimmt, wie oft Zabbix die Aktualisierung des Konfigurationscaches in Sekunden durchführt. Siehe auch die Optionen für runtime control.

Standard: 10
 Bereich: 1-3600

DBHost

Der Datenbank-Hostname.
Bei MySQL führt `localhost` oder eine leere Zeichenfolge zur Verwendung eines Sockets. Bei PostgreSQL wird bei einer leeren Zeichenfolge der standardmäßige Unix-Domain-Socket verwendet; wenn ein Pfad gesetzt ist (z. B. /var/run/pgbouncer), wird der angegebene Unix-Domain-Socket verwendet.

Standard: `localhost`

DBName

Der Datenbankname.

Erforderlich: Ja

DBPassword

Das Datenbankpasswort. Kommentieren Sie diese Zeile aus, wenn kein Passwort verwendet wird.

DBPort

Der Datenbank-Port, wenn nicht der standardmäßige Unix-Domain-Socket verwendet wird.[3]

Standard für MySQL: 3306

Standard für PostgreSQL: 5432

Bereich: 1024-65535

DBSchema

Der Name des Datenbankschemas. Wird für PostgreSQL verwendet.

DBSocket

Der Pfad zur MySQL-Socket-Datei.[3]

DBUser

Der Datenbankbenutzer.

DBTLSConnect

Wenn diese Option auf einen der folgenden Werte gesetzt wird, wird die Verwendung einer TLS-Verbindung zur Datenbank erzwungen:
required - Verbindung über TLS herstellen
verify_ca - Verbindung über TLS herstellen und Zertifikat verifizieren
verify_full - Verbindung über TLS herstellen, Zertifikat verifizieren und prüfen, dass die in DBHost angegebene Datenbankidentität mit ihrem Zertifikat übereinstimmt

Bei MySQL ab Version 5.7.11 und bei PostgreSQL werden die folgenden Werte unterstützt: `required`, `verify_ca`, `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

Der vollständige Pfadname einer Datei, die die Zertifikate der CA(s) der obersten Ebene für die Verifizierung des Datenbankzertifikats enthält.

Verbindlich: nein (ja, wenn DBTLSConnect auf *verify_ca* oder *verify_full* gesetzt ist)

DBTLSCertFile

Der vollständige Pfadname einer Datei, die das Zertifikat des Zabbix Server zur Authentifizierung bei der Datenbank enthält.

DBTLSKeyFile

Der vollständige Pfadname einer Datei, die den privaten Schlüssel zur Authentifizierung bei der Datenbank enthält.

DBTLSCipher

Die Liste der Verschlüsselungs-Chiffren, die der Zabbix Server für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Wird nur für MySQL unterstützt.

DBTLSCipher13

Die Liste der Verschlüsselungs-Ciphersuites, die der Zabbix Server für das TLS-v1.3-Protokoll zulässt. Wird nur für MySQL ab Version 8.0.16 unterstützt.

DebugLevel

Geben Sie den Debug-Level an:
0 - grundlegende Informationen über das Starten und Stoppen von Zabbix-Prozessen
1 - kritische Informationen;
2 - Fehlerinformationen;
3 - Warnungen;
4 - zum Debuggen (erzeugt viele Informationen);
5 - erweitertes Debugging (erzeugt noch mehr Informationen).
Siehe auch die Optionen zur Laufzeitsteuerung.

Standard: 3
 Bereich: 0-5

EnableGlobalScripts

Aktiviert globale Skripte auf dem Zabbix Server.
 Hinweis: Die Ausführung globaler Skripte ist standardmäßig aktiviert. Für Neuinstallationen ab Zabbix 7.0 ist EnableGlobalScripts jedoch explizit auf 0 (deaktiviert) gesetzt.

Standard: 1
 Werte: 0 - deaktivieren; 1 - aktivieren

ExportDir

Das Verzeichnis für den Echtzeit-Export von Ereignissen, Verlauf und Trends im newline-delimited-JSON-Format. Wenn gesetzt, aktiviert es den Echtzeit-Export.

ExportFileSize

Die maximale Größe pro Exportdatei in Byte. Wird für die Rotation verwendet, wenn `ExportDir` gesetzt ist.

Standard: 1G
 Bereich: 1M-1G

ExportType

Die Liste der durch Kommas getrennten Entitätstypen (Ereignisse, Verlauf, Trends) für den Echtzeit-Export (standardmäßig alle Typen). Nur gültig, wenn ExportDir gesetzt ist.
Beachten Sie, dass dies ein Konfigurationsfehler ist und der Server nicht startet, wenn ExportType angegeben ist, ExportDir jedoch nicht.

Beispiel für den Export von Verlauf und Trends:

```
ExportType=history,trends
```

Beispiel nur für den Ereignisexport:

```
ExportType=events
```

ExternalScripts

Der Speicherort externer Skripte (hängt von der Installationsvariablen `datadir` zur Kompilierzeit ab).

Standard: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Der Speicherort von fping6. Stellen Sie sicher, dass die fping6-Binärdatei 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

Der Speicherort von fping. Stellen Sie sicher, dass die fping-Binärdatei Root gehört und das SUID-Flag gesetzt ist.

Standard: `/usr/sbin/fping`

FrontendAllowedIP

Eine durch Kommas getrennte Liste von IP-Adressen oder CIDR-Bereichen, von denen aus Verbindungen vom Frontend erlaubt sind. Verbindungen zum Frontend werden nur von den hier aufgeführten Adressen akzeptiert, wenn dieser Parameter gesetzt ist. Standardmäßig werden alle Verbindungen für Frontend-Anfragen 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 zu erlauben.

Beispiel:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Der Knotenname des Hochverfügbarkeitsclusters. Wenn leer, arbeitet der Server im Standalone-Modus und es wird ein Knoten mit leerem Namen erstellt.

HistoryCacheSize

Die Größe des Verlaufs-Caches in Byte. Die Größe des Shared Memory zum Speichern von Verlaufsdaten.

Standard: 16M
 Bereich: 128K-16G

HistoryIndexCacheSize

Die Größe des Verlaufsindex-Caches in Byte. Die Größe des Shared Memory zum Indizieren der im Verlaufs-Cache gespeicherten Verlaufsdaten. Für das Zwischenspeichern eines Datenpunkts benötigt der Index-Cache ungefähr 100 Byte.

Standard: 4M
 Bereich: 128K-16G

HistoryStorageDateIndex

Aktiviert die Vorverarbeitung von Verlaufswerten im Verlaufsspeicher, um Werte basierend auf dem Datum in verschiedenen Indizes zu speichern.

Standard: 0
 Werte: 0 - deaktivieren; 1 - aktivieren

HistoryStorageURL

Die HTTP[S]-URL des Verlaufsspeichers. Dieser Parameter wird für die Einrichtung von Elasticsearch verwendet.

HistoryStorageTypes

Eine durch Kommas getrennte Liste von Werttypen, die an den Verlaufsspeicher gesendet werden sollen. Dieser Parameter wird für die Einrichtung von Elasticsearch verwendet.

Standard: uint,dbl,str,log,text,json

HousekeepingFrequency

Dieser Parameter bestimmt, wie oft Zabbix das Housekeeping-Verfahren in Stunden ausführt. Housekeeping entfernt veraltete Informationen aus der Datenbank.
Hinweis: Um zu verhindern, dass der Housekeeper überlastet wird (zum Beispiel, wenn die Verlaufs- und Trendzeiträume stark verkürzt werden), werden in einem Housekeeping-Zyklus pro Datenpunkt nicht mehr als 4 × HousekeepingFrequency Stunden veralteter Informationen gelöscht. Wenn HousekeepingFrequency also 1 ist, werden pro Zyklus nicht mehr als 4 Stunden veralteter Informationen gelöscht (beginnend mit dem ältesten Eintrag).
Hinweis: Um die Last beim Server-Start zu verringern, wird Housekeeping für 30 Minuten nach dem Start des Servers verschoben. Wenn HousekeepingFrequency also 1 ist, wird das allererste Housekeeping-Verfahren nach dem Start des Servers nach 30 Minuten ausgeführt und danach jeweils mit einer Verzögerung von einer Stunde wiederholt.
Es ist möglich, automatisches Housekeeping zu deaktivieren, indem HousekeepingFrequency auf 0 gesetzt wird. In diesem Fall kann das Housekeeping-Verfahren nur über die Runtime-Control-Option *housekeeper_execute* gestartet werden, und der Zeitraum veralteter Informationen, der in einem Housekeeping-Zyklus gelöscht wird, 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 für runtime control sowie Details zum Housekeeping-Verfahren.

Standard: 1
 Bereich: 0-24

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden. Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Platzhalterzeichen Asterisk für den Musterabgleich unterstützt. Siehe besondere Hinweise zu Einschränkungen.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Die IP-Adresse (oder der Hostname) des Zabbix Java gateway. Nur erforderlich, wenn Java-Poller gestartet werden.

JavaGatewayPort

Der Port, auf dem das Zabbix Java gateway lauscht.

Standard: 10052
 Bereich: 1024-32767

ListenBacklog

Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
Der Standardwert ist eine fest kodierte Konstante, die vom System abhängt.
Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementierungsspezifische Maximum“ gekürzt werden.

Standard: SOMAXCONN
 Bereich: 0 - INT_MAX

ListenIP

Eine Liste von durch Kommas getrennten IP-Adressen, auf denen der Trapper lauschen soll.
Wenn dieser Parameter fehlt, lauscht der Trapper auf allen Netzwerkschnittstellen.

Standard: 0.0.0.0

ListenPort

Der Port, auf dem der Trapper lauscht.

Standard: 10051
 Bereich: 1024-32767

LoadModule

Das Modul, das beim Start des Servers geladen werden soll. Module werden verwendet, um die Funktionalität des Servers zu erweitern. Das Modul muss sich in dem durch LoadModulePath angegebenen Verzeichnis befinden, oder dem Modulnamen muss der Pfad vorangestellt werden. 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

Der vollständige Pfad zum Speicherort der Server-Module. Der Standardwert hängt von den Kompilierungsoptionen ab.

LogFile

Der Name der Protokolldatei.

Verbindlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe der 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
 Erforderlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogSlowQueries

Legt fest, wie lange eine Datenbankabfrage dauern darf, bevor sie in Millisekunden protokolliert wird.
0 - langsame Abfragen nicht protokollieren.
Diese Option wird ab DebugLevel=3 aktiviert.

Standard: 0
 Bereich: 0-3600000

LogType

Der Typ der Log-Ausgabe:
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.

Standard: `file`

MaxConcurrentChecksPerPoller

Die maximale Anzahl asynchroner Prüfungen, die gleichzeitig von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller ausgeführt werden können. Siehe StartHTTPAgentPollers, StartAgentPollers und StartSNMPPollers.

Standard: 1000
 Bereich: 1-1000

MaxHousekeeperDelete

Die Tabelle „housekeeper" enthält „tasks" für das Housekeeping-Verfahren im Format: `[housekeeperid]`, `[object]`, `[objectid]`.
 Pro task und Housekeeping-Zyklus werden aus den Tabellen `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` und `problem` nicht mehr als `MaxHousekeeperDelete` Zeilen gelöscht.
 Wenn der Wert auf 0 gesetzt ist, wird überhaupt keine Begrenzung verwendet. In diesem Fall müssen Sie wissen, was Sie tun, damit die Datenbank nicht überlastet wird.[2]
Dieser Parameter gilt nur für das Löschen von Daten, die von bereits gelöschten Datenpunkten zurückgeblieben sind.
 Siehe auch die Details zum Housekeeping-Verfahren.

Standard: 5000
 Bereich: 0-1000000

NodeAddress

IP oder Hostname mit optionalem Port, um zu überschreiben, wie sich das Frontend mit dem Server verbinden soll.
Format: <address>[:<port>]

Wenn keine IP oder kein Hostname 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: HANodeName-Parameter; Aktivieren der Hochverfügbarkeit.

Standard: `localhost:10051`

PidFile

Name der PID-Datei.

Standard: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Legt fest, wie oft Zabbix Probleme für gelöschte Auslöser in Sekunden löscht.
 Siehe auch die Details zum Housekeeping-Verfahren.

Standard: 60
 Bereich: 1-3600

ProxyConfigFrequency

Legt fest, wie oft der Zabbix Server Konfigurationsdaten in Sekunden an einen Zabbix Proxy sendet. Wird nur für Proxys im passiven Modus verwendet.

Standard: 10
 Bereich: 1-604800

ProxyDataFrequency

Legt fest, wie oft der Zabbix Server Verlaufsdaten von einem Zabbix Proxy in Sekunden anfordert. Wird nur für Proxys im passiven Modus verwendet.

Standard: 1
 Bereich: 1-3600

ServiceManagerSyncFrequency

Legt fest, wie oft Zabbix die Konfiguration eines Service-Managers in Sekunden synchronisiert.

Standard: 60
 Bereich: 1-3600

SMSDevices

Eine Liste von durch Kommas getrennten Modemdateien, die vom Zabbix Server verwendet werden dürfen.
Das Senden von SMS ist nicht möglich, wenn dieser Parameter nicht gesetzt ist.

Beispiel:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Temporäre Datei, die für die Übergabe von Daten vom SNMP-Trap-Daemon an den Server verwendet wird.
Muss mit der Einstellung in der Konfigurationsdatei von zabbix_trap_receiver.pl oder SNMPTT übereinstimmen.

Standard: `/tmp/zabbix_traps.tmp`

SocketDir

Verzeichnis zum Speichern von IPC-Sockets, die von internen Zabbix-Services verwendet werden.

Standard: `/tmp`

SourceIP

Quell-IP-Adresse 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 von Skript-Datenpunkten
- JavaScript-HTTP-Anfragen der Vorverarbeitung
- das Senden von Benachrichtigungs-E-Mails (Verbindungen zum SMTP-Server)
- webhook-Benachrichtigungen (JavaScript-HTTP-Verbindungen)
- Verbindungen zum Vault

SSHKeyLocation

Speicherort der öffentlichen und privaten Schlüssel für SSH-Prüfungen und Aktionen.

SSLCertLocation

Speicherort der SSL-Client-Zertifikatsdateien für die Client-Authentifizierung.
Dieser Parameter wird nur in der Web-Überwachung verwendet.

SSLKeyLocation

Speicherort der privaten SSL-Schlüsseldateien für die Client-Authentifizierung.
Dieser Parameter wird nur für das Web-Monitoring verwendet.

SSLCALocation

Überschreibt den Speicherort der Zertifizierungsstellen-(CA-)Dateien für die SSL-Serverzertifikatsprüfung. Falls nicht gesetzt, wird das systemweite Verzeichnis verwendet.
Beachten Sie, dass der Wert dieses Parameters als libcurl-Option CURLOPT_CAPATH gesetzt wird. Bei libcurl-Versionen vor 7.42.0 hat dies nur dann eine Wirkung, wenn libcurl für die Verwendung von OpenSSL kompiliert wurde. Weitere Informationen finden Sie auf der cURL-Webseite.
Dieser Parameter wird im Web-Monitoring und bei der SMTP-Authentifizierung verwendet.

StartAgentPollers

Die Anzahl der vorab geforkten Instanzen von Zabbix-Agent-Pollern. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartAlerters

Die Anzahl der vorab geforkten Instanzen von Alertern.

Standard: 3
 Bereich: 1-100

StartBrowserPollers

Die Anzahl der vorab geforkten Instanzen von Browser-Datenpunkt-Pollern.

Standard: 1
 Bereich: 0-1000

StartConnectors

Die Anzahl der vorab geforkten Instanzen von connector workers. Der Connector-Manager-Prozess wird automatisch gestartet, wenn ein Connector-Worker gestartet wird.

Standard: 0
 Bereich: 0-1000

StartDBSyncers

Die Anzahl der vorab geforkten Instanzen von Verlaufssynchronisierern.
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

Die Anzahl der vorab geforkten Instanzen von Discovery-Workern[1].

Standard: 5
 Bereich: 0-1000

StartEscalators

Die Anzahl der vorab geforkten Instanzen von Eskalatoren.

Standard: 1
 Bereich: 1-100

StartHistoryPollers

Die Anzahl der vorab per Fork gestarteten Instanzen von History-Pollern.
Nur für berechnete Prüfungen erforderlich.

Standard: 5
 Bereich: 0-1000

StartHTTPAgentPollers

Die Anzahl der vorab geforkten Instanzen von HTTP-Agent-Pollern. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartHTTPPollers

Die Anzahl der vorab geforkten Instanzen von HTTP-Pollern[1].

Standard: 1
 Bereich: 0-1000

StartIPMIPollers

Die Anzahl der vorab geforkten Instanzen von IPMI-Pollern.

Standard: 0
 Bereich: 0-1000

StartJavaPollers

Die Anzahl der vorab geforkten Instanzen von Java-Pollern[1].

Standard: 0
 Bereich: 0-1000

StartLLDProcessors

Die Anzahl der vorab geforkten Instanzen von Low-Level-Discovery-(LLD)-Workern[1].
Der LLD-Manager-Prozess wird automatisch gestartet, wenn ein LLD-Worker gestartet wird.

Standard: 2
 Bereich: 1-100

StartODBCPollers

Die Anzahl der vorab geforkten Instanzen von ODBC-Pollern[1].

Standard: 1
 Bereich: 0-1000

StartPingers

Die Anzahl der vorab geforkten Instanzen von ICMP-Pingern[1].

Standard: 1
 Bereich: 0-1000

StartPollersUnreachable

Die Anzahl der vorab geforkten Instanzen von Pollern für nicht erreichbare Hosts (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

Die Anzahl der vorab geforkten Instanzen von Pollern[1].

Standard: 5
 Bereich: 0-1000

StartPreprocessors

Die Anzahl der vorab gestarteten Instanzen von Vorverarbeitungs-worker[1]-Threads sollte nicht kleiner als die Anzahl der verfügbaren CPU-Kerne eingestellt werden. Mehr worker sollten eingestellt werden, wenn die Vorverarbeitung nicht CPU-gebunden ist und viele Netzwerkanfragen enthält.

Standard: 16
 Bereich: 1-1000

StartProxyPollers

Die Anzahl der vorab per Fork gestarteten Instanzen von Pollern für passive Proxys[1].

Standard: 1
 Bereich: 0-250

StartReportWriters

Die Anzahl der vorab geforkten Instanzen von Report-Writern.
Wenn auf 0 gesetzt, ist die geplante Berichtserstellung deaktiviert.
Der Report-Manager-Prozess wird automatisch gestartet, wenn ein Report-Writer gestartet wird.

Standard: 0
 Bereich: 0-100

StartSNMPPollers

Die Anzahl der vorab geforkten Instanzen von SNMP-Pollern. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartSNMPTrapper

Wenn auf 1 gesetzt, wird ein SNMP trapper-Prozess gestartet.

Standard: 0
 Bereich: 0-1

StartTimers

Die Anzahl der vorab geforkten Instanzen von Timern.
Timer verarbeiten Wartungszeiträume.

Standard: 1
 Bereich: 1-1000

StartTrappers

Die Anzahl der vorab geforkten Instanzen von Trappern[1].
Trapper akzeptieren eingehende Verbindungen von Zabbix sender, aktiven Agenten und aktiven Proxys.

Standard: 5
 Bereich: 0-1000

StartVMwareCollectors

Die Anzahl der vorab geforkten Instanzen des VMware collector.

Standard: 0
 Bereich: 0-250

StatsAllowedIP

Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen externer Zabbix-Instanzen. Statistikanfragen werden nur von den hier aufgeführten Adressen akzeptiert. 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' 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.

Beispiel:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Timeout

Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Proxy, Agent, Web-Service und Legacy-SNMP-Prüfungen (einzelne OID-Nummer oder Zeichenfolge) gewartet wird.

Dieser Parameter definiert die Dauer für verschiedene Kommunikationsvorgänge:

- Ausführung entfernter Befehle auf dem Zabbix Agent
- Ausführung von SSH-/Telnet-Befehlen
- Anfragen an den Zabbix-Web-Service
- Kommunikations-Timeout für Medientyp-Testanfragen und die Methode `history.push`
- Neuplanung von Datenpunkten, wenn eine IPMI-Schnittstelle nicht mehr verfügbar ist
- Senden einer Antwort an den Zabbix Proxy, wenn der Datenaustausch aufgrund von Berechtigungs- oder Verschlüsselungsproblemen fehlschlägt
- Zeitlimit für asynchrone IPC-Sockets und Runtime-Control-Optionen
- JMX-Verbindungen
- Abrufen von Statistiken von einem entfernten Zabbix Proxy oder Server
- Senden von Antworten an das Zabbix Frontend
- DNS-Anfragen asynchroner Poller
- Antwort für den Heartbeat aktiver Prüfungen
- Abrufen von Zabbix-Agent-Daten (Werten) von aktiven Agents
- Abrufen von Daten von 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 Prüfungen verwendet, für die im Frontend Einstellungen für flexible timeout konfiguriert sind (global, auf Proxy-Ebene oder pro Datenpunkt).
Beispielsweise verwenden SNMP-Datenpunkte `walk[OID]` und `get[OID]` das im Frontend konfigurierte Timeout; Legacy-SNMP-Prüfungen verwenden weiterhin den Timeout-Wert des Servers.

Standard: 3
 Bereich: 1-30

TLSCAFile

Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname einer Datei, die das Serverzertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCipherAll

Der GnuTLS-Prioritätsstring oder der OpenSSL-Cipher-String (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien für Cipher-Suites bei zertifikats- und PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien für Chiffresuites bei zertifikat- und PSK-basierter 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

Der GnuTLS-Prioritätsstring oder der OpenSSL-Chiffrierstring (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für zertifikatbasierte 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

Die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für zertifikatbasierte Verschlüsselung.

TLSCipherPSK

Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien der Chiffre-Suite für PSK-basierte 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

Die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien für Chiffriersuiten für PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Der vollständige Pfadname einer Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSFrontendAccept

Welche eingehenden Verbindungen vom Frontend akzeptiert werden.

Es können mehrere Werte angegeben werden, durch Komma getrennt:

- unencrypted - Verbindungen ohne Verschlüsselung akzeptieren.
- cert - mit TLS und einem Zertifikat gesicherte Verbindungen akzeptieren.

Standard: `unencrypted`

TLSFrontendCertIssuer

Zulässiger Aussteller des Frontend-Zertifikats.

TLSFrontendCertSubject

Zulässiger Betreff des Frontend-Zertifikats.

TLSKeyFile

Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Server enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSListen

Steuert TLS auf dem Trapper-Socket.

Unterstützte Werte:

- required - nur TLS-Verbindungen akzeptieren

TmpDir

Das temporäre Verzeichnis.

Standard: `/tmp`

TrapperTimeout

Gibt das Timeout in Sekunden an für:

- den Abruf von Verlaufsdaten vom Zabbix Proxy
- das Senden von Konfigurationsdaten an den Zabbix Proxy
- die Ausführung globaler Skripte oder entfernter Befehle auf dem Zabbix Server

Standard: 300
 Bereich: 1-300

TrendCacheSize

Die Größe des Trend-Caches in Byte.
Die Größe des Shared Memory zum Speichern von Trenddaten.

Standard: `4M`
 Bereich: 128K-16G

TrendFunctionCacheSize

Die Größe des Trendfunktions-Caches in Byte.
Die Größe des Shared Memory zum Zwischenspeichern berechneter Trendfunktionsdaten.

Standard: `4M`
 Bereich: 128K-2G

UnavailableDelay

Legt fest, wie oft der Host während des Zeitraums der Nichtverfügbarkeit in Sekunden auf Verfügbarkeit geprüft wird.

Standard: 60
 Bereich: 1-3600

UnreachableDelay

Legt fest, wie oft der Host während des Zeitraums der Unerreichbarkeit in Sekunden auf Verfügbarkeit geprüft wird.

Standard: 15
 Bereich: 1-3600

UnreachablePeriod

Legt fest, nach wie vielen Sekunden der Unerreichbarkeit ein Host als nicht verfügbar behandelt wird.

Standard: 45
 Bereich: 1-3600

Benutzer

Privilegien auf einen bestimmten, auf dem System vorhandenen Benutzer reduzieren.
Hat nur Wirkung, wenn als 'root' ausgeführt und AllowRoot deaktiviert ist.

Standard: `zabbix`

ValueCacheSize

Die Größe des Verlaufswert-Caches in Byte.
Die Shared-Memory-Größe für das Caching von Anfragen zu Datenpunkt-Verlaufsdaten.
Wenn der Wert auf 0 gesetzt wird, wird der Wert-Cache deaktiviert (nicht empfohlen).
Wenn dem Wert-Cache der Shared Memory ausgeht, wird alle 5 Minuten eine Warnmeldung in das Server-Log geschrieben.

Standard: `8M`
 Bereich: 0,128K-64G

Vault

Gibt 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`

VaultDBPath

Vault-Pfad oder Abfrage, je nach Vault, aus dem die Zugangsdaten für die Datenbank anhand von Schlüsseln abgerufen werden.

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

Ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage, abhängig vom Vault. Die am besten geeigneten Standardwerte werden verwendet, wenn nichts angegeben ist.
Beachten Sie, dass für HashiCorp nach dem Mountpoint automatisch „data“ angehängt wird, wenn VaultPrefix nicht angegeben ist.

Beispielpräfix für Hashicorp:

```
v1/secret/data/zabbix/
```

Beispielpräfix für Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Der Name der SSL-Zertifikatsdatei, 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

Der Name der SSL-Private-Key-Datei, die für die Client-Authentifizierung verwendet wird.
 Die Private-Key-Datei 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

Das HashiCorp-Vault-Authentifizierungstoken, das ausschließlich für den Zabbix Server mit Nur-Lese-Berechtigung für die in Vault-Makros angegebenen Pfade und mit Nur-Lese-Berechtigung für den im optionalen Konfigurationsparameter VaultDBPath angegebenen Pfad erzeugt worden sein sollte.
Es ist ein Fehler, wenn VaultToken und die Umgebungsvariable VAULT_TOKEN gleichzeitig definiert sind.

Verpflichtend: Ja, wenn Vault auf *HashiCorp* gesetzt ist; andernfalls nein

VaultURL

Die HTTP[S]-URL des Vault-Servers. Das systemweite Verzeichnis für CA-Zertifikate wird verwendet, wenn SSLCALocation nicht angegeben ist.

Standard: `https://127.0.0.1:8200`

VMwareCacheSize

Die Größe des Shared Memory zum Speichern von VMware-Daten.
Eine interne VMware-Prüfung zabbix[vmware,buffer,...] kann verwendet werden, um die Nutzung des VMware-Cache zu überwachen (siehe Interne Prüfungen).
Beachten Sie, dass Shared Memory nicht zugewiesen wird, wenn keine vmware collector-Instanzen für den Start konfiguriert sind.

Standard: `8M`
 Bereich: 256K-2G

VMwareFrequency

Die Verzögerung in Sekunden zwischen den Datenerfassungen von einem einzelnen VMware-Dienst.
Diese Verzögerung sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Überwachungsdatenpunkts gesetzt werden.

Standard: 60
 Bereich: 10-86400

VMwarePerfFrequency

Die Verzögerung in Sekunden zwischen dem Abruf von Leistungszählerstatistiken von einem einzelnen VMware-Dienst. Diese Verzögerung sollte auf das kleinste Aktualisierungsintervall eines VMware-Monitoring-Datenpunkts gesetzt werden, der VMware-Leistungszähler verwendet.

Standard: 60
 Bereich: 10-86400

VMwareTimeout

Die maximale Anzahl von Sekunden, die ein VMware-Collector auf eine Antwort vom VMware-Dienst (vCenter oder ESX-Hypervisor) wartet.

Standard: 10
 Bereich: 1-300

WebServiceURL

Die HTTP[S]-URL zum Zabbix-Webservice im Format `http[s]://host:port/report`.

Beispiel:

```
WebServiceURL=http://localhost:10053/report
```

Hinweis: Das Schema (`http://`) darf nur bei Nicht-TLS-Verbindungen (HTTP) weggelassen werden; wenn TLS konfiguriert ist, muss `https://` verwendet werden.

WebDriverURL

HTTP[S]-URL der WebDriver-Schnittstelle.

Beispiel (verwendet mit dem eigenständigen Selenium WebDriver-Server):

```
WebDriverURL=http://localhost:4444
```

Fußnoten

[1] Beachten Sie, dass zu viele Datenerfassungsprozesse (Poller, Poller für nicht erreichbare Hosts, ODBC-Poller, HTTP-Poller, Java-Poller, Pinger, Trapper, Proxy-Poller) zusammen mit IPMI-Manager, SNMP-Trapper, Präprozessor-Workern und Discovery-Workern das Dateideskriptor-Limit pro Prozess für den Präprozessor-Manager ausschöpfen können.

> **Warning:**
> Das Ausschöpfen des Dateideskriptor-Limits führt dazu, dass der Zabbix Server stoppt, typischerweise kurz nach dem Start, manchmal jedoch auch erst später. Um solche Probleme zu vermeiden, prüfen Sie die Zabbix-Server-Konfigurationsdatei, um die Anzahl gleichzeitiger Prüfungen und Prozesse zu optimieren. Stellen Sie außerdem bei Bedarf sicher, dass das Dateideskriptor-Limit ausreichend hoch gesetzt ist, indem Sie die Systemlimits prüfen und anpassen.

[2] Wenn viele Datenpunkte gelöscht werden, erhöht dies die Last auf der Datenbank, da der Housekeeper alle Verlaufsdaten entfernen muss, die diese Datenpunkte hatten. Wenn wir beispielsweise nur 1 Datenpunktprototyp aus der Vorlage entfernen müssen, diese Vorlage jedoch mit 50 Hosts verknüpft ist und für jeden Host der Prototyp zu 100 realen Datenpunkten erweitert wird, müssen insgesamt 5000 Datenpunkte entfernt werden (1*50*100). Wenn für MaxHousekeeperDelete der Wert 500 gesetzt ist (MaxHousekeeperDelete=500), muss der Housekeeper-Prozess in einem Zyklus bis zu 2500000 Werte (5000*500) für die gelöschten Datenpunkte aus den Verlaufs- und Trendtabellen entfernen.

[3] DBSocket und DBPort schließen sich in der Server-Konfiguration gegenseitig aus. Geben Sie nur einen von beiden an oder lassen Sie beide undefiniert.

2 Zabbix Proxy

Übersicht

Die von der Zabbix-Proxy-Konfigurationsdatei (zabbix_proxy.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
AllowRoot	Erlaubt, den Proxy als 'root' auszuführen.
AllowUnsupportedDBVersions	Erlaubt dem Proxy, mit nicht unterstützten Datenbankversionen zu arbeiten.
CacheSize	Die Größe des Konfigurations-Caches.
ConfigFrequency	Dieser Parameter ist **veraltet** (verwenden Sie stattdessen ProxyConfigFrequency). Wie oft der Proxy Konfigurationsdaten vom Zabbix-Server in Sekunden abruft.
DataSenderFrequency	Der Proxy sendet die gesammelten Daten alle N Sekunden an den Server.
DBHost	Der Hostname der Datenbank.
DBName	Der Datenbankname oder der Pfad zur Datenbankdatei für SQLite3.
DBPassword	Das Datenbankpasswort.
DBPort	Der Datenbankport, wenn kein lokaler Socket verwendet wird.
DBSchema	Der Name des Datenbankschemas. Wird für PostgreSQL verwendet.

Parameter	Beschreibung
DBSocket	Der Pfad zur MySQL-Socket-Datei.
DBUser	Der Datenbankbenutzer.
DBTLSConnect	Wenn diese Option auf den angegebenen Wert gesetzt wird, wird die Verwendung einer TLS-Verbindung zur Datenbank erzwungen.
DBTLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) zur Überprüfung des Datenbankzertifikats enthält.
DBTLSCertFile	Der vollständige Pfadname einer Datei, die das Zabbix-Proxy-Zertifikat zur Authentifizierung bei der Datenbank enthält.
DBTLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel zur Authentifizierung bei der Datenbank enthält.
DBTLSCipher	Die Liste der Verschlüsselungs-Chiffren, die der Zabbix-Proxy für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Nur für MySQL unterstützt.
DBTLSCipher13	Die Liste der Verschlüsselungs-Chiffresuites, die der Zabbix-Proxy für das TLS-v1.3-Protokoll zulässt. Nur für MySQL unterstützt, ab Version 8.0.16.
DebugLevel	Die Debug-Stufe.
EnableRemoteCommands	Ob Remote-Befehle vom Zabbix-Server erlaubt sind.
ExternalScripts	Der Speicherort externer Skripte.
Fping6Location	Der Speicherort von fping6.
FpingLocation	Der Speicherort von fping.
HistoryCacheSize	Die Größe des Verlaufs-Caches.
HistoryIndexCacheSize	Die Größe des Verlaufsindex-Caches.
Hostname	Ein eindeutiger Proxy-Name, bei dem Groß-/Kleinschreibung beachtet wird.
HostnameItem	Der Datenpunkt, der zum Setzen von Hostname verwendet wird, wenn dieser nicht definiert ist.
HousekeepingFrequency	Wie oft Zabbix die Housekeeping-Prozedur in Stunden ausführt.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
JavaGateway	Die IP-Adresse (oder der Hostname) des Zabbix Java gateway.
JavaGatewayPort	Der Port, auf dem das Zabbix Java gateway lauscht.
ListenBacklog	Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
ListenIP	Eine durch Kommas getrennte Liste von IP-Adressen, auf denen der Trapper lauschen soll.
ListenPort	Der Port, auf dem der Trapper lauscht.
LoadModule	Das Modul, das beim Start des Proxy geladen wird.
LoadModulePath	Der vollständige Pfad zum Speicherort der Proxy-Module.
LogFile	Der Name der Protokolldatei.
LogFileSize	Die maximale Größe der Protokolldatei.
LogRemoteCommands	Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.
LogSlowQueries	Wie lange eine Datenbankabfrage dauern darf, bevor sie protokolliert wird.
LogType	Der Typ der Protokollausgabe.
MaxConcurrentChecksPerPoller	Die maximale Anzahl asynchroner Prüfungen, die gleichzeitig von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller ausgeführt werden können.
PidFile	Der Name der PID-Datei.
ProxyBufferMode	Gibt den Speichermechanismus für Verlaufs-, Discovery- und Autoregistrierungsdaten an (Festplatte/Speicher/Hybrid).
ProxyConfigFrequency	Wie oft der Proxy Konfigurationsdaten vom Zabbix-Server in Sekunden abruft.
ProxyLocalBuffer	Der Proxy behält Daten N Stunden lokal, auch wenn die Daten bereits mit dem Server synchronisiert wurden.
ProxyMemoryBufferAge	Das maximale Alter von Daten im Proxy-Speicherpuffer in Sekunden.
ProxyMemoryBufferSize	Die Größe des Shared-Memory-Caches für gesammelte Verlaufs-, Discovery- und Autoregistrierungsdaten.
ProxyMode	Der Betriebsmodus des Proxy (aktiv/passiv).
ProxyOfflineBuffer	Der Proxy behält Daten N Stunden lang, falls keine Verbindung zum Zabbix-Server besteht.
Server	Wenn ProxyMode auf den aktiven Modus gesetzt ist: IP-Adresse oder DNS-Name des Zabbix-Servers (address:port) oder Cluster (address:port;address2:port), von dem Konfigurationsdaten abgerufen und an den Daten gesendet werden. Wenn ProxyMode auf den passiven Modus gesetzt ist: Durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen des Zabbix-Servers.
SNMPTrapperFile	Die temporäre Datei, die zur Übergabe von Daten vom SNMP-Trap-Daemon an den Proxy verwendet wird.
SocketDir	Das Verzeichnis zum Speichern der von internen Zabbix-Diensten verwendeten IPC-Sockets.
SourceIP	Die Quell-IP-Adresse.
SSHKeyLocation	Der Speicherort der öffentlichen und privaten Schlüssel für SSH-Prüfungen und Aktionen.

Parameter	Beschreibung
SSLCertLocation	Der Speicherort der SSL-Clientzertifikatdateien für die Client-Authentifizierung.
SSLKeyLocation	Der Speicherort der SSL-Dateien mit privaten Schlüsseln für die Client-Authentifizierung.
SSLCALocation	Überschreibt den Speicherort der Zertifizierungsstellen-Dateien (CA) für die Überprüfung von SSL-Serverzertifikaten.
StartAgentPollers	Die Anzahl vorab geforkter Instanzen asynchroner Zabbix-Agent-Poller.
StartBrowserPollers	Die Anzahl vorab geforkter Instanzen von Browser-Datenpunkt-Pollern.
StartDBSyncers	Die Anzahl vorab geforkter Instanzen von Verlaufssynchronisierern.
StartDiscoverers	Die Anzahl vorab geforkter Instanzen von Discovery-Workern.
StartHTTPAgentPollers	Die Anzahl vorab geforkter Instanzen asynchroner HTTP-Agent-Poller.
StartHTTPPollers	Die Anzahl vorab geforkter Instanzen von HTTP-Pollern.
StartIPMIPollers	Die Anzahl vorab geforkter Instanzen von IPMI-Pollern.
StartJavaPollers	Die Anzahl vorab geforkter Instanzen von Java-Pollern.
StartODBCPollers	Die Anzahl vorab geforkter Instanzen von ODBC-Pollern.
StartPingers	Die Anzahl vorab geforkter Instanzen von ICMP-Pingern.
StartPollersUnreachable	Die Anzahl vorab geforkter Instanzen von Pollern für nicht erreichbare Hosts (einschließlich IPMI und Java).
StartPollers	Die Anzahl vorab geforkter Instanzen von Pollern.
StartPreprocessors	Die Anzahl vorab gestarteter Instanzen von Präprozessierungs-Workern.
StartSNMPPollers	Die Anzahl vorab geforkter Instanzen asynchroner SNMP-Poller.
StartSNMPTrapper	Wenn auf 1 gesetzt, wird ein SNMP-Trapper-Prozess gestartet.
StartTrappers	Die Anzahl vorab geforkter Instanzen von Trappern.
StartVMwareCollectors	Die Anzahl vorab geforkter VMware-Collector-Instanzen.
StatsAllowedIP	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen externer Zabbix-Instanzen. Die Statistikabfrage wird nur von den hier aufgeführten Adressen akzeptiert.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix-Proxy, Agent, Webservice sowie bei SNMP-Prüfungen gewartet wird (außer bei SNMP-`walk[OID]`- und `get[OID]`-Datenpunkten).
TLSAccept	Welche eingehenden Verbindungen vom Zabbix-Server akzeptiert werden sollen.
TLSCAFile	Der vollständige Pfadname einer 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 einer Datei, die das Serverzertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherAll13	Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikatsbasierte Verschlüsselung.
TLSCipherCert13	Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für PSK-basierte Verschlüsselung.
TLSCipherPSK13	Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl von Chiffresuites für PSK-basierte Verschlüsselung.
TLSConnect	Wie sich der Proxy mit dem Zabbix-Server verbinden soll.
TLSCRLFile	Der vollständige Pfadname einer Datei, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Proxy enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSListen	Steuert TLS auf dem Trapper-Socket.
TLSPSKFile	Der vollständige Pfadname einer Datei, die den vorinstallierten Schlüssel des Proxy enthält und für verschlüsselte Kommunikation mit dem Zabbix-Server verwendet wird.
TLSPSKIdentity	Die Identitätszeichenfolge des vorinstallierten Schlüssels, verwendet für verschlüsselte Kommunikation mit dem Zabbix-Server.
TLSServerCertIssuer	Der zulässige Aussteller des Serverzertifikats.
TLSServerCertSubject	Der zulässige Betreff des Serverzertifikats.
TmpDir	Das temporäre Verzeichnis.

Parameter	Beschreibung
TrapperTimeout	Gibt das Timeout in Sekunden an für: - das Abrufen von Konfigurationsdaten vom Zabbix-Server; - die Ausführung globaler Skripte oder die Ausführung von Remote-Befehlen.
UnavailableDelay	Wie oft ein Host während des Zeitraums der Nichtverfügbarkeit auf Verfügbarkeit geprüft wird.
UnreachableDelay	Wie oft ein Host während des Zeitraums der Nichterreichbarkeit auf Verfügbarkeit geprüft wird.
UnreachablePeriod	Nach wie vielen Sekunden der Nichterreichbarkeit der Host als nicht verfügbar behandelt wird.
User	Legt die Berechtigungen auf einen bestimmten, vorhandenen Benutzer im System ab.
Vault	Der Vault-Anbieter.
VaultDBPath	Der Speicherort, von dem Datenbankzugangsdaten anhand von Schlüsseln abgerufen werden sollen.
VaultPrefix	Benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage.
VaultTLSCertFile	Der Name der SSL-Zertifikatdatei, die für die Client-Authentifizierung verwendet wird.
VaultTLSKeyFile	Der Name der SSL-Datei mit privatem Schlüssel, die für die Client-Authentifizierung verwendet wird.
VaultToken	Das Authentifizierungstoken für HashiCorp Vault.
VaultURL	Die HTTP[S]-URL des Vault-Servers.
VMwareCacheSize	Die Shared-Memory-Größe zum Speichern von VMware-Daten.
VMwareFrequency	Die Verzögerung in Sekunden zwischen Datenerfassungen von einem einzelnen VMware-Dienst.
VMwarePerfFrequency	Die Verzögerung in Sekunden zwischen dem Abruf von Leistungszählerstatistiken von einem einzelnen VMware-Dienst.
VMwareTimeout	Die maximale Anzahl von Sekunden, die ein VMware-Collector auf eine Antwort vom VMware-Dienst wartet.
WebDriverURL	Die HTTP[S]-URL der WebDriver-Schnittstelle.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass ein Parameter obligatorisch ist.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standardwerten, 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 deaktiviert und der Proxy von „root“ gestartet wird, versucht der Proxy stattdessen, zum Benutzer „zabbix“ zu wechseln. Hat keine Auswirkung, 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

CacheSize

Die Größe des Konfigurations-Cache in Byte. Die Größe des Shared Memory zum Speichern von Host- und Datenpunkt-Daten.

Standard: 32M
 Bereich: 128K-64G

ConfigFrequency

Dieser Parameter ist **veraltet** (verwenden Sie stattdessen ProxyConfigFrequency).
Gibt an, wie oft der Proxy Konfigurationsdaten vom Zabbix Server in Sekunden abruft.
Parameter für aktive Proxys. Wird für passive Proxys ignoriert (siehe Parameter ProxyMode).

Standard: 3600
 Bereich: 1-604800

DataSenderFrequency

Der Proxy sendet gesammelte Daten alle N Sekunden an den Server. Beachten Sie, dass ein aktiver Proxy den Zabbix Server weiterhin jede Sekunde auf Aufgaben für Remote-Befehle abfragt.
Parameter für aktive Proxys. Wird für passive Proxys ignoriert (siehe Parameter ProxyMode).

Standard: 1
 Bereich: 1-3600

DBHost

Der Datenbank-Hostname.
Bei MySQL führen `localhost` oder eine leere Zeichenfolge zur Verwendung eines Sockets. Bei PostgreSQL führt nur eine leere Zeichenfolge zu einem Versuch, einen Socket zu verwenden.

Standard: `localhost`

DBName

Der Datenbankname oder der Pfad zur Datenbankdatei für SQLite3 (die Mehrprozessarchitektur von Zabbix erlaubt nicht die Verwendung einer In-Memory-Datenbank, z. B. `:memory:`, `file::memory:?cache=shared` oder `file:memdb1?mode=memory&cache=shar`
Versuchen Sie nicht, dieselbe Datenbank zu verwenden, die der Zabbix Server verwendet.

Verbindlich: Ja

DBPassword

Das Datenbankpasswort. Kommentieren Sie diese Zeile aus, wenn kein Passwort verwendet wird. Wird bei SQLite ignoriert.

DBPort

Der Datenbank-Port, wenn kein lokaler Socket verwendet wird.[1] Wird für SQLite ignoriert.

Standard für MySQL: 3306

Standard für PostgreSQL: 5432

Bereich: 1024-65535

DBSchema

Der Name des Datenbankschemas. Wird für PostgreSQL verwendet.

DBSocket

Der Pfad zur MySQL-Socket-Datei.[1]
Der Datenbank-Port, wenn kein lokaler Socket verwendet wird. Wird für SQLite ignoriert.

Standard: 3306

DBUser

Der Datenbankbenutzer. Wird für SQLite ignoriert.

DBTLSConnect

Das Setzen dieser Option erzwingt die Verwendung einer TLS-Verbindung zur Datenbank:
required - Verbindung über TLS herstellen
verify_ca - Verbindung über TLS herstellen und Zertifikat verifizieren
verify_full - Verbindung über TLS herstellen, Zertifikat verifizieren und prüfen, dass die durch DBHost angegebene Datenbankidentität mit ihrem Zertifikat übereinstimmt
Unter MySQL ab 5.7.11 und PostgreSQL werden die folgenden Werte unterstützt: ”required”, ”verify”, ”verify_full”.
Unter 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

Der vollständige Pfadname einer Datei, die die Zertifikate der übergeordneten CA(s) für die Zertifikatsprüfung der Datenbank enthält.

Verbindlich: nein (ja, wenn DBTLSConnect auf *verify_ca* oder *verify_full* gesetzt ist)

DBTLSCertFile

Der vollständige Pfadname einer Datei, die das Zabbix-Proxy-Zertifikat zur Authentifizierung bei der Datenbank enthält.

DBTLSKeyFile

Der vollständige Pfadname einer Datei, die den privaten Schlüssel für die Authentifizierung bei der Datenbank enthält.

DBTLSCipher

Die Liste der Verschlüsselungs-Chiffren, die der Zabbix Proxy für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Wird nur für MySQL unterstützt.

DBTLSCipher13

Die Liste der Verschlüsselungs-Ciphersuites, die der Zabbix Proxy für das TLS-v1.3-Protokoll zulässt. Wird nur für MySQL unterstützt, ab Version 8.0.16.

DebugLevel

Geben Sie den Debug-Level an:
0 - grundlegende Informationen über das Starten und Stoppen 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

EnableRemoteCommands

Ob Remote-Befehle vom Zabbix Server erlaubt sind.

Standard: 0
 Werte: 0 - nicht erlaubt; 1 - erlaubt

ExternalScripts

Der Speicherort externer Skripte (abhängig von der Installationsvariablen `datadir` zur Kompilierzeit).

Standard: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Der Speicherort von fping6. Stellen Sie sicher, dass die fping6-Binärdatei Root als Eigentümer hat 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

Der Speicherort von fping. Stellen Sie sicher, dass die fping-Binärdatei Root gehört und das SUID-Flag gesetzt ist.

Standard: `/usr/sbin/fping`

HistoryCacheSize

Die Größe des Verlaufs-Caches in Byte. Die Größe des Shared Memory zum Speichern von Verlaufsdaten.

Standard: `16M`
 Bereich: 128K-16G

HistoryIndexCacheSize

Die Größe des Verlaufsindex-Caches in Byte. Die Größe des Shared Memory zum Indizieren der im Verlaufs-Cache gespeicherten Verlaufsdaten. Für das Zwischenspeichern eines Datenpunkts benötigt der Index-Cache ungefähr 100 Byte.

Standard: `4M`
 Bereich: 128K-16G

Hostname

Ein eindeutiger, zwischen Groß- und Kleinschreibung unterscheidbarer Proxy-Name. Stellen Sie sicher, dass der Proxy-Name dem Server bekannt ist.
Zulässige Zeichen: alphanumerische Zeichen, ’.’, ’ ’, ’_’ und ’-’. Maximale Länge: 128

Standard: Durch HostnameItem festgelegt

HostnameItem

Der Datenpunkt, der zum Setzen von Hostname verwendet wird, wenn dieser nicht definiert ist (dies wird auf dem Proxy ähnlich wie auf einem Agent ausgeführt). Wird ignoriert, wenn Hostname gesetzt ist.
Unterstützt keine UserParameters, Leistungsindikatoren oder Aliase, unterstützt jedoch system.run[].

Standard: system.hostname

HousekeepingFrequency

Wie oft Zabbix die Housekeeping-Prozedur ausführt (in Stunden). Housekeeping entfernt veraltete Informationen aus der Datenbank.
Hinweis: Um die Last beim Start des Proxy zu verringern, wird Housekeeping nach dem Start des Proxy um 30 Minuten verschoben. Wenn HousekeepingFrequency also 1 ist, wird die allererste Housekeeping-Prozedur 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 die Housekeeping-Prozedur nur über die Runtime-Control-Option *housekeeper_execute* gestartet werden.
Siehe auch die Optionen zur Laufzeitsteuerung sowie Details zur Housekeeping-Prozedur.

Standard: 1
 Bereich: 0-24

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Platzhalterzeichen Asterisk für den Musterabgleich unterstützt.
Siehe besondere Hinweise zu Einschränkungen.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Die IP-Adresse (oder der Hostname) des Zabbix Java gateway. Nur erforderlich, wenn Java-Poller gestartet werden.

JavaGatewayPort

Der Port, auf dem das Zabbix Java gateway lauscht.

Standard: 10052
 Bereich: 1024-32767

ListenBacklog

Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
Der Standardwert ist eine fest kodierte Konstante, die vom System abhängt.
Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementierungsspezifische Maximum“ gekürzt werden.

Standard: SOMAXCONN
 Bereich: 0 - INT_MAX

ListenIP

Eine Liste von durch Kommas getrennten IP-Adressen, auf denen der Trapper lauschen soll.
Der Trapper lauscht auf allen Netzwerkschnittstellen, wenn dieser Parameter fehlt.

Standard: 0.0.0.0

ListenPort

Der Port, auf dem der Trapper lauscht.

Standard: 10051
 Bereich: 1024-32767

LoadModule

Das Modul, das beim Start des Proxy geladen werden soll. Module werden verwendet, um die Funktionalität des Proxy zu erweitern. Das Modul muss sich in dem durch LoadModulePath angegebenen Verzeichnis befinden, oder dem Modulnamen muss ein Pfad vorangestellt werden. 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

Der vollständige Pfad zum Speicherort der Proxy-Module. Der Standardwert hängt von den Kompilierungsoptionen ab.

LogFile

Der Name der Protokolldatei.

Verbindlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe einer Logdatei in MB.
0 - automatische Logrotation deaktivieren.
Hinweis: Wenn das Größenlimit der Logdatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Logdatei gekürzt und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogRemoteCommands

Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

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

LogSlowQueries

Wie lange eine Datenbankabfrage dauern darf, bevor sie protokolliert wird (in Millisekunden).
0 - langsame Abfragen nicht protokollieren.
Diese Option wird ab DebugLevel=3 aktiviert.

Standard: 0
 Bereich: 0-3600000

MaxConcurrentChecksPerPoller

Die maximale Anzahl asynchroner Prüfungen, die gleichzeitig von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller ausgeführt werden können. Siehe StartHTTPAgentPollers, StartAgentPollers und StartSNMPPollers.

Standard: 1000
 Bereich: 1-1000

PidFile

Der Name der PID-Datei.

Standard: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Gibt den Speichermechanismus für Verlaufs-, Netzwerkdiscovery- und Autoregistrierungsdaten an: *disk* - Daten werden in der Datenbank gespeichert und aus der Datenbank hochgeladen; *memory* - Daten werden im Speicher abgelegt und aus dem Speicher hochgeladen. Wenn dem Puffer der Speicher ausgeht, werden die alten Daten verworfen. Beim Herunterfahren wird der Puffer verworfen. *hybrid* - der Proxy-Puffer arbeitet normalerweise wie im Modus *memory*, bis ihm der Speicher ausgeht oder der älteste Datensatz das konfigurierte Alter überschreitet. In diesem Fall wird der Puffer in die Datenbank geschrieben und arbeitet wie im Modus *disk*, bis alle Daten hochgeladen wurden; danach arbeitet er wieder mit dem Speicher. Beim Herunterfahren wird der Speicherpuffer in die Datenbank geschrieben.

Siehe auch: Proxy memory buffer.

Standard: `disk`
 Werte: disk; memory; hybrid

ProxyConfigFrequency

Wie oft der Proxy Konfigurationsdaten vom Zabbix Server in Sekunden abruft.
Parameter für aktive Proxys. Wird für passive Proxys ignoriert (siehe Parameter ProxyMode).

Standard: 10
 Bereich: 1-604800

ProxyLocalBuffer

Der Proxy speichert Daten für N Stunden lokal, auch wenn die Daten bereits mit dem Server synchronisiert wurden.
Dieser Parameter kann verwendet werden, wenn lokale Daten von Drittanbieteranwendungen genutzt werden.

Standard: 0
 Bereich: 0-720

ProxyMemoryBufferAge

Das maximale Alter von Daten im Proxy-Speicherpuffer in Sekunden. Wenn aktiviert (nicht null) und Datensätze im Proxy-Speicherpuffer älter sind, erzwingt dies, 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

Die Größe des Shared-Memory-Caches für gesammelte Verlaufs-, Discovery- und Autoregistrierungsdaten in Byte. Wenn aktiviert (nicht null), hält der Proxy Verlaufs-, Discovery- und Autoregistrierungsdaten im Speicher, sofern der Cache nicht voll ist oder gespeicherte Datensätze nicht älter sind als das definierte ProxyMemoryBufferAge. Dieser Parameter kann nicht zusammen mit dem Parameter ProxyLocalBuffer verwendet werden.

Standard: 0
 Bereich: 0;128K-2G

ProxyMode

Der Betriebsmodus des Proxy.
0 - Proxy im aktiven Modus
1 - Proxy im passiven Modus
Beachten Sie, dass (sensible) Proxy-Konfigurationsdaten für Parteien verfügbar werden können, die Zugriff auf den Trapper-Port des Zabbix-Server 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

Der Proxy speichert Daten für N Stunden, falls keine Verbindung zum Zabbix Server besteht.
Ältere Daten gehen verloren.

Standard: 1
 Bereich: 1-720

Server

Wenn ProxyMode auf den *aktiven Modus* gesetzt ist:
IP-Adresse oder DNS-Name des Zabbix-Servers (Adresse:Port) oder Cluster (Adresse:Port;Adresse2:Port), von dem Konfigurationsdaten abgerufen und an den Daten gesendet werden.
Wenn

kein Port angegeben ist, wird der Standardport verwendet.
Cluster-Knoten müssen durch ein Semikolon getrennt werden.

Wenn ProxyMode auf den *passiven Modus* gesetzt ist:
Liste von durch Kommas getrennten IP-Adressen, optional in CIDR-Notation, oder DNS-Namen des Zabbix-Servers. 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' 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
```

Verbindlich: ja

SNMPTrapperFile

Eine temporäre Datei, die verwendet wird, um Daten vom SNMP-Trap-Daemon an den Proxy zu übergeben.
Muss mit der Datei in zabbix_trap_receiver.pl oder der SNMPTT-Konfigurationsdatei übereinstimmen.

Standard: `/tmp/zabbix_traps.tmp`

SocketDir

Das Verzeichnis zum Speichern von IPC-Sockets, die von internen Zabbix-Diensten verwendet werden.

Standard: `/tmp`

SourceIP

Die Quell-IP-Adresse für:

- ausgehende Verbindungen zum Zabbix-Server
- agentenlose Verbindungen (VMware, SSH, JMX, SNMP, Telnet und einfache Prüfungen)
- HTTP-Agent-Verbindungen
- JavaScript-HTTP-Anfragen von Skript-Datenpunkten
- JavaScript-HTTP-Anfragen der Vorverarbeitung
- Verbindungen zum Vault

SSHKeyLocation

Der Speicherort der öffentlichen und privaten Schlüssel für SSH-Prüfungen und Aktionen.

SSLCertLocation

Der Speicherort der SSL-Client-Zertifikatsdateien für die Client-Authentifizierung.
Dieser Parameter wird nur bei der Web-Überwachung verwendet.

SSLKeyLocation

Der Speicherort der privaten SSL-Schlüsseldateien für die Client-Authentifizierung.
Dieser Parameter wird nur in der Web-Überwachung verwendet.

SSLCALocation

Der Speicherort der Dateien der Zertifizierungsstelle (CA) für die Verifizierung von SSL-Serverzertifikaten.
Beachten Sie, dass der Wert dieses Parameters als libcurl-Option CURLOPT_CAPATH gesetzt wird. Bei libcurl-Versionen vor 7.42.0 hat dies nur dann eine Wirkung, wenn libcurl für die Verwendung von OpenSSL kompiliert wurde. Weitere Informationen finden Sie auf der cURL-Webseite.
Dieser Parameter wird in der Web-Überwachung und bei der SMTP-Authentifizierung verwendet.

StartAgentPollers

Die Anzahl der vorab geforkten Instanzen von Zabbix-Agent-Pollern. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartBrowserPollers

Die Anzahl der vorab geforkten Instanzen von Browser-Datenpunkt-Pollern.

Standard: 1
 Bereich: 0-1000

StartDBSyncers

Die Anzahl der vorab geforkten Instanzen von Verlaufs-Synchronisierern.
Hinweis: Seien Sie vorsichtig, wenn Sie diesen Wert ändern; eine Erhöhung kann mehr schaden als nützen.

Standard: 4
 Bereich: 1-100

StartDiscoverers

Die Anzahl der vorab geforkten Instanzen von Discovery-Workern.

Standard: 5
 Bereich: 0-1000

StartHTTPAgentPollers

Die Anzahl der vorab geforkten Instanzen von HTTP-Agent-Pollern. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartHTTPPollers

Die Anzahl der vorab geforkten Instanzen von HTTP-Pollern.

Standard: 1
 Bereich: 0-1000

StartIPMIPollers

Die Anzahl der vorab geforkten Instanzen von IPMI-Pollern.

Standard: 0
 Bereich: 0-1000

StartJavaPollers

Die Anzahl der vorab geforkten Instanzen von Java-Pollern.

Standard: 0
 Bereich: 0-1000

StartODBCPollers

Die Anzahl der vorab geforkten Instanzen von ODBC-Pollern.

Standard: 1
 Bereich: 0-1000

StartPingers

Die Anzahl der vorab geforkten Instanzen von ICMP-Pingern.

Standard: 1
 Bereich: 0-1000

StartPollersUnreachable

Die Anzahl der vorab geforkten Instanzen von Pollern für nicht erreichbare Hosts (einschließlich IPMI und Java). Mindestens ein Poller für nicht erreichbare Hosts muss laufen, wenn reguläre, IPMI- oder Java-Poller gestartet werden.

Standard: 1
 Bereich: 0-1000

StartPollers

Die Anzahl der vorab geforkten Instanzen von Pollern.

Standard: 5
 Bereich: 0-1000

StartPreprocessors

Die Anzahl der vorab gestarteten Instanzen von Vorverarbeitungs-worker-Threads sollte nicht kleiner als die Anzahl der verfügbaren CPU-Kerne eingestellt werden. Es sollten mehr worker eingestellt werden, wenn die Vorverarbeitung nicht CPU-gebunden ist und viele Netzwerkanfragen enthält.

Standard: 16
 Bereich: 1-1000

StartSNMPPollers

Die Anzahl der vorab geforkten Instanzen von SNMP-Pollern. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartSNMPTrapper

Wenn auf 1 gesetzt, wird ein Prozess für den SNMP trapper gestartet.

Standard: 0
 Bereich: 0-1

StartTrappers

Die Anzahl der vorab geforkten Instanzen von Trappern.
Trapper akzeptieren eingehende Verbindungen von Zabbix sender und aktiven Agenten.

Standard: 5
 Bereich: 0-1000

StartVMwareCollectors

Die Anzahl der vorab geforkten Instanzen des VMware collector.

Standard: 0
 Bereich: 0-250

StatsAllowedIP

Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen externer Zabbix-Instanzen. Die Statistikabfrage wird nur von den hier aufgeführten Adressen akzeptiert. Wenn dieser Parameter nicht gesetzt ist, werden keine Statistikabfragen 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
```

Timeout

Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Server, Agent, Web-Service und Legacy-SNMP-Prüfungen (einzelne OID-Nummer oder Zeichenfolge) gewartet wird.

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
- Zeitlimit für asynchrone IPC-Sockets und Runtime-Control-Optionen
- DNS-Anfragen des asynchronen Pollers
- Antwort für den Heartbeat aktiver Prüfungen
- Abruf von Zabbix-Agent-Daten (Werten) von aktiven Agents
- Abruf von Daten von Zabbix sender
- Senden der Liste aktiver Prüfungen an den Zabbix Agent

Dieses Timeout wird **nicht** für Prüfungen verwendet, für die im Frontend Einstellungen für flexible timeout konfiguriert sind (global, auf 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

Welche eingehenden Verbindungen vom Zabbix Server akzeptiert werden sollen. Wird für einen passiven Proxy verwendet, bei einem aktiven Proxy ignoriert. Mehrere Werte können angegeben werden, durch Komma getrennt:
unencrypted - Verbindungen ohne Verschlüsselung akzeptieren (Standard)
psk - Verbindungen mit TLS und einem Pre-shared Key (PSK) akzeptieren
cert - Verbindungen mit TLS und einem Zertifikat akzeptieren

Verbindlich: ja für passiven Proxy, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch bei *unencrypted*-Verbindung); andernfalls nein

TLSCAFile

Der vollständige Pfadname einer Datei, die die Zertifikate der CA(s) der obersten Ebene für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname einer Datei, die das Proxy-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCipherAll

Der GnuTLS-Prioritätsstring oder der OpenSSL-(TLS 1.2)-Cipher-String. Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Die Chiffre-Zeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien der Chiffre-Suite für zertifikat- 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

Der GnuTLS-Prioritätsstring oder der OpenSSL-(TLS 1.2)-Cipher-String. Überschreibt die standardmäßigen Auswahlkriterien der Cipher-Suite für zertifikatbasierte 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

Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien für Chiffriersuiten bei zertifikatbasierter Verschlüsselung.

TLSCipherPSK

Der GnuTLS-Prioritätsstring oder der OpenSSL-Cipher-String (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien der Cipher-Suite für PSK-basierte 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

Die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien für Chiffren-Suites für PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Wie sich der Proxy mit dem Zabbix Server verbinden soll. Wird für einen aktiven Proxy verwendet, bei einem passiven Proxy ignoriert. Es kann nur ein Wert angegeben werden:
unencrypted - Verbindung ohne Verschlüsselung herstellen (Standard)
psk - Verbindung über TLS und einen Pre-Shared Key (PSK) herstellen
cert - Verbindung über TLS und ein Zertifikat herstellen

Verbindlich: ja für aktiven Proxy, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch bei einer *unencrypted*-Verbindung); andernfalls nein

TLSCRLFile

Der vollständige Pfadname einer Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Proxy enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSListen

Steuert TLS auf dem Trapper-Socket.

Unterstützte Werte:

- required - nur TLS-Verbindungen akzeptieren

TLSPSKFile

Der vollständige Pfadname einer Datei, die den vorab geteilten Schlüssel des Proxy enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSPSKIdentity

Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSServerCertIssuer

Der zulässige Aussteller des Server-Zertifikats.

TLSServerCertSubject

Das zulässige Betreff-Feld des Server-Zertifikats.

TmpDir

Das temporäre Verzeichnis.

Standard: `/tmp`

TrapperTimeout

Gibt das Timeout in Sekunden an für:
 - den Abruf von Konfigurationsdaten vom Zabbix Server;
 - die Ausführung globaler Skripte oder die Ausführung entfernter Befehle.

Standard: 300
 Bereich: 1-300

UnavailableDelay

Wie oft ein Host während des Zeitraums der Nichtverfügbarkeit in Sekunden auf Verfügbarkeit geprüft wird.

Standard: 60
 Bereich: 1-3600

UnreachableDelay

Wie oft ein Host während des Zeitraums der Unerreichbarkeit in Sekunden auf Verfügbarkeit geprüft wird.

Standard: 15
 Bereich: 1-3600

UnreachablePeriod

Nach wie vielen Sekunden der Unerreichbarkeit ein Host als nicht verfügbar behandelt wird.

Standard: 45
 Bereich: 1-3600

Benutzer

Privilegien auf einen bestimmten, auf dem System vorhandenen Benutzer reduzieren.
Hat nur dann eine Wirkung, wenn der Prozess als „root“ ausgeführt wird und AllowRoot deaktiviert ist.

Standard: `zabbix`

Vault

Der Vault-Anbieter:
HashiCorp - HashiCorp KV Secrets Engine Version 2
CyberArk - CyberArk Central Credential Provider
Muss mit dem im Frontend festgelegten Vault-Anbieter übereinstimmen.

Standard: `HashiCorp`

VaultDBPath

Vault-Pfad oder Abfrage, je nach Vault, aus dem die Zugangsdaten für die Datenbank anhand von Schlüsseln abgerufen werden.

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

Ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage, abhängig vom Vault. Wenn nicht angegeben, werden die am besten geeigneten Standardwerte verwendet.
Beachten Sie, dass bei HashiCorp nach dem Mountpoint automatisch `data` angehängt wird, wenn VaultPrefix nicht angegeben ist.
Beachten Sie, dass bei HashiCorp nach dem Mountpoint automatisch `data` angehängt wird, wenn VaultPrefix nicht angegeben ist.

Beispielpräfix für Hashicorp:

```
v1/secret/data/zabbix/
```

Beispielpräfix für Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Der Name der SSL-Zertifikatsdatei, 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

Der Name der SSL-Private-Key-Datei, die für die Client-Authentifizierung verwendet wird.
Die Private-Key-Datei 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

Das HashiCorp-Vault-Authentifizierungstoken, das ausschließlich für den Zabbix Proxy mit schreibgeschützter Berechtigung für den im optionalen Konfigurationsparameter VaultDBPath angegebenen Pfad generiert worden sein sollte.
Es ist ein Fehler, wenn VaultToken und die Umgebungsvariable VAULT_TOKEN gleichzeitig definiert sind.

Verbindlich: Ja, wenn Vault auf *HashiCorp* gesetzt ist; andernfalls nein

VaultURL

Die HTTP[S]-URL des Vault-Servers. Das systemweite Verzeichnis für CA-Zertifikate wird verwendet, wenn SSLCALocation nicht angegeben ist.

Standard: `https://127.0.0.1:8200`

VMwareCacheSize

Die Größe des Shared Memory zum Speichern von VMware-Daten.
Eine interne VMware-Prüfung zabbix[vmware,buffer,...] kann verwendet werden, um die Nutzung des VMware-Cache zu überwachen (siehe Interne Prüfungen).
Beachten Sie, dass Shared Memory nicht zugewiesen wird, wenn keine vmware collector-Instanzen für den Start konfiguriert sind.

Standard: `8M`
 Bereich: 256K-2G

VMwareFrequency

Die Verzögerung in Sekunden zwischen den Datenerfassungen von einem einzelnen VMware-Service.
Diese Verzögerung sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Überwachungs-Datenpunkts gesetzt werden.

Standard: 60
 Bereich: 10-86400

VMwarePerfFrequency

Die Verzögerung in Sekunden zwischen dem Abruf von Leistungszählerstatistiken von einem einzelnen VMware-Dienst.
Diese Verzögerung sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Monitoring-Datenpunkts gesetzt werden, der VMware-Leistungszähler verwendet.

Standard: 60
 Bereich: 10-86400

VMwareTimeout

Die maximale Anzahl von Sekunden, die ein VMware-Collector auf eine Antwort vom VMware-Service (vCenter oder ESX-Hypervisor) wartet.

Standard: 10
 Bereich: 1-300

WebDriverURL

HTTP[S]-URL der WebDriver-Schnittstelle.

Beispiel (verwendet mit dem eigenständigen Selenium WebDriver-Server):

```
WebDriverURL=http://localhost:4444
```

Fußnoten

[1] DBSocket und DBPort schließen sich in der Proxy-Konfiguration gegenseitig aus. Geben Sie nur einen von beiden an oder lassen Sie beide undefiniert.

3 Zabbix Agent (UNIX)

Übersicht

Die von der Zabbix-Agent-Konfigurationsdatei (zabbix_agentd.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
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
AllowRoot	Erlaubt, dass der Agent als 'root' ausgeführt wird.
BufferSend	Daten nicht länger als N Sekunden im Puffer behalten.
BufferSize	Die maximale Anzahl von Werten im Speicherpuffer.
DebugLevel	Die Debug-Stufe.
DenyKey	Verweigert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
EnableRemoteCommands	Ob Remote-Befehle vom Zabbix Server erlaubt sind.
HeartbeatFrequency	Die Häufigkeit von Heartbeat-Nachrichten in Sekunden.
HostInterface	Ein optionaler Parameter, der die Host-Schnittstelle definiert.
HostInterfaceItem	Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Ermitteln der Host-Schnittstelle verwendet wird.
HostMetadata	Ein optionaler Parameter, der die Host-Metadaten definiert.
HostMetadataItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Ermitteln der Host-Metadaten verwendet wird.
Hostname	Ein optionaler Parameter, der den Hostnamen definiert.
HostnameItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Ermitteln des Hostnamens verwendet wird.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
ListenBacklog	Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
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.
LoadModule	Das Modul, das beim Start des Agent geladen wird.
LoadModulePath	Der vollständige Pfad zum Speicherort der Agent-Module.
LogFile	Der Name der Protokolldatei.
LogFileSize	Die maximale Größe der Protokolldatei.
LogRemoteCommands	Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.
LogType	Der Typ der Protokollausgabe.
MaxLinesPerSecond	Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an den Zabbix Server oder Proxy sendet, wenn aktive Prüfungen vom Typ 'log' und 'logrt' verarbeitet werden.
PidFile	Der Name der PID-Datei.
RefreshActiveChecks	Wie oft die Liste der aktiven Prüfungen aktualisiert wird.
Server	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix Servern und Zabbix Proxys.
ServerActive	Die Adresse des Zabbix Servers/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden.
SourceIP	Die Quell-IP-Adresse.
StartAgents	Die Anzahl der vorab geforkten Instanzen von zabbix_agentd, die passive Prüfungen verarbeiten.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Proxy oder Server gewartet wird.
TLSAccept	Welche eingehenden Verbindungen akzeptiert werden.
TLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

Parameter	Beschreibung
TLSCertFile	Der vollständige Pfadname einer Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Cipher-Zeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl der Cipher-Suite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherAll13	Die Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl der Cipher-Suite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Cipher-Zeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl der Cipher-Suite für zertifikatsbasierte Verschlüsselung.
TLSCipherCert13	Die Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl der Cipher-Suite für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK	Die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Cipher-Zeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl der Cipher-Suite für PSK-basierte Verschlüsselung.
TLSCipherPSK13	Die Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die Standardkriterien zur Auswahl der Cipher-Suite für PSK-basierte Verschlüsselung.
TLSConnect	Wie der Agent eine Verbindung zum Zabbix Server oder Proxy herstellen soll.
TLSCRLFile	Der vollständige Pfadname einer Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Der vollständige Pfadname einer Datei, die den vorinstallierten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.
TLSPSKIdentity	Die Identitätszeichenfolge des vorinstallierten Schlüssels, verwendet für verschlüsselte Kommunikation mit dem Zabbix Server.
TLSServerCertIssuer	Der zulässige Aussteller des Server-(Proxy-)Zertifikats.
TLSServerCertSubject	Der zulässige Betreff des Server-(Proxy-)Zertifikats.
UnsafeUserParameters	Erlaubt, dass alle Zeichen in Argumenten für benutzerdefinierte Parameter übergeben werden.
User	Legt die Berechtigungen auf einen bestimmten, vorhandenen Benutzer des Systems ab.
UserParameter	Ein benutzerdefinierter Parameter zur Überwachung.
UserParameterDir	Der Standardsuchpfad für UserParameter-Befehle.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass ein Parameter verpflichtend ist.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standardwerten, 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.

Details zu den Parametern

Alias

Legt einen Alias für einen Datenpunktschlüssel fest. Er kann verwendet werden, um einen langen und komplexen Datenpunktschlüssel durch einen kürzeren und einfacheren zu ersetzen.
 Es können mehrere *Alias*-Parameter vorhanden sein. Mehrere Parameter mit demselben *Alias*-Schlüssel sind nicht zulässig.
 Verschiedene *Alias*-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliasse 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: Abrufen der CPU-Auslastung mit Standard- und benutzerdefinierten Parametern.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Dies ermöglicht die Verwendung des Schlüssels **cpu.util**, um den prozentualen Wert der CPU-Auslastung mit Standardparametern abzurufen, sowie die Verwendung von **cpu.util[all, idle, avg15]**, um spezifische Daten zur CPU-Auslastung zu erhalten.

Beispiel 3: Ausführen mehrerer Regeln zur Low-Level-Discovery, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun ist es möglich, mehrere Discovery-Regeln mit **vfs.fs.discovery** und unterschiedlichen Parametern für jede Regel einzurichten, 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 Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schlüsselabgleich können in Kombination mit DenyKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge des Auftretens verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

AllowRoot

Erlaubt dem Agent, als „root" zu laufen. Wenn deaktiviert und der Agent von „root" gestartet wird, versucht der Agent stattdessen, zum Benutzer „zabbix" zu wechseln. Hat keine Auswirkung, wenn er unter einem normalen Benutzer gestartet wird.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

BufferSend

Daten nicht länger als N Sekunden im Puffer behalten.

Standard: 5
 Bereich: 1-3600

BufferSize

Die maximale Anzahl von Werten im Speicherpuffer. Der Agent sendet alle gesammelten Daten an den Zabbix Server oder Proxy, wenn der Puffer voll ist.

Standard: 100
 Bereich: 2-65535

DebugLevel

Geben Sie den Debug-Level an:
0 - grundlegende Informationen über das Starten und Stoppen 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

DenyKey

Verweigert die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schlüsselabgleich können in Kombination mit AllowKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge ihres Auftretens verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

EnableRemoteCommands

Ob Remote-Befehle vom Zabbix Server erlaubt sind. Dieser Parameter ist **veraltet**; verwenden Sie stattdessen AllowKey=system.run[*] oder DenyKey=system.run[*].
Er ist ein interner Alias für die Parameter AllowKey/DenyKey, abhängig vom Wert:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

HeartbeatFrequency

Die Häufigkeit von Heartbeat-Nachrichten in Sekunden. Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Ein optionaler Parameter, der die Host-Schnittstelle (IP-Adresse oder DNS-Name) definiert, die während des Prozesses 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 einer DNS-Adresse. Weitere Details finden Sie unter DNS als Standardschnittstelle verwenden.

Falls nicht definiert, wird der Wert von HostInterfaceItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert das Limit von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Ein optionaler Parameter, der einen Datenpunkt definiert, der verwendet wird, um die Host-Schnittstelle (IP-Adresse oder DNS-Name) während des Prozesses der Host-Autoregistrierung zu bestimmen. Dieser Wert wird nur verwendet, wenn HostInterface nicht definiert ist. Weitere Details 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 für AllowKey/DenyKey unterstützt.

HostMetadata

Ein optionaler Parameter, der die Metadaten definiert, die verwendet werden, um den Host während des Prozesses der Autoregistrierung des Hosts zu identifizieren oder zu unterscheiden (aktiver Agent). HostMetadata ermöglicht es, zwischen Hosts über den Hostnamen hinaus zu unterscheiden.

Falls nicht definiert, wird der Wert von HostMetadataItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der angegebene Wert das Limit von 2034 Byte 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 sind, aber keine gültigen IP-Adressen oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Bereich: 0-2034 Byte

HostMetadataItem

Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen von Host-Metadaten verwendet wird. Diese Option wird nur verwendet, wenn HostMetadata nicht definiert ist.

Der Wert von HostMetadataItem wird bei jedem Versuch der Autoregistrierung abgerufen und nur im Prozess der Host-Autoregistrierung verwendet (aktiver Agent). HostMetadataItem ermöglicht es, zwischen Hosts über den Hostnamen hinaus zu unterscheiden.

Benutzerparameter und Aliasse werden unterstützt. Der Datenpunkt system.run[] wird unabhängig von den Einstellungen für AllowKey/DenyKey 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 sind, aber keine gültigen IP-Adressen oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Hostname

Eine Liste von durch Kommas getrennten, eindeutigen, groß-/kleinschreibungssensitiven Hostnamen. Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen. Der Wert wird von 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

Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen des Hostnamens verwendet wird.
Diese Option wird nur verwendet, wenn Hostname nicht definiert ist.
Benutzerparameter oder Aliasse werden nicht unterstützt, aber der Datenpunkt system.run[] wird unabhängig von den Werten von AllowKey/DenyKey unterstützt.

Standard: `system.hostname`

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden. Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Asterisk-Platzhalterzeichen für den Musterabgleich unterstützt.
Siehe besondere Hinweise zu Einschränkungen.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
Der Standardwert ist eine fest kodierte Konstante, die vom System abhängt.
Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementierungsspezifische Maximum“ gekürzt werden.

Standard: SOMAXCONN
 Bereich: 0 - INT_MAX

ListenIP

Eine Liste von durch Kommas getrennten IP-Adressen, auf denen der Agent lauschen soll.

Standard: 0.0.0.0

ListenPort

Der Agent lauscht auf diesem Port auf Verbindungen vom Server.

Standard: 10050
 Bereich: 1024-32767

LoadModule

Das Modul, 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 werden. 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

Der vollständige Pfad zum Speicherort der Agent-Module. Der Standardwert hängt von den Kompilierungsoptionen ab.

LogFile

Der Name der Protokolldatei.

Verbindlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe einer 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

LogRemoteCommands

Aktiviert 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

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

MaxLinesPerSecond

Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an den Zabbix Server oder Proxy sendet, wenn aktive Prüfungen vom Typ „log“ und „logrt“ verarbeitet werden. Der angegebene Wert wird durch den Parameter „maxlines“ überschrieben, der im Datenpunkt-Schlüssel „log“ oder „logrt“ 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

PidFile

Der Name der PID-Datei.

Standard: /tmp/zabbix_agentd.pid

RefreshActiveChecks

Wie oft die Liste der aktiven Prüfungen aktualisiert wird, in Sekunden. Beachten Sie, dass nach einem fehlgeschlagenen Aktualisieren der aktiven Prüfungen der nächste Aktualisierungsversuch in 60 Sekunden erfolgt.

Standard: 5
 Bereich: 1-86400

Server

Eine durch Kommas getrennte Liste von 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, dann 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. Beachten Sie, dass „IPv4-kompatible IPv6-Adressen" (Präfix 0000::/96) unterstützt werden, aber durch RFC4291 als veraltet eingestuft sind. Leerzeichen sind erlaubt.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Verpflichtend: ja, wenn StartAgents nicht explizit auf 0 gesetzt ist

ServerActive

Die Adresse des Zabbix Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden. Die Server-/Proxy-Adresse ist eine IP-Adresse oder ein DNS-Name mit einem optionalen, durch einen Doppelpunkt getrennten Port.
Die Cluster-Konfiguration besteht aus einer oder mehreren, durch Semikolon getrennten Adressen von Mitgliedern einer Server- oder Proxy-Gruppe. Es können mehrere Zabbix Server/Cluster und Zabbix Proxys angegeben werden, getrennt durch Komma. Sofern keine Proxy-Gruppen verwendet werden, sollte nicht mehr als ein Zabbix Proxy von jedem Zabbix Server/Cluster angegeben werden. Wenn ein Zabbix Proxy angegeben ist, sollte der Zabbix Server/Cluster für diesen Proxy nicht angegeben werden.
Es können mehrere durch Komma getrennte Adressen angegeben werden, um mehrere unabhängige Zabbix Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn kein Port 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 bei 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 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

Die Quell-IP-Adresse für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy;
- den Verbindungsaufbau bei der Ausführung einiger Datenpunkte (web.page.get, net.tcp.port usw.).

StartAgents

Die Anzahl der vorab geforkten Instanzen von zabbix_agentd, die passive Prüfungen verarbeiten. Wenn auf 0 gesetzt, sind passive Prüfungen deaktiviert und der Agent lauscht auf keinem TCP-Port.

Standard: 10
 Bereich: 0-100

Timeout

Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Proxy oder Server gewartet werden soll.

Dieser Parameter definiert die Dauer verschiedener Kommunikationsvorgänge, darunter:

- das Warten auf eine Antwort vom Zabbix Server;
- das Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunkt-Konfiguration und Datenpunkt-Daten bei 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.

Dieser Timeout wird **nicht** für jene Agent-Prüfungen verwendet, die im Frontend konfigurierbare Timeout-Einstellungen haben (auf globaler, Proxy- oder Datenpunkt-Ebene).

Standard: 3
 Bereich: 1-30

TLSAccept

Welche eingehenden Verbindungen akzeptiert werden sollen. Wird für passive Prüfungen verwendet. Mehrere Werte können angegeben werden, durch Komma getrennt:
unencrypted - Verbindungen ohne Verschlüsselung akzeptieren (Standard)
psk - Verbindungen mit TLS und einem vorinstallierten Schlüssel (PSK) akzeptieren
cert - Verbindungen mit TLS und einem Zertifikat akzeptieren

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch für eine *unencrypted*-Verbindung); andernfalls nein

TLSCAFile

Der vollständige Pfadname der Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname der Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSCipherAll

Der GnuTLS-Prioritätsstring oder der OpenSSL-(TLS 1.2)-Cipher-String.
Überschreibt die standardmäßigen Auswahlkriterien der Cipher-Suite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für zertifikat- 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

Der GnuTLS-Prioritäts-String oder der OpenSSL-(TLS 1.2)-Cipher-String. Überschreibt die standardmäßigen Auswahlkriterien der Cipher-Suite für zertifikatsbasierte 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

Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien für Chiffriersuiten bei zertifikatsbasierter Verschlüsselung.

TLSCipherPSK

Der GnuTLS-Prioritätsstring oder der OpenSSL-Cipher-String (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien der Cipher-Suite für PSK-basierte 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

Die Chiffrezeichenfolge für OpenSSL 1.1.1 oder neuer in TLS 1.3. Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Wie der Agent eine Verbindung zum Zabbix Server oder Proxy herstellen soll. Wird für aktive Prüfungen verwendet. Es kann nur ein Wert angegeben werden:
unencrypted - Verbindung ohne Verschlüsselung herstellen (Standard)
psk - Verbindung über TLS und mit einem vorab geteilten Schlüssel (PSK) herstellen
cert - Verbindung über TLS und mit einem Zertifikat herstellen

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch bei einer *unencrypted*-Verbindung); andernfalls nein

TLSCRLFile

Der vollständige Pfadname der Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Der vollständige Pfadname der Datei, die den vorab geteilten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSPSKIdentity

Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSServerCertIssuer

Der zulässige Aussteller des Server-(Proxy-)Zertifikats.

TLSServerCertSubject

Der zulässige Zertifikatbetreff des Servers (Proxy).

UnsafeUserParameters

Erlaubt, alle Zeichen in Argumenten an benutzerdefinierte Parameter zu übergeben. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbruchzeichen nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

Benutzer

Privilegien auf einen bestimmten, auf dem System vorhandenen Benutzer reduzieren.
Wirkt sich nur aus, wenn als 'root' ausgeführt wird und AllowRoot deaktiviert ist.

Standard: `zabbix`

UserParameter

Ein benutzerdefinierter Parameter zur Überwachung. Es kann mehrere benutzerdefinierte Parameter geben.
Format: UserParameter=<key>,<shell command>
Beachten Sie, dass der Shell-Befehl keine leere Zeichenfolge oder nur ein Zeilenende zurückgeben darf. Shell-Befehle können relative Pfade haben, wenn der Parameter UserParameterDir angegeben ist.

Beispiel:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Der standardmäßige Suchpfad für UserParameter-Befehle. Falls verwendet, 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
```

Siehe auch

1. Unterschiede in der Konfiguration des Zabbix Agent für aktive und passive Prüfungen ab Version 2.0.0

4 Zabbix Agent 2 (UNIX)

Übersicht

Zabbix Agent 2 ist eine neue Generation des Zabbix Agent und kann anstelle des Zabbix Agent verwendet werden.

Die von der Konfigurationsdatei des Zabbix Agent 2 (zabbix_agent2.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
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
BufferSend	Daten nicht länger als N Sekunden im Puffer behalten.
BufferSize	Die maximale Anzahl von Werten im Speicherpuffer.
ControlSocket	Der Control-Socket, der zum Senden von Laufzeitbefehlen mit der Option '-R' verwendet wird.
DebugLevel	Die Debug-Stufe.
DenyKey	Verweigert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
EnablePersistentBuffer	Aktiviert die Verwendung eines lokalen persistenten Speichers für aktive Datenpunkte.
ForceActiveChecksOnStart	Führt aktive Prüfungen unmittelbar nach dem Neustart für die erste empfangene Konfiguration aus.
HeartbeatFrequency	Die Häufigkeit von Heartbeat-Nachrichten in Sekunden.
HostInterface	Ein optionaler Parameter, der die Host-Schnittstelle definiert.
HostInterfaceItem	Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Ermitteln der Host-Schnittstelle verwendet wird.
HostMetadata	Ein optionaler Parameter, der die Host-Metadaten definiert.
HostMetadataItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Ermitteln der Host-Metadaten verwendet wird.
Hostname	Ein optionaler Parameter, der den Hostnamen definiert.
HostnameItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Ermitteln des Hostnamens verwendet wird.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
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.
LogFileSize	Die maximale Größe der Protokolldatei.
LogType	Der Typ der Protokollausgabe.
PersistentBufferFile	Die Datei, in der Zabbix Agent 2 die SQLite-Datenbank speichern soll.
PersistentBufferPeriod	Der Zeitraum, für den Daten gespeichert werden sollen, wenn keine Verbindung zum Server oder Proxy besteht.
PidFile	Der Name der PID-Datei.
Plugins.<PluginName>.System.Capacity	Die Begrenzung der Prüfungen pro Plugin, die gleichzeitig ausgeführt werden können.
Plugins.Log.MaxLinesPerSecond	Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an Zabbix Server oder Proxy sendet, wenn aktive Prüfungen vom Typ 'log' und 'logrt' verarbeitet werden.
Plugins.SystemRun.LogRemoteCommands	Aktiviert die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
PluginSocket	Der Pfad zum UNIX-Socket für die Kommunikation mit ladbaren Plugins.
PluginTimeout	Das Timeout für Verbindungen mit ladbaren Plugins in Sekunden.
RefreshActiveChecks	Wie oft die Liste der aktiven Prüfungen aktualisiert wird.
Server	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix Servern und Zabbix Proxys.
ServerActive	Die Adresse des Zabbix Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden.
SourceIP	Die Quell-IP-Adresse.
StatusPort	Falls gesetzt, lauscht der Agent auf diesem Port auf HTTP-Statusanfragen (http://localhost:<port>/status).
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Proxy oder Server gewartet werden soll.

Parameter	Beschreibung
TLSAccept	Welche eingehenden Verbindungen akzeptiert werden sollen.
TLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Der vollständige Pfadname einer Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Die OpenSSL-Chiffrenzeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl der Chiffresuite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherAll13	Die OpenSSL-Chiffrenzeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer. Überschreibt die Standardkriterien zur Auswahl der Chiffresuite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert	Die OpenSSL-Chiffrenzeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl der Chiffresuite für zertifikatsbasierte Verschlüsselung.
TLSCipherCert13	Die OpenSSL-Chiffrenzeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer. Überschreibt die Standardkriterien zur Auswahl der Chiffresuite für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK	Die OpenSSL-Chiffrenzeichenfolge (TLS 1.2). Überschreibt die Standardkriterien zur Auswahl der Chiffresuite für PSK-basierte Verschlüsselung.
TLSCipherPSK13	Die OpenSSL-Chiffrenzeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer. Überschreibt die Standardkriterien zur Auswahl der Chiffresuite für PSK-basierte Verschlüsselung.
TLSConnect	Wie sich der Agent mit Zabbix Server oder Proxy verbinden soll.
TLSCRLFile	Der vollständige Pfadname einer Datei, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Der vollständige Pfadname einer Datei, die den vorab geteilten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit Zabbix Server verwendet wird.
TLSPSKIdentity	Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für verschlüsselte Kommunikation mit Zabbix Server verwendet wird.
TLSServerCertIssuer	Der zulässige Aussteller des Server-(Proxy-)Zertifikats.
TLSServerCertSubject	Das zulässige Subjekt des Server-(Proxy-)Zertifikats.
UnsafeUserParameters	Erlaubt, alle Zeichen in Argumenten an benutzerdefinierte Parameter zu übergeben.
UserParameter	Ein benutzerdefinierter zu überwachender Parameter.
UserParameterDir	Der Standard-Suchpfad für UserParameter-Befehle.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass ein Parameter obligatorisch ist.

Beachten Sie:

- Die Standardwerte entsprechen den Prozessstandardwerten, 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

Legt einen Alias für einen Datenpunktschlüssel fest. Er kann verwendet werden, um einen langen und komplexen Datenpunktschlüssel durch einen kürzeren und einfacheren zu ersetzen.
 Es können mehrere *Alias*-Parameter vorhanden sein. Mehrere Parameter mit demselben *Alias*-Schlüssel sind nicht zulässig.
 Verschiedene *Alias*-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliasse können in *HostMetadataItem*, aber 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]
```

Jetzt kann der Kurzschlüssel **zabbix.userid** zum Abrufen von Daten verwendet werden.

Beispiel 2: Abrufen der CPU-Auslastung mit Standard- und benutzerdefinierten Parametern.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Dies ermöglicht die Verwendung des Schlüssels **cpu.util**, um den prozentualen Wert der CPU-Auslastung mit Standardparametern abzurufen, sowie die Verwendung von **cpu.util[all, idle, avg15]**, um spezifische Daten zur CPU-Auslastung zu erhalten.

Beispiel 3: Ausführen mehrerer Regeln zur Low-Level-Discovery, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Jetzt ist es möglich, mehrere Discovery-Regeln mit **vfs.fs.discovery** und unterschiedlichen Parametern für jede Regel einzurichten, 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 Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schüsselabgleich können in Kombination mit DenyKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge ihres Auftretens verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

BufferSend

Das Zeitintervall in Sekunden, das bestimmt, wie oft Werte aus dem Puffer an den Zabbix Server gesendet werden. Beachten Sie, dass die Daten früher gesendet werden, wenn der Puffer voll ist.

Standard: 5
 Bereich: 1-3600

BufferSize

Die maximale Anzahl von Werten im Speicherpuffer. 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

Der Control-Socket, der verwendet wird, um Laufzeitbefehle mit der Option '-R' zu senden.

Standard: `/tmp/agent.sock`

DebugLevel

Geben Sie den Debug-Level an:
0 - grundlegende Informationen über das Starten und Stoppen 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

DenyKey

Verweigert die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schlüsselabgleich können in Kombination mit AllowKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge ihres Auftretens verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

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

ForceActiveChecksOnStart

Aktive Prüfungen unmittelbar nach dem Neustart für die zuerst empfangene Konfiguration ausführen. Auch als Konfigurationsparameter pro Plugin verfügbar, zum Beispiel: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

HeartbeatFrequency

Die Häufigkeit von Heartbeat-Nachrichten in Sekunden. Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Ein optionaler Parameter, der die Host-Schnittstelle (IP-Adresse oder DNS-Name) definiert, die während des Prozesses der Host-Autoregistrierung verwendet wird. Dieser Wert wird verwendet, um die Schnittstelle im neu erstellten Host zu befüllen, und ermöglicht die explizite Konfiguration entweder einer IP- oder einer DNS-Adresse. Weitere Details finden Sie unter DNS als Standardschnittstelle verwenden.

Falls nicht definiert, wird der Wert von HostInterfaceItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert das Limit von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Ein optionaler Parameter, der einen Datenpunkt definiert, der verwendet wird, um die Host-Schnittstelle (IP-Adresse oder DNS-Name) während des Prozesses der Host-Autoregistrierung zu bestimmen.
Dieser Wert wird nur verwendet, wenn HostInterface nicht definiert ist.
Weitere Details 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 für AllowKey/DenyKey unterstützt.

HostMetadata

Ein optionaler Parameter, der die Metadaten definiert, die verwendet werden, um den Host während des Autoregistrierungs-prozesses zu identifizieren oder zu unterscheiden. HostMetadata ermöglicht es, zwischen Hosts über den Hostnamen hinaus zu unterscheiden.

Falls nicht definiert, wird der Wert von HostMetadataItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der angegebene Wert das Limit von 2034 Byte 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 sind, aber keine gültigen IPs oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Bereich: 0-2034 Byte

HostMetadataItem

Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen von Host-Metadaten verwendet wird. Diese Option wird nur verwendet, wenn HostMetadata nicht definiert ist.

Der Wert von HostMetadataItem wird bei jedem Versuch der Autoregistrierung abgerufen und nur im Prozess der Host-Autoregistrierung verwendet. HostMetadataItem ermöglicht die Unterscheidung zwischen Hosts über den Hostnamen hinaus.

Benutzerparameter und Aliasse werden unterstützt. Der Datenpunkt system.run[] wird unabhängig von den Einstellungen AllowKey/DenyKey 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 sind, aber keine gültigen IP-Adressen oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Hostname

Eine Liste von durch Kommas getrennten, eindeutigen Hostnamen, bei denen Groß-/Kleinschreibung beachtet wird. Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen. Der Wert wird von 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

Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Abrufen des Host-Namens verwendet wird. 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/DenyKey unterstützt.

Standard: `system.hostname`

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden. Während der Installation erstellt Zabbix das Include-Verzeichnis in /usr/local/etc, sofern dies nicht während der Kompilierung geändert wurde. Der Pfad kann relativ zum Speicherort der Datei *zabbix_agent2.conf* angegeben werden.
Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Asterisk-Platzhalterzeichen für den Musterabgleich unterstützt.
Siehe besondere Hinweise zu Einschränkungen.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Eine Liste von durch Kommas getrennten IP-Adressen, auf denen der Agent lauschen soll. Die erste IP-Adresse wird an den Zabbix Server gesendet, falls eine Verbindung zu ihm hergestellt wird, um die Liste der aktiven Prüfungen abzurufen.

Standard: `0.0.0.0`

ListenPort

Der Agent lauscht auf diesem Port auf Verbindungen vom Server.

Standard: 10050
 Bereich: 1024-32767

LogFile

Der Name der Protokolldatei.

Standard: `/tmp/zabbix_agent2.log`
 Erforderlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe einer 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

Der Typ der Log-Ausgabe:
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

Standard: `file`

PersistentBufferFile

Die Datei, in der Zabbix Agent 2 die SQLite-Datenbank speichern soll. Muss ein vollständiger Dateiname sein. Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

Der Zeitraum, für den Daten gespeichert werden sollen, wenn keine Verbindung zum Server oder Proxy besteht. Ältere Daten gehen verloren. Protokolldaten bleiben erhalten. Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (*EnablePersistentBuffer=1*).

Standard: 1h
 Bereich: 1m-365d

PidFile

Der Name der PID-Datei.

Standard: `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

Das Limit für Prüfungen pro Plugin `<PluginName>`, die gleichzeitig ausgeführt werden können.

Standard: 1000 Bereich: 1-1000

Plugins.Log.MaxLinesPerSecond

Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an den Zabbix Server oder Proxy sendet, wenn aktive Prüfungen vom Typ „log“ und „logrt“ verarbeitet werden. Der angegebene Wert wird durch den Parameter „maxlines“ überschrieben, der im Datenpunkt-Schlüssel „log“ und „logrt“ angegeben ist.
Hinweis: Zabbix verarbeitet 10-mal mehr neue Zeilen als in *MaxLinesPerSecond* festgelegt, um die erforderliche Zeichenfolge in Log-Datenpunkten zu finden.

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. Es werden keine Protokolleinträge erstellt, wenn system.run[] lokal durch die Parameter HostMetadataItem, HostInterfaceItem oder HostnameItem gestartet wird.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

PluginSocket

Der Pfad zum UNIX-Socket für die Kommunikation mit ladbaren Plugins.

Standard: `/tmp/agent.plugin.sock`

PluginTimeout

Das Zeitlimit für Verbindungen mit ladbaren Plugins, in Sekunden.

Standard: `Timeout<br>` Bereich: 1-30

RefreshActiveChecks

Wie oft die Liste der aktiven Prüfungen aktualisiert wird, in Sekunden. Beachten Sie, dass nach einem fehlgeschlagenen Aktualisieren der aktiven Prüfungen die nächste Aktualisierung in 60 Sekunden versucht wird.

Standard: 5
 Bereich: 1-86400

Server

Eine Liste von durch Kommas getrennten IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern oder 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. Leerzeichen sind erlaubt. Wenn dieser Parameter nicht angegeben ist, 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

Die Adresse des Zabbix Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden. Die Server/Proxy-Adresse ist eine IP-Adresse oder ein DNS-Name sowie ein optionaler, durch einen Doppelpunkt getrennter Port.
Die Cluster-Konfiguration besteht aus einer oder mehreren, durch Semikolon getrennten Adressen von Mitgliedern einer Server- oder Proxy-Gruppe. Es können mehrere Zabbix Server/Cluster und Zabbix Proxys angegeben werden, getrennt durch Komma. Sofern keine Proxy-Gruppen verwendet werden, 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 Zabbix Cluster für diesen Proxy nicht angegeben werden.
Es können mehrere durch Komma getrennte Adressen angegeben werden, um mehrere unabhängige Zabbix Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn kein Port 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 bei 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 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

Die Quell-IP-Adresse für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy.
- das Herstellen von Verbindungen bei der Ausführung einiger Datenpunkte (web.page.get, net.tcp.port usw.).

StatusPort

Falls gesetzt, lauscht der Agent auf diesem Port auf HTTP-Statusanfragen (http://localhost:<port>/status).

Bereich: 1024-32767

Timeout

Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Proxy oder Server gewartet wird.

Dieser Parameter definiert die Dauer verschiedener Kommunikationsvorgänge, darunter:

- Warten auf eine Antwort vom Zabbix Server;
- Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunkt-Konfiguration und Datenpunkt-Daten bei aktiven Checks;
- Abrufen von Log-Daten über logfile;
- Senden von Heartbeat-Nachrichten;
- maximale Dauer für `vfs.*`-Prüfungen;
- Verwendung als Fallback in Szenarien, in denen ein Server oder Proxy älter als Version 7.0 Checks ohne Timeouts sendet.

Dieses Timeout wird **nicht** für jene Agent-Checks verwendet, deren Timeout-Einstellungen im Frontend konfigurierbar sind (global, auf Proxy-Ebene oder pro Datenpunkt).

Standard: 3
 Bereich: 1-30

TLSAccept

Die zu akzeptierenden eingehenden Verbindungen. Wird für passive Prüfungen verwendet. Es können mehrere Werte angegeben werden, durch Komma getrennt:
unencrypted - Verbindungen ohne Verschlüsselung akzeptieren (Standard)
psk - Verbindungen mit TLS und einem Pre-Shared Key (PSK) akzeptieren
cert - Verbindungen mit TLS und einem Zertifikat akzeptieren

Erforderlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch für eine *unencrypted*-Verbindung); andernfalls nein

TLSCAFile

Der vollständige Pfadname der Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname der Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSCipherAll

Die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien für Chiffresuiten bei zertifikats- und PSK-basierter Verschlüsselung.

Beispiel:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Die OpenSSL-Chiffrezeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer. Überschreibt die standardmäßigen Auswahlkriterien für Chiffriersuiten für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Die OpenSSL-Chiffrenzeichenfolge (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für zertifikatsbasierte Verschlüsselung.

Beispiel:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Beachten Sie, dass dieser Parameter nicht zusammen mit `TLSAccept=cert,psk` verwendet werden kann; für Zertifikatsverbindungen (`TLSConnect=cert`) verwenden Sie stattdessen TLSCipherAll.

TLSCipherCert13

Die OpenSSL-Chiffrezeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer. Überschreibt die standardmäßigen Auswahlkriterien der Chiffresuite für zertifikatbasierte Verschlüsselung.

Beachten Sie, dass dieser Parameter nicht zusammen mit `TLSAccept=cert,psk` verwendet werden kann; für eine Zertifikatverbindung (`TLSConnect=cert`) verwenden Sie stattdessen TLSCipherAll13.

TLSCipherPSK

Die OpenSSL-Chiffrezeichenfolge (TLS 1.2). Überschreibt die standardmäßigen Auswahlkriterien der Chiffre-Suite für PSK-basierte Verschlüsselung.

Beispiel:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Die OpenSSL-Chiffrezeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer. Überschreibt die standardmäßigen Auswahlkriterien der Chiffriersuite für PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Wie der Agent eine Verbindung zum Zabbix Server oder Proxy herstellen soll. Wird für aktive Prüfungen verwendet. Es kann nur ein Wert angegeben werden:
unencrypted - Verbindung ohne Verschlüsselung herstellen (Standard)
psk - Verbindung über TLS und einen Pre-Shared Key (PSK) herstellen
cert - Verbindung über TLS und ein Zertifikat herstellen

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch bei einer *unencrypted*-Verbindung); andernfalls nein

TLSCRLFile

Der vollständige Pfadname der Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Der vollständige Pfadname der Datei, die den vorab geteilten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSPSKIdentity

Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSServerCertIssuer

Der zulässige Aussteller des Server-(Proxy-)Zertifikats.

TLSServerCertSubject

Der zulässige Zertifikat-Betreff des Servers (Proxy).

UnsafeUserParameters

Erlaubt, alle Zeichen in Argumenten für benutzerdefinierte Parameter zu übergeben. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbruchzeichen nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

UserParameter

Ein benutzerdefinierter Parameter zur Überwachung. Es kann mehrere benutzerdefinierte Parameter geben.
Format: UserParameter=<key>,<shell command>
Beachten Sie, dass der Shell-Befehl keine leere Zeichenfolge oder nur ein Zeilenende zurückgeben darf. Shell-Befehle können relative Pfade haben, wenn der Parameter UserParameterDir angegeben ist.

Beispiel:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Der standardmäßige Suchpfad für UserParameter-Befehle. Falls verwendet, ändert der Agent vor der Ausführung eines Befehls sein Arbeitsverzeichnis 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
```

5 Zabbix Agent (Windows)

Übersicht

Die von der Windows-Zabbix-Agent-Konfigurationsdatei (zabbix_agentd.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
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
BufferSend	Daten nicht länger als N Sekunden im Puffer behalten.
BufferSize	Die maximale Anzahl von Werten im Speicherpuffer.
DebugLevel	Die Debug-Stufe.
DenyKey	Verweigert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
EnableRemoteCommands	Ob Remote-Befehle vom Zabbix-Server erlaubt sind.
HeartbeatFrequency	Die Häufigkeit von Heartbeat-Nachrichten in Sekunden.
HostInterface	Ein optionaler Parameter, der die Host-Schnittstelle definiert.
HostInterfaceItem	Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Ermitteln der Host-Schnittstelle verwendet wird.
HostMetadata	Ein optionaler Parameter, der die Host-Metadaten definiert.
HostMetadataItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Ermitteln der Host-Metadaten verwendet wird.
Hostname	Ein optionaler Parameter, der den Hostnamen definiert.
HostnameItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Ermitteln des Hostnamens verwendet wird.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
ListenBacklog	Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
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.
LogFileSize	Die maximale Größe der Protokolldatei.
LogRemoteCommands	Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.
LogType	Der Typ der Protokollausgabe.
MaxLinesPerSecond	Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an den Zabbix-Server oder Proxy sendet, wenn aktive Prüfungen vom Typ 'log' und 'logrt' verarbeitet werden.
PerfCounter	Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungszählers <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
PerfCounterEn	Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungszählers <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt. Im Vergleich zu PerfCounter müssen die perfcounter-Pfade auf Englisch sein.
RefreshActiveChecks	Wie oft die Liste der aktiven Prüfungen aktualisiert wird.
Server	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern und Zabbix-Proxys.
ServerActive	Die Adresse des Zabbix-Servers/Proxys oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden.
SourceIP	Die Quell-IP-Adresse.
StartAgents	Die Anzahl der vorab geforkten Instanzen von zabbix_agentd, die passive Prüfungen verarbeiten.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix-Proxy oder -Server gewartet wird.
TLSAccept	Welche eingehenden Verbindungen akzeptiert werden sollen.
TLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) zur Verifizierung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Der vollständige Pfadname einer Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

Parameter	Beschreibung
TLSConnect	Wie der Agent eine Verbindung zum Zabbix-Server oder Proxy herstellen soll.
TLSCRLFile	Der vollständige Pfadname einer Datei, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Der vollständige Pfadname einer Datei, die den vorab geteilten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix-Server verwendet wird.
TLSPSKIdentity	Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für verschlüsselte Kommunikation mit dem Zabbix-Server verwendet wird.
TLSServerCertIssuer	Der zulässige Aussteller des Server-(Proxy-)Zertifikats.
TLSServerCertSubject	Das zulässige Subjekt des Server-(Proxy-)Zertifikats.
UnsafeUserParameters	Erlaubt, alle Zeichen in Argumenten für benutzerdefinierte Parameter zu übergeben.
UserParameter	Ein benutzerdefinierter Parameter zur Überwachung.
UserParameterDir	Der Standard-Suchpfad für UserParameter-Befehle.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass ein Parameter verpflichtend ist.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standardwerten, 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

Legt einen Alias für einen Datenpunkt-Schlüssel fest. Er kann verwendet werden, um einen langen und komplexen Datenpunkt-Schlüssel durch einen kürzeren und einfacheren zu ersetzen.
 Es können mehrere *Alias*-Parameter vorhanden sein. Mehrere Parameter mit demselben *Alias*-Schlüssel sind nicht zulässig.
 Verschiedene *Alias*-Schlüssel können auf denselben Datenpunkt-Schlüssel verweisen.
 Aliasse können in *HostMetadataItem* verwendet werden, jedoch nicht im Parameter *HostnameItem* oder *PerfCounter*.

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[*]
```

Dies ermöglicht die Verwendung des Schlüssels **cpu.load**, um die CPU-Auslastung mit Standardparametern abzurufen, sowie die Verwendung von **cpu.load[percpu,avg15]**, um spezifische Daten zur CPU-Auslastung abzurufen.

Beispiel 3: Ausführen mehrerer Regeln zur Low-Level-Discovery, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun ist es möglich, mehrere Discovery-Regeln mit **vfs.fs.discovery** und unterschiedlichen Parametern für jede Regel einzurichten, 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 Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schlüsselabgleich können in Kombination mit DenyKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

BufferSend

Daten nicht länger als N Sekunden im Puffer behalten.

Standard: 5
 Bereich: 1-3600

BufferSize

Die maximale Anzahl von Werten im Speicherpuffer. Der Agent sendet alle gesammelten Daten an den Zabbix Server oder Proxy, wenn der Puffer voll ist.

Standard: 100
 Bereich: 2-65535

DebugLevel

Geben Sie den Debug-Level an:
0 - grundlegende Informationen über das Starten und Stoppen 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

DenyKey

Verweigert die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schlüsselabgleich können in Kombination mit AllowKey definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge des Auftretens nacheinander verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

EnableRemoteCommands

Ob Remote-Befehle vom Zabbix Server erlaubt sind. Dieser Parameter ist **veraltet**; verwenden Sie stattdessen AllowKey=system.run[*] oder DenyKey=system.run[*].
Er ist ein interner Alias für die Parameter AllowKey/DenyKey, abhängig vom Wert:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

HeartbeatFrequency

Die Häufigkeit von Heartbeat-Nachrichten in Sekunden. Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Ein optionaler Parameter, der die Host-Schnittstelle (IP-Adresse oder DNS-Name) definiert, die während des Prozesses der Host-Autoregistrierung verwendet wird. Dieser Wert wird verwendet, um die Schnittstelle im neu erstellten Host zu befüllen, und ermöglicht die explizite Konfiguration entweder einer IP- oder einer DNS-Adresse. Weitere Details finden Sie unter DNS als Standardschnittstelle verwenden.

Falls nicht definiert, wird der Wert von HostInterfaceItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert das Limit von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Ein optionaler Parameter, der einen Datenpunkt definiert, der verwendet wird, um die Host-Schnittstelle (IP-Adresse oder DNS-Name) während des Prozesses der Host-Autoregistrierung zu bestimmen.
Dieser Wert wird nur verwendet, wenn HostInterface nicht definiert ist.
Weitere Details 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/DenyKey unterstützt.

HostMetadata

Ein optionaler Parameter, der die Metadaten definiert, die verwendet werden, um den Host während des Autoregistrierungs-Prozesses des Hosts (aktiver Agent) zu identifizieren oder zu unterscheiden. HostMetadata ermöglicht es, zwischen Hosts über den Hostnamen hinaus zu unterscheiden.

Falls nicht definiert, wird der Wert von HostMetadataItem bezogen.

Der Agent gibt einen Fehler aus und startet nicht, 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 sind, aber keine gültigen IP-Adressen oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt — die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Bereich: 0-2034 Byte

HostMetadataItem

Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen von Host-Metadaten verwendet wird. Diese Option wird nur verwendet, wenn HostMetadata nicht definiert ist.

Der Wert von HostMetadataItem wird bei jedem Versuch der Autoregistrierung abgerufen und nur im Prozess der Host-Autoregistrierung verwendet (aktiver Agent). HostMetadataItem ermöglicht die Unterscheidung zwischen Hosts über den Hostnamen hinaus.

Benutzerparameter, Leistungsindikatoren und Aliase werden unterstützt. Der Datenpunkt system.run[] wird unabhängig von den Einstellungen AllowKey/DenyKey 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 sind, aber keine gültigen IP-Adressen oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt — die Ausgabe wird am ersten Zeilenumbruch abgeschnitten.

Hostname

Eine Liste von durch Kommas getrennten, eindeutigen Hostnamen, bei denen Groß- und Kleinschreibung beachtet wird. Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen. Der Wert wird von 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

Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen des Host-Namens verwendet wird. Diese Option wird nur verwendet, wenn Hostname nicht definiert ist. Benutzerparameter, Leistungsindikatoren oder Aliase werden nicht unterstützt, aber der Datenpunkt system.run[] wird unabhängig von den Werten von AllowKey/DenyKey unterstützt.
Siehe auch eine detailliertere Beschreibung.

Standard: `system.hostname`

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden (standardmäßig unter `C:\Program Files\Zabbix Agent`, wenn der Zabbix Agent mit Windows-MSI-Installer-Paketen installiert wurde; im während der Installation angegebenen Ordner, wenn der Zabbix Agent als ZIP-Archiv installiert wurde). Alle eingebundenen Dateien müssen eine korrekte Syntax haben, andernfalls wird der Agent nicht gestartet.
Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Asterisk-Platzhalterzeichen für den Musterabgleich unterstützt.
Siehe besondere Hinweise zu Einschränkungen.

Beispiel:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange.
Der Standardwert ist eine fest kodierte Konstante, die vom System abhängt.
Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementierungsspezifische Maximum“ gekürzt werden.

Standard: `SOMAXCONN`
 Bereich: 0 - INT_MAX

ListenIP

Eine Liste von durch Kommas getrennten IP-Adressen, auf denen der Agent lauschen soll.

Standard: `0.0.0.0`

ListenPort

Der Agent lauscht auf diesem Port auf Verbindungen vom Server.

Standard: 10050
 Bereich: 1024-32767

LogFile

Der Name der Agent-Protokolldatei.

Standard: `c:\zabbix_agentd.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 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

Aktiviert 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

Der Typ der Protokollausgabe:
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.

Standard: `file`

MaxLinesPerSecond

Die maximale Anzahl neuer Zeilen, 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, der im Datenpunkt-Schlüssel „log“, „logrt“ oder „eventlog“ angegeben ist.
Hinweis: Zabbix verarbeitet 10-mal mehr neue Zeilen als in *MaxLinesPerSecond* festgelegt, um die erforderliche Zeichenfolge in Log-Datenpunkten zu finden.

Standard: 20
 Bereich: 1-1000

PerfCounter

Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert für den Systemleistungsindikator <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
Syntax: <parameter_name>,”<perf_counter_path>”,<period>

Wenn Sie beispielsweise die durchschnittliche Anzahl von Prozessor-Interrupts 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 beim Erstellen eines Datenpunkts als Datenpunktschlüssel zu verwenden. Für die Berechnung des Durchschnittswerts werden jede Sekunde Stichproben erfasst.
Sie können ”typeperf -qx” ausführen, um die Liste aller in Windows verfügbaren Leistungsindikatoren abzurufen.

PerfCounterEn

Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert für den Systemleistungsindikator <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt. Im Vergleich zu PerfCounter müssen die Pfade der Leistungsindikatoren bei perfcounter 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 von Prozessor-Interrupts pro Sekunde für die letzte Minute erhalten möchten, können Sie einen neuen Parameter ”interrupts” wie folgt definieren:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Bitte beachten Sie die doppelten Anführungszeichen um den Pfad des Leistungsindikators. Der Parametername (interrupts) ist beim Erstellen eines Datenpunkts als Schlüssel des Datenpunkts zu verwenden. Für die Berechnung des Durchschnittswerts werden jede Sekunde Stichproben genommen.
Die Liste der englischen Zeichenfolgen finden Sie, indem Sie den folgenden Registrierungsschlüssel anzeigen: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Wie oft die Liste der aktiven Prüfungen aktualisiert wird, in Sekunden. Beachten Sie, dass nach einem fehlgeschlagenen Aktualisieren der aktiven Prüfungen die nächste Aktualisierung in 60 Sekunden versucht wird.

Standard: 5
 Bereich: 1-86400

Server

Eine Liste von durch Kommas getrennten IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern oder Zabbix-Proxys. Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, dann 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. Beachten Sie, dass „IPv4-kompatible IPv6-Adressen" (Präfix 0000::/96) unterstützt werden, aber durch RFC4291 als veraltet eingestuft sind. Leerzeichen sind erlaubt.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Verbindlich: ja, wenn StartAgents nicht explizit auf 0 gesetzt ist

ServerActive

Die Adresse des Zabbix Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden. Die Server/Proxy-Adresse ist eine IP-Adresse oder ein DNS-Name sowie optional ein durch einen Doppelpunkt getrennter Port.
Die Cluster-Konfiguration besteht aus einer oder mehreren durch Semikolon getrennten Adressen von Mitgliedern einer Server- oder Proxy-Gruppe. Es können mehrere Zabbix Server/Cluster und Zabbix Proxys angegeben werden, getrennt durch Komma. Sofern keine Proxy-Gruppen verwendet werden, 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.
Es können mehrere durch Komma getrennte Adressen angegeben werden, um mehrere unabhängige Zabbix Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn kein Port 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 bei 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 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
```

Bereich: (*)

SourceIP

Die Quell-IP-Adresse für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy;
- das Herstellen von Verbindungen bei der Ausführung einiger Datenpunkte (web.page.get, net.tcp.port usw.).

StartAgents

Die Anzahl der vorab geforkten Instanzen von zabbix_agentd, die passive Prüfungen verarbeiten. Wenn auf 0 gesetzt, sind passive Prüfungen deaktiviert und der Agent lauscht auf keinem TCP-Port.

Standard: 10
 Bereich: 0-100 (*)

Timeout

Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Proxy oder Server gewartet werden soll.

Dieser Parameter definiert die Dauer verschiedener Kommunikationsvorgänge, darunter:

- Warten auf eine Antwort vom Zabbix Server;
- Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunkt-Konfiguration und Datenpunkt-Daten bei aktiven Prüfungen;
- Abrufen von Protokolldaten über die Überwachung von Logdateien oder Windows-Ereignisprotokollen;
- Senden von Heartbeat-Nachrichten;
- maximale Dauer für `vfs.*`-Prüfungen;
- Verwendung durch Zabbix Agent-Module;
- 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 im Frontend konfigurierbare Timeout-Einstellungen haben (auf globaler Ebene, Proxy-Ebene oder pro Datenpunkt).

Standard: 3
 Bereich: 1-30

TLSAccept

Die eingehenden Verbindungen, die akzeptiert werden sollen. Wird für passive Prüfungen verwendet. Mehrere Werte können angegeben werden, durch Komma getrennt:
unencrypted - Verbindungen ohne Verschlüsselung akzeptieren (Standard)
psk - Verbindungen mit TLS und einem vorab geteilten Schlüssel (PSK) akzeptieren
cert - Verbindungen mit TLS und einem Zertifikat akzeptieren

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch für eine *unencrypted*-Verbindung); andernfalls nein

TLSCAFile

Der vollständige Pfadname der Datei, die die Zertifikate der CA(s) der obersten Ebene für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname der Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSConnect

Wie der Agent eine Verbindung zum Zabbix Server oder Proxy herstellen soll. Wird für aktive Prüfungen verwendet. Es kann nur ein Wert angegeben werden:
unencrypted - Verbindung ohne Verschlüsselung herstellen (Standard)
psk - Verbindung mit TLS und einem Pre-Shared Key (PSK) herstellen
cert - Verbindung mit TLS und einem Zertifikat herstellen

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch bei einer *unencrypted*-Verbindung); andernfalls nein

TLSCRLFile

Der vollständige Pfadname der Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Der vollständige Pfadname der Datei, die den vorab geteilten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSPSKIdentity

Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSServerCertIssuer

Der zulässige Aussteller des Server-(Proxy-)Zertifikats.

TLSServerCertSubject

Der zulässige Zertifikat-Betreff des Servers (Proxy).

UnsafeUserParameters

Erlaubt, alle Zeichen in Argumenten an benutzerdefinierte Parameter zu übergeben. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbruchzeichen nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

UserParameter

Ein benutzerdefinierter Parameter zur Überwachung. Es kann mehrere benutzerdefinierte Parameter geben.
Format: UserParameter=<key>,<shell command>
Beachten Sie, dass der Shell-Befehl keine leere Zeichenfolge oder nur ein Zeilenende zurückgeben darf. Shell-Befehle können relative Pfade haben, wenn der Parameter UserParameterDir angegeben ist.

Beispiel:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Der Standard-Suchpfad für UserParameter-Befehle. Falls verwendet, 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
```

Note:
(*) Die Anzahl der in ServerActive aufgeführten aktiven Server plus die Anzahl der in StartAgents angegebenen vorab geforkten Instanzen für passive Prüfungen muss kleiner als 64 sein.

6 Zabbix Agent 2 (Windows)

Übersicht

Zabbix Agent 2 ist eine neue Generation des Zabbix Agent und kann anstelle des Zabbix Agent verwendet werden.

Die von der Windows-Konfigurationsdatei für Zabbix Agent 2 (zabbix_agent2.conf) unterstützten Parameter sind in diesem Abschnitt aufgeführt.

Die Parameter sind ohne zusätzliche Informationen aufgeführt. Klicken Sie auf den Parameter, um die vollständigen Details anzuzeigen.

Parameter	Beschreibung
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
BufferSend	Daten nicht länger als N Sekunden im Puffer behalten.
BufferSize	Die maximale Anzahl von Werten im Speicherpuffer.
ControlSocket	Der Control-Socket, der zum Senden von Laufzeitbefehlen mit der Option '-R' verwendet wird.
DebugLevel	Die Debug-Stufe.
DenyKey	Verweigert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
EnablePersistentBuffer	Aktiviert die Verwendung eines lokalen persistenten Speichers für aktive Datenpunkte.
ForceActiveChecksOnStart	Führt aktive Prüfungen unmittelbar nach dem Neustart für die erste empfangene Konfiguration aus.
HeartbeatFrequency	Die Häufigkeit von Heartbeat-Nachrichten in Sekunden.
HostInterface	Ein optionaler Parameter, der die Host-Schnittstelle definiert.
HostInterfaceItem	Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Abrufen der Host-Schnittstelle verwendet wird.
HostMetadata	Ein optionaler Parameter, der die Host-Metadaten definiert.
HostMetadataItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen der Host-Metadaten verwendet wird.
Hostname	Ein optionaler Parameter, der den Hostnamen definiert.
HostnameItem	Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen des Hostnamens verwendet wird.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
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.
LogFileSize	Die maximale Größe der Protokolldatei.
LogType	Der Typ der Protokollausgabe.
PersistentBufferFile	Die Datei, in der Zabbix Agent 2 die SQLite-Datenbank speichern soll.
PersistentBufferPeriod	Der Zeitraum, für den Daten gespeichert werden sollen, wenn keine Verbindung zum Server oder Proxy besteht.
Plugins.<PluginName>.System.Capacity	Die Begrenzung von Prüfungen pro Plugin, die gleichzeitig ausgeführt werden können.
Plugins.Log.MaxLinesPerSecond	Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an den Zabbix Server oder Proxy sendet, wenn aktive Prüfungen vom Typ 'log' und 'logrt' verarbeitet werden.
Plugins.SystemRun.LogRemoteCommands	Aktiviert die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
PluginSocket	Der Pfad zum UNIX-Socket für die Kommunikation mit ladbaren Plugins.
PluginTimeout	Das Timeout für Verbindungen mit ladbaren Plugins in Sekunden.

Parameter	Beschreibung
PerfCounter	Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungszählers <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
PerfCounterEn	Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungszählers <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt. Im Vergleich zu PerfCounter müssen die perfcounter-Pfade auf Englisch angegeben werden.
RefreshActiveChecks	Wie oft die Liste der aktiven Prüfungen aktualisiert wird.
Server	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix Servern und Zabbix Proxys.
ServerActive	Die Adresse des Zabbix Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden.
SourceIP	Die Quell-IP-Adresse.
StatusPort	Falls gesetzt, lauscht der Agent auf diesem Port auf HTTP-Statusanfragen (http://localhost:<port>/status).
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Proxy oder Server gewartet werden soll.
TLSAccept	Welche eingehenden Verbindungen akzeptiert werden sollen.
TLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Der vollständige Pfadname einer Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll13	Die OpenSSL-Chiffrezeichenfolge (TLS 1.3). Überschreibt die Standardkriterien zur Auswahl der Cipher Suite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert13	Die OpenSSL-Chiffrezeichenfolge (TLS 1.3). Überschreibt die Standardkriterien zur Auswahl der Cipher Suite für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK13	Die OpenSSL-Chiffrezeichenfolge (TLS 1.3). Überschreibt die Standardkriterien zur Auswahl der Cipher Suite für PSK-basierte Verschlüsselung.
TLSConnect	Wie sich der Agent mit dem Zabbix Server oder Proxy verbinden soll.
TLSCRLFile	Der vollständige Pfadname einer Datei, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Der vollständige Pfadname einer Datei, die den vorinstallierten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.
TLSPSKIdentity	Die Identitätszeichenfolge des vorinstallierten Schlüssels, die für verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.
TLSServerCertIssuer	Der zulässige Aussteller des Server-(Proxy-)Zertifikats.
TLSServerCertSubject	Der zulässige Betreff des Server-(Proxy-)Zertifikats.
UnsafeUserParameters	Erlaubt, alle Zeichen in Argumenten an benutzerdefinierte Parameter zu übergeben.
UserParameter	Ein benutzerdefinierter Parameter zur Überwachung.
UserParameterDir	Der Standardsuchpfad für UserParameter-Befehle.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass ein Parameter obligatorisch ist.

Beachten Sie:

- Die Standardwerte entsprechen den Prozessstandardwerten, 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

Legt einen Alias für einen Datenpunktschlüssel fest. Er kann verwendet werden, um einen langen und komplexen Datenpunktschlüssel durch einen kürzeren und einfacheren zu ersetzen.
 Es können mehrere *Alias*-Parameter vorhanden sein. Mehrere Parameter mit demselben *Alias*-Schlüssel sind nicht zulässig.
 Verschiedene *Alias*-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliasse können in *HostMetadataItem*, aber 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[*]
```

Dies ermöglicht die Verwendung des Schlüssels **cpu.load**, um die CPU-Auslastung mit Standardparametern abzurufen, sowie die Verwendung von **cpu.load[percpu,avg15]**, um spezifische Daten zur CPU-Auslastung abzurufen.

Beispiel 3: Ausführen mehrerer Regeln zur Low-Level-Discovery, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun ist es möglich, mehrere Discovery-Regeln mit **vfs.fs.discovery** und unterschiedlichen Parametern für jede Regel einzurichten, 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 Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schüsselabgleich können in Kombination mit DenyKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge ihres Auftretens verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

BufferSend

Das Zeitintervall in Sekunden, das bestimmt, wie oft Werte aus dem Puffer an den Zabbix Server gesendet werden.
Beachten Sie, dass die Daten früher gesendet werden, wenn der Puffer voll ist.

Standard: 5
 Bereich: 1-3600

BufferSize

Die maximale Anzahl von Werten im Speicherpuffer. 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

Der Control-Socket, der verwendet wird, um Laufzeitbefehle mit der Option '-R' zu senden.

Standard: `\\.\pipe\agent.sock`

DebugLevel

Geben Sie den Debug-Level an:
0 - grundlegende Informationen über das Starten und Stoppen 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

DenyKey

Verweigert die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen "*" unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
Mehrere Regeln für den Schlüsselabgleich können in Kombination mit AllowKey definiert werden. Die Parameter werden einzeln entsprechend ihrer Reihenfolge des Auftretens verarbeitet. Siehe auch: Einschränken von Agent-Prüfungen.

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

ForceActiveChecksOnStart

Aktive Prüfungen unmittelbar nach dem Neustart für die zuerst empfangene Konfiguration ausführen. Auch als Konfigurationsparameter pro Plugin verfügbar, zum Beispiel: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

HeartbeatFrequency

Die Häufigkeit von Heartbeat-Nachrichten in Sekunden. Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Ein optionaler Parameter, der die Host-Schnittstelle (IP-Adresse oder DNS-Name) definiert, die während des Prozesses der Host-Autoregistrierung 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 Details finden Sie unter DNS als Standard-schnittstelle verwenden.

Falls nicht definiert, wird der Wert von HostInterfaceItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert das Limit von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Ein optionaler Parameter, der einen Datenpunkt definiert, der verwendet wird, um die Host-Schnittstelle (IP-Adresse oder DNS-Name) während des Prozesses der Host-Autoregistrierung zu bestimmen. Dieser Wert wird nur verwendet, wenn HostInterface nicht definiert ist. Weitere Details 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/DenyKey unterstützt.

HostMetadata

Ein optionaler Parameter, der die Metadaten definiert, die verwendet werden, um den Host während des Prozesses der Autoregistrierung des Hosts zu identifizieren oder zu unterscheiden (aktiver Agent). HostMetadata ermöglicht es, zwischen Hosts über den Hostnamen hinaus zu unterscheiden.

Falls nicht definiert, wird der Wert von HostMetadataItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der angegebene Wert das Limit von 2034 Byte 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 sind, aber keine gültigen IPs oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt — die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Bereich: 0-2034 Byte

HostMetadataItem

Ein optionaler Parameter, der einen Zabbix-Agent-Datenpunkt definiert, der zum Abrufen von Host-Metadaten verwendet wird. Diese Option wird nur verwendet, wenn HostMetadata nicht definiert ist.

Der Wert von HostMetadataItem wird bei jedem Versuch der automatischen Registrierung abgerufen und nur beim Prozess der automatischen Host-Registrierung verwendet (aktiver Agent). HostMetadataItem ermöglicht die Unterscheidung zwischen Hosts über den Hostnamen hinaus.

Benutzerparameter und Aliasse werden unterstützt. Der Datenpunkt system.run[] wird unabhängig von den Einstellungen AllowKey/DenyKey unterstützt.

Während einer Anfrage zur automatischen Registrierung 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 sind, aber keine gültigen IP-Adressen oder DNS-Namen darstellen, ebenfalls zurückgewiesen und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt — die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Hostname

Eine Liste von durch Kommas getrennten, eindeutigen, groß-/kleinschreibungssensitiven Hostnamen. Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen. Der Wert wird von 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

Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Abrufen des Host-Namens verwendet wird. 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/DenyKey unterstützt.

Standard: `system.hostname`

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden (standardmäßig unter `C:\Program Files\Zabbix Agent 2`, wenn der Zabbix Agent mit Windows-MSI-Installer-Paketen installiert wurde; im während der Installation angegebenen Ordner, wenn der Zabbix Agent als ZIP-Archiv installiert wurde). Alle eingebundenen Dateien müssen eine korrekte Syntax haben, andernfalls wird der Agent nicht gestartet. Der Pfad kann relativ zum Speicherort der Datei *zabbix_agent2.conf* sein (z. B. `Include=.\zabbix_agent2.d\plugins.d\*.conf`).
Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Asterisk-Platzhalterzeichen für den Musterabgleich unterstützt.
Siehe besondere Hinweise zu Einschränkungen.

Beispiel:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Eine durch Kommas getrennte Liste von IP-Adressen, auf denen der Agent lauschen soll. Die erste IP-Adresse wird an den Zabbix-Server gesendet, falls eine Verbindung zu ihm hergestellt wird, um die Liste der aktiven Prüfungen abzurufen.

Standard: `0.0.0.0`

ListenPort

Der Agent lauscht auf diesem Port auf Verbindungen vom Server.

Standard: 10050
 Bereich: 1024-32767

LogFile

Der Name der Agent-Protokolldatei.

Standard: `c:\zabbix_agent2.log`
 Verbindlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe einer Protokolldatei 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

LogType

Der Typ der Protokollausgabe:
file - schreibt das Protokoll in die durch den Parameter LogFile angegebene Datei;
console - schreibt das Protokoll in die Standardausgabe.

Standard: `file`

PersistentBufferFile

Die Datei, in der Zabbix Agent 2 die SQLite-Datenbank speichern soll. Muss ein vollständiger Dateiname sein. Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

Der Zeitraum, für den Daten gespeichert werden sollen, wenn keine Verbindung zum Server oder Proxy besteht. Ältere Daten gehen verloren. Protokolldaten bleiben erhalten. Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (*EnablePersistentBuffer=1*).

Standard: 1h
 Bereich: 1m-365d

Plugins.<PluginName>.System.Capacity

Die Begrenzung der Prüfungen pro `<PluginName>`-Plugin, die gleichzeitig ausgeführt werden können.

Standard: 1000 Bereich: 1-1000

Plugins.Log.MaxLinesPerSecond

Die maximale Anzahl neuer Zeilen, die der Agent pro Sekunde an den Zabbix Server oder Proxy sendet, wenn aktive Prüfungen von 'log', 'logrt' und 'eventlog' verarbeitet werden. Der angegebene Wert wird durch den Parameter 'maxlines' überschrieben, der

im Datenpunktschlüssel 'log', 'logrt' oder 'eventlog' 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

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 durch die Parameter HostMetadataItem, HostInterfaceItem oder HostnameItem gestartet wird.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

PluginSocket

Der Pfad zum UNIX-Socket für die Kommunikation mit ladbaren Plugins.

Standard: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Das Timeout für Verbindungen mit ladbaren Plugins, in Sekunden.

Standard: `Timeout`
 Bereich: 1-30

PerfCounter

Definiert einen neuen Parameter <parameter_name>, der der Durchschnittswert für den Systemleistungsindikator <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) ist.
Syntax: <parameter_name>,"<perf_counter_path>",<period>

Wenn Sie beispielsweise die durchschnittliche Anzahl von Prozessor-Interrupts 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 beim Erstellen eines Datenpunkts als Datenpunktschlüssel zu verwenden. Stichproben zur Berechnung des Durchschnittswerts werden jede Sekunde genommen.
Sie können "typeperf -qx" ausführen, um die Liste aller in Windows verfügbaren Leistungsindikatoren zu erhalten.

PerfCounterEn

Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert für den Systemleistungsindikator <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt. Im Vergleich zu PerfCounter müssen die Pfade der Leistungsindikatoren bei perfcounter 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 von Prozessorinterrupts pro Sekunde für die letzte Minute erhalten möchten, können Sie einen neuen Parameter "interrupts" wie folgt definieren:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Bitte beachten Sie die doppelten Anführungszeichen um den Pfad des Leistungsindikators. Der Parametername (interrupts) ist beim Erstellen eines Datenpunkts als Schlüssel des Datenpunkts zu verwenden. Stichproben zur Berechnung des Durchschnittswerts werden jede Sekunde erfasst.
Die Liste der englischen Zeichenfolgen finden Sie unter folgendem Registrierungsschlüssel: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Wie oft die Liste der aktiven Prüfungen aktualisiert wird, in Sekunden. Beachten Sie, dass nach einem fehlgeschlagenen Aktualisieren der aktiven Prüfungen der nächste Aktualisierungsversuch in 60 Sekunden erfolgt.

Standard: 5
 Bereich: 1-86400

Server

Eine Liste von durch Kommas getrennten IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern oder 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. Leerzeichen sind erlaubt. Wenn dieser Parameter nicht angegeben ist, 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

Die Adresse des Zabbix Server/Proxy oder die Cluster-Konfiguration, von der aktive Prüfungen abgerufen werden. Die Server/Proxy-Adresse ist eine IP-Adresse oder ein DNS-Name mit einem optionalen, durch Doppelpunkt getrennten Port.
Die Cluster-Konfiguration besteht aus einer oder mehreren, durch Semikolon getrennten Adressen von Mitgliedern einer Server- oder Proxy-Gruppe. Es können mehrere Zabbix Server/Cluster und Zabbix Proxys angegeben werden, getrennt durch Komma. Sofern keine Proxy-Gruppen verwendet werden, 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.
Es können mehrere durch Komma getrennte Adressen angegeben werden, um mehrere unabhängige Zabbix Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn kein Port 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 bei 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 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

Die Quell-IP-Adresse für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy;
- den Verbindungsaufbau bei der Ausführung einiger Datenpunkte (web.page.get, net.tcp.port usw.).

StatusPort

Falls gesetzt, lauscht der Agent auf diesem Port auf HTTP-Statusanfragen (http://localhost:<port>/status).

Bereich: 1024-32767

Timeout

Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit Zabbix Proxy oder Server gewartet wird.

Dieser Parameter definiert die Dauer verschiedener Kommunikationsvorgänge, darunter:

- Warten auf eine Antwort vom Zabbix Server;
- Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunkt-Konfiguration und Datenpunkt-Daten bei aktiven Prüfungen;
- Abrufen von Protokolldaten über die Überwachung von Logdateien oder des Windows-Ereignisprotokolls;
- Senden von Heartbeat-Nachrichten;
- maximale Dauer für `vfs.*`-Prüfungen;
- 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, deren Timeout-Einstellungen im Frontend konfigurierbar sind (global, auf Proxy-Ebene oder pro Datenpunkt).

Standard: 3
 Bereich: 1-30

TLSAccept

Die eingehenden Verbindungen, die akzeptiert werden sollen. Wird für passive Prüfungen verwendet. Es können mehrere Werte angegeben werden, durch Komma getrennt:
unencrypted - Verbindungen ohne Verschlüsselung akzeptieren (Standard)
psk - Verbindungen mit TLS und einem Pre-Shared Key (PSK) akzeptieren
cert - Verbindungen mit TLS und einem Zertifikat akzeptieren

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch für eine *unencrypted*-Verbindung); andernfalls nein

TLSCAFile

Der vollständige Pfadname der Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname der Datei, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSCipherAll13

Die OpenSSL-Chiffrenzeichenfolge (TLS 1.3). Überschreibt die standardmäßigen Auswahlkriterien für Chiffren-Suites für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Die OpenSSL-Chiffrezeichenfolge (TLS 1.3). Überschreibt die standardmäßigen Auswahlkriterien für Chiffresuiten bei zertifikatsbasierter Verschlüsselung.

Beachten Sie, dass dieser Parameter nicht zusammen mit `TLSAccept=cert,psk` verwendet werden kann; für Zertifikatsverbindungen (`TLSConnect=cert`) verwenden Sie stattdessen TLSCipherAll13.

TLSCipherPSK13

Die OpenSSL-Chiffrenzeichenfolge (TLS 1.3). Überschreibt die standardmäßigen Auswahlkriterien für Chiffren-Suites bei PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Wie der Agent eine Verbindung zum Zabbix Server oder Proxy herstellen soll. Wird für aktive Prüfungen verwendet. Es kann nur ein Wert angegeben werden:
unencrypted - Verbindung ohne Verschlüsselung herstellen (Standard)
psk - Verbindung über TLS und einen Pre-Shared Key (PSK) herstellen
cert - Verbindung über TLS und ein Zertifikat herstellen

Verbindlich: ja, wenn TLS-Zertifikats- oder PSK-Parameter definiert sind (auch für eine *unencrypted*-Verbindung); andernfalls nein

TLSCRLFile

Der vollständige Pfadname der Datei, die gesperrte Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Der vollständige Pfadname der Datei, die den vorab geteilten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSPSKIdentity

Die Identitätszeichenfolge des vorab geteilten Schlüssels, die für die verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSServerCertIssuer

Der zulässige Aussteller des Server-(Proxy-)Zertifikats.

TLSServerCertSubject

Der zulässige Zertifikat-Betreff des Servers (Proxy).

UnsafeUserParameters

Erlaubt, alle Zeichen in Argumenten an benutzerdefinierte Parameter zu übergeben. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbruchzeichen nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

UserParameter

Ein benutzerdefinierter Parameter zur Überwachung. Es kann mehrere benutzerdefinierte Parameter geben.
Format: UserParameter=<key>,<shell command>
Beachten Sie, dass der Shell-Befehl keine leere Zeichenfolge oder nur ein Zeilenende zurückgeben darf. Shell-Befehle können relative Pfade haben, wenn der Parameter UserParameterDir angegeben ist.

Beispiel:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Der standardmäßige Suchpfad für UserParameter-Befehle. Falls verwendet, ändert der Agent vor der Ausführung eines Befehls sein Arbeitsverzeichnis 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
```

7 Configuration parameters of plugins

See configuration parameters for Zabbix agent 2 plugins:

- Ceph plugin
- Docker plugin
- Ember+ plugin
- Memcached plugin
- Modbus plugin
- MongoDB plugin
- MQTT plugin
- MSSQL plugin
- MySQL plugin
- NVIDIA GPU plugin
- Oracle plugin
- PostgreSQL plugin
- Redis plugin
- S.M.A.R.T. plugin

Plugin configuration principles

This page describes plugin configuration principles and best practices.

All plugins are configured using *Plugins.** parameter, which can either be part of the Zabbix agent 2 configuration file or a plugin's own configuration file. If a plugin uses a separate configuration file, path to this file should be specified in the Include parameter of Zabbix agent 2 configuration file.

A typical plugin parameter has the following structure:

Plugins.<PluginName>.<Parameter>=<Value>

Additionally, there are two specific groups of parameters:

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* used for defining default parameter values.
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* used for defining separate sets of parameters for different monitoring targets via named sessions.

All parameter names should adhere to the following requirements:

- it is recommended to capitalize the names of your plugins
- the parameter should be capitalized
- special characters are not allowed
- nesting isn't limited by a maximum level
- the number of parameters is not limited

For example, to perform active checks that do not have *Scheduling* update interval immediately after the agent restart only for the Uptime plugin, set `Plugins.Uptime.System.ForceActiveChecksOnStart=1` in the configuration file. Similarly, to set custom limit for concurrent checks for the CPU plugin, set the `Plugins.CPU.System.Capacity=N` in the configuration file.

Default values

You can set default values for the connection-related parameters (URI, username, password, etc.) in the configuration file in the format:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

For example, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, etc.

If a value for such parameter is not provided in an item key or in the named session parameters, the plugin will use the default value. If a default parameter is also undefined, hardcoded defaults will be used.

> **Note:**
> If an item key does not have any parameters, Zabbix agent 2 will attempt to collect the metric using values defined in the default parameters section.

Named sessions

Named sessions represent an additional level of plugin parameters and can be used to specify separate sets of authentication parameters for each of the instances being monitored. Each named session parameter should have the following structure:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

A session name can be used as a connString item key parameter instead of specifying a URI, username, and/or password separately.

In item keys, the first parameter can be either a connString or a URI. If the first key parameter doesn't match any session name, it will be treated as a URI. Note that passing embedded URI credentials in the item key is not supported, use named session parameters instead.

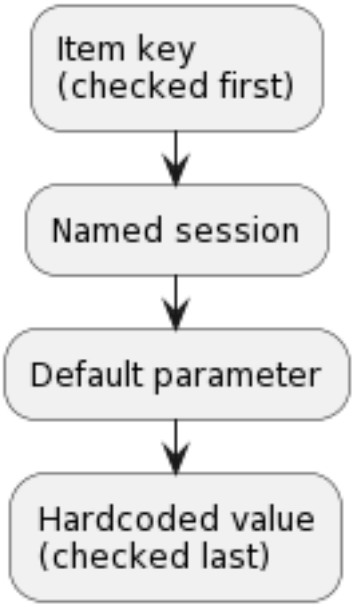
The list of available named session parameters depends on the plugin.

It is possible to override session parameters by specifying new values in the item key parameters (see example).

If a parameter is not defined for the named session, Zabbix agent 2 will use the value defined in the default plugin parameter.

Parameter priority

Zabbix agent 2 plugins search for connection-related parameter values in the following order:



1. The first item key parameter is compared to session names. If no match is found, it is treated as an actual value; in this case, step 3 will be skipped. If a match is found, the parameter value (usually, a URI) must be defined in the named session.
2. Other parameters will be taken from the item key if defined.
3. If an item key parameter (for example, password) is empty, plugin will look for the corresponding named session parameter.
4. If the session parameter is also not specified, the value defined in the corresponding default parameter will be used.
5. If all else fails, the plugin will use the hardcoded default value.

Example 1

Monitoring of two instances "MySQL1" and "MySQL2".

Configuration parameters:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

As a result of this configuration, each session name may be used as a connString in an item key, e.g., `mysql.ping[MySQL1]` or `mysql.ping[MySQL2]`.

Example 2

Providing some of the parameters in the item key.

Configuration parameters:

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

Item key: `pgsql.ping[session1,new_username,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the following parameters:

- URI from session parameter: `192.0.2.234:5432`
- Username from the item key: `new_username`
- Password from session parameter (since it is omitted in the item key): `session_password`
- Database name from the item key: `postgres`

Example 3

Collecting a metric using default configuration parameters.

Configuration parameters:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Item key: `pgsql.ping[,,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the parameters:

- Default URI: `192.0.2.234:5432`
- Default username: `zabbix`
- Default password: `password`
- Database name from the item key: `postgres`

Connections

Some plugins support gathering metrics from multiple instances simultaneously. Both local and remote instances can be monitored. TCP and Unix-socket connections are supported.

It is recommended to configure plugins to keep connections to instances in an open state. The benefits are reduced network congestion, latency, and CPU and memory usage due to the lower number of connections. The client library takes care of this.

> **Note:**
> Time period for which unused connections should remain open can be determined by *Plugins.<PluginName>.KeepAlive* parameter. Example: *Plugins.Memcached.KeepAlive*

1 Ceph plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ceph plugin.

It is recommended to specify them in their own configuration file (e.g. `ceph.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with ”#” are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.ApiKey				Default API key for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Mode		native / restful	restful (deprecated)	Default mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Default.User				Default username for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Uri			https://localhost:8003	Default URI for connecting to Ceph; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.InsecureSkipVerify		false / true	false	Determines whether an http client should verify the server’s certificate chain and host name. If *true*, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks (should be used only for testing).
Plugins.Ceph.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Ceph.Sessions.<SessionName>.ApiKey				Named session API key. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Mode		native / restful	restful (deprecated)	Named session mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Uri				Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

2 Docker plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Docker plugin.

It is recommended to specify them in their own configuration file (e.g. `docker.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Docker daemon unix-socket location. Must contain a scheme (only `unix://` is supported).
Plugins.Docker.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

3 Ember+ plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ember+ plugin.

It is recommended to specify them in their own configuration file (e.g. `emberplus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The Ember+ plugin is a loadable plugin and is available and fully described in the Ember+ plugin repository.

This plugin is currently only available to be built from the source (for both Unix and Windows).

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.EmberPlus.Default.Uri			tcp://localhost:9998	The default URI to connect. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.EmberPlus.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9998	The URI to connect, for the named session. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.EmberPlus.System.Path				Path to the Ember+ plugin executable. Example usage: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-`
Plugins.EmberPlus.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

4 Memcached plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Memcached plugin.

It is recommended to specify them in their own configuration file (e.g. `memcached.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Memcached.Default.Password				Default password for connecting to Memcached; used if no value is specified in an item key or named session.

Plugins.Memcached.Default.Uri			tcp://localhost:11211	Default URI for connecting to Memcached; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Default.User				Default username for connecting to Memcached; used if no value is specified in an item key or named session.
Plugins.Memcached.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Memcached.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description

Plugins.Memcached.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Memcached.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

5 Modbus plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Modbus plugin.

It is recommended to specify them in their own configuration file (e.g. `modbus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint				Endpoint is a connection string consisting of a protocol scheme, a host address and a port or serial port name and attributes. **<SessionName>** - define name of a session for using in item keys.
Plugins.Modbus.Sessions.<SessionName>.SlaveID				Slave ID of a named session. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.SlaveID=20` *Note* that this named session parameter is checked only if the value provided in the item key slave ID parameter is empty.
Plugins.Modbus.Sessions.<SessionName>.Timeout				Timeout of a named session in seconds. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.Timeout=2`

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

6 MongoDB plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MongoDB plugin.

It is recommended to specify them in their own configuration file (e.g., `mongo.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MongoDB plugin is a loadable plugin and is available and fully described in the MongoDB plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

When connecting to MongoDB, the plugin checks its configuration in a specific order, which determines which value to use. The order, from highest to lowest priority, is as follows:

1. Parameters specified in the connection URI (e.g., query parameters defined after ? in `Plugins.MongoDB.Sessions.<session_name` have the highest priority).
2. Parameters in the Zabbix agent 2 item key (e.g., mongodb.collection.stats).
3. Parameters defined in a named session (e.g., `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions`
4. Default values (e.g., `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

For example:

- If you specify the database for authenticating a connection to the MongoDB server in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=a` then the plugin will consider the `*.<SessionName>.AuthSource` parameter to be set to `admin`, even if the `*.<SessionName>.AuthSource` parameter is set to a different value.
- If you specify the URI in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongo` but provide the username and the password in the Zabbix agent 2 item key, the plugin will use the username and password from the item key, even if `Plugins.MongoDB.Sessions.<SessionName>.User` and `Plugins.MongoDB.Sessions.<SessionNam` are configured.

Attention:
If the `mongodb+srv://` scheme is used, the URI may retrieve parameters from a DNS TXT record. Parameters obtained from this TXT record override all other configurations, including URI parameters, Zabbix Agent 2 item key parameters, and session parameters.
 Only one TXT record may exist for the hostname used in the connection string.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.System.Path				Path to the MongoDB plugin executable. Example usage: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).
Plugins.MongoDB.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.Uri			tcp://127.0.0.1:27017	Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Must match the URI format. Supported schemes: mongodb:// and mongodb+srv://; the tcp:// scheme is deprecated and kept for backward compatibility with existing configurations. A port can be omitted (default=27017). Examples: mongodb://127.0.0.1:27017, mongodb+srv://example.com, localhost.
Plugins.MongoDB.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.Direct			false	Connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. **<SessionName>** - define name of a session for using in item keys. Supported values: true - the plugin will connect only to the specified server, ignoring other members of the replica set; false - the plugin will automatically discover all nodes in the replica set. If the tcp:// (deprecated) scheme is used, the parameter is set to true by default.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism				Authentication mechanism for connecting to MongoDB server. **<SessionName>** - define name of a session for using in item keys. Supported values: SCRAM-SHA-1, SCRAM-SHA-256, and MONGODB-X509.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource			admin	Database for authenticating connection to MongoDB server. **<SessionName>** - define name of a session for using in item keys. If the MONGODB-X509 authentication mechanism is used, this parameter is automatically set to $external; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet				Replica set name, used for automatic discovery of MongoDB replica set nodes. **<SessionName>** - define name of a session for using in item keys. If this parameter is used with the mongodb+srv:// scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the Plugins.MongoDB.Sessions.<SessionName>.Direct parameter is set to true, the agent will fail to start.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference				Order for connecting to MongoDB replica set nodes or fallback. **<SessionName>** - define name of a session for using in item keys. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: `required` - require TLS connection; `verify\_ca` - verify certificates; `verify\_full` - verify certificates and IP address. Supported since plugin version 1.2.1.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Default.Uri			`tcp://127.0.0.1:27017`	Default URI for connecting to MongoDB. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`. Deprecated scheme: `tcp://` (used as the default for backward compatibility with existing configurations). A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Default.User				Default username for connecting to MongoDB.
Plugins.MongoDB.Default.Password				Default password for connecting to MongoDB.
Plugins.MongoDB.Default.Direct			`false`	Default connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Default.AuthMechanism				Default authentication mechanism for connecting to MongoDB server. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`. If no value is specified either here, in an item key, in a named session, or in the URI, a SCRAM-based mechanism is used (if a username is provided; otherwise, no authentication will be used).
Plugins.MongoDB.Default.AuthSource			`admin`	Default database for authenticating connection to MongoDB server. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Default.ReplicaSet				Default replica set name; used for automatic discovery of MongoDB replica set nodes. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Default.ReadPreference				Default order for connecting to MongoDB replica set nodes or fallback. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Default.TLSConnect				Default encryption type for communications between Zabbix agent 2 and monitored databases. Supported values: `required` - requires TLS connection. `verify\_ca` - verifies certificates. `verify\_full` - verifies certificates and IP address. If no value is specified either here, in an item key, in a named session, or in the URI, a TLS connection will not be established.
Plugins.MongoDB.Default.TLSCAFile				Default pathname of a file containing the top-level CA(s) certificates for peer certificate verification. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSCertFile				Default pathname of a file containing the MongoDB client certificate or certificate chain. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSKeyFile				Default pathname of a file containing the MongoDB client private key used for encrypted communications between Zabbix agent 2 and monitored databases.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

7 MQTT plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MQTT plugin.

It is recommended to specify them in their own configuration file (e.g. `mqtt.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.Password				Default password for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Default.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.
Plugins.MQTT.Default.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.
Plugins.MQTT.Default.TLSKeyFile				Full pathname of a file containing the MQTT private key for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.

Plugins.MQTT.Default.Topic				Default topic for MQTT subscription; used if no value is specified in an item key or named session.
				
				The topic may contain wildcards ("+","#")
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Default.Url			tcp://localhost:1883	Default MQTT broker connection string; used if no value is specified in an item key or named session.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Default.User				Default username for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the MQTT private key used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.

Plugins.MQTT.Sessions.<SessionName>.Topic				Named session topic for MQTT subscription.
				<SessionName> - define name of a session for using in item keys.
				
				The topic may contain wildcards ("+","#")
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Sessions.<SessionName>.Url				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 MSSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MSSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mssql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MSSQL plugin is a loadable plugin and is available and fully described in the MSSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.CustomQueriesDir			`/usr/local/sbin/zabbix/custom-queries/mssql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Specifies the file path to a directory containing user-defined .sql files with custom queries that the plugin can execute. The plugin loads all available .sql files in the configured directory at startup. This means that any changes to the custom query files will not be reflected until the plugin is restarted. The plugin is started and stopped together with Zabbix agent 2.
Plugins.MSSQL.CustomQueriesEnabled			false	If set, enables the execution of the `mssql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.MSSQL.Default.CACertPath				The default file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server. The certificate must be in PEM format.
Plugins.MSSQL.Default.Database				The default database name to connect to.
Plugins.MSSQL.Default.Encrypt				Specifies the default connection encryption type. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted.
Plugins.MSSQL.Default.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server by default.
Plugins.MSSQL.Default.Password				The password to be sent to a protected MSSQL server by default.
Plugins.MSSQL.Default.TLSMinVersion				The minimum TLS version to use by default. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`.
Plugins.MSSQL.Default.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it by default. Possible values: `true`, `false`.
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	The default URI to connect. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.MSSQL.Default.User				The default username to be sent to a protected MSSQL server.
Plugins.MSSQL.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				The file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server for the named session. The certificate must be in PEM format. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Database				The database name to connect to for the named session. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				Specifies the connection encryption type for the named session. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Password				The password to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				The minimum TLS version to use for the named session. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it for the named session. Possible values: `true`, `false`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Uri			sqlserver://localhost:1433	The URI to connect, for the named session. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.User				The username to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.System.Path				Path to the MSSQL plugin executable. Global setting for the MSSQL plugin. Applied to all connections. Example usage: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
Plugins.MSSQL.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

9 MySQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MySQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mysql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;

- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.CallTimeout	no	1-30	global timeout	The maximum amount of time in seconds to wait for a request to be done.
Plugins.Mysql.CustomQueriesPath	no		`/usr/local/share/zabbix/custom-queries/mysql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing `.sql` files with custom queries.
Plugins.Mysql.CustomQueriesEnabled	no		false	If set, enables the execution of the `mysql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Mysql.Default.Password	no			Default password for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.Default.TLSCAFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSCertFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Default.TLSKeyFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the database private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Parameter	Mandatory	Range	Default	Description

Plugins.Mysql.Default.Uri	no		tcp://localhost:3306	Default URI for connecting to MySQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User	no			Default username for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.KeepAlive	no	60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Mysql.Sessions.<SessionName>.Password	no			Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	no (yes, if Plugins.Mysql.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.

Plugins.Mysql.Sessions.<SessionName>.Uri	no			Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User	no			Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Timeout	no	1-30	global timeout	The maximum time in seconds for waiting when a connection has to be established.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

10 NVIDIA GPU plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the NVIDIA GPU plugin.

It is recommended to specify them in their own configuration file (e.g. `nvidiagpu.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The NVIDIA GPU plugin is a loadable plugin and is available and fully described in the NVIDIA GPU plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.NVIDIA.System.Path				Path to the NVIDIA GPU plugin executable. Example usage: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout		1-30	global timeout	Specifies the maximum time (in seconds) to wait for a server response during connection attempts and subsequent operations in the session. The global item-type timeout or individual item timeout will override this value if greater.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

11 Oracle plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Oracle plugin.

It is recommended to specify them in their own configuration file (e.g. `oracle.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CallTimeout		1-30	global timeout	The maximum wait time in seconds for a request to be completed.
Plugins.Oracle.ConnectTimeout		1-30	global timeout	The maximum wait time in seconds for a connection to be established.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CustomQueriesPath			/usr/local/share/zabbix/custom_queries/oracle for Unix systems *:\Program Files\Zabbix Agent 2\Custom Queries\Oracle for Windows systems, where * is the drive name from the ProgramFiles environment variable	Full path to a directory containing *.sql files with custom queries. Example: /etc/zabbix/oracle/sql
Plugins.Oracle.CustomQueriesEnabled			false	If set, enables the execution of the oracle.custom.query item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Oracle.Default.Password				Default password for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Service				Default service name for connecting to Oracle (SID is not supported); used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Uri				Default URI for connecting to Oracle; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only tcp and tcps schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS key) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR (TNS value)
Plugins.Oracle.Default.User				Default username for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Oracle.ResolveTNS		true / false	true	The option specifies how to interpret the connection string (ConnString) for connecting to the Oracle server. If it is set to false the default scheme and port will be added to ConnString (if absent). If the option is set to true, the default scheme and port will be omitted (unless explicitly specified in ConnString), and ConnString will be passed to the Oracle client as is. If the Oracle client finds ConnString in the tnsnames.ora file, the connection description found will be used to connect to the Oracle server.
Plugins.Oracle.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.Sessions.<SessionName>.Service	no			Named session service name to be used for connection (SID is not supported). **<SessionName>** - define name of a session for using in item keys.
Plugins.Oracle.Sessions.<SessionName>.Uri	no			Named session connection string for Oracle. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `tcp` and `tcps` schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: `tcp://127.0.0.1:1521` `localhost` `zbx_tns_example` (TNS key) `(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR` (TNS value)
Plugins.Oracle.Sessions.<SessionName>.User	no			Named session username. **<SessionName>** - define name of a session for using in item keys.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

12 PostgreSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the PostgreSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `postgresql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The PostgreSQL plugin is a loadable plugin and is available and fully described in the PostgreSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.CacheMode			prepare	Cache mode for the PostgreSQL connection. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that "describe" is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.CallTimeout		1-30	global timeout	Maximum wait time (in seconds) for a request to be completed.
Plugins.PostgreSQL.CustomQueriesPath			`/usr/local/share/zabbix/custom_queries/postgresql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing files with custom queries.
Plugins.PostgreSQL.CustomQueriesEnabled			false	If set, enables the execution of the `postgresql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.PostgreSQL.Default.Database				Default database for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.Password				Default password for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.TLSCAFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSCertFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.TLSKeyFile	No (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Plugins.PostgreSQL.Default.Uri	No			Default URI for connecting to PostgreSQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	No			Default username for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.KeepAlive	No	60-900	300	Maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.PostgreSQL.Sessions.<SessionName>.CachePreparedMode	No		prepare	Cache mode for the PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that ”describe” is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	No			Database for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	No	Must match the password format.		Password for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	No (yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate peer certificate verification. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the PostgreSQL certificate or certificate chain. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	No			Encryption type for PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the PostgreSQL private key. **<SessionName>** - define name of a session for using in item keys.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.System.Path	Yes			Path to the PostgreSQL plugin executable. Example usage: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

13 Redis plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Redis plugin.

It is recommended to specify them in their own configuration file (e.g. `redis.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.Password				Default password for connecting to Redis; used if no value is specified in an item key or named session.
Plugins.Redis.Default.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Default.TLSCAFile				Default full pathname of a file containing the top-level CA(s) certificates.
Plugins.Redis.Default.TLSCertFile				Default full pathname of a file containing the agent's certificate.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.TLSKeyFile				Default full pathname of a file containing the agent's private key.

Plugins.Redis.Default.Uri			tcp://localhost:6379	Default URI for connecting to Redis; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User			default	Default user to send to the protected Redis server; used if no value is specified in an item key or named session.
Plugins.Redis.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Redis.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. **<SessionName>** - define the session name to be used in item keys. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	(yes, if TLSConnect is `verify_ca` or `verify_full`)			Full pathname of a file containing the top-level CA(s) certificates for Redis server certificate verification. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent's certificate for client authentication. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the agent's private key for client authentication. **<SessionName>** - define the session name to be used in item keys.

Plugins.Redis.Sessions.<SessionName>.Uri			localhost:6379	Connection string of a named session.
				<SessionName> - define the session name to be used in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User			default	User to send to the protected Redis server. **<SessionName>** - define the session name to be used in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

14 SMART plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the SMART plugin.

It is recommended to specify them in their own configuration file (e.g. `smart.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- The path to the `smartctl` executable must be set either by adding its directory to the system's `PATH` environment variable or by configuring `Plugins.Smart.Path`; this applies to both Linux and Windows;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Smart.Path	no		smartctl	Path to the smartctl executable.
Plugins.Smart.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 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 Shutdown-Skripte binden 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	nein		0.0.0.0	IP-Adresse, auf der gelauscht wird.
LISTEN_PORT	zabbix.listenPort	nein	1024-32767	10052	Port, auf dem gelauscht wird.
PID_FILE	zabbix.pidFile	nein		/tmp/zabbix_java.pid	Name der PID-Datei. Wenn nicht angegeben, wird Zabbix Java Gateway als Konsolenanwendung gestartet.

Variable	Property	Mandatory	Range	Default	Description
PROPERTIES_FILE	zabbix.propertiesFile	nein			Name der Eigenschaftendatei. Kann verwendet werden, um zusätzliche Eigenschaften im Schlüssel-Wert-Format festzulegen, sodass sie in der Befehlszeile nicht sichtbar sind, oder um vorhandene zu überschreiben. Zum Beispiel: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	nein	1-1000	5	Anzahl der zu startenden Worker-Threads.
TIMEOUT	zabbix.timeout	nein	1-30	3	Wie lange auf Netzwerkoperationen gewartet wird (in Sekunden).

Warning:
Port 10052 ist nicht bei IANA registriert.

9 Zabbix-Webservice

Übersicht

Der Zabbix-Webservice ist ein Prozess, der für die Kommunikation mit externen Webservices verwendet wird.

Die von der Konfigurationsdatei des Zabbix-Webservice (zabbix_web_service.conf) unterstützten Parameter sind in diesem Abschnitt aufgeführt.

Die Parameter sind ohne zusätzliche Informationen aufgelistet. Klicken Sie auf den Parameter, um die vollständigen Details anzuzeigen.

Parameter	Beschreibung
AllowedIP	Eine Liste von durch Kommas getrennten IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern und Zabbix-Proxys.
DebugLevel	Die Debug-Stufe.
IgnoreURLCertErrors	Gibt die Behandlung von TLS-Zertifikatsvalidierungsfehlern beim Zugriff auf die Frontend-URL an.
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.
TLSCAFile	Der vollständige Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Der vollständige Pfadname einer Datei, die das Dienstzertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Schlüssel des Dienstes enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Beachten Sie:

- Die Standardwerte entsprechen den Prozessstandardwerten, 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

AllowedIP

Eine durch Kommas getrennte Liste von 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
```

Verbindlich: ja

DebugLevel

Geben Sie die Debug-Stufe an:
0 - grundlegende Informationen über das Starten und Stoppen 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

IgnoreURLCertErrors

Gibt die Behandlung von TLS-Zertifikatvalidierungsfehlern beim Zugriff auf die Frontend-URL an:
0 - Zertifikatfehler nicht ignorieren;
1 - Zertifikatfehler ignorieren.

Standard: 0
 Bereich: 0-1

Include

Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden. Während der Installation erstellt Zabbix das Include-Verzeichnis in /usr/local/etc, sofern dies nicht während der Kompilierung geändert wurde. Der Pfad kann relativ zum Speicherort der Datei *zabbix_web_service.conf* sein.
Um nur relevante Dateien im angegebenen Verzeichnis einzubinden, wird das Asterisk-Platzhalterzeichen für den Musterabgleich unterstützt.
Siehe besondere Hinweise zu Einschränkungen.

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
```

Verbindlich: Ja, wenn LogType auf *file* gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe einer Protokolldatei 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

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`

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 CA(s) der obersten Ebene für die Verifizierung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname der Datei, die das Dienstzertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Dienstes enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

10 Umgebungsvariablen

Übersicht

Umgebungsvariablen ermöglichen die Konfiguration von Zabbix-Komponenten, ohne Werte fest in Konfigurationsdateien zu hinterlegen. Dadurch lassen sich Konfigurationen in dynamischen Umgebungen wie Docker einfach verwalten, in denen Variablen zur Laufzeit übergeben werden können, um unterschiedliche Setups anzupassen.

Im einfachsten Fall können Sie den Wert des Konfigurationsparameters DebugLevel des Zabbix-Servers auf eine Umgebungsvariable setzen und diese dann verwenden, um den Server beim Start zu konfigurieren:

```
#### Zabbix-Server-Konfigurationsdatei:
DebugLevel=${NEW_DEBUG_LEVEL}

#### Starten des Zabbix-Servers:
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
- Web-Service
- Zabbix-Sender (bei Verwendung der -c, --config option)

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. zum Erhöhen des Agent-Protokollierungsgrads) 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 Neuladen von Umgebungsvariablen nicht. Während des Neuladens werden Variablen ignoriert, und nur Parameter mit regulären Werten werden neu geladen.
- Die aktuellen Umgebungsvariablen des Prozesses, die in Konfigurationsdateien verwendet wurden, werden gelöscht, nachdem die Zabbix-Komponente gestartet wurde. Dadurch wird sichergestellt, dass untergeordnete Prozesse (z. B. von Zabbix ausgeführte Remote-Skripte) nicht auf diese Variablen zugreifen können. Beachten Sie jedoch, dass die anfänglichen Variablen des Prozesses 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 wird bei Namen von Umgebungsvariablen die Groß-/Kleinschreibung nicht beachtet.

Beispiele

Die folgenden Beispiele zeigen, wie Sie Umgebungsvariablen mit Zabbix-Komponenten konfigurieren und verwenden können.

Beispiel 1: Konfigurieren und Testen des Zabbix-Agenten

1. Setzen Sie Umgebungsvariablen in der Konfigurationsdatei des Agenten:

```
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 Agenten 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 runtime commands (z.B. um die Protokollierungsstufe des Agenten zu erhöhen), 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
```

Dies liegt daran, dass der Agent seine Konfigurationsdatei verwendet, um Laufzeitbefehle auszuführen; wenn Umgebungsvariablen weggelassen werden, werden die Standardwerte der Konfigurationsparameter verwendet.

Alternativ können Sie, nachdem Sie Umgebungsvariablen in der Konfigurationsdatei des Agenten gesetzt haben, diese den Prozessen zur Verfügung stellen (z. B. mit dem Befehl `export`). Dadurch wird das Risiko eines unerwarteten Verhaltens aufgrund fehlender oder falsch gesetzter 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-Agenten für einen Container

Wenn Sie Ihr eigenes benutzerdefiniertes Image für Zabbix-Komponenten (z.B. Zabbix-Agent) erstellen und konfigurieren, können Sie Konfigurationsparameter mit Hilfe von Umgebungsvariablen definieren und dann den Container mit diesen Variablen starten.

1. Setzen Sie bei der Erstellung des Images Umgebungsvariablen in der Konfigurationsdatei des Agenten:

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
```

```
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. Nach der Erstellung des Container-Images starten Sie den Agenten-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 runtime commands verwenden (z. B. um die Protokollierungsstufe des Agenten zu erhöhen), rufen Sie die Container-Shell auf und führen Sie den Laufzeitbefehl:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
Der Laufzeitbefehl `userparameter_reload` unterstützt nicht das Nachladen von Umgebungsvariablen. Während des Neuladens werden Variablen ignoriert, und nur Parameter mit regulären Werten werden neu geladen.

11 Einbindung

Übersicht

Zusätzliche Dateien oder Verzeichnisse können mit dem Parameter `Include` in die Konfiguration von Server, Proxy, Agent und Web-Service eingebunden werden.

Hinweise zur Einbindung

Wenn der Parameter `Include` zum Einbinden einer Datei verwendet wird, muss die Datei lesbar sein.

Wenn der Parameter `Include` zum Einbinden eines Verzeichnisses verwendet wird:

- Alle Dateien im Verzeichnis müssen lesbar sein.
- Es sollte keine bestimmte Reihenfolge der Einbindung angenommen werden (d. h. Dateien werden nicht in alphabetischer Reihenfolge eingebunden). Definieren Sie daher keinen Parameter in mehreren "Include"-Dateien (z. B. um eine allgemeine Einstellung mit einer spezifischen zu überschreiben).
- Alle Dateien im Verzeichnis werden in die Konfiguration eingebunden.
- Achten Sie auf Sicherungskopien von Dateien, die von einigen Texteditoren automatisch erstellt werden. Wenn zum Beispiel beim Bearbeiten der Datei "include/my_specific.conf" eine Sicherungskopie "include/my_specific_conf.BAK" erstellt wird, werden beide Dateien eingebunden. Verschieben Sie "include/my_specific.conf.BAK" aus dem Verzeichnis "Include". Unter Linux kann der Inhalt des Verzeichnisses "Include" mit dem Befehl "ls -al" auf unnötige Dateien überprüft werden.

Wenn der Parameter `Include` zum Einbinden von Dateien mithilfe eines Musters verwendet wird:

- Alle Dateien, die dem Muster entsprechen, müssen lesbar sein.
- Es sollte keine bestimmte Reihenfolge der Einbindung angenommen werden (d. h. Dateien werden nicht in alphabetischer Reihenfolge eingebunden). Definieren Sie daher keinen Parameter in mehreren "Include"-Dateien (z. B. um eine allgemeine Einstellung mit einer spezifischen zu überschreiben).

3 Protokolle

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte in diesem Abschnitt zuzugreifen.

1 Server-Proxy-Datenaustauschprotokoll

Übersicht

Der Datenaustausch zwischen Server und Proxy basiert auf dem JSON-Format.

Anfrage- und Antwortnachrichten müssen mit Header und Datenlänge beginnen.

Passiver Proxy

Konfigurationsanfrage

Der Server sendet zunächst eine leere Anfrage `proxy config`. Diese Anfrage wird alle `ProxyConfigFrequency` (Schritte des Server-Konfigurationsparameters) Sekunden gesendet.

Der Proxy antwortet mit der aktuellen Proxy-Version, dem Sitzungs-Token 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*	Sitzungs-Token der Proxy-Konfiguration.
config_revision	*number*	Revisionsnummer 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 von Feldnamen.
-	*string*	Feldname.
data	*array*	Array von Zeilen.
-	*array*	Array von Spalten.
-	*string,number*	Spaltenwert, dessen Typ 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 zusammen mit den Konfigurationsdaten gesendet (optional).
del_hostids	*array*	Array entfernter Host-IDs (optional).
-	*number*	Host-Kennung.
del_macro_hostids	*array*	Array von Host-IDs, bei denen alle Makros entfernt wurden (optional).
-	*number*	Host-Kennung.
proxy→server:		
response	*string*	Information über den Erfolg der Anfrage ('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", "ipmi_password"
"data": [
```

```
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
```

```
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
```

```
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


proxy→server:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Datenanforderung

Die Anfrage `proxy data` wird verwendet, um Verfügbarkeitsdaten von Host-Schnittstellen sowie Verlaufs-, Discovery- und Autoregistrierungsdaten vom Proxy abzurufen. Diese Anfrage wird alle `ProxyDataFrequency` Sekunden (Konfigurationsparameter des Servers) gesendet.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Token der Datensitzung.
interface availability	*array*	*(optional)* Array von Datenobjekten zur 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 Verlaufsdatenobjekten.
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, eindeutig innerhalb einer Datensitzung).

name	value type	description
timestamp	*number*	*(optional)* Zeitstempel von Datenpunkten vom Typ Log.
source	*string*	*(optional)* Quellwert des Eventlog-Datenpunkts.
severity	*number*	*(optional)* Schweregradwert des Eventlog-Datenpunkts.
eventid	*number*	*(optional)* eventid-Wert des Eventlog-Datenpunkts.
state	*string*	*(optional)* Datenpunktstatus: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Letzte Loggröße von Datenpunkten vom Typ Log.
mtime	*number*	*(optional)* Änderungszeit von Datenpunkten vom Typ Log.
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-Serviceprüfung **1**, *SVC_LDAP* - LDAP-Serviceprüfung **2**, *SVC_SMTP* - SMTP-Serviceprüfung **3**, *SVC_FTP* - FTP-Serviceprüfung **4**, *SVC_HTTP* - HTTP-Serviceprüfung **5**, *SVC_POP* - POP-Serviceprüfung **6**, *SVC_NNTP* - NNTP-Serviceprüfung **7**, *SVC_IMAP* - IMAP-Serviceprüfung **8**, *SVC_TCP* - TCP-Portverfügbarkeitsprüfung **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-Serviceprüfung **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)* Datenpunktschlüssel für Discovery-Prüfungen 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 registration	*array*	*(optional)* Array von Autoregistrierungs-Datenobjekten.
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

name	value type	description
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').
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 einem benutzerdefinierten 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
      }
  ]
}
```

Aufgabenanfrage

Der Austausch `proxy tasks` verarbeitet die Übermittlung und Bestätigung von Aufgaben für Remote-Befehle 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 Anfrage `proxy tasks`. Der Proxy antwortet mit seiner aktuellen Version und allen ausstehenden Aufgabenergebnissen (einschließlich Zeitstempeln). Der Server bestätigt dann den Empfang, indem er ein `response` zurückgibt, und kann neue Aufgaben einschließen, die der Proxy ausführen soll.

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:		
response	*string*	Information über den Erfolg 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

Konfigurationsanfrage

Die Anfrage `proxy config` wird von einem aktiven Proxy gesendet, um Proxy-Konfigurationsdaten abzurufen. Diese Anfrage wird alle `ProxyConfigFrequency` Sekunden gesendet (Proxy-Konfigurationsparameter).

name	value type	description
proxy→server:		
request	*string*	'proxy config'
host	*string *	Proxy-Name.
version	*string*	Proxy-Version (<major>.<minor>.<build>).
session	*string*	Sitzungs-Token der Proxy-Konfiguration.
config_revision	*number*	Revisionsnummer der Proxy-Konfiguration.
hostmap_revision	*number*	Aktuelle Revisionsnummer der Host-zu-Proxy-Zuordnung.
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 Änderungen).
fields	*array*	Array von Feldnamen.
-	*string*	Feldname.
data	*array*	Array von Zeilen.
-	*array*	Array von Spalten.

name	value type	description
-	*string,number*	Spaltenwert, dessen Typ vom Spaltentyp im Datenbankschema abhängt.
macro.secrets	*object*	Informationen zu geheimen Makros; nicht vorhanden, wenn es keine Änderungen an Vault-Makros gibt (optional).
proxy_group	*string*	Der Name der Proxy-Gruppe, zu der der Proxy gehört.
config_revision	*number*	Revision des Konfigurations-Caches - wird zusammen mit den Konfigurationsdaten gesendet (optional).
del_hostids	*array*	Array entfernter Host-IDs (optional).
-	*number*	Host-Kennung.
del_macro_hostids	*array*	Array von Host-IDs, bei denen alle Makros entfernt wurden (optional).
-	*number*	Host-Kennung.

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, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
```

```
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
```

```
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Datenanforderung

Die Anfrage `proxy data` wird vom Proxy gesendet, um die Verfügbarkeit von Host-Schnittstellen sowie Verlaufs-, Discovery- und Autoregistrierungsdaten bereitzustellen. Diese Anfrage wird alle `DataSenderFrequency` Sekunden gesendet (Konfigurationsparameter des Proxy). Beachten Sie, dass ein aktiver Proxy den Zabbix Server weiterhin jede Sekunde auf Remote-Command-Aufgaben abfragt (mit einer leeren Anfrage `proxy data`).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Proxy-Name.
session	*string*	Datensitzungs-Token.
interface availability	*array*	*(optional)* Array von Datenobjekten zur 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 Verlaufsdatenobjekten.
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, eindeutig innerhalb einer Datensitzung).
timestamp	*number*	*(optional)* Zeitstempel von Datenpunkten vom Typ Log.
source	*string*	*(optional)* Quellwert des Eventlog-Datenpunkts.
severity	*number*	*(optional)* Schweregradwert des Eventlog-Datenpunkts.
eventid	*number*	*(optional)* eventid-Wert des Eventlog-Datenpunkts.
state	*string*	*(optional)* Datenpunktstatus: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Letzte Log-Größe von Datenpunkten vom Typ Log.
mtime	*number*	*(optional)* Änderungszeit von Datenpunkten vom Typ Log.
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-Serviceprüfung **1**, *SVC_LDAP* - LDAP-Serviceprüfung **2**, *SVC_SMTP* - SMTP-Serviceprüfung **3**, *SVC_FTP* - FTP-Serviceprüfung **4**, *SVC_HTTP* - HTTP-Serviceprüfung **5**, *SVC_POP* - POP-Serviceprüfung **6**, *SVC_NNTP* - NNTP-Serviceprüfung **7**, *SVC_IMAP* - IMAP-Serviceprüfung **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-Serviceprüfung **15**, *SVC_TELNET* - Prüfung der Telnet-Verfügbarkeit
ip	*string*	IP-Adresse des Hosts.

name	value type	description
dns	*string*	DNS-Name des Hosts.
port	*number*	*(optional)* Portnummer des Dienstes.
key_	*string*	*(optional)* Datenpunktschlüssel für Discovery-Prüfungen 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
autoregistration	*array*	*(optional)* Array von Autoregistrierungs-Datenobjekten.
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 des Remote-Commands
status	*number*	Ausführungsstatus des Remote-Commands: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - Remote-Command erfolgreich abgeschlossen **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - Remote-Command 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 (Verlauf, Autoregistrierung, Host-Verfügbarkeit, Netzwerk-Discovery). Mögliche Werte: **enabled** - normaler Betrieb **disabled** - der Server akzeptiert keine Daten (möglicherweise aufgrund eines internen Cache-Limits)
tasks	*array*	*(optional)* Array von Aufgaben.
type	*number*	Aufgabentyp: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - Remote-Command
clock	*number*	Zeitpunkt der Aufgabenerstellung.
ttl	*number*	Zeit in Sekunden, nach der die Aufgabe abläuft.
commandtype	*number*	Typ des Remote-Commands: **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 einem benutzerdefinierten Skript)
command	*string*	Auszuführender Remote-Command.

name	value type	description
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-Commands.
authtype	*number*	*(optional)* Authentifizierungstyp für SSH-Commands.
username	*string*	*(optional)* Benutzername für Telnet- und SSH-Commands.
password	*string*	*(optional)* Passwort für Telnet- und SSH-Commands.
publickey	*string*	*(optional)* Öffentlicher Schlüssel für SSH-Commands.
privatekey	*string*	*(optional)* Privater Schlüssel für SSH-Commands.
parent_taskid	*number*	ID der übergeordneten Aufgabe.
hostid	*number*	ID des Ziel-Hosts.

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
}]
}
```


2 Zabbix Agent/Agent2-Protokoll

Weitere Informationen zu den Protokollen von Zabbix Agent und Zabbix Agent 2 finden Sie auf der Seite Passive and active agent checks.

4 Zabbix agent 2 plugin protocol

Zabbix agent 2 protocol is based on code, size and data model.

Code

Typ	Größe	Kommentare
Byte	4	Payload-Typ, derzeit wird nur JSON unterstützt.

Größe

Type	Größe	Kommentare
Byte	4	Größe der aktuellen Nutzlast in Byte.

Payload-Daten

Typ	Größe	Kommentare
Byte	Durch das Feld *Größe* definiert	JSON-formatierte Daten.

Definition der Payload-Daten

Allgemeine Daten

Diese Parameter sind in allen Anfragen/Antworten vorhanden:

Name	Type	Comments
id	uint32	Für Anfragen – die fortlaufende Kennung, die verwendet wird, um Anfragen mit Antworten zu verknüpfen. Eindeutig innerhalb einer Anfragerichtung (d. h. vom Agent zum Plugin oder vom Plugin zum Agent). Für Antworten – ID der entsprechenden Anfrage.
type	uint32	Der Anfragetyp.

Log-Anfrage

Eine von einem Plugin gesendete Anfrage, um eine Protokollmeldung in die Agent-Protokolldatei zu schreiben.

direction	plugin → agent
response	no

Für Log-Anfragen spezifische Parameter:

Name	Type	Comments
severity	uint32	Der Schweregrad der Meldung (Log-Level).
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 zu erhalten und ein Plugin zu registrieren.

direction	Agent → Plugin
response	ja

Für Registrierungsanfragen spezifische Parameter:

Name	Type	Comments
version	string	Die Protokollversion <major>.<minor>

Beispiel:

```
{"id":1,"type":2,"version":"1.0"}
```

Registrierungsantwort

Antwort des Plugins auf die Registrierungsanfrage.

Richtung	Plugin → Agent
Antwort	n/v

Für Registrierungsantworten spezifische Parameter:

Name	Type	Kommentare
name	string	Der Plugin-Name.
metrics	array of strings (optional)	Die im Plugin verwendeten Metriken mit Beschreibungen. Gibt RegisterMetrics() zurück. Fehlt, wenn ein Fehler zurückgegeben wird.
interfaces	uint32 (optional)	Die Bitmaske der vom Plugin unterstützten Schnittstellen. Fehlt, wenn ein Fehler zurückgegeben wird.
error	string (optional)	Eine Fehlermeldung, die zurückgegeben wird, wenn ein Plugin nicht gestartet werden kann. Fehlt, 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 zur Ausführung der Funktion Start der Runner-Schnittstelle.

Richtung	Agent → Plugin
Antwort	nein

Die Anfrage hat keine spezifischen Parameter; sie enthält nur die Parameter der allgemeinen Daten.

Beispiel:

```
{"id":3,"type":4}
```

Anfrage beenden

Eine vom Agent gesendete Anfrage, um ein Plugin herunterzufahren.

direction	Agent → Plugin
response	nein

Die Anfrage hat keine spezifischen Parameter; sie enthält nur Parameter der allgemeinen Daten.

Beispiel:

```
{"id":3,"type":5}
```

Export-Anfrage

Eine Anfrage zur Ausführung der Export-Funktion der Exporter-Schnittstelle.

direction	Agent → plugin
response	nein

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. Fehlt, wenn ein Fehler zurückgegeben wird.
error	string (optional)	Fehlermeldung, wenn die Export-Funktion nicht erfolgreich ausgeführt wurde. Fehlt, wenn value zurückgegeben wird.

Beispiele:

```
{"id":5,"type":7,"value":"response"}
```

oder

```
{"id":5,"type":7,"error":"error message"}
```

Anfrage konfigurieren

Eine Anfrage zur Ausführung der Funktion *Configure* der Schnittstelle *Configurator*.

Richtung	Agent → Plugin
Antwort	n/v

Für *Configure*-Anfragen spezifische Parameter:

Name	Type	Comments
global_options	JSON object	JSON-Objekt mit globalen Agent-Konfigurationsoptionen.
private_options	JSON object (optional)	JSON-Objekt mit privaten Plugin-Konfigurationsoptionen, falls angegeben.

Beispiel:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Anfrage validieren

Eine Anfrage zur Ausführung der Funktion *Validate* der Schnittstelle *Configurator*.

direction	Agent → plugin
response	ja

Spezifische Parameter für *Validate*-Anfragen:

Name	Type	Comments
private_options	JSON object (optional)	JSON-Objekt mit privaten Plugin-Konfigurationsoptionen, falls angegeben.

Beispiel:

```
{"id":7,"type":9,"private_options":{...}}
```

Antwort validieren

Antwort der *Validate*-Funktion der *Configurator*-Schnittstelle.

Richtung	plugin → Agent
Antwort	n/a

Für *Validate*-Antworten spezifische Parameter:

Name	Type	Kommentare
error	string (optional)	Eine Fehlermeldung, die zurückgegeben wird, wenn die Validate-Funktion nicht erfolgreich ausgeführt wird. Fehlt, wenn sie erfolgreich ausgeführt wurde.

Beispiel:

```
{"id":8,"type":10}
```

oder

```
{"id":8,"type":10,"error":"error message"}
```

5 Zabbix-Sender-Protokoll

Übersicht

Zabbix Server und Zabbix Proxy verwenden ein JSON-basiertes Kommunikationsprotokoll, um Daten von Zabbix sender zu empfangen. Daten können mithilfe eines Trapper-Datenpunkts oder eines HTTP-Agent-Datenpunkts mit aktivierter Trapper-Funktion empfangen werden.

Anfrage- und Antwortnachrichten müssen mit Header und Datenlänge beginnen.

Zabbix-Sender-Anfrage

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Antwort des Zabbix-Servers

```
{
    "response": "success",
    "info": "verarbeitet: 1; fehlgeschlagen: 0; gesamt: 1; aufgewendete Sekunden: 0.060753"
}
```

Zabbix sender-Anfrage mit einem Zeitstempel

Alternativ kann Zabbix sender eine Anfrage mit einem 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
}
```

Antwort des Zabbix-Servers

```
{
    "response": "success",
    "info": "verarbeitet: 2; fehlgeschlagen: 0; gesamt: 2; aufgewendete Sekunden: 0.060904"
}
```

6 Überschrift

Übersicht

Der Header ist in allen Anfrage- und Antwortnachrichten zwischen Zabbix-Komponenten vorhanden. Er ist erforderlich, um die Nachrichtenlänge zu bestimmen, ob sie komprimiert ist oder nicht und ob es sich um ein großes Paket handelt oder nicht.

Das Zabbix-Kommunikationsprotokoll hat ein Paketgrößenlimit von 1 GB pro Verbindung. 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, beginnt der Zabbix Server automatisch, das große Paketformat (Flag 0x04) zu verwenden, 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-Konfigurationssynchronisierer Pakete verarbeiten kann, die größer als 1 GB sind.

Struktur

Der Header besteht aus vier Feldern. Alle Zahlen im Header sind im Little-Endian-Format formatiert.

Feld	Größe	Größe (großes Paket)	Beschreibung
<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	Wenn Komprimierung verwendet wird (0x02-Flag) - die Länge der unkomprimierten Daten Wenn keine Komprimierung verwendet wird - 00 00 00 00

Beispiele

Hier sind einige Codeschnipsel, die zeigen, wie man den Zabbix-Protokoll-Header zu den Daten hinzufügt, die man senden möchte, um das Paket zu erhalten, das man an Zabbix senden sollte, damit es korrekt interpretiert wird. Diese Codeschnipsel gehen davon aus, dass die Daten nicht größer als 1 GB sind, so dass das Format für große Pakete nicht verwendet wird.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

or

```
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;
```

or

```
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;
```

or

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
```

```
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

7 Newline-delimited-JSON-Exportprotokoll

Dieser Abschnitt enthält Details zum Exportprotokoll im newline-delimited-JSON-Format, das verwendet wird in:

- Datenexport in Dateien
- Streaming zu externen Systemen

Folgendes kann exportiert werden:

- Auslöser-Ereignisse
- Datenpunkt-Werte
- Trends (nur Export in Dateien)

Alle Dateien haben die Erweiterung .ndjson. Jede Zeile der Exportdatei ist ein JSON-Objekt.

Auslöser-Ereignisse

Die folgenden Informationen werden für ein Problemereignis exportiert:

Field		Type	Description
clock		number	Anzahl der Sekunden seit der Epoch bis zu dem Zeitpunkt, an dem das Problem erkannt wurde (Ganzzahlanteil).
ns		number	Anzahl der Nanosekunden, die zu `clock` addiert werden müssen, um eine genaue Problemerkennungszeit 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 - Katastrophe).
hosts		array	Liste der Hosts, die am Auslöser-Ausdruck beteiligt sind; es sollte mindestens ein Element im Array vorhanden sein.
-		object	
	host	string	Host-Name.
	name	string	Sichtbarer Host-Name.
groups		array	Liste der Hostgruppen aller Hosts, die am Auslöser-Ausdruck beteiligt sind; es sollte mindestens ein Element im Array vorhanden sein.
-		string	Name der Hostgruppe.
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 Epoch bis zu dem Zeitpunkt, an dem das Problem behoben wurde (Ganzzahlanteil).
ns	number	Anzahl der Nanosekunden, die zu `clock` addiert werden müssen, um eine genaue Problembehebungszeit zu erhalten.
value	number	0 (immer).
eventid	number	ID des Wiederherstellungsereignisses.
p_eventid	number	ID des Problemereignisses.

Beispiele

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 von Mehrfachproblemen):

```
{"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}
```

Datenpunktwerte

Die folgenden Informationen werden für einen erfassten Datenpunktwert exportiert:

Feld	Type	Beschreibung
host	object	Host-Name des Datenpunkt-Hosts.
host	string	Host-Name.
name	string	Sichtbarer 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	Sichtbarer Datenpunktname.
clock	number	Anzahl der Sekunden seit der Epoch bis zu dem Zeitpunkt, an dem der Wert erfasst wurde (Ganzzahlanteil).
ns	number	Anzahl der Nanosekunden, die zu `clock` addiert werden müssen, um den genauen Zeitpunkt der Werterfassung zu erhalten.
timestamp (*Log* only)	number	0, falls nicht verfügbar.
source (*Log* only)	string	Leere Zeichenfolge, falls nicht verfügbar.
severity (*Log* only)	number	0, falls nicht verfügbar.

Feld	Type	Beschreibung
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 - numerischer Gleitkommawert, 1 - Zeichen, 2 - Log, 3 - numerisch vorzeichenlos, 4 - Text, 5 - binär, 6 - JSON

Beispiele

Numerischer (unsigned) Wert:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Numerischer (float) Wert:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Zeichen, Textwert:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Log-Wert:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Trends

Die folgenden Informationen werden für einen berechneten Trendwert exportiert:

Feld	Typ	Beschreibung
host	object	Host-Name des Datenpunkt-Hosts.
host	string	Host-Name.
name	string	Sichtbarer 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	Sichtbarer Datenpunkt-Name.
clock	number	Anzahl der Sekunden seit der Epoch bis zu dem Zeitpunkt, an dem der Wert erfasst wurde (Ganzzahlteil).
count	number	Anzahl der für eine bestimmte Stunde erfassten Werte.
min	number	Minimaler Datenpunkt-Wert für eine bestimmte Stunde.
avg	number	Durchschnittlicher Datenpunkt-Wert für eine bestimmte Stunde.
max	number	Maximaler Datenpunkt-Wert für eine bestimmte Stunde.
type	number	Werttyp: 0 - numerischer Gleitkommawert, 3 - numerisch ohne Vorzeichen

Beispiele

Numerischer (unsigned) Wert:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Numerischer (float) Wert:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

4 Datenpunkte

Bitte verwenden Sie die Seitenleiste, um auf die Inhalte in diesem Abschnitt zuzugreifen.

1 Parameter von vm.memory.size

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 gelesen 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 Einheit, 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, die vom Kernel zum Zwischenspeichern von Datenstrukturen für den eigenen Gebrauch verwendet wird
- **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 die Festplatte ausgelagert.

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 auf 3.10 unter 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 bei der Berechnung die Low-Watermark 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 notwendigerweise dem Gesamtwert. Zum Beispiel unter FreeBSD:
* Active-, inactive-, wired- und cached-Speicher werden als verwendet betrachtet, da sie nützliche Informationen speichern.
* Gleichzeitig werden inactive-, cached- und free-Speicher als verfügbar betrachtet, da diese Speicherarten Prozessen, die mehr Speicher anfordern, sofort zugewiesen werden können.

Somit ist inactive-Speicher gleichzeitig verwendet und verfügbar. Aus diesem Grund ist der Datenpunkt *vm.memory.size[used]* nur zu Informationszwecken vorgesehen, während *vm.memory.size[available]* zur Verwendung in Auslösern vorgesehen ist.

Siehe auch

1. Zusätzliche Details zur Speicherberechnung in verschiedenen Betriebssystemen

2 Passive and active agent checks

Überblick

Dieser Abschnitt enthält Details zu passiven und aktiven Prüfungen, die vom Zabbix Agent und Zabbix Agent 2 durchgeführt werden.

Zabbix verwendet ein JSON-basiertes Kommunikationsprotokoll für die Kommunikation mit den Agents.

Die Protokolle von Zabbix Agent und Zabbix Agent 2 sind seit Zabbix 7.0 vereinheitlicht. Der Unterschied zwischen Anfragen/Antworten von Zabbix Agent und Zabbix Agent 2 wird durch den Wert des Tags „variant“ ausgedrückt.

Passive Prüfungen

Eine passive Prüfung ist eine einfache Datenanfrage. Der Zabbix Server oder Proxy fragt bestimmte Daten ab (zum Beispiel CPU-Auslastung), und der Zabbix Agent sendet das Ergebnis an den Server zurück.

Passive 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.

Der Agent-Poller versucht, eine Verbindung zu allen Adressen herzustellen, die von der DNS-Abfrage zurückgegeben werden. Dadurch wird sichergestellt, dass der Poller, falls eine IP-Adresse nicht erreichbar ist, die nächste verfügbare Adresse ausprobiert, was die Wahrscheinlichkeit einer erfolgreichen Verbindung erhöht. Diese Verbesserung gilt sowohl für den Zabbix Server als auch für den Proxy.

Die maximale Parallelität asynchroner Prüfungen beträgt 1000 (definiert durch MaxConcurrentChecksPerPoller).

Die Anzahl asynchroner Agent-Poller wird durch den Parameter StartAgentPollers festgelegt.

Server-Anfrage

Für die Definition von Header und Datenlänge siehe Protokolldetails.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Feld		Type	Pflichtfeld	Wert
request		*string*	ja	`"passive checks"`
data		*array of object*	ja	Passiver Prüfungs-Datenpunkt.
	key	*string*	ja	Datenpunktschlüssel mit erweiterten Makros.
	timeout	*number*	ja	Kommunikations-Timeout.

Agent-Antwort

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Feld	Type	Pflichtfeld	Wert
version	*string*	ja	Die Versionsnummer des Agent.
variant	*number*	ja	Die Agent-Variante (*1* - Zabbix Agent, *2* - Zabbix Agent 2).
data	*array of object*	ja	Enthält das Ergebnis der Prüfung.
value	*string*	nein	Der Datenpunktwert, wenn die Prüfung erfolgreich war.
error	*string*	nein	Die Fehlermeldung, wenn die Prüfung nicht erfolgreich war.

Zum Beispiel für unterstützte Datenpunkte:

1. Der Server öffnet eine TCP-Verbindung
2. Der Server sendet **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”agent.ping”,”timeout”:3}]}**
3. Der Agent liest die Anfrage und antwortet mit **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”value”:1}]}**
4. Der Server verarbeitet die Daten, um den Wert zu erhalten, in unserem Fall „1“
5. Die TCP-Verbindung wird geschlossen

Für nicht unterstützte Datenpunkte:

1. Der Server öffnet eine TCP-Verbindung
2. Der Server sendet **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”vfs.fs.size[/nono]”,”timeout”:3}**
3. Der Agent liest die Anfrage und antwortet mit **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”error”:”Unsup item key.”}]}**
4. Der Server verarbeitet die Daten und ändert den Status des Datenpunkts auf „nicht unterstützt“ mit der angegebenen Fehlermeldung
5. Die TCP-Verbindung wird geschlossen

Failover auf das alte Protokoll

Um sicherzustellen, dass Zabbix Server oder Proxy mit Agents aus Versionen vor 7.2 arbeiten können, die das Klartextprotokoll verwenden, wurde ein Failover auf das alte Protokoll implementiert.

Passive Prüfungen werden nach einem Neustart oder wenn die Schnittstellenkonfiguration geändert wird, mit dem JSON-Protokoll (7.0 und höher) durchgeführt. Wenn als Antwort kein gültiges JSON empfangen wird (der Agent hat „ZBX_NOTSUPPORTED“ gesendet), speichert Zabbix die Schnittstelle als altes Protokoll im Cache und **wiederholt** die Prüfung, indem nur der Datenpunktschlüssel gesendet wird.

Beachten Sie, dass Zabbix Server/Proxy stündlich erneut versucht, mit allen Schnittstellen über das neue Protokoll zu arbeiten, 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 Konfigurationsdatei des Agent angegeben. Die Häufigkeit, mit der diese Prüfungen abgefragt werden, wird durch den Parameter „RefreshActiveChecks“ in derselben Konfigurationsdatei festgelegt. Schlägt das Aktualisieren aktiver Prüfungen jedoch fehl, wird der Vorgang nach fest codierten 60 Sekunden erneut versucht.

Note:
Seit Zabbix 6.4 erhält der Agent (im aktiven Modus) nicht mehr standardmäßig alle zwei Minuten eine vollständige Kopie der Konfiguration vom Server/Proxy. Stattdessen wird zur Verringerung des Netzwerkverkehrs und des Ressourcenverbrauchs standardmäßig alle 5 Sekunden eine inkrementelle Konfigurationssynchronisierung durchgeführt, bei der der Server/Proxy **nur dann** eine vollständige Kopie der Konfiguration bereitstellt, wenn der Agent sie noch nicht erhalten hat oder sich die Host-Konfiguration, globale Makros oder globale reguläre Ausdrücke geändert haben.

Der Agent sendet dann periodisch die neuen Werte an den/die Server. Wenn der Agent Remote-Befehle 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 erwägen, nur aktive Prüfungen zu verwenden, da Sie in diesem Fall die Firewall nicht ändern müssten, um eingehende Erstverbindungen zuzulassen.

Abrufen der Liste von Datenpunkten

Agent-Anfrage

Die Anfrage für aktive Prüfungen wird verwendet, um die aktiven Prüfungen abzurufen, die vom Agent verarbeitet werden sollen. Diese Anfrage wird vom Agent beim Start und danach in den 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	Der Konfigurationsparameter HostMetadata oder der Metrikwert von HostMetadataItem.
interface	*string*	no	Der Konfigurationsparameter HostInterface oder der Metrikwert von HostInterfaceItem.
ip	*string*	no	Die erste IP des Konfigurationsparameters ListenIP, falls gesetzt.
port	*number*	no	Der Wert des Konfigurationsparameters ListenPort, falls gesetzt und nicht der Standard-Listening-Port des Agent.
version	*string*	yes	Die Versionsnummer des Agent.
variant	*number*	yes	Die Agent-Variante (*1* - Zabbix-Agent, *2* - Zabbix-Agent 2).
config_revision	*number*	no	Konfigurationskennung für die inkrementelle Konfigurationssynchronisierung.
session	*string*	no	Sitzungskennung für die inkrementelle Konfigurationssynchronisierung.

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	Fehlerinformationen im Fehlerfall.
data	*array of objects*	no	Aktive Prüfungs-Datenpunkte. Wird weggelassen, wenn die Host-Konfiguration unverändert ist.
key	*string*	no	Datenpunktschlüssel mit expandierten Makros.
itemid	*number*	no	Datenpunktkennung.
delay	*string*	no	Aktualisierungsintervall des Datenpunkts. Flexible/zeitgesteuerte Intervalle werden seit Zabbix 7.0 sowohl von Zabbix-Agent als auch von Zabbix-Agent 2 unterstützt.
lastlogsize	*number*	no	lastlogsize des Datenpunkts.
mtime	*number*	no	mtime des Datenpunkts.
timeout	*string*	no	Timeout des Datenpunkts.
refresh_unsupported	*number*	no	Aktualisierungsintervall für nicht unterstützte Datenpunkte.
regexp	*array of objects*	no	Globale reguläre Ausdrücke.
name	*string*	no	Name des globalen regulären Ausdrucks.
expression	*string*	no	Globaler regulärer Ausdruck.
expression_type	*number*	no	Typ des globalen regulären Ausdrucks.
exp_delimiter	*string*	no	Trennzeichen des globalen regulären Ausdrucks.
case_sensitive	*number*	no	Einstellung zur Groß-/Kleinschreibung des globalen regulären Ausdrucks.
commands	*array of objects*	no	Auszuführende Remote-Befehle. Enthalten, wenn die Ausführung von Remote-Befehlen durch eine Operation einer Aktion oder die manuelle Ausführung eines Skripts ausgelöst wurde. Beachten Sie, dass die Ausführung von Remote-Befehlen auf einem aktiven Agent seit Zabbix-Agent 7.0 unterstützt wird. Ältere aktive Agents ignorieren alle Remote-Befehle, die in der Server-Antwort auf aktive Prüfungen enthalten sind.
command	*string*	no	Remote-Befehl.
id	*number*	no	Kennung des Remote-Befehls.
wait	*number*	no	Ausführungsmodus des Remote-Befehls („0“ (nowait) für Befehle aus Operationen von Aktionen; „1“ (wait) für Befehle aus der manuellen Ausführung von Skripten).
config_revision	*number*	no	Konfigurationskennung für die inkrementelle Konfigurationssynchronisierung. Wird weggelassen, wenn die Host-Konfiguration unverändert ist. Wird erhöht, wenn sich die Host-Konfiguration ändert.

Der Server muss mit Erfolg antworten.

Zum Beispiel:

1. Agent öffnet eine TCP-Verbindung
2. Agent fragt die Liste der Prüfungen ab
3. Server antwortet mit einer Liste von Datenpunkten und auszuführenden Remote-Befehlen
4. Agent verarbeitet die Antwort
5. Die TCP-Verbindung wird geschlossen
6. Agent startet die periodische Datenerfassung und führt Remote-Befehle aus (unterstützt seit Zabbix-Agent 7.0)

Attention:
Beachten Sie, dass (sensible) Konfigurationsdaten für Parteien verfügbar werden können, die Zugriff auf den Trapper-Port des Zabbix-Server haben, wenn eine aktive Prüfung verwendet wird. Dies ist möglich, weil sich jeder als aktiver Agent ausgeben und Konfigurationsdaten von Datenpunkten anfordern kann; eine Authentifizierung findet nicht statt, sofern Sie nicht die Optionen für Verschlüsselung verwenden.

Senden gesammelter Daten

Agent sendet

Die Datenanfrage des Agent enthält die gesammelten Datenpunktwerte sowie 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	Die Wertkennung (inkrementeller Zähler, der zur Prüfung auf doppelte Werte bei Netzwerkproblemen verwendet wird).
itemid	*number*	yes	Die Datenpunktkennung.
value	*string*	no	Der Datenpunktwert.
lastlogsize	*number*	no	Das lastlogsize des Datenpunkts.
mtime	*number*	no	Das mtime des Datenpunkts.
state	*number*	no	Der Status des Datenpunkts.
source	*string*	no	Die Quelle des Ereignisprotokolls des Werts.
eventid	*number*	no	Die eventid des Ereignisprotokolls des Werts.
severity	*number*	no	Die severity des Ereignisprotokolls des Werts.
timestamp	*number*	no	Der Zeitstempel des Ereignisprotokolls des Werts.
clock	*number*	yes	Der Zeitstempel des Werts (Sekunden seit der Epoch).
ns	*number*	yes	Die Nanosekunden des Zeitstempels des Werts.

Field	Type	Mandatory	Value
commands	*array of objects*	no	Ausführungsergebnis von Remote-Befehlen. Beachten Sie, dass die Ausführung von Remote-Befehlen auf einem aktiven Agent seit Zabbix-Agent 7.0 unterstützt wird. Ältere aktive Agenten ignorieren alle Remote-Befehle, die in der Server-Antwort für aktive Prüfungen enthalten sind.
id	*number*	no	Kennung des Remote-Befehls.
value	*string*	no	Ausführungsergebnis des Remote-Befehls, wenn die Ausführung erfolgreich war.
error	*string*	no	Fehlermeldung zur Ausführung des Remote-Befehls, wenn die Ausführung fehlgeschlagen ist.
session	*string*	yes	Eindeutige Sitzungskennung, die bei jedem Start des Agent generiert wird.
host	*string*	yes	Host-Name.
version	*string*	yes	Die Versionsnummer des Agent.
variant	*number*	yes	Die Variante des Agent (*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 Datensitzung (identifiziert durch das Sitzungs- Token) eindeutig ist. Diese ID wird verwendet, um doppelte Werte zu verwerfen, die in Umgebungen mit schlechter Konnektivität gesendet werden könnten.

Server-Antwort

Die Antwort auf Agent-Daten wird vom Server nach der Verarbeitung der Agent-Datenanfrage an den Agent 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 Datenpunktverarbeitung.

Attention:
Wenn das Senden einiger Werte auf dem Server fehlschlägt (zum Beispiel, weil der Host oder der Datenpunkt deaktiviert oder gelöscht wurde), wird der Agent nicht erneut versuchen, 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. Die 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 von Zabbix agent/ 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
}
```

Feld	Typ	Obligatorisch	Wert
request	*string*	ja	`active check heartbeat`
host	*string*	ja	Der Host-Name.
heartbeat_freq	*number*	ja	Die Heartbeat-Frequenz des Agent (Konfigurationsparameter HeartbeatFrequency).
version	*string*	ja	Die Versionsnummer des Agent.
variant	*number*	ja	Die Variante des Agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Umleitungsantwort

Wenn ein Host neu zugewiesen wurde, kann der Server den Agent anweisen, seine Heartbeat-Nachricht (und nachfolgende aktive Prüfungen) an eine andere Proxy- oder Server-Instanz umzuleiten.

```
{
  "response": "failed",
  "redirect": {
    "revision": 2,
    "address": "192.0.2.0:10055"
  }
}
```

Feld		Typ	Obligatorisch	Wert
response		*string*	ja	`success` | `failed`
redirect		*object*	ja	Umleitungsanweisungen.
	revision	*number*	ja	Kennung der Konfigurationsrevision.
	address	*string*	ja	Adresse des Ziel-Servers/Proxy.

Älteres XML-Protokoll

Note:
Zabbix akzeptiert bis zu 16 MB XML-Base64-kodierte Daten, aber ein einzelner dekodierter Wert sollte nicht länger als 64 KB sein, andernfalls wird er während der Dekodierung auf 64 KB gekürzt.

3 Minimale Berechtigungsstufe 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.

Allgemeine vom Agent unter Windows unterstützte Datenpunkte

Datenpunktschlüssel	Benutzergruppe	
	Empfohlen	Technisch mindestens 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

Datenpunktschlüssel	Benutzergruppe	
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 Datenpunktschlüssel

Datenpunktschlüssel	Benutzergruppe	
	Empfohlen	Technisch minimal 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

4 Kodierung von zurückgegebenen Werten

Der Zabbix Server erwartet jeden zurückgegebenen Textwert in der UTF8-Kodierung. Dies betrifft alle Arten von Prüfungen: Zabbix Agent, SSH, Telnet usw.

Verschiedene 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 verschiedenen Datenbank-Backends in solchen Fällen beschrieben.

MySQL

Wenn ein Wert ein Nicht-ASCII-Zeichen in einer Nicht-UTF8-Kodierung enthält, wird dieses Zeichen und alles Folgende verworfen, wenn die Datenbank diesen Wert speichert. In die *zabbix_server.log* werden keine Warnmeldungen geschrieben.
Relevant mindestens für MySQL-Version 5.1.61

PostgreSQL

Wenn ein Wert ein Nicht-ASCII-Zeichen in einer Nicht-UTF8-Kodierung 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 *zabbix_server.log* geschrieben.
Relevant mindestens für PostgreSQL-Version 9.1.3

5 Unterstützung für große Dateien

Die Unterstützung für große Dateien, oft als LFS abgekürzt, bezeichnet die Fähigkeit, auf 32-Bit-Betriebssystemen mit Dateien zu arbeiten, die größer als 2 GB sind. Die Unterstützung für große Dateien betrifft mindestens die Überwachung von Protokolldateien und alle vfs.file.* Datenpunkte. Die Unterstützung für große Dateien hängt von den Fähigkeiten eines Systems zum Zeitpunkt der Kompilierung von Zabbix ab, ist jedoch auf einem 32-Bit-Solaris aufgrund der Inkompatibilität mit procfs und swapctl vollständig deaktiviert.

6 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]`.

7 Hinweise zum memtype-Parameter in proc.mem-Datenpunkten

Übersicht

Der Parameter **memtype** wird auf den Plattformen Linux, AIX, FreeBSD und Solaris unterstützt.

Drei gängige Werte von „memtype“ werden auf all diesen Plattformen unterstützt: `pmem`, `rss` und `vsize`. Zusätzlich werden auf einigen Plattformen plattformspezifische „memtype“-Werte unterstützt.

AIX

In der Tabelle finden Sie die unter AIX für den Parameter „memtype“ 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 Resident Set	pi_trss + pi_drss	ps -o rssize
size	Größe des Prozesses (Code + Daten)	pi_dvm	Spalte SIZE von „ps gvw“
dsize	Datengröße	pi_dsize	

Unterstützter Wert	Beschreibung	Quelle in der Struktur procentry64	Versucht kompatibel zu sein mit
tsize	Textgröße (Code)	pi_tsize	Spalte TSIZ von „ps gvw“
sdsize	Datengröße aus der Shared Library	pi_sdsize	
drss	Größe des Data Resident Set	pi_drss	
trss	Größe des Text Resident Set	pi_trss	

Hinweise für AIX:

1. Wenn Sie Parameter für den Datenpunkt-Schlüssel proc.mem[] unter AIX auswählen, versuchen Sie, möglichst enge 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 nur des Parameters für die Befehlszeile (regulärer Ausdruck für die Übereinstimmung) dazu führt, dass der Zabbix Agent sich selbst mit einbezieht – wahrscheinlich nicht das, was Sie möchten.

2. Verwenden Sie nicht `ps -ef`, um Prozesse zu durchsuchen – es zeigt nur Nicht-Kernel-Prozesse an. Verwenden Sie `ps -Af`, um alle Prozesse anzuzeigen, die vom Zabbix Agent gesehen werden.
3. Gehen wir das Beispiel von `topasrec` durch, um zu sehen, wie 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[]` hat folgende Argumente:

```
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]
```

Das 1. Kriterium ist ein Prozessname (Argument <name>). In unserem Beispiel wird der Zabbix Agent ihn als „topasrec“ sehen. Für eine Übereinstimmung müssen Sie entweder „topasrec“ angeben oder das Feld leer lassen.
Das 2. Kriterium ist ein Benutzername (Argument <user>). Für eine Übereinstimmung müssen Sie entweder „root“ angeben oder das Feld leer lassen.
Das 3. Kriterium, das bei der Prozessauswahl verwendet wird, ist ein Argument <cmdline>. Der Zabbix Agent wird seinen 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“ sehen. Für eine Übereinstimmung müssen Sie entweder einen regulären Ausdruck angeben, der mit dieser Zeichenkette übereinstimmt, oder das Feld leer lassen.

Die Argumente <mode> und <memtype> werden nach Anwendung der drei oben genannten Kriterien verwendet.

FreeBSD

In der Tabelle finden Sie die unter FreeBSD für den Parameter „memtype“ 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 physischen Speichers	aus rss berechnet	ps -o pmem
rss	Größe des Resident Set	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 Textsegments (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	Größe des Stacks	kp_eproc.e_vm.vm_ssize oder ki_ssize	ps -o ssiz

Linux

In der Tabelle sind die unter Linux für den Parameter „memtype“ unterstützten Werte aufgeführt.

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 der Resident Set Size	VmRSS
data	Größe des Datensegments	VmData
exe	Größe des Codesegments	VmExe
hwm	Spitzenwert der Resident Set Size	VmHWM
lck	Größe des gesperrten Speichers	VmLck
lib	Größe der gemeinsam genutzten Bibliotheken	VmLib
peak	Spitzenwert der Größe des virtuellen Speichers	VmPeak
pin	Größe der fixierten Seiten	VmPin
pte	Größe der Seitentabelleneinträge	VmPTE
size	Größe des Prozesscodes + Daten- + Stack-Segmente	VmExe + VmData + VmStk
stk	Größe des Stack-Segments	VmStk
swap	Größe des verwendeten Swap-Speichers	VmSwap

Hinweise für Linux:

1. Nicht alle „memtype“-Werte werden von älteren Linux-Kernels unterstützt. Zum Beispiel unterstützen Linux-2.4-Kernels die Werte `hwm`, `pin`, `peak`, `pte` und `swap` nicht.
2. Wir haben festgestellt, dass die Selbstüberwachung des aktiven Prüfprozess des Zabbix Agent mit `proc.mem[...,...,...,...,data` einen Wert anzeigt, der 4 kB größer ist als der in der Zeile `VmData` in der Datei /proc/<pid>/status des Agent gemeldete Wert. Zum Zeitpunkt der Selbstmessung erhöht sich das Datensegment des Agent um 4 kB und kehrt dann auf die vorherige Größe zurück.

Solaris

In der Tabelle finden Sie die unter Solaris für den Parameter „memtype“ 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 der Resident Set Size Sie kann unterschätzt werden – siehe die Beschreibung von rss in „man ps“.	pr_rssize	ps -o rss

Fußnoten

[1] Standardwert.

8 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><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.

9 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 im service-Parameter angegebene Dienste detailliert beschrieben.

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.

10 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 läuft	pmem: Prozentsatz des realen Speichers
user: Benutzer (real), unter dem der Prozess läuft	group: Gruppe (real), unter der der Prozess läuft	rss: Resident Set Size
group: Gruppe (real), unter der der Prozess läuft	uid: Benutzer-ID	data: Größe des Datensegments
uid: Benutzer-ID	gid: ID der Gruppe, unter der der Prozess läuft	exe: Größe des Codesegments
gid: ID der Gruppe, unter der der Prozess läuft	tid: Thread-ID	lib: Größe der gemeinsam genutzten Bibliotheken
vsize: Größe des virtuellen Speichers	tname: Thread-Name	lck: Größe des gesperrten Speichers
pmem: Prozentsatz des realen Speichers	cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (Wert wird in Clock-Ticks angegeben — durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)	pin: Größe der fixierten Seiten
rss: Resident Set Size	cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (Wert wird in Clock-Ticks angegeben — durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)	pte: Größe der Seitentabelleneinträge
data: Größe des Datensegments	state: Thread-Status	size: Größe von Prozess-Code + Daten- + Stack-Segmenten
exe: Größe des Codesegments	ctx_switches: Anzahl der Kontextwechsel	stk: Größe des Stack-Segments

mode=process	mode=thread	mode=summary
hwm: maximale Resident Set Size	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 wird in Clock-Ticks angegeben — 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 wird in Clock-Ticks angegeben — 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 fixierten Seiten		threads: Anzahl der Threads
pte: Größe der Seitentabelleneinträge		page_faults: Anzahl der Seitenfehler
size: Größe von Prozess-Code + Daten- + Stack-Segmenten		pss: Proportional Set Size Memory
stk: Größe des Stack-Segments		
swap: Größe des verwendeten Swap-Speichers		
cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (Wert wird in Clock-Ticks angegeben — durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)		
cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (Wert wird in Clock-Ticks angegeben — 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: Proportional Set Size Memory		

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

mode=process	mode=thread	mode=summary
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)
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

mode=process	mode=thread	mode=summary
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
io_write_op: IO-Schreiboperationen		
io_other_b: übertragene IO-Bytes, außer Lese- und Schreiboperationen		
io_other_op: IO-Operationen, außer Lese- und Schreiboperationen		

11 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 erreichbar wird.

Nicht erreichbare Schnittstelle

Eine Host-Schnittstelle wird nach einer fehlgeschlagenen Prüfung (Netzwerkfehler, Timeout) durch Zabbix-, SNMP-, IPMI- oder JMX-Agenten als nicht erreichbar behandelt. Seit Zabbix 6.2.0 beeinflussen auch aktive Prüfungen des Zabbix Agent die Verfügbarkeit der Schnittstelle. Wenn aktive Prüfungen nicht verfügbar werden, tragen sie zum allgemeinen Verfügbarkeitsstatus der Agent-Schnittstelle bei.

Ab dem Moment, in dem eine Schnittstelle nicht erreichbar wird, definiert **UnreachableDelay**, wie oft sie mithilfe eines der Datenpunkte (einschließlich LLD-Regeln) erneut geprüft wird.
Diese erneuten Prüfungen werden von Pollern für nicht erreichbare Hosts durchgeführt (oder von IPMI-Pollern bei IPMI-Prüfungen). Standardmäßig beträgt das Intervall zwischen aufeinanderfolgenden Erreichbarkeitsprüfungen 15 Sekunden.

> **Attention:**
> Prüfungen, die von asynchronen Pollern durchgeführt werden, werden nicht an Poller für nicht erreichbare Hosts übergeben.

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 Log-Meldungen 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 des Zeitraums der Nichterreichbarkeit. Standardmäßig beträgt UnreachablePeriod 45 Sekunden.
Dieser Wert sollte um ein Mehrfaches größer sein als UnreachableDelay, 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.

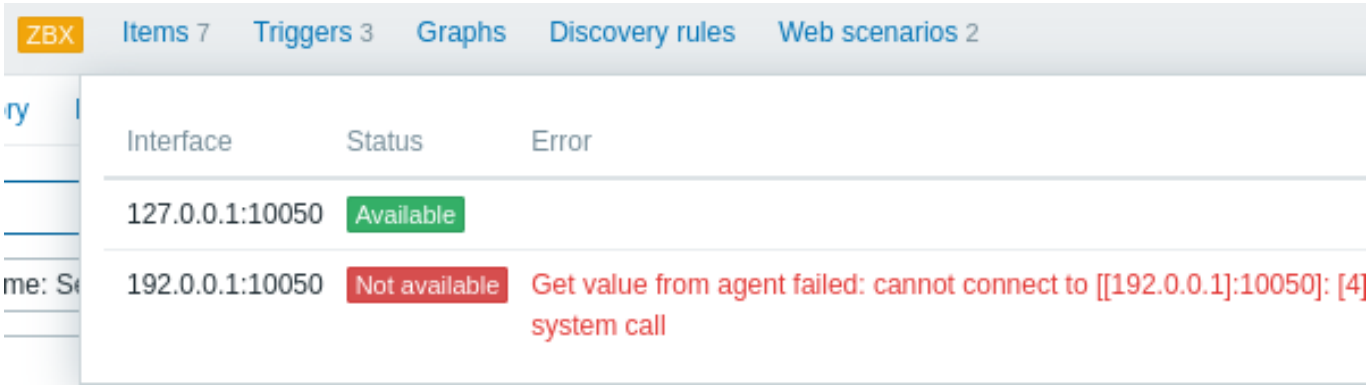
Nicht verfügbare Schnittstelle

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
```

12 Remote-Überwachung von Zabbix-Statistiken

Überblick

Es ist möglich, einige interne Metriken des Zabbix Server und Proxy für eine andere Zabbix-Instanz oder ein Drittanbieter-Tool per Fernzugriff verfügbar zu machen. Dies kann nützlich sein, damit Supporter/Dienstleister die Zabbix Server/Proxys ihrer Kunden per Fernzugriff überwachen können oder damit in Organisationen, in denen Zabbix nicht das Haupt-Monitoring-Tool ist, interne Zabbix-Metriken von einem Drittanbieter-System in einer übergreifenden Monitoring-Umgebung überwacht werden können.

Interne Zabbix-Statistiken werden für einen konfigurierbaren Satz von Adressen bereitgestellt, die im neuen Parameter 'StatsAllowedIP' des server/proxy aufgelistet sind. Anfragen werden nur von diesen Adressen akzeptiert.

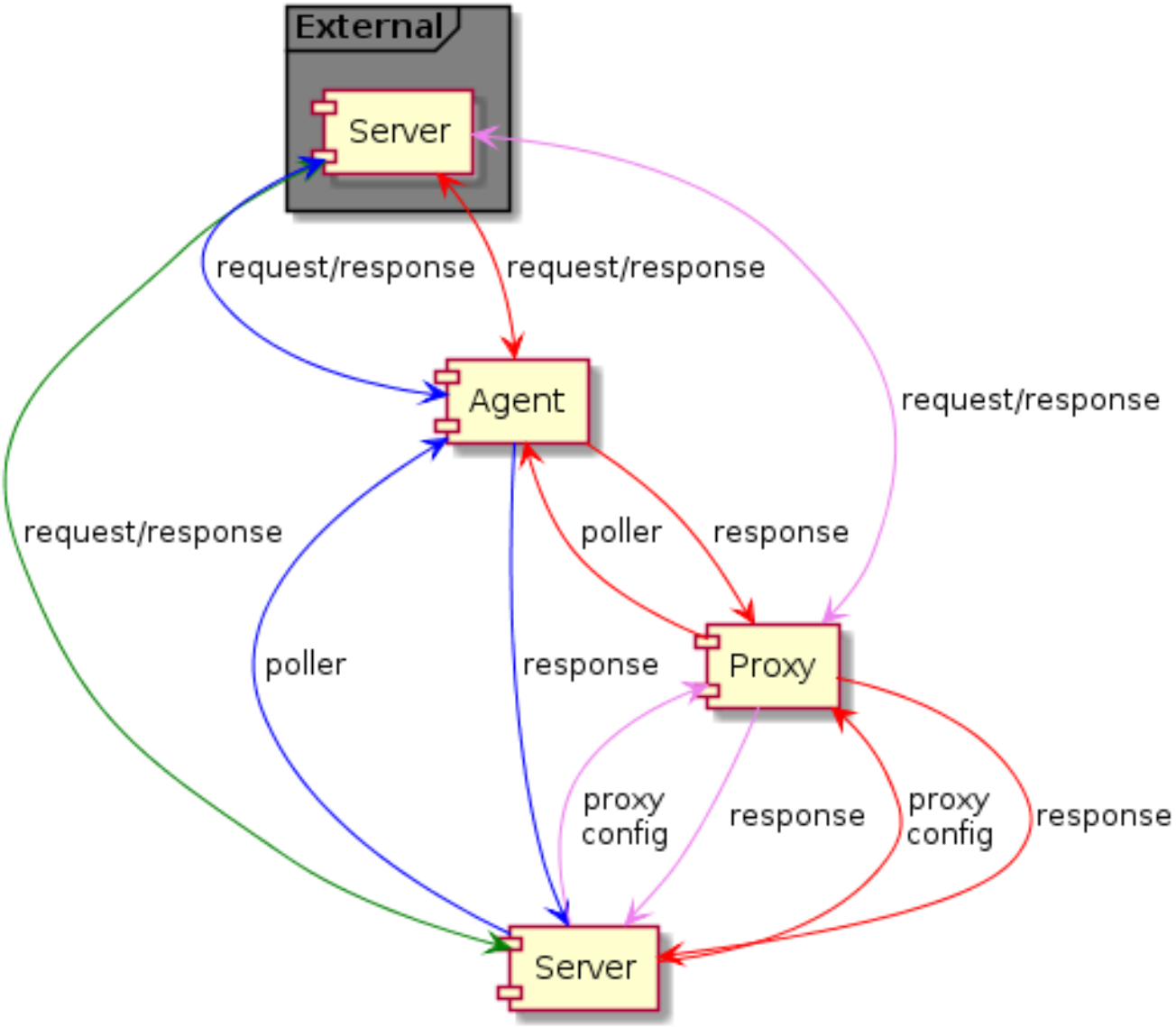
Datenpunkte

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 Daten-

punkte 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 dafür vorgesehen sind, interne Warteschlangenstatistiken einer anderen Zabbix-Instanz per Fernabfrage abzurufen:

- `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 Agent-basierte interne Warteschlangen-abfragen an einen entfernten Zabbix-Server/Proxy

Siehe auch: Interne Datenpunkte, Zabbix-Agent-Datenpunkte

13 Konfiguration 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. Installieren Sie KDC und Client-Dienstprogramme:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Beantworten Sie während der Paketeinrichtung die Eingabeaufforderungen, z. B.:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (or your FQDN)
Administrative server for your Kerberos realm: localhost (or your FQDN)
```

2. Ordnen Sie einen leicht lesbaren Hostnamen zu (optional, für lokale Tests).

Bearbeiten Sie /etc/hosts und fügen Sie einen Eintrag für Ihren DC und Webserver hinzu, falls Sie kein DNS haben:

```
sudo vi /etc/hosts
```

Beispiel für eine Zeile, die Sie hinzufügen könnten:

```
192.168.1.100  dc01.example.com dc01
```

3. Konfigurieren Sie den Kerberos-Client und den KDC-Realm:

```
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 Hostname→Realm-Zuordnung funktioniert. Abweichungen hier verursachen Fehler vom Typ `Server not found in Kerberos database`.

4. Initialisieren Sie die Kerberos-Datenbank (einmalig, auf dem KDC-Host). Legen Sie bei Aufforderung ein sicheres Master-Passwort fest:

```
sudo krb5_newrealm
```

5. Erstellen Sie den Principal `HTTP/host.fqdn@REALM` mit genau dem Hostnamen, den Clients verwenden werden; bevorzugen Sie Kleinbuchstaben (z. B. `HTTP/web.example.com@EXAMPLE.COM`). Eine Abweichung bei Groß-/Kleinschreibung oder Name verursacht `Server not found in Kerberos database`.

```
sudo kadmin.local
```

Innerhalb von `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 auf den Web-Host (oder belassen Sie sie lokal, wenn es dieselbe Maschine ist) und setzen Sie Berechtigungen, die von Apache verwendet werden können:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verify
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Installieren und aktivieren Sie das Apache-GSSAPI-Modul:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Nicht alle mod_auth_gssapi-Versionen unterstützen jede `Gssapi*`-Direktive. Wenn Apache mit `Invalid command 'GssapiCredStore'` fehlschlägt, entfernen Sie die nicht unterstützte Direktive oder aktualisieren Sie das Modul.

7. Konfigurieren Sie einen VirtualHost (passen Sie `DocumentRoot` / den Pfad zu Ihrer Zabbix-UI an):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

Innerhalb von `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>
```

Starten Sie Apache neu:

```
sudo systemctl restart apache2
```

8. Aktivieren/starten Sie die KDC-Dienste und prüfen Sie die lauschenden Ports (KDC-Host):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # or: sudo netstat -tnlp | grep :80
```

9. Beziehen Sie ein TGT zum Testen (führen Sie dies als der Benutzer aus, der das Ticket verwenden wird).

In der Ticketliste sollte `krbtgt/EXAMPLE.COM@EXAMPLE.COM` angezeigt werden. Führen Sie `kinit` als derselbe OS-Benutzer aus, der das Ticket benötigt (z. B. `zabbix` für Web-Prüfungen oder `www-data`/Apache für interaktive Browser-SSO-Tests). Tickets, die für einen anderen OS-Benutzer ausgestellt wurden, sind nicht sichtbar, sofern `KRB5CCNAME` und die Berechtigungen nicht angepasst werden.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Testen Sie den SPNEGO-Austausch mit curl (von einem Client mit gültigem TGT). Ein `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-UI HTTP-authentifizierte Anmeldungen akzeptieren soll, aktivieren Sie HTTP-Authentifizierung im Zabbix-Frontend (ui/conf/zabbix.conf.php):

```
$ALLOW_HTTP_AUTH = true;
```

Gehen Sie in der Web-UI zu *Benutzer* > *Authentifizierung* und wechseln Sie zur Registerkarte *HTTP settings*. Aktivieren Sie das Kontrollkästchen *Enable HTTP authentication* und klicken Sie im Pop-up auf *Ok*. Wählen Sie im Dropdown *Default login form* die Option "HTTP login form". Entscheiden Sie, ob *Case-sensitive login* zu Ihrer Verzeichnisrichtlinie passt. Klicken Sie zum Abschluss auf die Schaltfläche *Update*.

12. Browser-Konfiguration (Firefox wird als Beispiel verwendet): Setzen Sie `network.negotiate-auth.trusted-uris` auf den/die Host(s), die Negotiate ausführen (`dc01.example.com`), damit der Browser Kerberos-Tokens automatisch sendet.

Innerhalb von `about:config`:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Beim Aufruf von `http://dc01.example.com` sollten Sie nun direkt ohne Formular bei Zabbix angemeldet werden.

13. Halten Sie Schlüssel/Tickets aktuell. Die Standardlebensdauer eines Kerberos-Tickets beträgt ungefähr 10 Stunden. Fügen Sie einen cron-/systemd-Timer hinzu, um Abläufe zu vermeiden:

```
####for the web service
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####for the monitoring user
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Prüfungen zur Systempflege:

- `klist -k /etc/apache2/http.keytab` — prüfen Sie, ob der Service-Principal in der keytab vorhanden ist.
- `sudo tail -f /var/log/apache2/error.log` — beobachten Sie auf GSSAPI-Fehler (`gss_acquire_cred[_from]() failed to get server creds` bedeutet keytab/Berechtigungen oder fehlender Principal).
- Wenn `curl --negotiate` 401/403 zurückgibt, bedeutet das oft einen falschen Principal, kein Ticket, einen Host-Header-Konflikt oder ein Problem mit Dateisystemberechtigungen; prüfen Sie die Logs und die Domain-Zuordnungen in `/etc/krb5.conf`.

Sicherheits- und Dateiberechtigungshinweise

Keytab-Dateien dürfen nur von dem Konto lesbar sein, das sie benötigt. Beispielberechtigungen: 0400 mit Eigentümer `zabbix:zabbix` für eine Keytab-Datei eines zabbix-Benutzers 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 in die Domäne aufgenommene Maschinenprinzipale.

Wenn Sie Tests oder Skripte ausführen, die `KRB5CCNAME` setzen oder Keytabs kopieren, überprüfen Sie nach dem Vorgang Eigentümer und Berechtigungen doppelt — dass ein Webserver Anmeldedaten ablehnt, ist häufig ein Problem mit Dateiberechtigungen.

14 Parameter von modbus.get

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

Parameter	Beschreibung	Standardwerte	Beispiel
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
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

15 Erstellen benutzerdefinierter Leistungsindikatornamen für VMware

Übersicht

Der VMware-Leistungszählerpfad hat das Format `group/counter[rollup]`, wobei:

- `group` - die Leistungszählergruppe, zum Beispiel *cpu*
- `counter` - der Name des Leistungszählers, zum Beispiel *usagemhz*
- `rollup` - der Rollup-Typ des Leistungszählers, zum Beispiel *average*

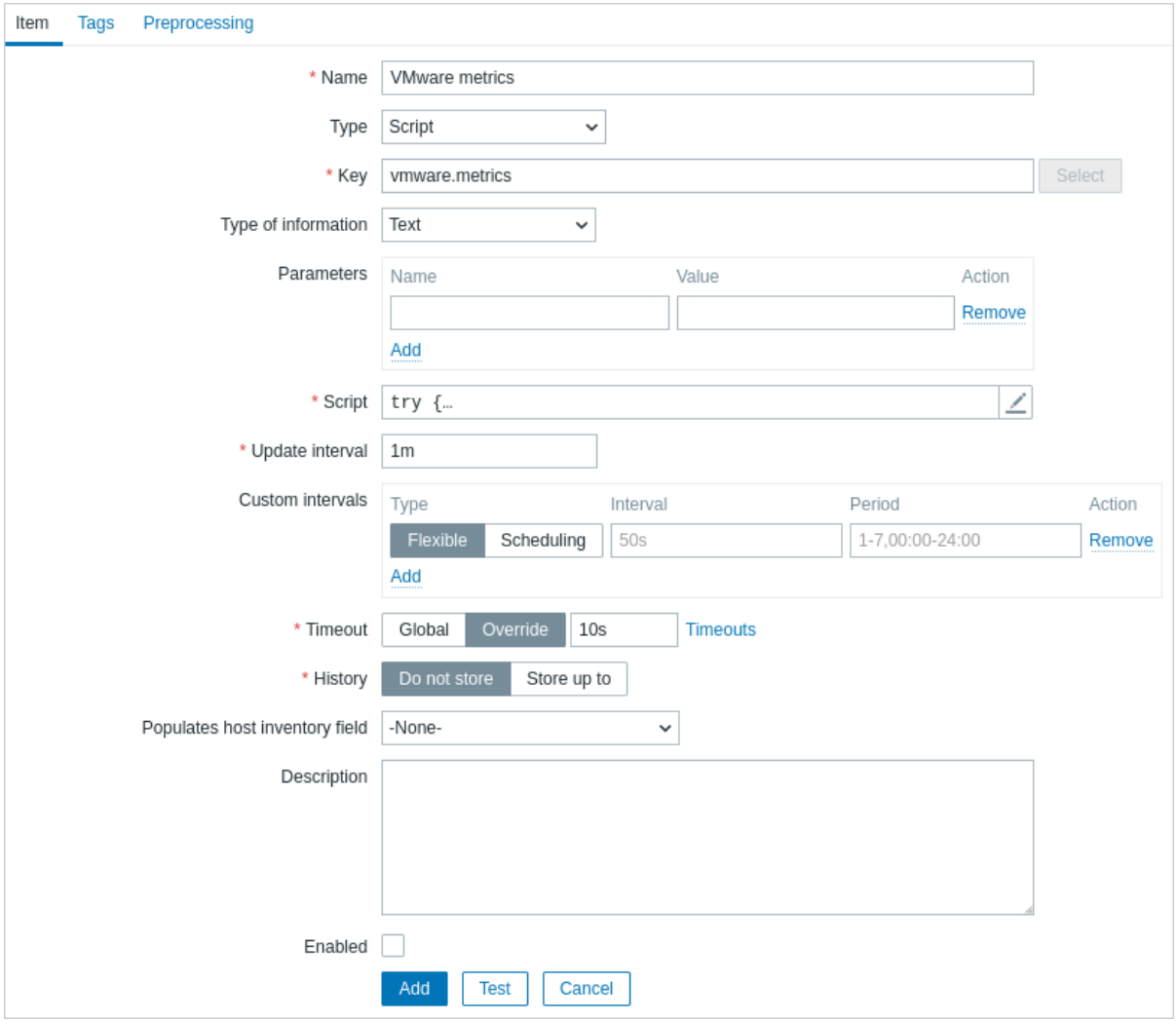
Das obige Beispiel würde also den folgenden Zählerpfad ergeben: `cpu/usagemhz[average]`

Die Beschreibungen der Leistungszählergruppen, Zählernamen und Rollup-Typen finden Sie in der VMware-Dokumentation.

Es ist möglich, interne Namen abzurufen und benutzerdefinierte Leistungszählernamen zu erstellen, indem in Zabbix ein Skript-Datenpunkt verwendet wird.

Konfiguration

1. Erstellen Sie einen deaktivierten Script-Datenpunkt auf dem Haupt-VMware-Host (auf dem der Datenpunkt **eventlog[]** vorhanden ist) mit den folgenden Parametern:



- *Name*: VMware-Metriken
- *Typ*: Script
- *Schlüssel*: vmware.metrics
- *Informationstyp*: Text
- *Script*: Kopieren Sie das unten bereitgestellte Script und fügen Sie es ein
- *Timeout*: 10
- *Historie*: Nicht speichern
- *Aktiviert*: 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;
```

Sobald der Datenpunkt konfiguriert ist, 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 Zählerpfad 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]`

16 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 Schlüssel 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 gebaut wurde (falls nicht verfügbar, auf 0 gesetzt)
  - **value** - menschenlesbares Datum und Uhrzeit, wann das Paket gebaut 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": ""
        }
    }
]
```


17 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 basierend auf den im Datenpunktschlüssel angegebenen Parametern enthält.

Zum Beispiel 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[192.0.2.0,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[192.0.2.0,example.com]` einen Fehler zurückgeben:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


Die folgenden Arten von Fehlercodes sind möglich:

Szenario	"zbx_error_code"	"zbx_error_msg"
Keine Fehler und die DNS-Antwort wurde empfangen und geparst.	0	
DNS ist nicht verfügbar.	-1	"Communication error"
Während des JSON-Parsings tritt ein Fehler auf	-2	"Received unexpected response"

Mit zusätzlichen Parametern kann der Datenpunkt `net.dns.get[192.0.2.0,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": "192.0.2.0"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Siehe auch

Weitere Informationen über DNS-Einträge finden Sie unter:

- Domänennamen - Implementierung und Spezifikation
- Domain Name System (DNS) Parameter

18 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
```

19 Unterstützung für große JSON-Werte

Um eine zuverlässige Verarbeitung großer Werte mit dem JSON-Datentyp (1 MiB oder mehr) sicherzustellen, prüfen 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` (auf manchen Systemen ist standardmäßig möglicherweise 1 MB eingestellt):

```
[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 Konfigurationsdateien von Zabbix Server/Proxy und Agent). Andernfalls werden Nachrichten zwischen ihnen möglicherweise nicht vollständig übertragen, und es können Fehler wie *„message length does not match expected length“* auftreten. Prüfen Sie bei Bedarf außerdem weitere Timeouts, etwa das Skript-Timeout oder das Timeout für den Datenpunkt-Test.

5 Unterstützte Funktionen

Siehe Auslöserausdruck.

6 Makros

Es ist möglich, die sofort einsatzbereiten Unterstützten Makros und Benutzermakros nach unterstütztem Ort zu verwenden.

1 Von Standort unterstützte Makros

Übersicht

Die Tabelle nach Position unterstützte Makros enthält eine Referenz aller Makros (integrierte, Ausdrucks- und benutzerdefinierbare), die an der jeweiligen Position unterstützt werden.

Wenn alle Makros aus einer zugehörigen Gruppe unterstützt werden, wird nur der Gruppenname angegeben. Klicken Sie auf jeden Makronamen oder Gruppennamen, um vollständige Details zu den Makros anzuzeigen.

Die Liste Makrodetails enthält eine detaillierte Liste aller **integrierten** Makros, gruppiert nach Anwendungsbereich. Beachten Sie, dass Sie zum Anpassen von Makrowerten (zum Beispiel zum Kürzen oder Extrahieren bestimmter Teilzeichenfolgen) Makrofunktionen verwenden können.

Nach Position unterstützte Makros

Position	Unterstützte Makros	Kommentare
Aktion		
Bedingung Zeitperiode, Standarddauer des Operationsschritts, Schrittdauer	Benutzerdefinierte Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt
Benachrichtigungen und Befehle		

Position	Unterstützte Makros	Kommentare
Auslöser-basiert	Aktions-Makros Datums-/Zeit-Makros {ESC.HISTORY} Ereignis-Makros, Ursachen-/Symptomereignis-Makros Ereignisaktualisierungs-Makros: {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} Ziel-Host-Makros (nur Befehle) Host-Inventar-Makros Datenpunkt-Makros (außer {ITEM.STATE}, {ITEM.STATE.ERROR}) Proxy-Makros 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} Benutzerdefinierte Makros	Siehe auch: Verwendung von Makros in Nachrichten
Problemaktualisierung	Aktions-Makros Datums-/Zeit-Makros {ESC.HISTORY} Ereignis-Makros, Ursachen-/Symptomereignis-Makros, Wiederherstellungsereignis-Makros Ereignisaktualisierungs-Makros: alle (außer {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Ausdrucksmakro: {?EXPRESSION} Funktions-Makros Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Ziel-Host-Makros (nur Befehle) Host-Inventar-Makros Datenpunkt-Makros (außer {ITEM.STATE}, {ITEM.STATE.ERROR}) Proxy-Makros 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} Benutzerdefinierte Makros	
Problemwiederherstellung	Wiederherstellungsereignis-Makros {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	

Position	Unterstützte Makros	Kommentare
Discovery	Aktions-Makros Datums-/Zeit-Makros Discovery-Makros Ereignis-Makros: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Ziel-Host-Makros (nur Befehle) Proxy-Makros	
Autoregistrierung	Aktions-Makros Datums-/Zeit-Makros Ereignis-Makros: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Host-Makros: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Ziel-Host-Makros (nur Befehle) Proxy-Makros	
Service-basiert	Aktions-Makros Datums-/Zeit-Makros {ESC.HISTORY} Ereignis-Makros: alle außer {EVENT.ACK.STATUS}, {EVENT.OPDATA} Service-Makros Benutzerdefinierte Makros	
Service-Wiederherstellung	Ereignis-Makros (außer {EVENT.ACK.STATUS}, {EVENT.OPDATA}), Wiederherstellungsereignis-Makros	
Service-Aktualisierung	Aktions-Makros Datums-/Zeit-Makros {ESC.HISTORY} Ereignis-Makros (außer {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Ereignisaktualisierungs-Makros: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Service-Makros Benutzerdefinierte Makros	
Interne Benachrichtigungen	Aktions-Makros Datums-/Zeit-Makros {ESC.HISTORY} Ereignis-Makros (außer {EVENT.NSEVERITY}, {EVENT.SEVERITY}), Wiederherstellungsereignis-Makros Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Host-Inventar-Makros Datenpunkt-Makros: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Proxy-Makros	
Datenpunkt-basiert	Datenpunkt-Makros: {ITEM.STATE}, {ITEM.STATE.ERROR}	
LLD-Regel-basiert	Benachrichtigungs-Makros für Low-Level-Discovery	
Auslöser-basiert	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} Benutzerdefinierte Makros	

Position	Unterstützte Makros	Kommentare
Warnskript-Parameter	Warnungs-Makros Benutzerdefinierte Makros (nur global)	
Konnektor		
Feld URL, Benutzername, Passwort, Bearer-Token, HTTP-Proxy, SSL-Zertifikat, SSL-Schlüsseldatei, SSL-Schlüsselpasswort	Benutzerdefinierte Makros (nur global)	
Feld Timeout	Benutzerdefinierte Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Graphname	Ausdrucksmakro: {?EXPRESSION}	In diesem Makro an dieser Position 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} Benutzerdefinierte Makros [1]	[1] Für die IP-Adresse muss ein einzelnes Makro das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt
Schnittstellen-Port	Benutzerdefinierte Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt
SNMP-Community SNMPv3-Kontextname, Sicherheitsname, Authentifizierungs-Passphrase, Privacy-Passphrase	Benutzerdefinierte Makros	

Position	Unterstützte Makros	Kommentare
IPMI-Benutzername, Passwort		
Datenpunkt, Datenpunkt-Prototyp, LLD-Regel		
Name	Benutzerdefinierte Makros	In Namen von LLD-Regeln nicht unterstützt.
Beschreibung	Benutzerdefinierte Makros	
Schlüsselparameter	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	Die unterstützten {HOST.*}-Makros werden zur Schnittstelle aufgelöst, die für den Datenpunkt ausgewählt ist. Wenn sie in Datenpunkten ohne Schnittstellen verwendet werden, werden sie in dieser Prioritätsreihenfolge zur Zabbix-Agent-, SNMP-, JMX- oder IPMI-Schnittstelle des Hosts aufgelöst oder zu 'UNKNOWN', wenn der Host keine Schnittstelle hat.
Felder Aktualisierungsintervall, Benutzerdefinierte Intervalle, Timeout (verfügbar für unterstützte Datenpunkttypen), Aufbewahren bis zu (für Verlauf/Trends)	Benutzerdefinierte Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Felder Verlorene Ressourcen löschen, Verlorene Ressourcen deaktivieren	Benutzerdefinierte Makros	Diese Felder werden nur für LLD-Regeln unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Parameter des Vorverarbeitungsschritts (einschließlich benutzerdefinierter Skripte), *Parameter der benutzerdefinierten Fehlerbehandlung* (*Felder Wert setzen auf* und *Fehler setzen auf*)	Benutzerdefinierte Makros	
Reguläre Ausdrücke für Filter	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	Diese Felder werden nur für LLD-Regeln unterstützt.

Position	Unterstützte Makros	Kommentare
Reguläre Ausdrücke für Override-Filter	Benutzerdefinierte Makros	Diese Felder werden nur für LLD-Regeln unterstützt.
Override-Operationen: *Aktualisierungsintervall*, *Speicherzeitraum für Verlauf*, *Speicherzeitraum für Trends* (für Datenpunkt-Prototypen)	Benutzerdefinierte Makros	Diese Felder werden nur für LLD-Regeln unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/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} Host-Inventar-Makros Datenpunkt-Makros: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Benutzerdefinierte Makros	
Skript-Datenpunkt *Parameternamen und -werte*		
Berechneter/Aggregat-Datenpunktformel (*Ausdruckskonstanten*, *Funktionsparameter*, *Datenpunktschlüsselparameter*, (nur Aggregat-Datenpunkt) *Filterbedingungen* (Hostgruppenname und Tag-Name))	Benutzerdefinierte Makros	
Datenbankmonitor-Datenpunkt *Felder Benutzername, Passwort, SQL-Abfrage*	Benutzerdefinierte Makros	
HTTP-Agent-Datenpunkt *Felder URL, Abfragefelder, Request-Body, 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} Benutzerdefinierte Makros	URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das darin enthaltene Makro als ”******” aufgelöst wird
HTTP-Agent-Datenpunkt *Feld HTTP-Proxy*	Datenpunkt-Makros: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Benutzerdefinierte Makros	
HTTP-Agent-Datenpunkt *Felder Erforderliche Statuscodes, HTTP-Authentifizierungs-Benutzername/-Passwort, SSL-Schlüsselpasswort*	Benutzerdefinierte Makros	
HTTP-Agent-Datenpunkt *Feld Erlaubte Hosts*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	
JMX-Agent-Datenpunkt *Feld Endpoint*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Benutzerdefinierte Makros	

Position	Unterstützte Makros	Kommentare
Trapper-Datenpunkt *Feld Erlaubte Hosts*		
SNMP-Agent-Datenpunkt *Feld SNMP-OID*	Benutzerdefinierte Makros	
SSH- und Telnet-Datenpunkt *Skript*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	
SSH-Agent-Datenpunkt *Felder Benutzername, Datei mit öffentlichem Schlüssel, Datei mit privatem Schlüssel, Passwort*	Benutzerdefinierte Makros	
Telnet-Agent-Datenpunkt *Felder Benutzername, Passwort*	Benutzerdefinierte 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} Host-Inventar-Makros Karten-Makros {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} Host-Inventar-Makros Karten-Makros 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 Position 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	Ausdrucksmakro: {?EXPRESSION} [1]	
Textfeld in Kartenformen	{MAP.NAME}	
Netzwerkerkennung		
Feld Aktualisierungsintervall	Benutzerdefinierte Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Felder SNMP-Community, SNMP-OID	Benutzerdefinierte Makros (nur global)	

Position	Unterstützte Makros	Kommentare
Felder SNMPv3-Kontextname, Sicherheitsname, Authentifizierungs-Passphrase, Privacy-Passphrase		
Proxy		
Adresse für aktive Agents > Port (wenn der Proxy zu einer Gruppe gehört) Für passiven Proxy-Modus: Schnittstellenfelder *Adresse* und *Port* *Timeouts für Datenpunkttypen* im Override-Modus	Benutzerdefinierte Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Proxy-Gruppe		
Felder Failover-Zeitraum, Mindestanzahl von Proxys	Benutzerdefinierte Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/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} Host-Inventar-Makros [2] {MANUALINPUT} Benutzernamen-Makros Benutzerdefinierte Makros	[1] Nur Bestätigungstext [2] Unterstützt für Zabbix-Server und Zabbix-Proxy
Befehle (Typ: Skript, manuelle **Ereignisaktion**)	Datums-/Zeit-Makros Ereignis-Makros, Ursachen-/Symptomereignis-Makros, Wiederherstellungsereignis-Makros (wenn eine Wiederherstellung stattgefunden hat) Ereignisaktualisierungs-Makros: {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} Host-Inventar-Makros Datenpunkt-Makros (außer {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Proxy-Makros 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} Benutzernamen-Makros Benutzerdefinierte Makros	

Position	Unterstützte Makros	Kommentare
Bestätigungstext (Typ: Skript, manuelle **Ereignisaktion**)	Ereignis-Makros: {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} Host-Inventar-Makros {MANUALINPUT} Benutzernamen-Makros Benutzerdefinierte 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} Host-Inventar-Makros Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Auslöser-Makros: {TRIGGER.ID} (nur Auslöser-Tag-Wert) Benutzerdefinierte Makros	Makros für Tag-Name und -Wert werden nur während des Prozesses der Ereignisgenerierung 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 Benutzerdefinierte Makros	
Ereignisname	Datums-/Zeit-Makros: {TIME}, {TIMESTAMP} Ausdrucksmakro: {?EXPRESSION} Funktions-Makros 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}	
Betriebsdaten	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Benutzerdefinierte Makros	
Ausdruck	{TRIGGER.VALUE} Benutzerdefinierte Makros [1]	[1] Nur in Konstanten und Funktionsparametern; geheime Makros werden nicht unterstützt
Felder Tag für Abgleich, Name des Menüeintrags, URL des Menüeintrags	Benutzerdefinierte Makros	URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das darin enthaltene Makro 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*} Benutzerdefinierte Makros	

Position	Unterstützte Makros	Kommentare
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}	
Befehle für Benutzerparameter	Positionsmakros/-referenzen: $1...$9	
Webszenario		
Felder Name, Agent, HTTP-Proxy	Benutzerdefinierte Makros	
Feld Aktualisierungsintervall	Benutzerdefinierte Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Felder Variablen, Header, SSL-Zertifikat, SSL-Schlüsseldatei	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	
Felder Benutzer, Passwort, SSL-Schlüsselpasswort	Benutzerdefinierte Makros	
Webszenario-Schritt *Felder Name, Variablen (nur Werte)*	Benutzerdefinierte Makros	
Webszenario-Schritt *Felder Timeout, Erforderliche Statuscodes*	Benutzerdefinierte Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Webszenario-Schritt *Felder URL, Header (Namen und Werte), Erforderliche Zeichenfolge*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	
Webszenario-Schritt *Feld Post*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
webhook-Medientyp		
Parameterwerte	Warnungs-Makros Alle Makros, die in Auslöser-basierten Problembenachrichtigungen unterstützt werden (siehe oben)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
JavaScript-Skript, *JavaScript*-Skriptparametername	Benutzerdefinierte Makros (nur global)	
Name des Menüeintrags, URL des Menüeintrags	{EVENT.TAGS.<tag name>}	
Widgets		

Position	Unterstützte Makros	Kommentare
Messwert-Widget *Feld Beschreibung*	Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Host-Inventar-Makros Datenpunkt-Makros (außer {ITEM.STATE.ERROR}) Benutzerdefinierte Makros (nur global)	
Waben-Widget *Feld Primäre/sekundäre Beschriftung*		
Datenpunktwert-Widget *Feld Beschreibung*		
Top-Hosts-Widget *Spalte Textdaten*	Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Host-Inventar-Makros Benutzerdefinierte Makros (nur global)	
URL-Widget (dynamisch) *Feld URL*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierte Makros	
Andere Positionen		
Feld Arbeitszeit	Benutzerdefinierte Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros/eine Mischung mit Text werden nicht unterstützt.
Feld Timeouts für Datenpunkttypen (globale Einstellung)		
Benutzermedium *Feld Aktiv wenn*	Benutzerdefinierte Makros (nur global)	
E-Mail-Medientyp *Felder Benutzername, Passwort*	Benutzerdefinierte Makros (nur global)	Geheime Makros werden empfohlen.
Geplante Berichte *Felder Betreff, Nachricht*	Datums-/Zeit-Makros: {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 durch den Tag-Namen verwiesen wird.
 Ein Tag-Name, der nicht alphanumerische Zeichen enthält (einschließlich nicht-englischer mehrbyteiger 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 durch den Tag-Namen verwiesen wird.
 Ein Tag-Name, der nicht alphanumerische Zeichen enthält (einschließlich nicht-englischer mehrbyteiger 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: Host-Name, Ereignisname, Schweregrad, Alter, Service-Tags und -Werte.
 Dieses Makro wird im Kontext des Ursacheereignisses verwendet und gibt Informationen über Symptomereignisse 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 Tag-Objekte des Wiederherstellungsereignisses 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}

Für Menschen lesbarer Name der Aktion(en), die während einer Problemaktualisierung durchgeführt wurden.
 Wird in die folgenden Werte aufgelöst: *bestätigt*, *nicht bestätigt*, *kommentiert*, *Schweregrad von (ursprünglicher Schweregrad) auf (aktualisierter Schweregrad) geändert* und *geschlossen* (abhängig davon, wie viele Aktionen in einer Aktualisierung durchgeführt werden).

{EVENT.UPDATE.ACTIONJSON}

Ein JSON-Array, das Details zu den Aktion(en) enthält, die während der Problemaktualisierung durchgeführt wurden.
 Mögliche Werte von JSON-Eigenschaften:
- true (für die Eigenschaften `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` und `symptom`);
- <message string> (für die Eigenschaft `message`);
- Zeitstempel (für die Eigenschaften `suppress_until` und `timestamp`) oder 0 (für die Eigenschaft `suppress_until`, wenn die Unterdrückung unbegrenzt gilt);
- 0, 1, 2, 3, 4, 5 (für die Schweregrad-Eigenschaften `old` und `new`).

 Zum Beispiel: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851199,"timestamp":1730822048}`.

{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 Datenpunkt-basierten Funktion im Auslöser-Ausdruck zum Zeitpunkt des Ereignisses.
 Es werden nur Funktionen gezählt, die `/host/key` als ersten Parameter haben.

 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{FUNCTION.RECOVERY.VALUE}

Der Wert der N-ten Datenpunkt-basierten Funktion im Wiederherstellungsausdruck zum Zeitpunkt des Ereignisses.*
 Es werden nur Funktionen gezählt, die `/host/keys` als ersten Parameter haben.

 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, Interface-IP-/DNS-Feldern und 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 ebenfalls keine JMX-Schnittstelle definiert ist, wird die Haupt-IPMI-Schnittstelle verwendet. Wenn der Host über keine Schnittstelle verfügt, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.CONN<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{HOST.DNS}

Der DNS-Name des Hosts.
 In globalen Skripten, in den IP-/DNS-Feldern von Schnittstellen und in Webszenarien wird das Makro zur primären 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 über keine Schnittstelle verfügt, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.DNS<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck 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 Verlaufsfunktion verwendet wird, zum Beispiel `last(/{HOST.HOST}/{ITEM.KEY})`.

 Dieses Makro kann mit einem numerischen Index als {HOST.HOST<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck 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 auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{HOST.IP}

Die IP-Adresse des Hosts.
 In globalen Skripten, in den IP-/DNS-Feldern von Schnittstellen 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 definiert ist, wird die Haupt-IPMI-Schnittstelle verwendet. Wenn der Host über keine Schnittstelle verfügt, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.IP<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{HOST.METADATA}

Die Host-Metadaten.

 Wird nur für die aktive Agent-Autoregistrierung verwendet.

{HOST.NAME}

Der sichtbare Host-Name.

 Dieses Makro kann mit einem numerischen Index als {HOST.NAME<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{HOST.PORT}

Der Port des Hosts (Agent).
 In globalen Skripten, Interface-IP-/DNS-Feldern und Webszenarien wird das Makro zur Hauptschnittstelle des Agent 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 definiert ist, wird die Haupt-IPMI-Schnittstelle verwendet. Wenn der Host über keine Schnittstelle verfügt, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.PORT<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.CONTACT}

Das Feld *Kontakt* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.CONTACT<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.DEPLOYMENT.STATUS}

Das Feld *Bereitstellungsstatus* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.DEPLOYMENT.STATUS<1-9>}, um im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HOST.NETWORKS}

Das Feld *Host-Netzwerke* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HOST.NETWORKS<1-9>}, um im Ausdruck eines Auslösers 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 *Hardware architecture* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.ARCH<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.EXPIRY}

Das Feld *Ablaufdatum der Hardwarewartung* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.EXPIRY<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.INSTALL}

Das Feld *Installationsdatum der Hardware* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.INSTALL<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.PURCHASE}

Das Feld *Kaufdatum der Hardware* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.PURCHASE<1-9>}, um im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.LOCATION}

Das Feld *Standort* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.LOCATION<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.LOCATION.LAT}

Das Feld *Breitengrad des Standorts* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.LOCATION.LAT<1-9>}, um im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 Auslöser-Ausdruck 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.MODEL}

Das Feld *Modell* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.MODEL<1-9>}, um im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.NOTES}

Das Feld *Notizen* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.NOTES<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 Auslöser-Ausdruck 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 Auslöser-Ausdruck 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.NOTES}

Das Feld *Anmerkungen zum primären Ansprechpartner* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.NOTES<1-9>}, um im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.EMAIL}

Das Feld *E-Mail des sekundären Ansprechpartners* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.NAME}

Das Feld *Name des sekundären POC* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.NAME<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.NOTES}

Das Feld *Notizen zum sekundären Ansprechpartner* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.NOTES<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.PHONE.A}

Das Feld *Sekundäre POC-Telefonnummer A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.PHONE.B}

Das Feld *Sekundäre POC-Telefonnummer B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.PHONE.B<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.SCREEN}

Das Feld *Sekundärer POC-Bildschirmname* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.SCREEN<1-9>}, um im Ausdruck eines Auslösers 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 auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ADDRESS.A}

Das Feld *Standortadresse A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ADDRESS.A<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ADDRESS.B}

Das Feld *Standortadresse B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ADDRESS.B<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ADDRESS.C}

Das Feld *Standortadresse C* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ADDRESS.C<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.CITY}

Das Feld *Standort Stadt* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.CITY<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.COUNTRY}

Das Feld *Standortland* 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.NOTES}

Das Feld *Standortnotizen* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.NOTES<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.RACK}

Das Feld *Standort-Rack-Position* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.RACK<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.STATE}

Das Feld *Standort Bundesland/Provinz* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.STATE<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ZIP}

Das Feld *Standort ZIP/Postleitzahl* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ZIP<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.A}

Das Feld *Software application 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 *Software application B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.B<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.C}

Das Feld *Software application C* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.C<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.D}

Das Feld *Software application D* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.D<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.E}

Das Feld *Software application E* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.E<1-9>}, um im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.TYPE}

Das Feld *Typ* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.TYPE<1-9>}, um im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 Auslöser-Ausdruck 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host 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 im Ausdruck eines Auslösers 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 Auslöser-Ausdruck 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 im Ausdruck eines Auslösers auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

Datenpunkt-Makros

{ITEM.DESCRIPTION}

Die Beschreibung des N-ten Datenpunkts im Auslöser-Ausdruck, der eine Benachrichtigung verursacht 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.DESCRIPTION.ORIG}

Die ursprüngliche Beschreibung (mit nicht aufgelösten Makros) des N-ten Datenpunkts im Auslöserausdruck, der eine Benachrichtigung verursacht 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 im 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 im Auslöserausdruck, der eine Benachrichtigung verursacht hat.
 Makrofunktionen werden für dieses Makro nicht unterstützt, wenn es als Platzhalter im ersten Parameter einer Verlaufsfunktion 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 Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.KEY.ORIG}

Der ursprüngliche Schlüssel (mit nicht aufgelösten Makros) des N-ten Datenpunkts im Auslöser-Ausdruck, der eine Benachrichtigung verursacht 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE}

Der neueste Wert des N-ten Datenpunkts im Auslöser-Ausdruck, der eine Benachrichtigung verursacht hat.
 Im Frontend wird er zu *UNKNOWN* aufgelöst, wenn der neueste Verlaufswert vor mehr als dem Zeitraum *Max history display period* erfasst wurde (festgelegt im Menüabschnitt Administration → General).
 Bei Verwendung im Problemnamen wird das Makro beim Anzeigen von Problemereignissen nicht zum neuesten Datenpunktwert aufgelöst; stattdessen bleibt der Datenpunktwert vom Zeitpunkt erhalten, zu dem das Problem aufgetreten ist.
Bei Verwendung in Benachrichtigungen wird das Makro in einigen Fällen möglicherweise nicht zum neuesten Datenpunktwert zum Zeitpunkt des Auslösens des Auslösers aufgelöst. Wenn ein Datenpunkt beispielsweise schnell zwei Werte, „A“ und „B“, empfängt und der Auslöser bei „A“ ausgelöst wird, kann in Benachrichtigungen aufgrund einer leichten Verarbeitungsverzögerung „B“ als neuester Wert angezeigt werden – der neueste Datenpunktwert hat sich zwischen dem Zeitpunkt, zu dem der Auslöser ausgelöst wurde, und dem Zeitpunkt, zu dem die Benachrichtigung erstellt wurde, geändert. Um dies zu vermeiden, können Sie das Makro {ITEM.VALUE} verwenden, das zum Wert zum Zeitpunkt des Auslösens des Auslösers aufgelöst wird und so sicherstellt, dass in der Benachrichtigung der korrekte Wert verwendet wird.
 Es ist ein Alias für `last(/{HOST.HOST}/{ITEM.KEY}).`
 Der aufgelöste Wert für Text-/Log-Datenpunkte wird im Frontend an den folgenden Stellen auf 20 Zeichen gekürzt:
- Betriebsdaten;
- Auslöserbeschreibung;
- Auslöser-URLs;
- Bezeichnungen von Auslöser-URLs;
- Beschreibung des Datenpunktwert-Widgets.
Um zu einem 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.AGE}

Die Zeit, die zwischen der Erfassung des neuesten Datenpunkt-Werts und der Auswertung des Makros verstrichen ist.
 Wird in einem menschenlesbaren 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.DATE}

Das Datum, an dem der letzte Datenpunktwert erfasst wurde.
 Wird im Format `YYYYMMDD` angezeigt.
 In Auslösernamen (in Monitoring -> Problems-Liste), Ereignisnamen sowie Tag-Namen und -Werten wird die Zeitzone des Servers 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 Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.TIME}

Die Uhrzeit, zu der der letzte Datenpunktwert erfasst wurde.
 Wird im Format `HHMMSS` angezeigt.
 In Auslösernamen (in Überwachung -> Problemliste), Ereignisnamen sowie Tagnamen und -werten wird die Zeitzone des Servers 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 Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.TIMESTAMP}

Der UNIX-Zeitstempel, zu dem der letzte Datenpunkt-Wert 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 Log-Ereignisses.
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 Log-Eintrag 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 Auslöser-Ausdruck

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 den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.NSEVERITY}

Der numerische Schweregrad 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 den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.SEVERITY}

Die verbale Severity 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 Auslöser-Ausdruck 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 Protokolleintrag in das Protokoll 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.TIMESTAMP}

Der UNIX-Zeitstempel, zu dem der Protokolleintrag in das Protokoll geschrieben wurde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.TIMESTAMP<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.NAME}

Der Name des Datenpunkts, wobei alle Makros aufgelöst sind.

 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 (mit nicht aufgelösten Makros) des Datenpunkts.

 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 letzte Status des N-ten Datenpunkts im Auslöser-Ausdruck, der eine Benachrichtigung verursacht hat.
 Mögliche Werte: *Nicht unterstützt*, *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 indizierte Makros.

{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 auf einen der folgenden Werte aufgelöst:

- den historischen Wert (zum Zeitpunkt des Ereignisses) des N-ten Datenpunkts im Auslöser-Ausdruck, wenn er im Kontext einer Änderung des Auslöser-Status verwendet wird, zum Beispiel bei der Anzeige von Ereignissen oder beim Senden von Benachrichtigungen;
- den letzten Wert des N-ten Datenpunkts im Auslöser-Ausdruck, wenn er ohne den Kontext einer Änderung des Auslöser-Status verwendet wird, zum Beispiel bei der Anzeige einer Liste von Auslösern in einem Pop-up-Auswahlfenster. In diesem Fall funktioniert er genauso wie {ITEM.LASTVALUE}.

Im ersten Fall wird er zu *UNKNOWN* aufgelöst, wenn der Verlaufswert bereits gelöscht wurde oder nie gespeichert wurde.
Im zweiten Fall, und nur im Frontend, wird er zu *UNKNOWN* aufgelöst, wenn der letzte Verlaufswert vor mehr als dem Zeitraum *Max history display period* erfasst wurde (festgelegt im Menüabschnitt Administration→General).
Der aufgelöste Wert

für Text-/Log-Datenpunkte wird vom Frontend an den folgenden Stellen auf 20 Zeichen gekürzt:
- Betriebsdaten;
- Auslöser-Beschreibung;
- Auslöser-URLs;
- Auslöser-URL-Beschriftungen;
- Beschreibung des Datenpunktwert-Widgets.
Um einen 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 Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.VALUE.AGE}

Die Zeit, die zwischen der Erfassung des Datenpunktwertes und der Makroauswertung verstrichen ist.
 Wird in einem menschenlesbaren 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 Datenpunktwert erfasst wurde.
 Wird im Format `YYYYMMDD` angezeigt.
 In Auslösernamen (in Überwachung -> Problemliste), Ereignisnamen sowie Tag-Namen und -Werten wird die Zeitzone des Servers 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 Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.VALUE.TIME}

Die Uhrzeit, zu der der Datenpunktwert erfasst wurde.
 Wird im Format `HHMMSS` angezeigt.
 In Auslösernamen (in Monitoring -> Problems list), Ereignisnamen sowie Tag-Namen und -Werten wird die Zeitzone des Servers 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 Auslöserausdruck 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 im Auslöser-Ausdruck, der eine Benachrichtigung ausgelöst hat.
 Mögliche Werte: *0* - Numerischer Gleitkommawert, *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 indizierte 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 letzte Status der Low-Level-Discovery-Regel.
 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.
 Wird auf eines der folgenden Elemente aufgelöst:

- Proxy des N-ten Datenpunkts im Auslöser-Ausdruck (in Auslöser-basierten Benachrichtigungen). Sie können hier indizierte Makros verwenden;
- Proxy, der die Discovery ausgeführt hat (in Discovery-Benachrichtigungen). Verwenden Sie hier {PROXY.DESCRIPTION} ohne Index;
- Proxy, bei dem sich ein aktiver Agent registriert hat (in Autoregistrierungs-Benachrichtigungen). Verwenden Sie hier {PROXY.DESCRIPTION} ohne Index.

Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {PROXY.DESCRIPTION<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{PROXY.NAME}

Der Proxy-Name.
 Wird auf eines der folgenden Elemente aufgelöst:

- Proxy des N-ten Datenpunkts im Auslöser-Ausdruck (in Auslöser-basierten Benachrichtigungen). Sie können hier indizierte Makros verwenden;
- Proxy, der die Discovery ausgeführt hat (in Discovery-Benachrichtigungen). Verwenden Sie hier {PROXY.NAME} ohne Index;
- Proxy, bei dem sich ein aktiver Agent registriert hat (in Autoregistrierungs-Benachrichtigungen). Verwenden Sie hier {PROXY.NAME} ohne Index.

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 Tag-Bereich der Service-Konfiguration definiert werden. Wird zu einer leeren Zeichenfolge erweitert, wenn keine Tags vorhanden sind.

{SERVICE.TAGSJSON}

Ein JSON-Array, das Tag-Objekte von Service-Ereignissen enthält.
 Tags von Service-Ereignissen können im Tag-Bereich der Service-Konfiguration 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 durch den Tag-Namen verwiesen wird.
 Die Service-Ereignis-Tags können im Abschnitt „Tags“ der Service-Konfiguration definiert werden.
 Ein Tag-Name, der nicht alphanumerische Zeichen enthält (einschließlich nicht-englischer mehrbyteiger 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.
 Datenpunkt-basierte Funktionen werden ausgewertet und zum Zeitpunkt der Ereigniserzeugung 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.
 Datenpunkt-basierte 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 den N-ten Datenpunkt, die N-te Funktion oder den N-ten Host im Feld *Expression* 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 zu den Werten des ersten, zweiten und dritten Datenpunkts im Auslöserausdruck zum Zeitpunkt des Ereignisses aufgelöst (falls vorhanden).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** werden zu den Werten der ersten, zweiten und dritten datenpunktbasierten Funktionen zum Zeitpunkt des Ereignisses aufgelöst (falls vorhanden).

Im Kontext von Auslösern beziehen sich indizierte Makros immer auf das Feld *Expression* der Auslöserkonfiguration, nicht auf die *Recovery expression*. Zum Beispiel wird **{ITEM.VALUE2}** in einem Wiederherstellungsereignis zum Wert des zweiten Datenpunkts aus dem Problemausdruck zum Zeitpunkt der Wiederherstellung aufgelöst.

Das Makro **{HOST.HOST<1-9>}** wird auch innerhalb des Ausdrucksmakros `{?func(/host/key,param)}` in **Grafiknamen** unterstützt. Zum Beispiel wird `{?func(/{HOST.HOST2}/key,param)}` in einem Grafiknamen zum Host des zweiten Datenpunkts in der Grafik aufgelöst.

Warning:
Indizierte Makros werden in keinem anderen Kontext als den hier genannten Fällen aufgelöst. Verwenden Sie in anderen Kontexten Makros ohne Index (**{HOST.HOST}, {HOST.IP}** usw.).

2 Benutzer-Makros, die vom Standort unterstützt werden

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:

Position	Mehrere Makros/Mischung mit Text[1]
Name des Datenpunkts	ja
Parameter des Datenpunktschlüssels	ja
Aktualisierungsintervall	nein
Benutzerdefinierte Intervalle	nein
Timeout (verfügbar für unterstützte Datenpunkt-Typen)	nein
Speichern bis zu (für Verlauf und Trends)	nein
Beschreibung	ja
Berechneter/aggregierter Datenpunkt	

Position	Mehrere Makros/Mischung mit Text[1]
Formel (Ausdruckskonstanten und Funktionsparameter; Parameter des Datenpunktschlüssels; (*nur aggregierter Datenpunkt*) Filterbedingungen (Hostgruppenname und Tag-Name))	ja
Datenbankmonitor	
Benutzername	ja
Passwort	ja
SQL-Abfrage	ja
HTTP-Agent	
URL[3]	ja
Abfragefelder	ja
Request-Body	ja
Header (Namen und Werte)	ja
Erforderliche Statuscodes	ja
HTTP-Proxy	ja
Benutzername für HTTP-Authentifizierung	ja
Passwort für HTTP-Authentifizierung	ja
SSL-Zertifikatsdatei	ja
SSL-Schlüsseldatei	ja
SSL-Schlüsselpasswort	ja
Erlaubte Hosts	ja
JMX-Agent	
JMX-Endpunkt	ja
Skript-Datenpunkt	
Parameternamen und -werte	ja
Browser-Datenpunkt	
Parameternamen und -werte	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
Tags[2]	
Tag-Namen	ja
Tag-Werte	ja
Vorverarbeitungsschritte	
Parameter (einschließlich benutzerdefinierter Skripte)	ja
Parameter für benutzerdefinierte 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:

Ort	Mehrere Makros/Mischung mit Text[1]
Schlüsselparameter	ja
Aktualisierungsintervall	nein
Benutzerdefiniertes Intervall	nein
Timeout (verfügbar für unterstützte Datenpunkt-Typen)	nein
Verlorene Ressourcen löschen	nein
Verlorene Ressourcen deaktivieren	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	
Zulässige 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

Ort	Mehrere Makros/Mischung mit Text[1]
Überschreibungen	
Filter: regulärer Ausdruck	ja
Operationen: Aktualisierungsintervall (für Datenpunkt-Prototypen)	nein
Operationen: Aufbewahrungszeitraum für Verlaufsdaten (für Datenpunkt-Prototypen)	nein
Operationen: Aufbewahrungszeitraum für Trenddaten (für Datenpunkt-Prototypen)	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
	Privacy-Passphrase	ja
	SNMP-OID	ja

Proxys

In einer Proxy-Konfiguration können Benutzermakros in den folgenden Feldern verwendet werden:

Ort	Mehrere Makros/Mischung mit Text[1]
Adresse für aktive Agents > *Port* (wenn der Proxy zu einer Gruppe gehört)	nein
Schnittstellenadresse und Port (für passiven Proxy)	nein
Timeouts für Datenpunkttypen	nein

Proxy-Gruppen

In einer Proxy-Gruppen-Konfiguration können Benutzermakros in den folgenden Feldern verwendet werden:

Ort	Mehrere Makros/Mischung mit Text[1]
Failover-Zeitraum	nein
Mindestanzahl 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
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 Stellen

Zusätzlich zu den hier aufgeführten Stellen können Benutzermakros in den folgenden Feldern verwendet werden:

Stelle	Mehrere Makros/Mischung mit Text[1]
Globale Skripte (URL, Skript, SSH, Telnet, IPMI), einschließlich Bestätigungstext webhooks	ja
JavaScript-Skript	nein
Name des JavaScript-Skriptparameters	nein
Wert des JavaScript-Skriptparameters	ja
Dashboards	
Spalte des Datentyps *Text* im Dashboard-Widget *Top hosts*	ja
Parameter *Description* in den Dashboard-Widgets *Item value* und *Gauge*	ja
Parameter *Text* für primäre/sekundäre Beschriftung im Dashboard-Widget *Honeycomb*	ja
Parameter *URL*[3] im Dashboard-Widget *URL*	ja

Stelle	Mehrere Makros/Mischung mit Text[1]
Benutzer → *Benutzer* → *Medien*	
Aktiv wenn	ja
Administration → *Allgemein* → *GUI*	
Arbeitszeit	nein
Administration → *Allgemein* → *Timeouts*	
Timeouts für Datenpunkttypen	nein
Administration → *Allgemein* → *Connectors*	
URL	ja
Benutzername	ja
Passwort	ja
Bearer-Token	ja
Timeout	nein
HTTP-Proxy	ja
SSL-Zertifikatsdatei	ja
SSL-Schlüsseldatei	ja
SSL-Schlüsselpasswort	ja
Warnungen → *Medientypen* → *Nachrichtenvorlagen*	
Betreff	ja
Nachricht	ja

Stelle	Mehrere Makros/Mischung mit Text[1]
Warnungen → Medientypen → Skript Skriptparameter	ja
Warnungen → Medientypen → Medientyp Felder *Username* und *Password* für den Medientyp *Email* (wenn *Authentication* auf "Username and password" gesetzt ist; geheime Makros empfohlen)	ja

Eine vollständige Liste aller in Zabbix unterstützten Makros finden Sie unter unterstützte Makros.

Fußnoten

[1] Wenn mehrere Makros in einem Feld oder mit Text gemischte Makros für den Ort nicht unterstützt werden, muss ein einzelnes Makro das gesamte Feld ausfüllen.

[2] In Tag-Namen und -Werten verwendete Makros werden nur während des Prozesses der Ereigniserzeugung aufgelöst.

[3] URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das darin enthaltene Makro als "******" aufgelöst wird.

7 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 die folgenden Zeitsuffixe in der Entitätskonfiguration:

- **s** - Sekunden *(bei Verwendung funktioniert dies genauso wie der Rohwert)*
- **m** - Minuten
- **h** - Stunden
- **d** - Tage
- **w** - Wochen
- **M** - Monate *(Trendfunktionen nur)*
- **y** - Jahre *(Trendfunktionen nur)*

Zeitsuffixe werden nur mit Ganzzahlen unterstützt. Zum Beispiel wird 1h unterstützt, aber 1,5h oder 1.5h werden nicht unterstützt; verwenden Sie stattdessen 90m.

Suffixe für Speichergrößen

Zabbix unterstützt die folgenden Suffixe für Speichergrößen:

- **K** - Kilobyte
- **M** - Megabyte
- **G** - Gigabyte
- **T** - Terabyte

Suffixe für Datenpunktwerte

Suffixe können auch verwendet werden, um numerische Datenpunktwerte in einem menschenlesbaren Format anzuzeigen.

Um dies zu aktivieren, verwenden Sie eines der folgenden Suffixe im Feld *Units*, wenn Sie einen Datenpunkt konfigurieren:

- **B** - Byte
- **Bps** - Byte pro Sekunde
- **s** - Sekunden, angezeigt mit bis zu drei größten Zeitangaben ungleich null
- **uptime** - verstrichene Zeit im Format 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 zusätzlich die folgenden Regeln:

- Für B und Bps verwendet Zabbix die Umrechnung zur Basis 2 (1K = 1024B) gemäß dem JEDEC-Standard.
- Für andere Einheiten (wie Hz, W usw.) verwendet Zabbix die Umrechnung zur Basis 10 (1K = 1000).
- Für s (Sekunden):
  - Das Format umfasst yyy mmm ddd hhh mmm sss ms; angezeigt werden nur bis zu drei der größten Zeitangaben ungleich null (z. B. 1M 10d 4h).
  - Wenn eine Einheit null ist und sich zwischen zwei Einheiten ungleich null befindet, wird sie ausgelassen (z. B. 10d 56m statt 10d 0h 56m).

Wenn *Units* verwendet werden, werden die folgenden Multiplikator-Suffixe automatisch auf Datenpunktwerte 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 Datenpunktwerte basierend auf den angegebenen Einheiten umgewandelt 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 codierte Stoppliste für Einheiten, bestehend aus `ms`, `rpm`, `RPM`, `%`. Diese Stoppliste ist veraltet; daher ist die korrekte Methode, die Umwandlung solcher Einheiten zu verhindern, `!ms`, `!rpm`, `!RPM`, `!%`.

8 Syntax für Zeiträume

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`

9 Befehlsausführung

Zabbix verwendet gemeinsame Funktionalität für externe Prüfungen, Benutzerparameter, system.run-Datenpunkte, benutzerdefinierte Alarmierungsskripte, 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.

Prüfung des Exit-Codes

Exit-Codes werden unter den folgenden Bedingungen geprüft:

- Nur für benutzerdefinierte Alarmierungsskripte, Remote-Befehle und Benutzerskripte, die auf Zabbix Server und Zabbix Proxy ausgeführt werden.
- Jeder Exit-Code, der sich von 0 unterscheidet, wird als Ausführungsfehler betrachtet.
- Inhalte von Standard Error und Standard Output bei fehlgeschlagenen Ausführungen werden gesammelt und sind im Frontend verfügbar (wo das Ausführungsergebnis angezeigt wird).
- Ein zusätzlicher Log-Eintrag 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 Log-Einträge für fehlgeschlagene Befehle/Skripte:

- Inhalte von Standard Error und Standard Output bei fehlgeschlagenen Ausführungen (falls vorhanden).
- "Process exited with code: N." (bei leerer Ausgabe und einem 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

10 Versionskompatibilität

Unterstützte Agenten

Um mit Zabbix 8.0 kompatibel zu sein, darf der Zabbix Agent nicht älter als Version 1.4 und nicht neuer als 8.0 sein.

Möglicherweise müssen Sie die Konfiguration älterer Agenten überprüfen, da sich einige Parameter geändert haben, zum Beispiel Parameter im Zusammenhang mit Protokollierung für Versionen vor 3.0.

Um die neueste Funktionalität, Metriken, verbesserte Leistung und den reduzierten Speicherverbrauch voll auszunutzen, verwenden Sie den neuesten unterstützten Agent.

Hinweise zu Windows XP

- Unter 32-Bit-Windows XP verwenden Sie keine Zabbix Agenten neuer als 6.0.x;
- Unter Windows XP/Server 2003 verwenden Sie keine Agent-Vorlagen, die neuer als Zabbix 4.0.x sind. Die neueren Vorlagen verwenden englische Leistungsindikatoren, die erst seit 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
Aktuell	Ja	Ja	Ja
Veraltet	Ja	Nein	Remote-Befehle (z. B. Shell-Skripte); Sofortige Prüfungen von Datenpunkt-Werten (d. h. *Jetzt ausführen*); Hinweis: Vorverarbeitungs-Tests mit einem echten Wert werden nicht unterstützt.
Nicht unterstützt	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.

12 Dynamische Linkbibliothek für Zabbix sender unter Windows

Übersicht

In einer Windows-Umgebung können Anwendungen Daten an den Zabbix Server/Proxy senden, indem sie die dynamische Linkbibliothek für Zabbix sender (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 beziehen.

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* die Option "No encryption" und unter *Packaging* die Option "Archive" auszuwählen. Laden Sie dann Zabbix Agent herunter (nicht Zabbix Agent 2).

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 Anweisungen).

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 linken Sie mit der Bibliothek zabbix_sender.lib.

Siehe auch

- Beispiel eines einfachen Zabbix-sender-Dienstprogramms, das mit der dynamischen Linkbibliothek für Zabbix sender implementiert wurde, um die Verwendung der Bibliothek zu veranschaulichen;
- Datei zabbix_sender.h für die Schnittstellenfunktionen der dynamischen Linkbibliothek für Zabbix sender. Diese Datei enthält eine Dokumentation, die den Zweck jeder Schnittstellenfunktion, ihre Argumente und den Rückgabewert erläutert.

13 Upgrade der Service-Überwachung

Übersicht In Zabbix 6.0 wurde die Funktionalität der Service-Überwachung erheblich überarbeitet (siehe Neuerungen in Zabbix 6.0.0 für die Liste der Änderungen).

Diese Seite beschreibt, wie Services und SLAs, die in früheren Zabbix-Versionen definiert wurden, während eines Upgrades auf Zabbix 6.0 oder neuer geändert werden.

Services In älteren Zabbix-Versionen hatten Services zwei Arten von Abhängigkeiten: weiche und harte. Nach einem Upgrade werden alle Abhängigkeiten gleich behandelt.

Wenn ein Service „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 Service „Child service" nach einem Upgrade zwei übergeordnete Services: „Parent service 1" und „Parent service 2".

Die Auslöser-basierte Zuordnung zwischen Problemen und Services wurde durch eine tag-basierte Zuordnung ersetzt. In Zabbix 6.0 und neuer verfügt das Service-Konfigurationsformular über den neuen Parameter *Problem tags*, mit dem ein oder mehrere Paare aus Tag-Name und Tag-Wert für den Problemabgleich angegeben werden können. Auslöser, die mit einem Service verknüpft waren, erhalten ein neues Tag `ServiceLink` : `<trigger ID>:<trigger name>` (der Tag-Wert wird auf 32 Zeichen gekürzt). Verknüpfte Services erhalten das `ServiceLink`-Problem-Tag mit demselben Wert.

Regeln zur Statusberechnung

Der „Statusberechnungsalgorithmus" wird anhand der folgenden Regeln aktualisiert:

- Nicht berechnen → Status auf OK setzen
- Problem, wenn mindestens ein untergeordneter Service ein Problem hat → Kritischster Status der untergeordneten Services
- Problem, wenn alle untergeordneten Services Probleme haben → Kritischster Status, wenn alle untergeordneten Services Probleme haben

SLAs Zuvor mussten SLA-Ziele für jeden Service separat definiert werden. Seit Zabbix 6.0 ist das SLA zu einer eigenständigen Entität geworden, die Informationen über den Service-Zeitplan, das erwartete Service Level Objective (SLO) und Ausfallzeiten enthält, die von der Berechnung ausgeschlossen werden sollen. Nach der Konfiguration kann ein SLA über Service-Tags mehreren Services zugewiesen werden.

Während eines Upgrades:

- Für jeden Service definierte identische SLAs werden gruppiert, und pro 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 Metrik in SLA-Berichten, wird für bestehende Services auf 01/01/2000 00:00 gesetzt.

14 Andere Probleme

Anmeldung und systemd

Wir empfehlen, einen *zabbix*-Benutzer als Systembenutzer zu erstellen, also ohne die Möglichkeit, sich anzumelden. Einige Benutzer ignorieren diese Empfehlung und verwenden dasselbe Konto zur Anmeldung (z. B. per SSH) an dem Host, auf dem Zabbix läuft. Dies kann beim Abmelden zum Absturz des Zabbix-Daemons führen. In diesem Fall erhalten Sie im Zabbix-Server-Log etwa Folgendes:

```
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-Log:

```
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=

Steuert, ob dem Benutzer gehörende System-V- und POSIX-IPC-Objekte entfernt werden sollen, wenn sich der
Benutzer vollständig abmeldet. Akzeptiert ein boolesches Argument. Falls aktiviert, darf der Benutzer nach
Beendigung der letzten Sitzung des Benutzers keine IPC-Ressourcen mehr verwenden. Dies umfasst System-V-Se
Shared Memory und Nachrichtenwarteschlangen sowie POSIX-Shared-Memory und Nachrichtenwarteschlangen. Beach
dass IPC-Objekte des root-Benutzers und anderer Systembenutzer von dieser Einstellung nicht betroffen sind
Standardwert ist "yes".
```

Für dieses Problem gibt es 2 Lösungen:

1. (empfohlen) Verwenden Sie das *zabbix*-Konto ausschließlich für Zabbix-Prozesse und erstellen Sie für andere Zwecke ein dediziertes 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.

Zabbix Frontend hinter Proxy verwenden

Wenn das Zabbix Frontend hinter einem Proxy-Server ausgeführt wird, muss der Cookie-Pfad in der Proxy-Konfigurationsdatei umgeschrieben werden, damit er dem per Reverse-Proxy weitergeleiteten Pfad entspricht. Siehe die folgenden Beispiele. Wenn der Cookie-Pfad nicht umgeschrieben wird, können bei Benutzern Autorisierungsprobleme auftreten, wenn sie versuchen, sich am Zabbix Frontend anzumelden.

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
# ..
```

16 Escaping-Beispiele

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.

21 Kurzanleitungen

Überblick

Dieser Abschnitt der Dokumentation enthält kurze Anleitungen zur Einrichtung von Zabbix für einige häufig benötigte Überwachungsziele.

Er wurde speziell für neue Zabbix-Benutzer konzipiert und kann als Wegweiser durch andere Abschnitte der Dokumentation verwendet werden, die die zur 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 unter Verwendung aktiver Prüfungen überwachen
- Websites mit Browser-Datenpunkten überwachen
- Website-Zertifikate mit Zabbix Agent 2 (passiv) überwachen
- Einen Netzwerk-Switch oder Router mit Zabbix überwachen

- Windows-Ereignisprotokolle unter Verwendung aktiver Prüfungen überwachen

1 Linux mit Zabbix Agent überwachen

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

Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihres Linux-Rechners zu aktivieren. Wenn Sie nach umfassenden Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Configuration 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.

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

Prüfen Sie Ihre Zabbix Server-Version:

```
zabbix_server -V
```

Installieren Sie den 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 vollständig 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 Datenanfrage. Der Zabbix Server oder Proxy fragt einige 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/von den Server(n) eine Liste von Datenpunkten zur unabhängigen 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 (active). 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 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 die IP-Adresse oder den DNS-Namen (und optional den Port) Ihres Zabbix-Servers zum Parameter `Server` hinzu. Zum Beispiel:

```
Server=192.0.2.0:10051
```

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

3. Starten Sie den Zabbix Agent neu.

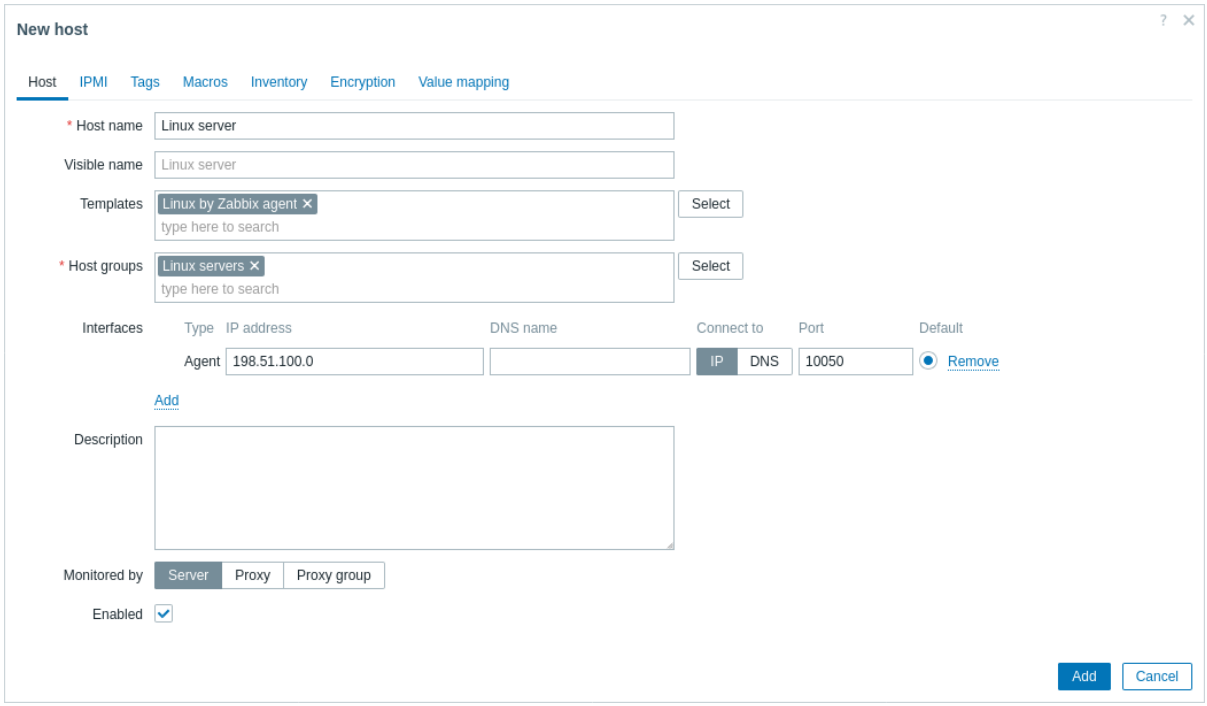
```
systemctl restart zabbix-agent
```

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 Host-Namen 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 Host-Gruppe ein oder wählen Sie eine aus (z. B. „Linux-Server“).
- 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 die IP-Adresse oder den DNS-Namen (und optional den Port) Ihres Zabbix-Servers zum Parameter ServerActive hinzu. Zum Beispiel:

```
ServerActive=192.0.2.0:10051
```

Der Zabbix Agent verwendet diese Adresse, um sich mit dem Trapper-Port des Zabbix-Servers (Standard: 10051) zu verbinden und die Konfigurationsdaten für aktive Prüfungen anzufordern.

3. Definieren Sie den Parameter Hostname, der mit dem Host-Namen übereinstimmen muss, der im Zabbix Frontend definiert wird. In unserem Beispiel ist das:

```
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. Genauer gesagt initiiert der Agent eine Verbindung zum Server und identifiziert sich mit dem Wert von Hostname. Der Server stellt dann die Überwachungskonfiguration für diesen Host bereit. Wenn diese Werte voneinander abweichen, erhält der Agent nicht die passende Konfiguration, was zu fehlenden Metriken oder Überwachungsproblemen führt.

4. Starten Sie den Zabbix Agent neu.

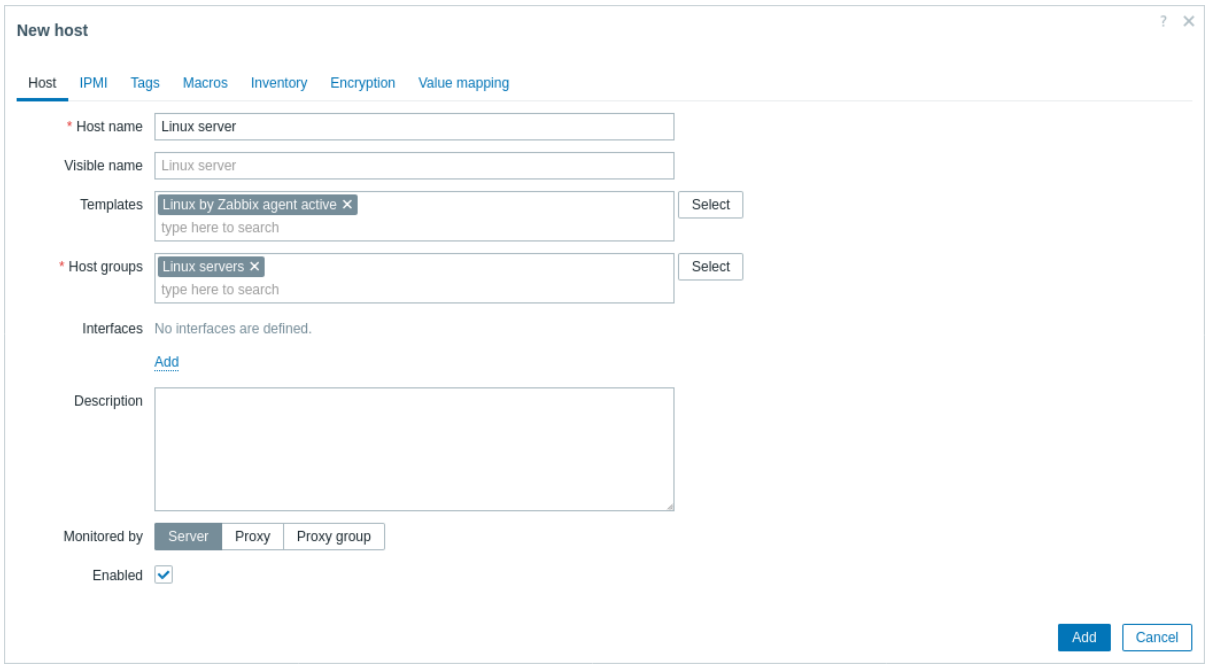
```
systemctl restart zabbix-agent
```

Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

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

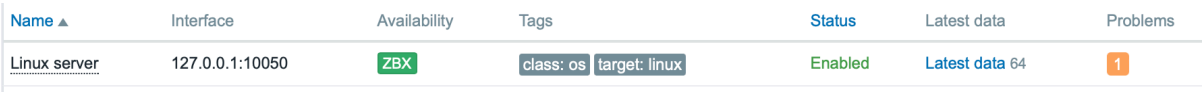
- Geben Sie im Feld *Host name* einen Host-Namen ein (z. B. „Linux server“), der mit dem zuvor in der Agent-Konfigurationsdatei definierten Wert des Parameters *Hostname* übereinstimmt.
- Geben Sie im Feld *Templates* die Vorlage „Linux by Zabbix agent active“ ein oder wählen Sie sie aus, die mit dem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



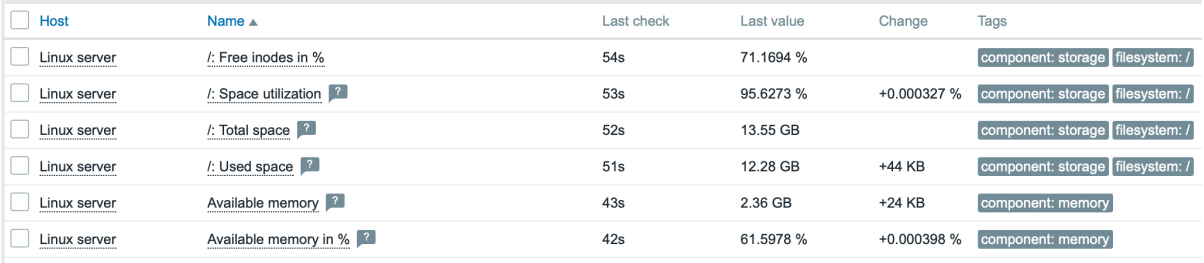
3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host repräsentiert den überwachten Linux-Rechner.

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

Um die erfassten Metriken anzuzeigen, öffnen Sie den Menüabschnitt *Monitoring->Hosts* menu section und klicken Sie neben dem Host auf *Latest data*.

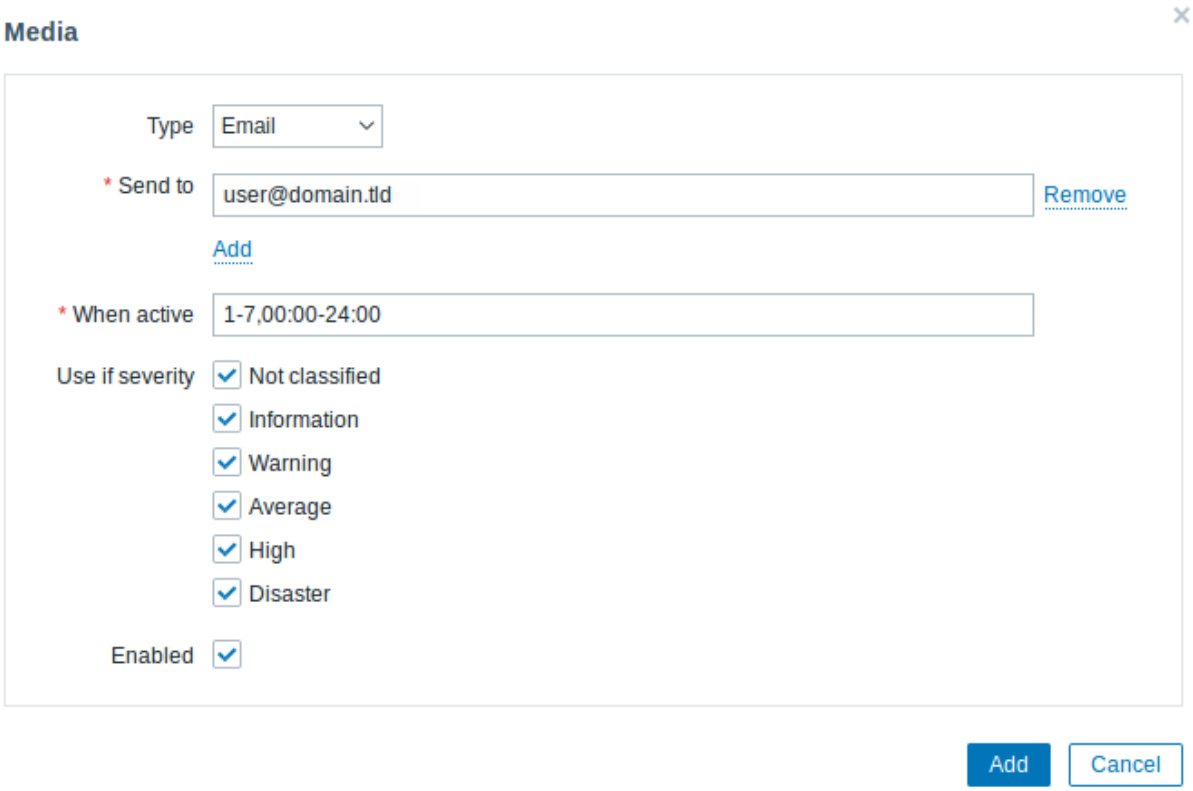


Dadurch wird eine Liste aller zuletzt vom Linux-Server-Host erfassten Metriken geöffnet.



Problembenachrichtigungen 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 fugen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung für Empfangen von Problembenachrichtigungen.

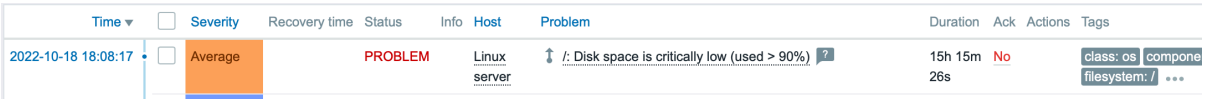
Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Testen Sie Ihre Konfiguration Unter Linux können Sie eine hohe CPU-Auslastung simulieren und dadurch einen Problemalarm erhalten, indem Sie Folgendes ausführen:

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

Möglicherweise müssen Sie mehrere md5sum-Prozesse ausführen, damit die CPU-Auslastung 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:

- Erstellen eines Datenpunkts - 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.
- Problemeskalationen - wie Sie mehrstufige Warnszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten gelöst wird, eine Nachricht an den Rechenzentrumsleiter senden).
- Installation aus Paketen - wie Sie Zabbix-Komponenten mithilfe offizieller RPM- und DEB-Pakete für verschiedene Linux-Distributionen installieren und dabei Zugriff auf die neuesten Funktionen und Fehlerbehebungen sicherstellen.

2 Monitor Windows with Zabbix agent

Einführung Diese Seite führt Sie durch die Schritte, die erforderlich sind, um die grundlegende Überwachung von Windows-Rechnern mit Zabbix zu starten.

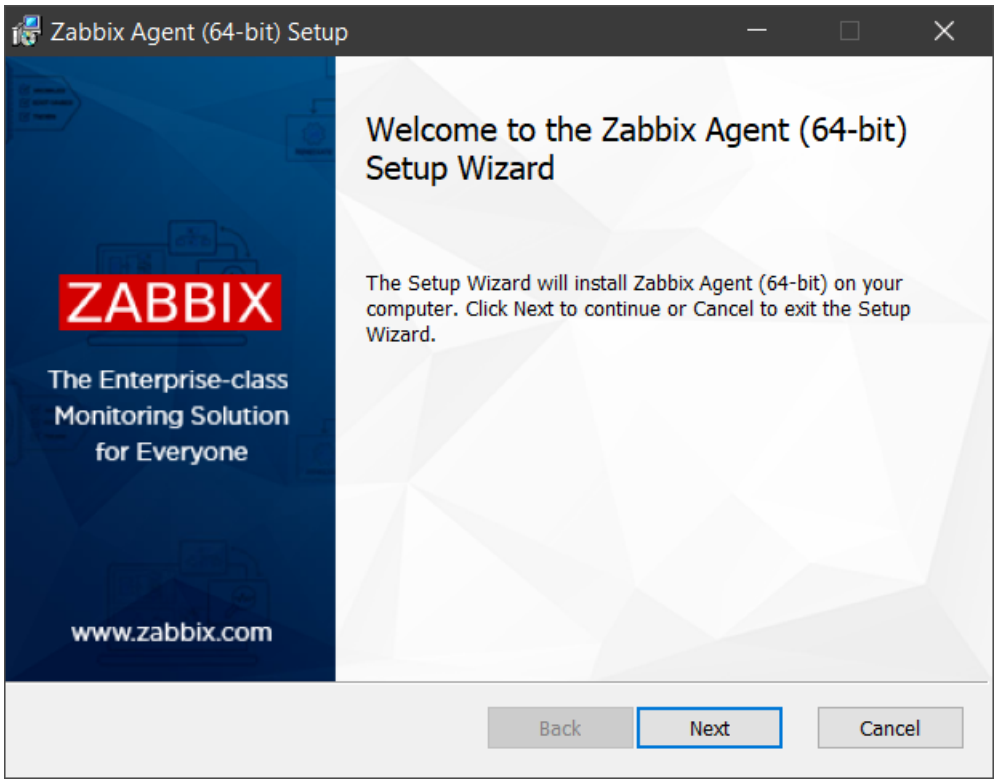
Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält den minimalen Satz an Schritten, der erforderlich ist, um die grundlegende Überwachung Ihres Windows-Rechners zu aktivieren.
Wenn Sie nach umfassenden Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, siehe den Abschnitt Configuration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Installationsanleitung fortfahren, müssen Sie Zabbix Server und 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 das Monitoring konfigurieren Der Zabbix Agent kann Metriken im aktiven oder passiven Modus erfassen (gleichzeitig).

> **Note:**
> Eine passive Prüfung ist eine einfache Datenanfrage. Der Zabbix Server oder Proxy fragt einige 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/von den Server(n) eine Liste von Datenpunkten zur unabhängigen Verarbeitung abrufen und die Daten anschließend gesammelt zurücksenden. Weitere Informationen finden Sie unter Passive and active agent checks.

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

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

Passive Prüfungen Zabbix frontend

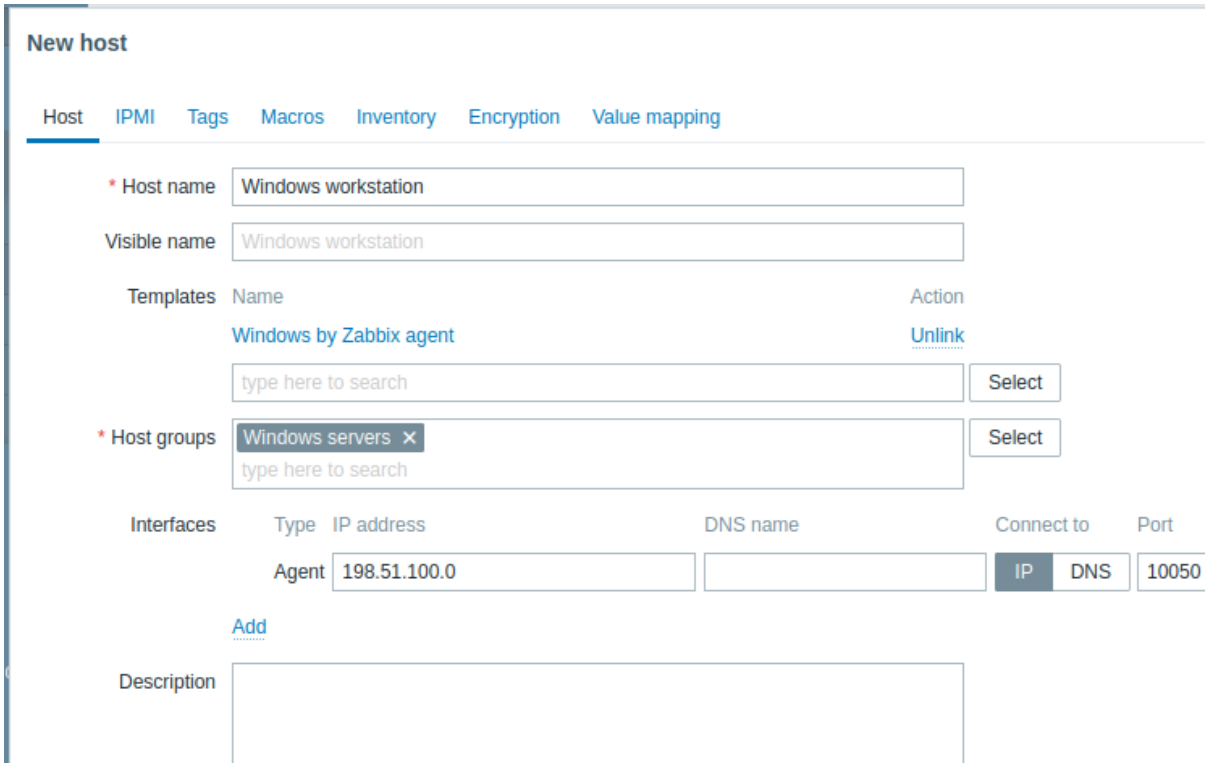
1. Log into Zabbix frontend.

2. Create a host in Zabbix web interface.

This host will represent your Windows machine.

3. In the *Interfaces* parameter, add *Agent* interface and specify the IP address or DNS name of the Windows machine where the agent is installed.

4. In the *Templates* parameter, type or select *Windows by Zabbix agent*.



Zabbix agent

For passive checks Zabbix agent needs to know the IP address or DNS name of Zabbix server. If you have provided correct information during the agent installation, the configuration file is already updated. Otherwise, you need to manually specify it. Go to the `C:\Program files\Zabbix Agent` folder, open the file *zabbix_agentd.conf* and add the IP/DNS of your Zabbix server to the *Server* parameter.

Example:

```
Server=192.0.2.22
```

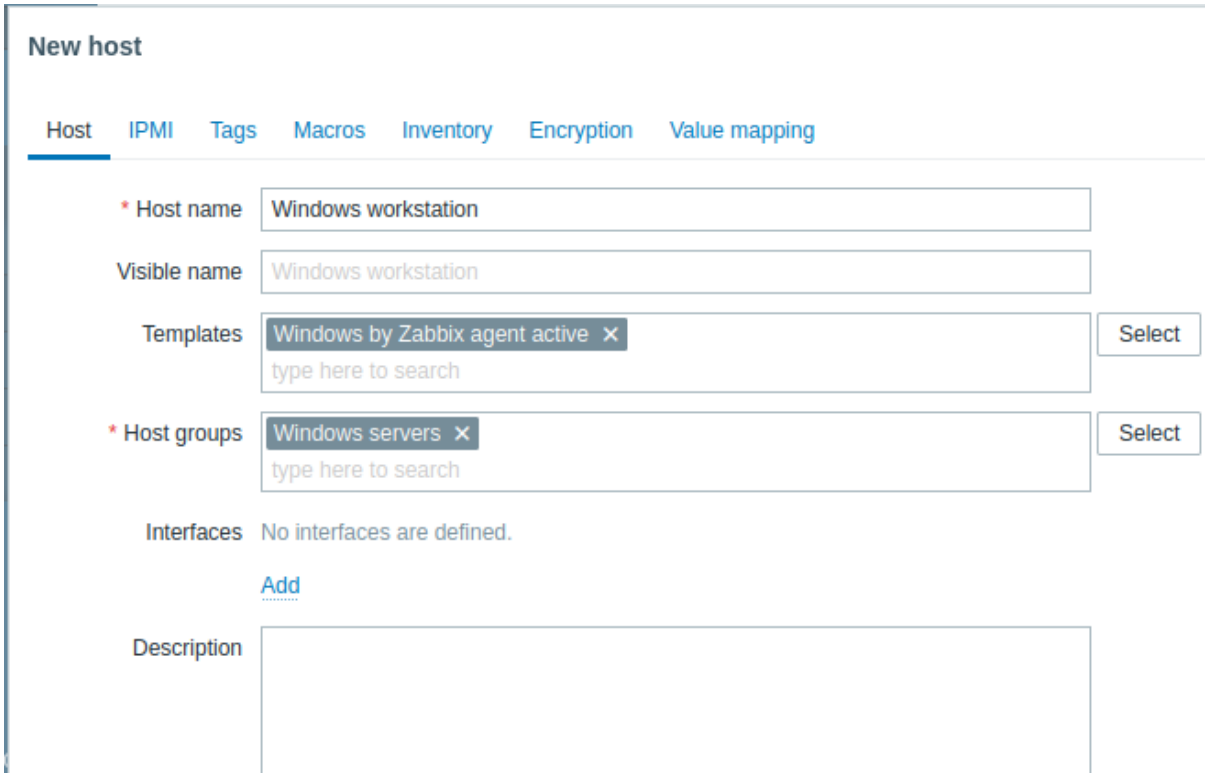
Aktive Prüfungen Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

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

Dieser Host wird Ihren Windows-Rechner repräsentieren.

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



Zabbix agent

In the `C:\Program files\Zabbix Agent` folder open the file *zabbix_agentd.conf* and add:

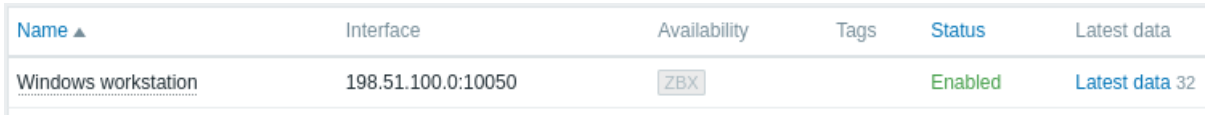
- The name of the host you created in Zabbix web interface to the *Hostname* parameter.
- The IP address or DNS name of your Zabbix server machine to the *ServerActive* parameter (might be prefilled if you have provided it during Zabbix agent setup).

Example:

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

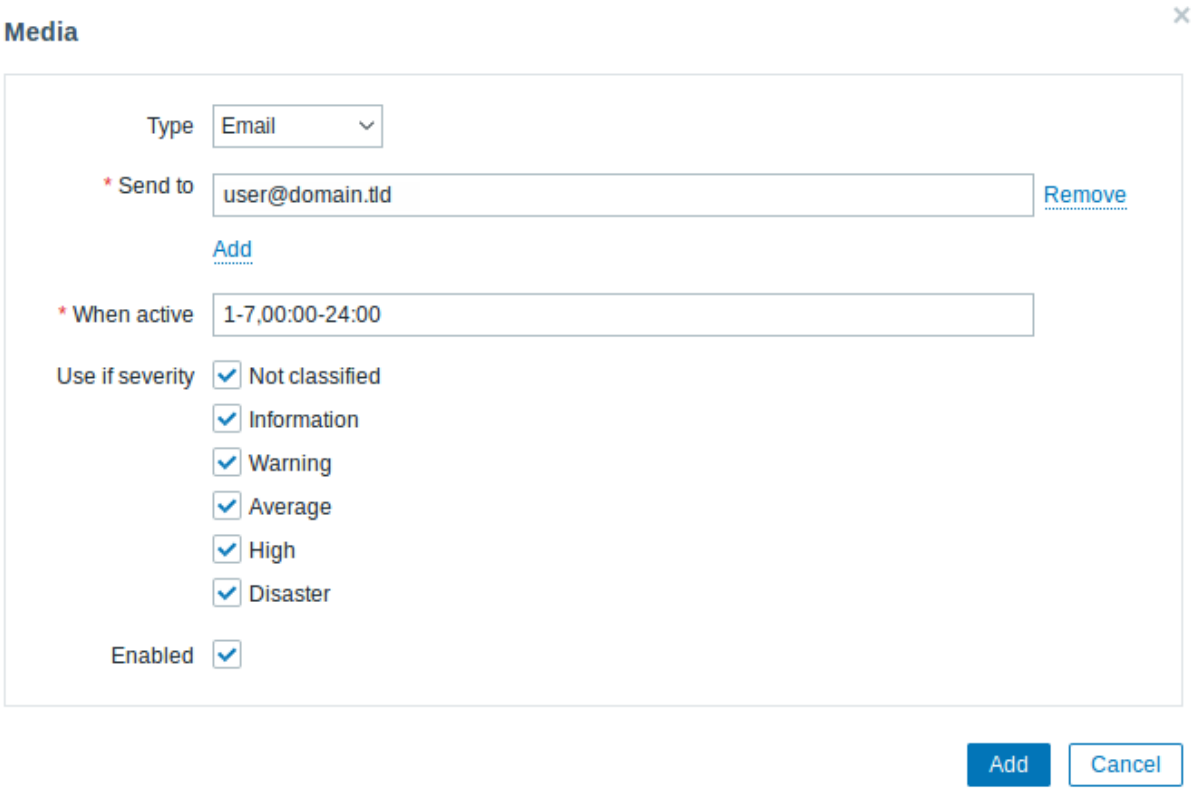
Gesammelte Metriken anzeigen Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix Ihren Windows-Rechner bereits.

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



Problemwarnungen 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 für Empfangen von Problembenachrichtigungen.

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

Note:
Unter Windows können Sie das Dienstprogramm CpuStres verwenden, um eine hohe CPU-Auslastung zu simulieren und dadurch eine Problemwarnung zu erhalten.

Siehe auch:

- Erstellen eines Datenpunkts - wie Sie die Überwachung zusätzlicher Metriken starten (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.
- Problemeskalationen - wie Sie mehrstufige Warnszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten behoben wird, eine Nachricht an den Rechenzentrum-sleiter senden).

3 Apache über HTTP überwachen

Einführung Diese Seite zeigt eine schnelle und einfache Möglichkeit, die Überwachung eines Apache-Webservers zu starten, ohne zusätzliche Software zu installieren.

Für wen diese Anleitung gedacht ist

Diese Anleitung 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 Möglichkeiten zur tiefgehenden Anpassung suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Configuration 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:

Führen Sie auf einem RHEL-basierten System Folgendes aus:

```
httpd -v
```

Führen Sie auf Debian/Ubuntu Folgendes aus:

```
apache2 -v
```

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

Führen Sie auf einem RHEL-basierten System Folgendes aus:

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

Führen Sie auf Debian/Ubuntu Folgendes aus:

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

Wenn status_module nicht in der Liste angezeigt wird, aktivieren Sie das Modul mit folgendem Befehl:

Führen Sie auf einem RHEL-basierten System Folgendes aus:

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

Führen Sie auf Debian/Ubuntu Folgendes 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

Führen Sie auf einem RHEL-basierten System Folgendes aus:

```
sudo systemctl restart httpd
```

Führen Sie auf Debian/Ubuntu Folgendes 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.

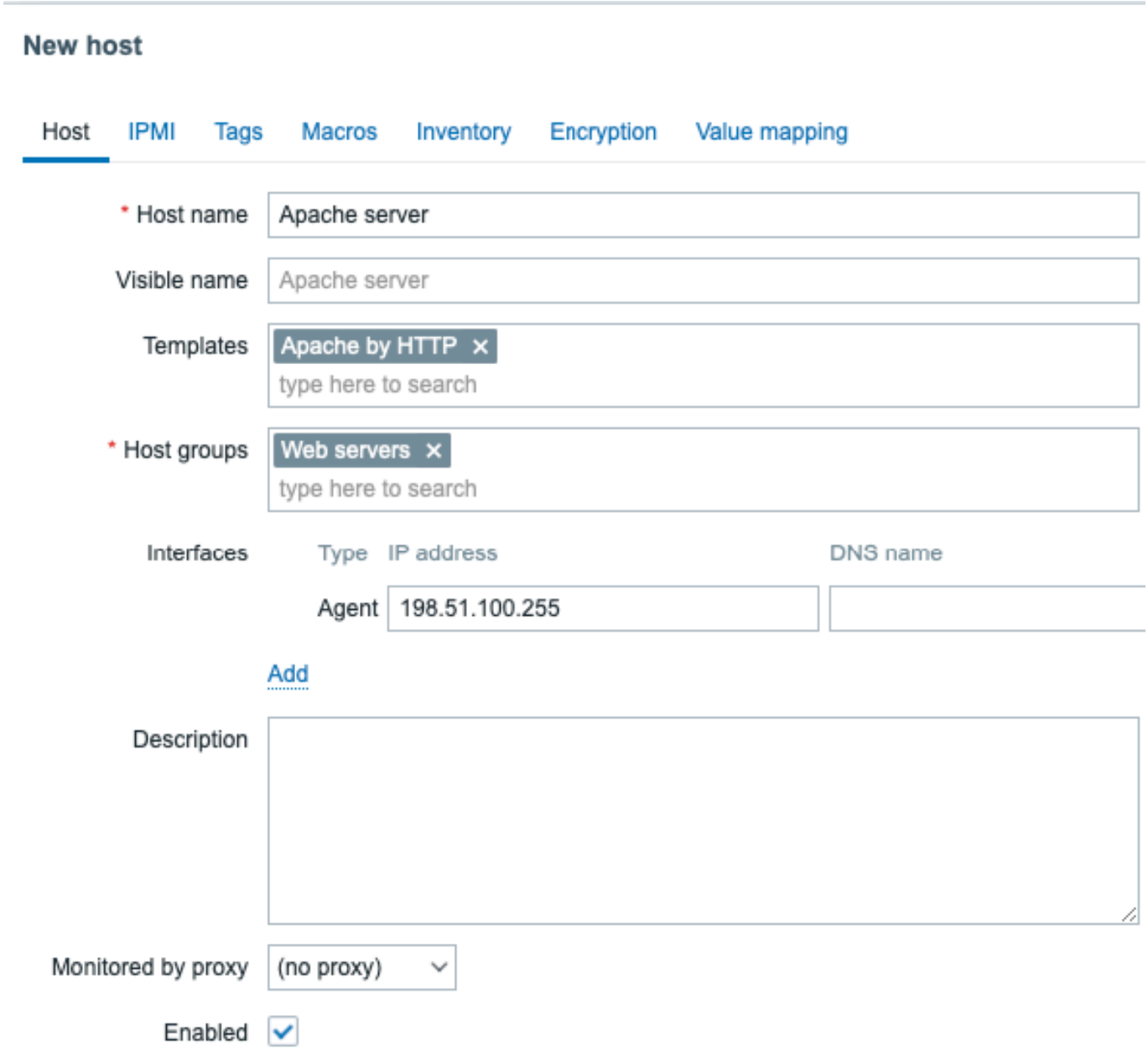
Zabbix für die Überwachung konfigurieren 1. Melden Sie sich im Zabbix Frontend an.

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

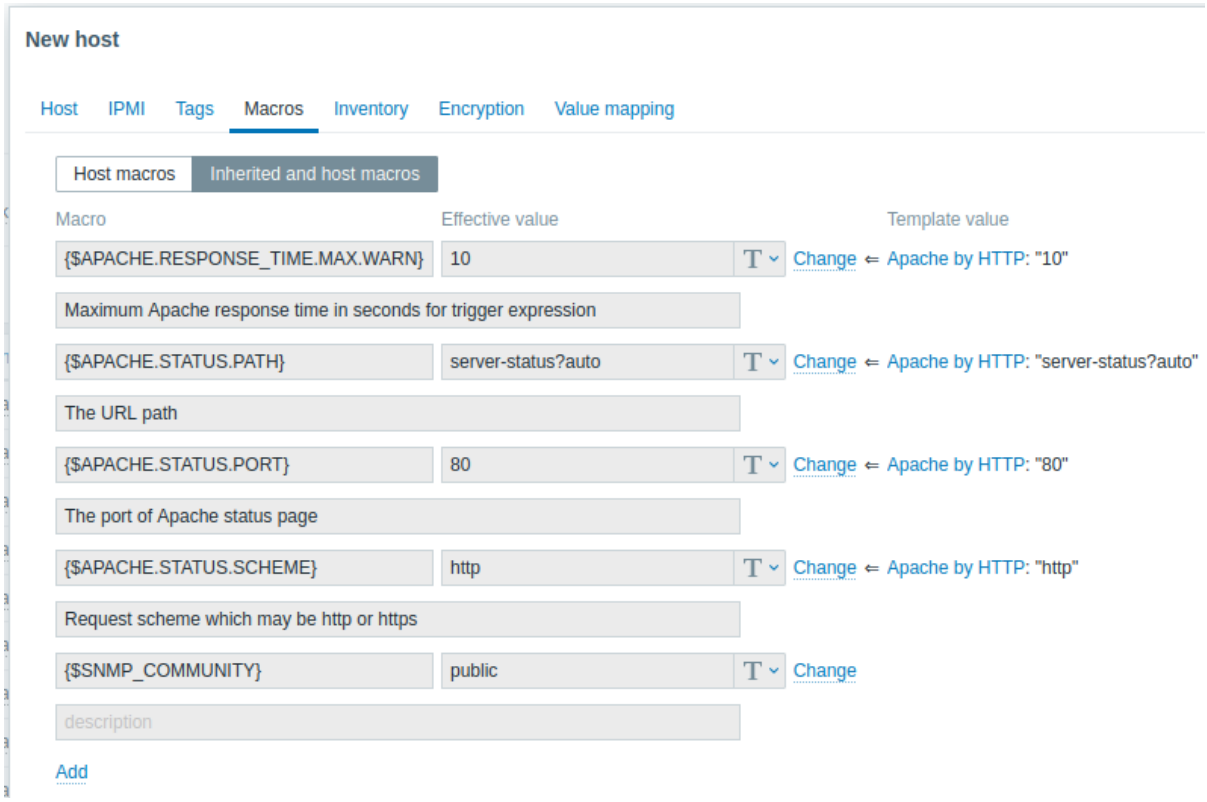
Dieser Host wird Ihren Apache-Server repräsentieren.

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

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



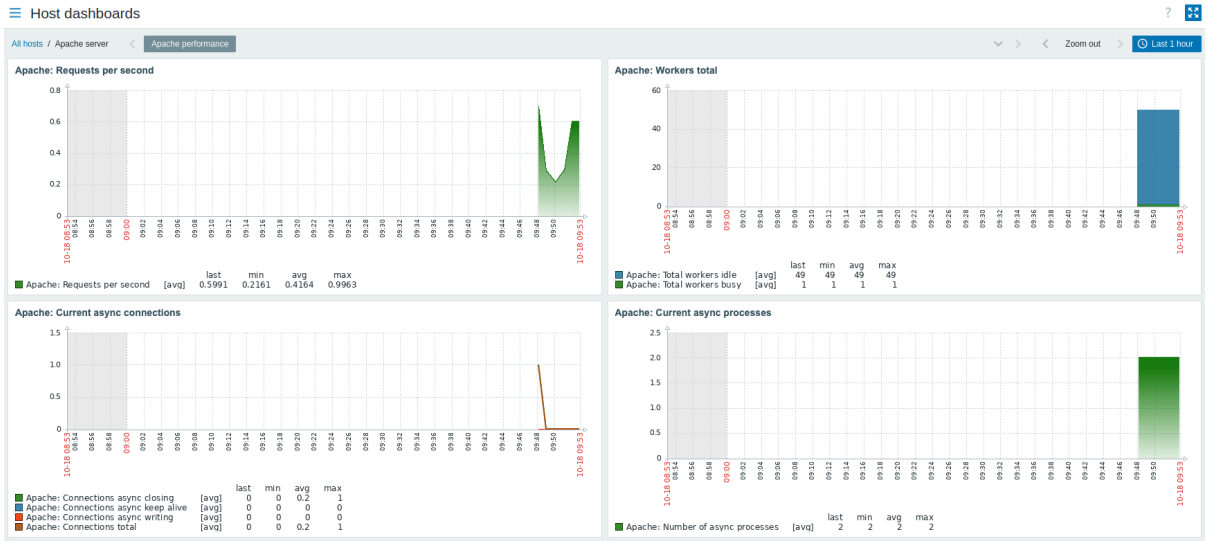
5. Wechseln Sie zur Registerkarte **Macros** und wählen Sie den Modus *Inherited and host macros*. Prüfen Sie, ob die Werte der Makros {$APACHE.STATUS.PORT} und {$APACHE.STATUS.SCHEME} zu Ihren Installationseinstellungen passen. Standardmäßig ist der Port 80 und das Schema http. Ändern Sie die Makrowerte,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



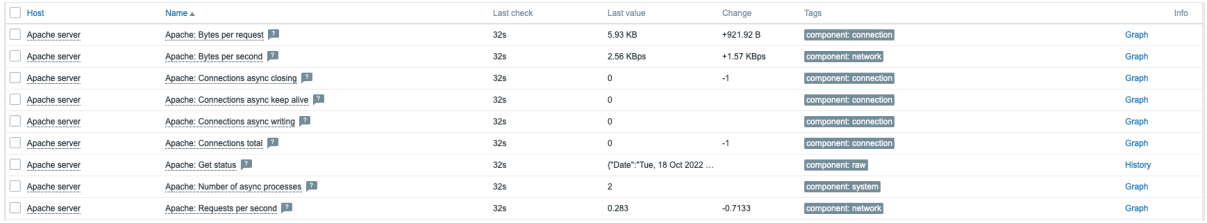
Gesammelte Metriken anzeigen Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren Apache-Webserver.

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

Dadurch gelangen Sie zum Dashboard des Hosts mit den wichtigsten Metriken, die von der Apache-Seite /server-status erfasst wurden.

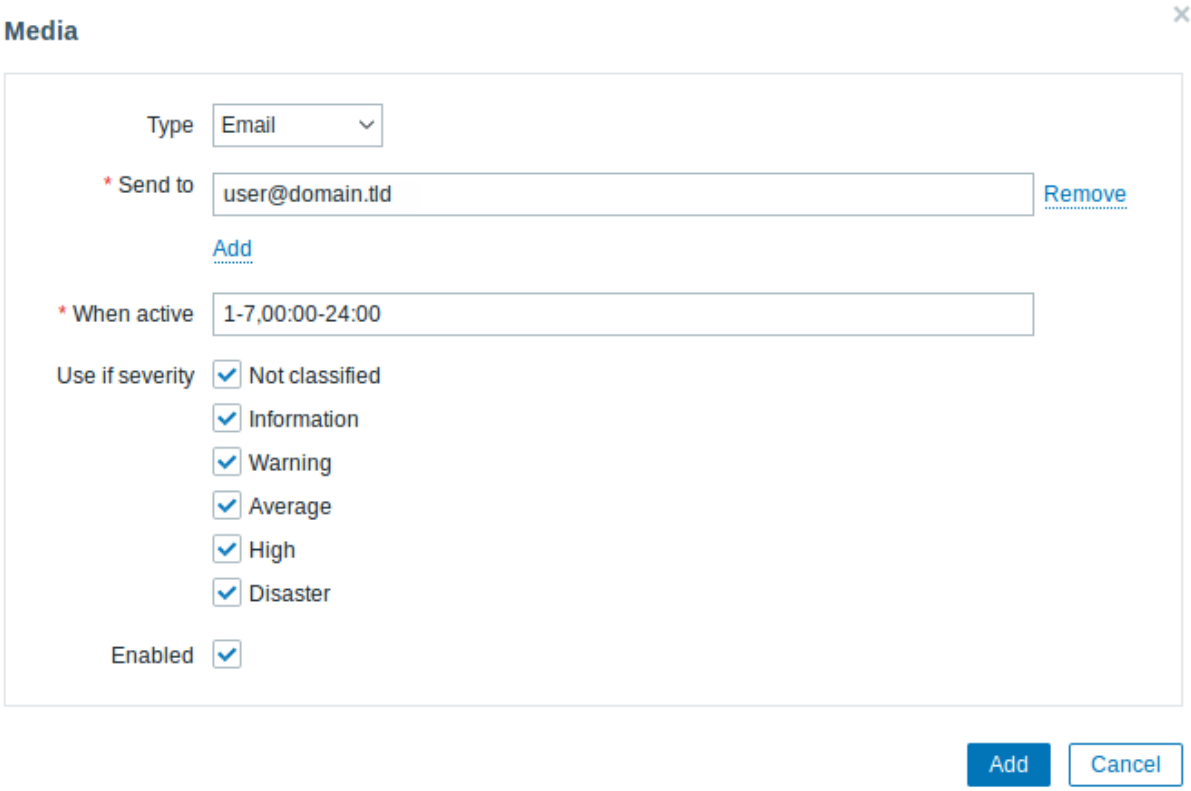


Alternativ können Sie unter *Monitoring->Hosts* auf *Latest data* klicken, um alle zuletzt erfassten Metriken in einer Liste anzuzeigen.



Problembenachrichtigungen einrichten Zabbix kann Sie mit verschiedenen Methoden über ein Problem in Ihrer Infrastruktur benachrichtigen.
Diese Anleitung enthält 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 für Empfangen von Problembenachrichtigungen.

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

Testen Sie Ihre Konfiguration So simulieren Sie ein echtes Problem und erhalten eine Test-Benachrichtigung zu einem Problem:

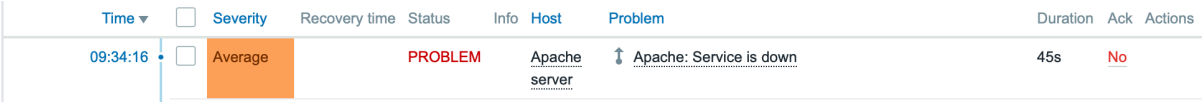
1. Öffnen Sie die Konfiguration des Hosts *Apache server* in Zabbix.

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

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 jetzt keine Verbindung 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 Apache weiter zu überwachen.

Siehe auch:

- Härtung des Webservers - empfohlene Einstellungen für mehr Sicherheit des Webservers.
- Erstellen eines Datenpunkts - wie Sie mit der Überwachung zusätzlicher Metriken beginnen.
- HTTP-Datenpunkte - wie Sie benutzerdefinierte Metriken mit HTTP-Agent überwachen.
- Problemeskalationen - wie Sie mehrstufige Warnszenarien erstellen
  (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten behoben wird, eine Nachricht
  an den Rechenzentrumsleiter senden).

4 MySQL mit Zabbix Agent 2 überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um die grundlegende Überwachung eines MySQL-Servers zu starten.

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 bietet jedoch auch Anleitungen für die anderen Ansätze, sodass Sie gerne den Ansatz wählen können, der Ihren Anforderungen am besten entspricht.

Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung eines MySQL-Servers zu aktivieren. Wenn Sie nach umfassenden Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Configuration im Zabbix-Handbuch.

Voraussetzungen

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

Abhängig von Ihrer Umgebung können einige Schritte in dieser Anleitung leicht abweichen. Diese Anleitung 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 der erforderlichen Zugriffsstufe (den Benutzer „zabbix“, der bei der Installation von Zabbix erstellt wurde), dieser Benutzer hat jedoch mehr Berechtigungen als für eine einfache Überwachung notwendig sind (Berechtigungen zum DROP von Datenbanken, zum DELETE von Einträgen aus Tabellen usw.). Daher muss ein MySQL-Benutzer erstellt werden, der *nur* zur Überwachung des MySQL-Servers dient.

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;
```

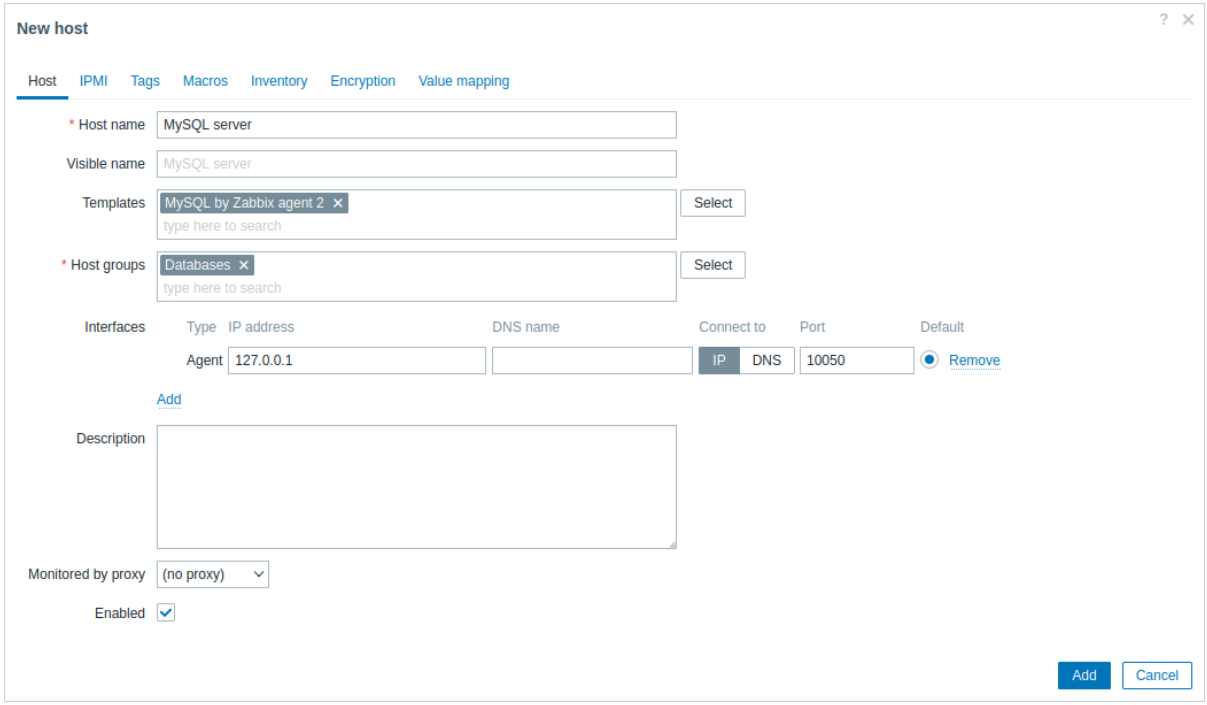
Sobald der Benutzer erstellt wurde, können Sie mit dem nächsten Schritt fortfahren.

Zabbix Frontend konfigurieren

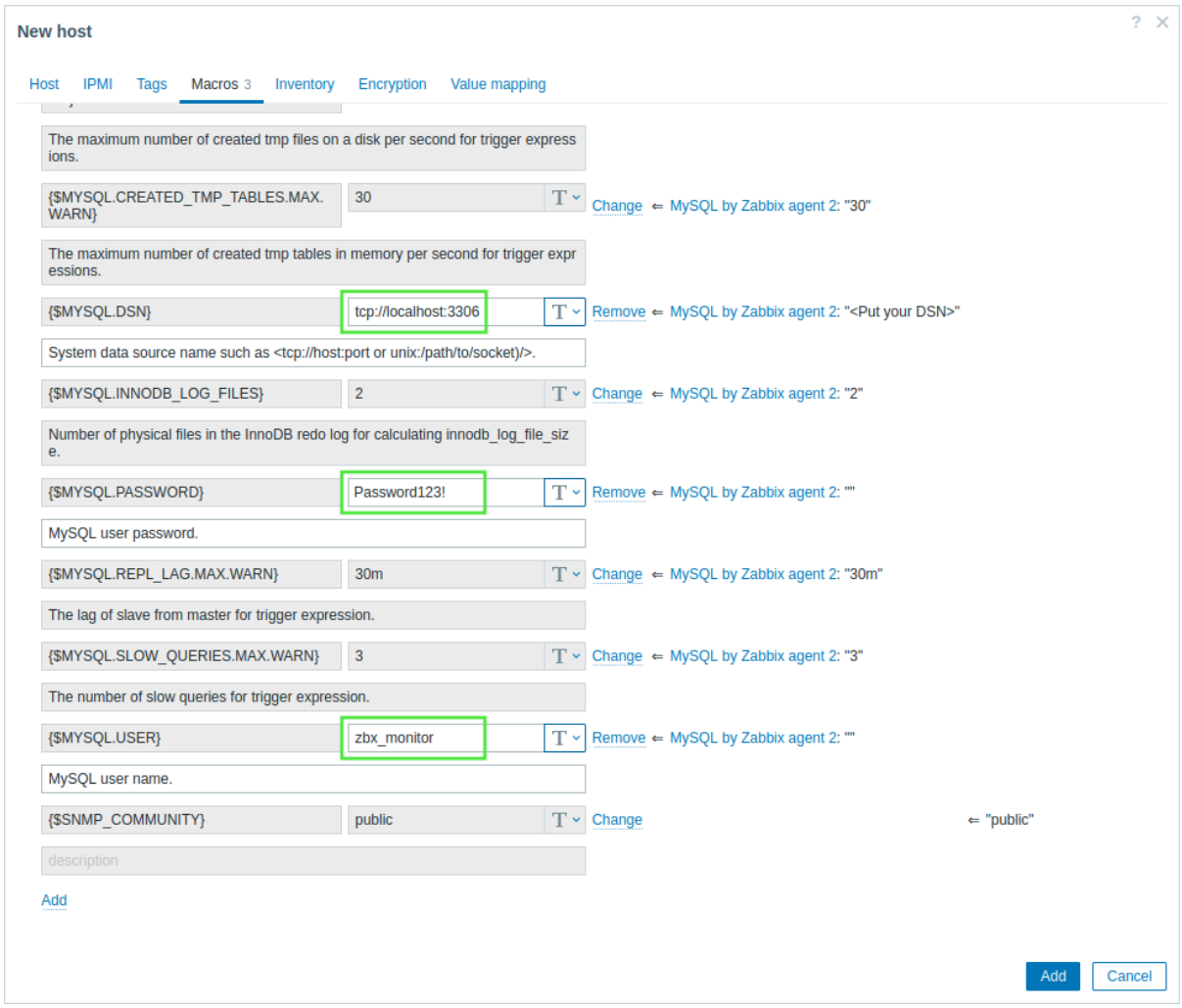
1. Melden Sie sich im Zabbix Frontend an.

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

- Geben Sie im Feld *Host name* einen Host-Namen 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 Host-Gruppe ein oder wählen Sie eine 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 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:
  - {$MYSQL.DSN} - legen Sie die Datenquelle des MySQL-Servers fest (die Verbindungszeichenfolge einer benannten Sitzung aus der Konfigurationsdatei des MySQL-Zabbix-Agent-2-Plugins). In diesem Leitfaden wird die Standard-Datenquelle „tcp://localhost:3306“ verwendet, um einen MySQL-Server zu überwachen, der auf demselben Rechner wie Zabbix Server und Zabbix Agent 2 installiert ist.
  - {$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.

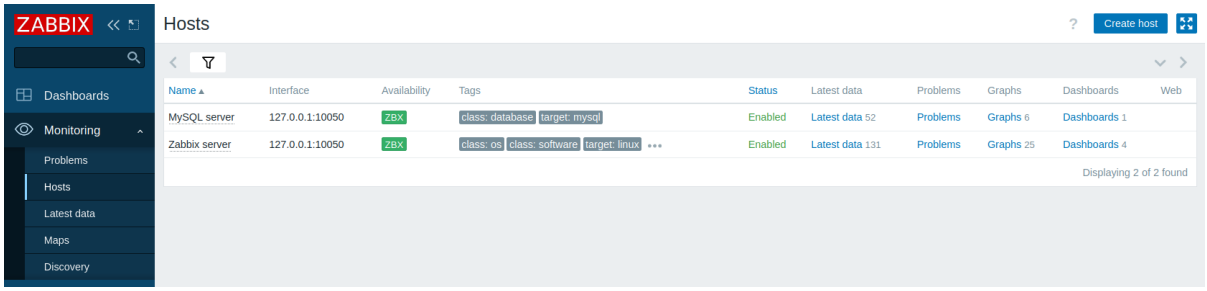


3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host wird Ihren MySQL-Server repräsentieren.

Erfasste Metriken anzeigen

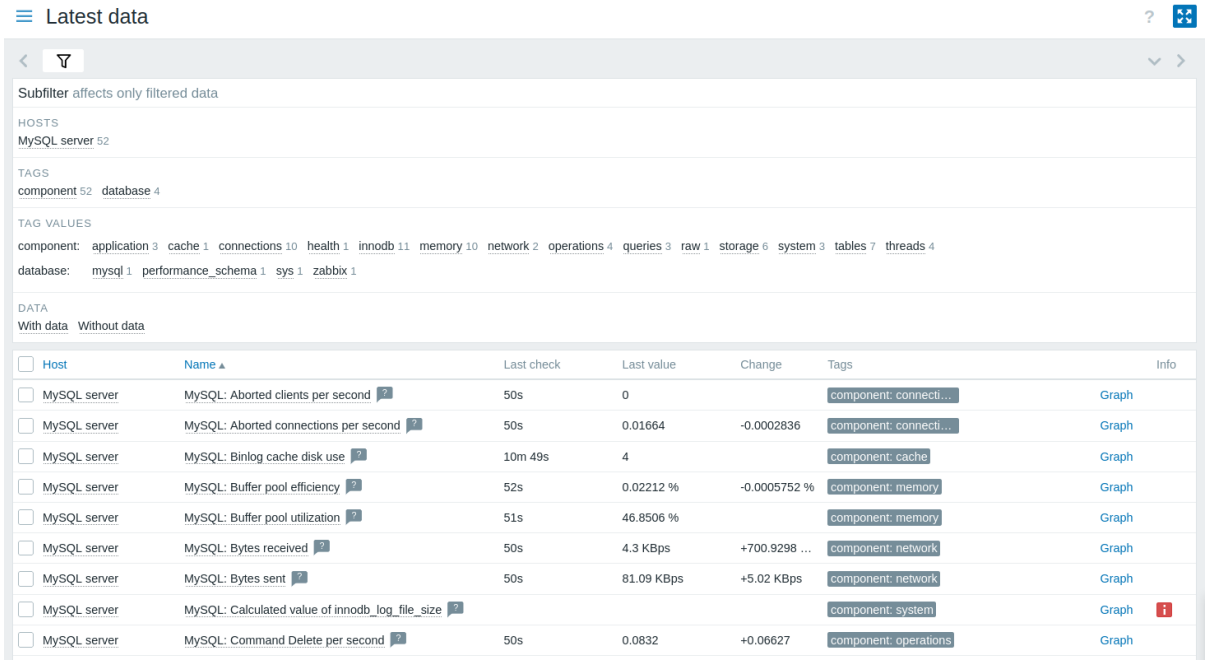
Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren MySQL-Server.

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



Dadurch gelangen Sie zum Host-Dashboard (auf Vorlagenebene konfiguriert) mit den wichtigsten Metriken, die vom MySQL-Server erfasst wurden.

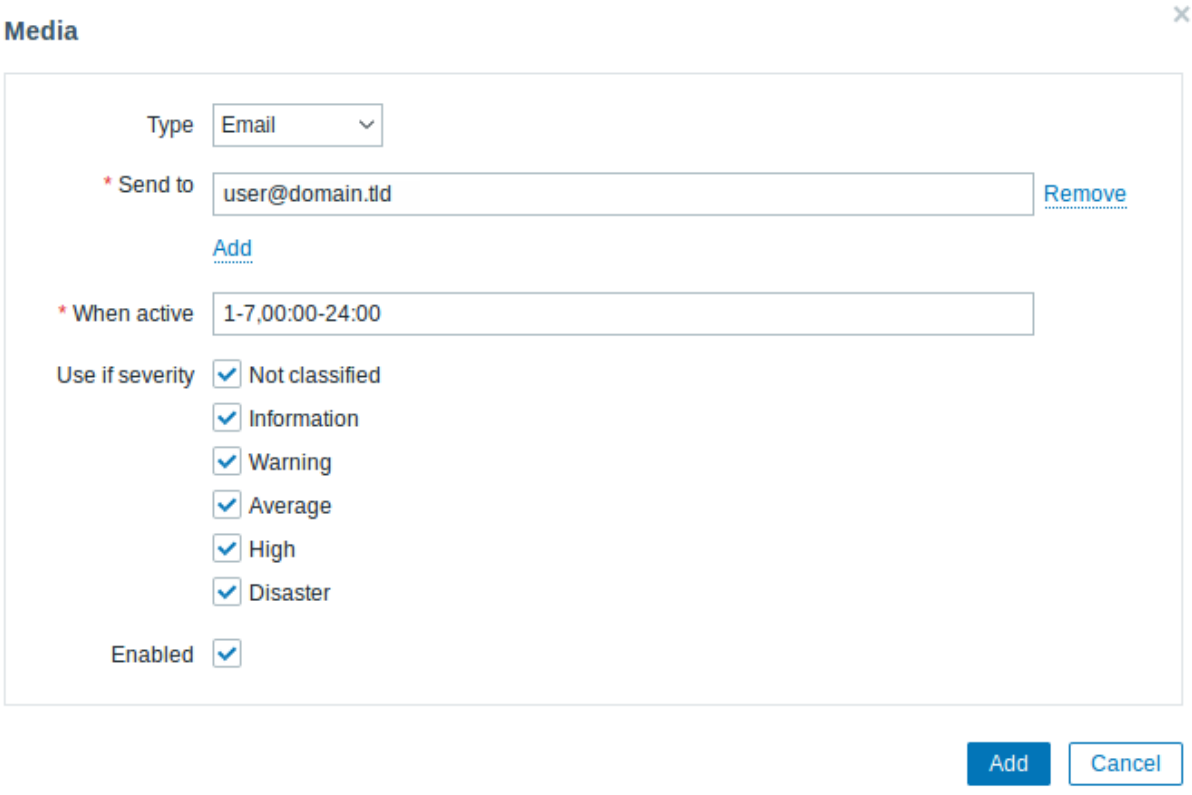
Alternativ können Sie im Menübereich *Monitoring → Hosts* auf *Latest data* klicken, um alle zuletzt erfassten Metriken in einer Liste anzuzeigen. Beachten Sie, dass für den Datenpunkt *MySQL: Calculated value of innodb_log_file_size* voraussichtlich keine Daten vorhanden sind, da der Wert aus den Daten 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 für 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.

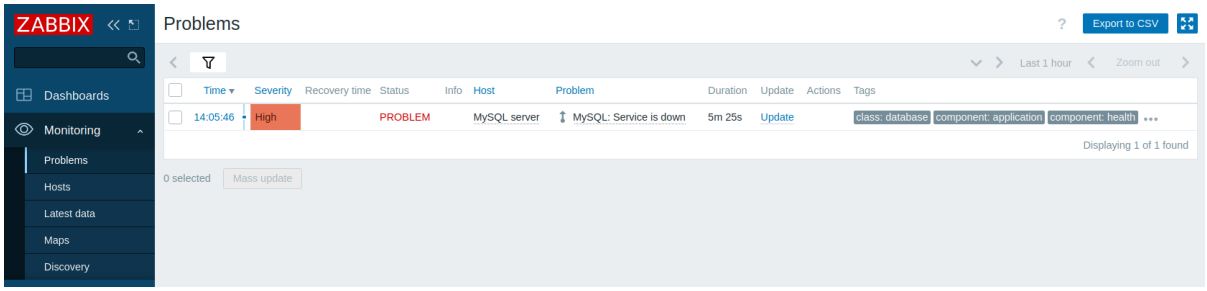
1. Öffnen Sie die Konfiguration Ihres MySQL-Server-Hosts in Zabbix.

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

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

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

5. In wenigen Augenblicken wird Zabbix das Problem „MySQL: Service is down“ erkennen, da keine Verbindung zum MySQL-Server hergestellt werden kann. Das Problem wird unter *Monitoring → Problems* angezeigt.



Wenn Benachrichtigungen konfiguriert sind, erhalten Sie außerdem eine Problembenachrichtigung.

6. Ändern 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 den Zabbix Agent oder den Standard Open Database Connectivity (ODBC) verwenden. Obwohl die Verwendung von Zabbix Agent 2 empfohlen wird, gibt es möglicherweise Umgebungen, 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: Der 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 der Überwachung eines MySQL-Servers mit Zabbix Agent 2 ähneln, gibt es einige wesentliche Unterschiede: Sie müssen den Zabbix Agent oder ODBC konfigurieren, um einen MySQL-Server überwachen zu können. Die folgenden Anweisungen führen Sie durch diese **Unterschiede**.

MySQL mit Zabbix Agent überwachen

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

Sobald 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 den Zabbix Agent so konfigurieren, dass er eine Verbindung zum MySQL-Server herstellen und diesen überwachen kann.
Dazu gehört die Konfiguration mehrerer Benutzerparameter zur Ausführung benutzerdefinierter Agent-Prüfungen sowie die Bereitstellung der erforderlichen Zugangsdaten für den Zabbix Agent, damit er sich als zuvor erstellter Benutzer „zbx_monitor“ mit dem MySQL-Server verbinden kann.

Zabbix Agent konfigurieren

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

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

> **Attention:**
> Das Verzeichnis für zusätzliche Konfigurationen des Zabbix Agent sollte sich im selben Verzeichnis befinden wie Ihre Zabbix-Agent-Konfigurationsdatei (*zabbix_agentd.conf*). Je nach Betriebssystem und Zabbix-Installation kann dieses Verzeichnis einen anderen Speicherort haben als in dieser Anleitung angegeben. Die Standardspeicherorte finden Sie im Parameter `Include` in der Zabbix-Agent-Konfigurationsdatei.

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

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

```
vi template_db_mysql.conf
```

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

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

```
systemctl restart zabbix-agent
```

Sobald 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 existiert, müssen Sie es erstellen; Standard: */var/lib/zabbix*).

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

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

```
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

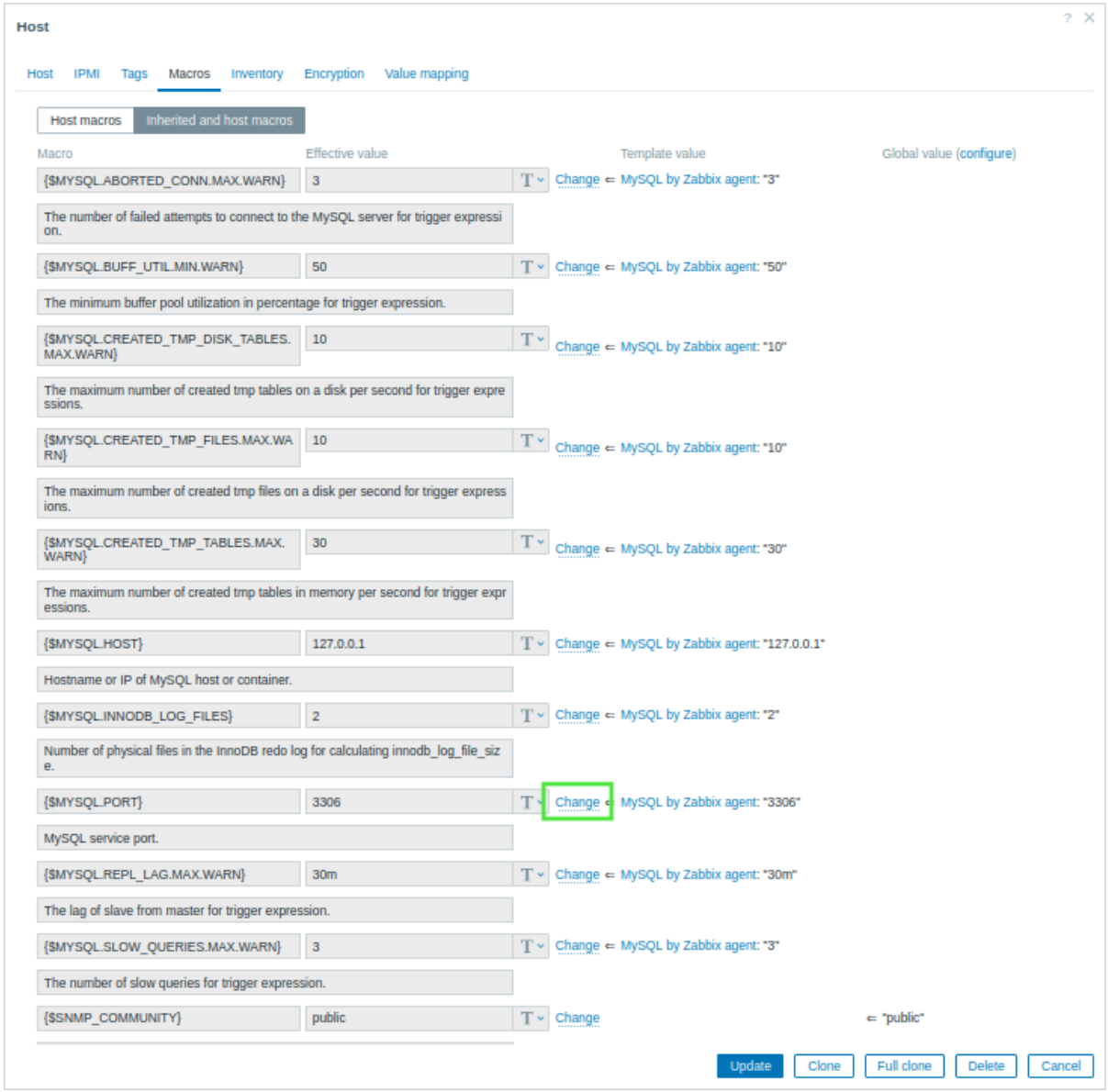
Um das Zabbix Frontend zu konfigurieren, folgen Sie den Anweisungen im Abschnitt *Zabbix Frontend konfigurieren* mit den folgenden Anpassungen:

- 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.

Sobald Sie das Zabbix Frontend konfiguriert haben, können Sie erfasste Metriken anzeigen und Problembenachrichtigungen einrichten.

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

- Klicken Sie im Abschnitt *Geerbte Makros und Host-Makros* der Host-Konfiguration des MySQL-Servers neben dem Makrowert {$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 Zabbix Server und Zabbix Frontend herunterladen und installieren.

Sobald 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 gehören die Installation einer der am häufigsten verwendeten Open-Source-Implementierungen der ODBC-API – unixODBC – und eines unixODBC-Treibers sowie die Bearbeitung der ODBC-Treiberkonfigurationsdatei.

ODBC konfigurieren

1. Installieren Sie unixODBC. Die empfohlene Methode zur Installation von unixODBC ist die Verwendung der Standard-Paket-Repositories 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 sich in der Datei *odbcinst.ini* befindet. 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 Adresse des MySQL-Servers, um einen MySQL-Server zu überwachen, der auf demselben Rechner wie der ODBC-Treiber installiert ist. Beachten Sie den Namen der Datenquelle (DSN) "test", der benötigt wird, wenn Sie das 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; sie wird mit dem Host verknüpft.
- Die Konfiguration von *Schnittstellen* ist nicht erforderlich.
- Der Makrowert {$MYSQL.DSN} im Abschnitt *Geerbte Makros und Host-Makros* der Host-Konfiguration des MySQL-Servers sollte auf den DSN-Namen aus der Datei *odbc.ini* gesetzt werden.

Sobald Sie das Zabbix Frontend konfiguriert haben, können Sie erfasste Metriken anzeigen, Problembenachrichtigungen einrichten und Ihre Konfiguration testen.

Siehe auch

- Erstellen eines Datenpunkts – wie Sie die Überwachung zusätzlicher Metriken starten.
- Problemeskalationen – wie Sie mehrstufige Warnszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten behoben wird, eine Nachricht an den Rechenzentrumsleiter senden).
- ODBC-Überwachung – wie Sie ODBC auf anderen Linux-Distributionen einrichten und mit ODBC zusätzliche datenbankbezogene Metriken überwachen.
- 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*.

5 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 diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende VMware-Überwachung zu aktivieren. Wenn Sie nach Möglichkeiten zur tiefgehenden Anpassung suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Überwachung virtueller Maschinen 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.

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

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

Zabbix Server konfigurieren

Um VMware zu überwachen, müssen die Zabbix-Prozesse des *vmware collector* aktiviert werden. Weitere Informationen dazu, wie die VMware-Überwachung durchgeführt wird, finden Sie unter Überwachung virtueller Maschinen.

1. Öffnen Sie die Zabbix-Server-Konfigurationsdatei.

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

2. Suchen Sie den Parameter `StartVMwareCollectors` in der Zabbix-Server-Konfigurationsdatei und setzen Sie ihn auf 2 oder mehr (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
```

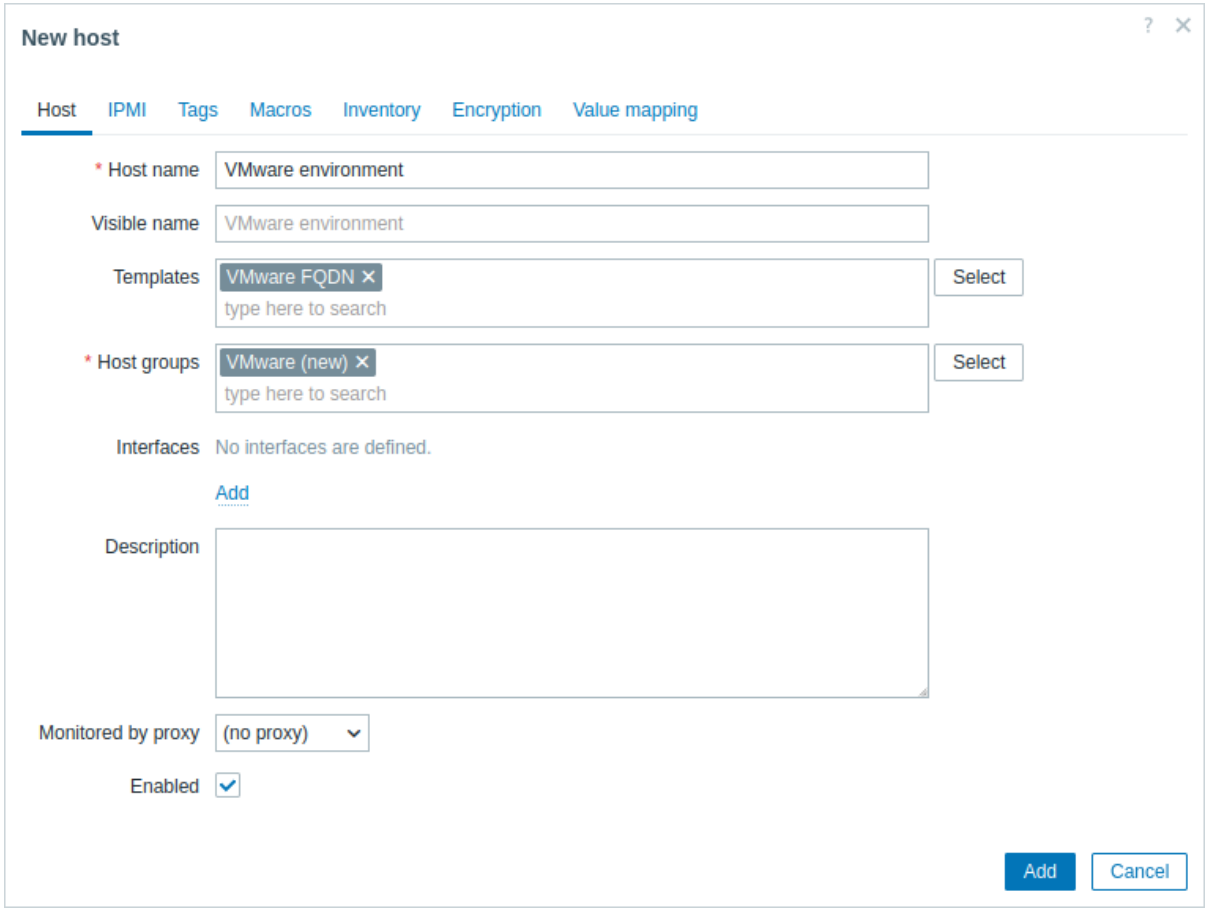
Sobald die Prozesse des *vmware collector* gestartet wurden, fahren Sie mit dem nächsten Schritt fort.

Zabbix Frontend konfigurieren

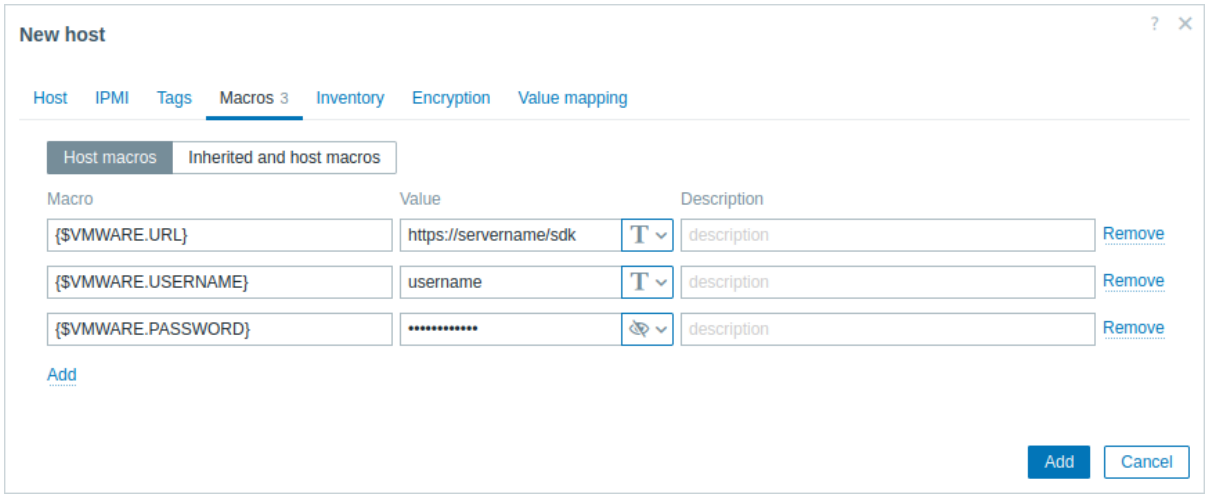
1. Melden Sie sich im Zabbix Frontend an.

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

- Geben Sie im Feld *Host name* einen Host-Namen 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 Überwachung virtueller Maschinen.
- Geben Sie im Feld *Host groups* eine Host-Gruppe ein oder wählen Sie sie aus (zum Beispiel eine neue Host-Gruppe „VMware“).



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



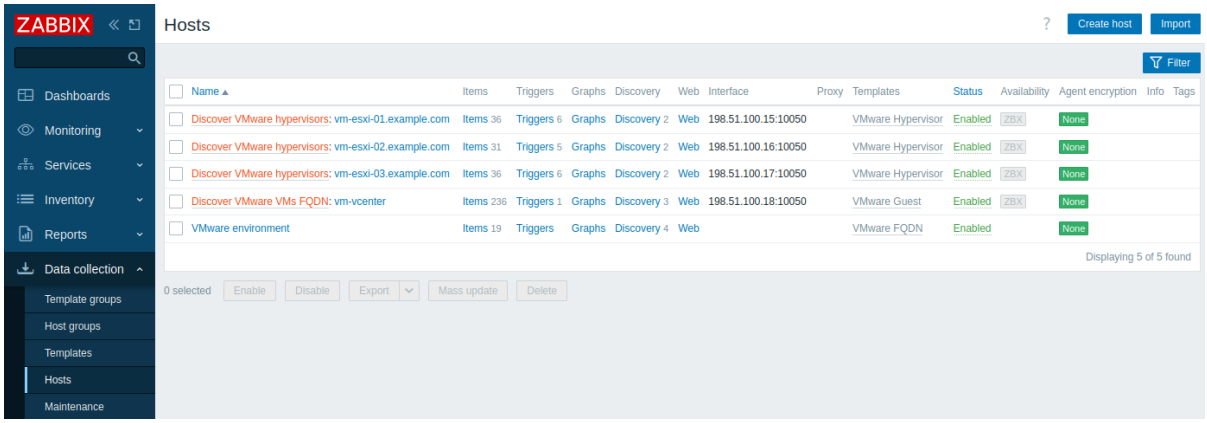
3. Klicken Sie auf die Schaltfläche *Add*, um den Host zu erstellen. Dieser Host wird Ihre VMware-Umgebung repräsentieren.

Erfasste Metriken anzeigen

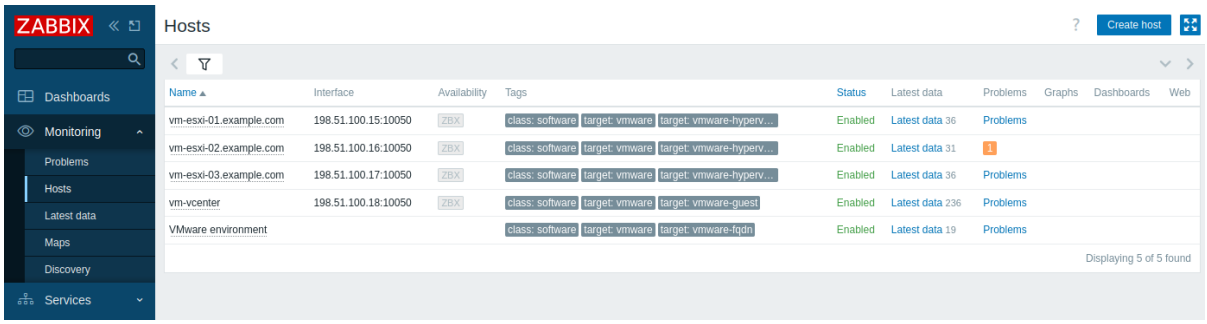
Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihre VMware-Umgebung.

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

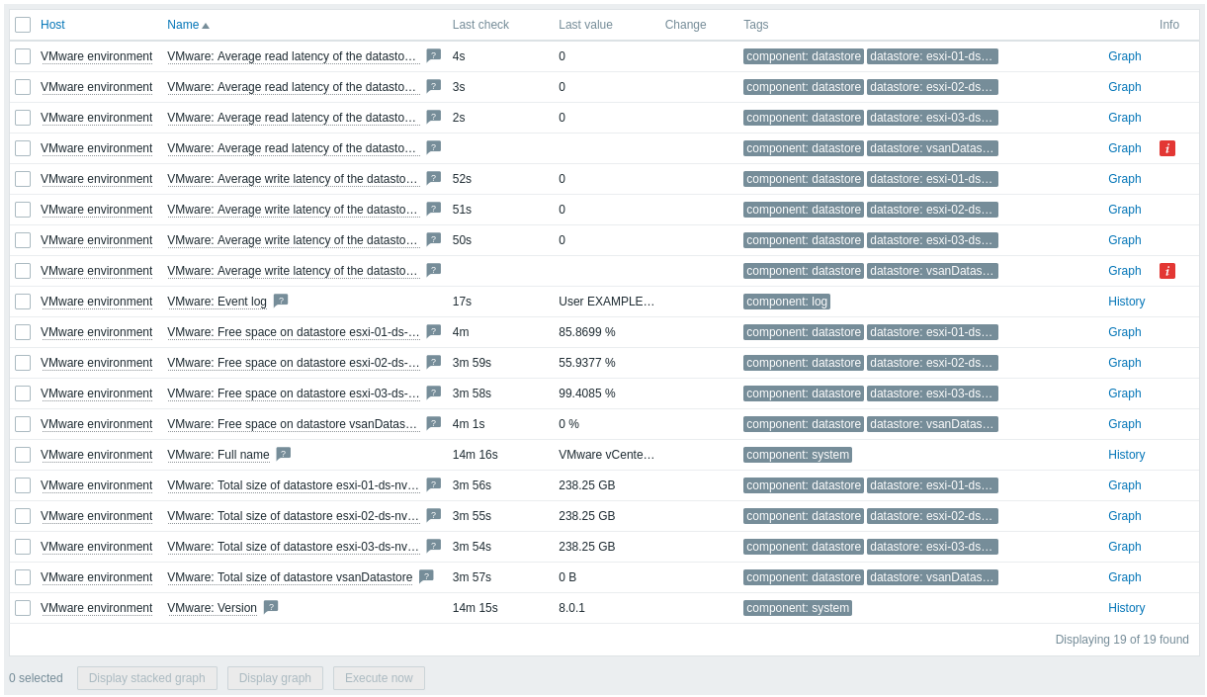
Um die erstellten Hosts anzuzeigen, wechseln Sie zum Menüabschnitt *Datenerfassung → Hosts*.



Um die erfassten Metriken anzuzeigen, wechseln Sie zum Menüabschnitt *Überwachung → Hosts* und klicken Sie neben dem erstellten Host „VMware environment“ oder einem der Hosts, die für die erkannten Entitäten erstellt wurden, auf *Neueste Daten*.



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

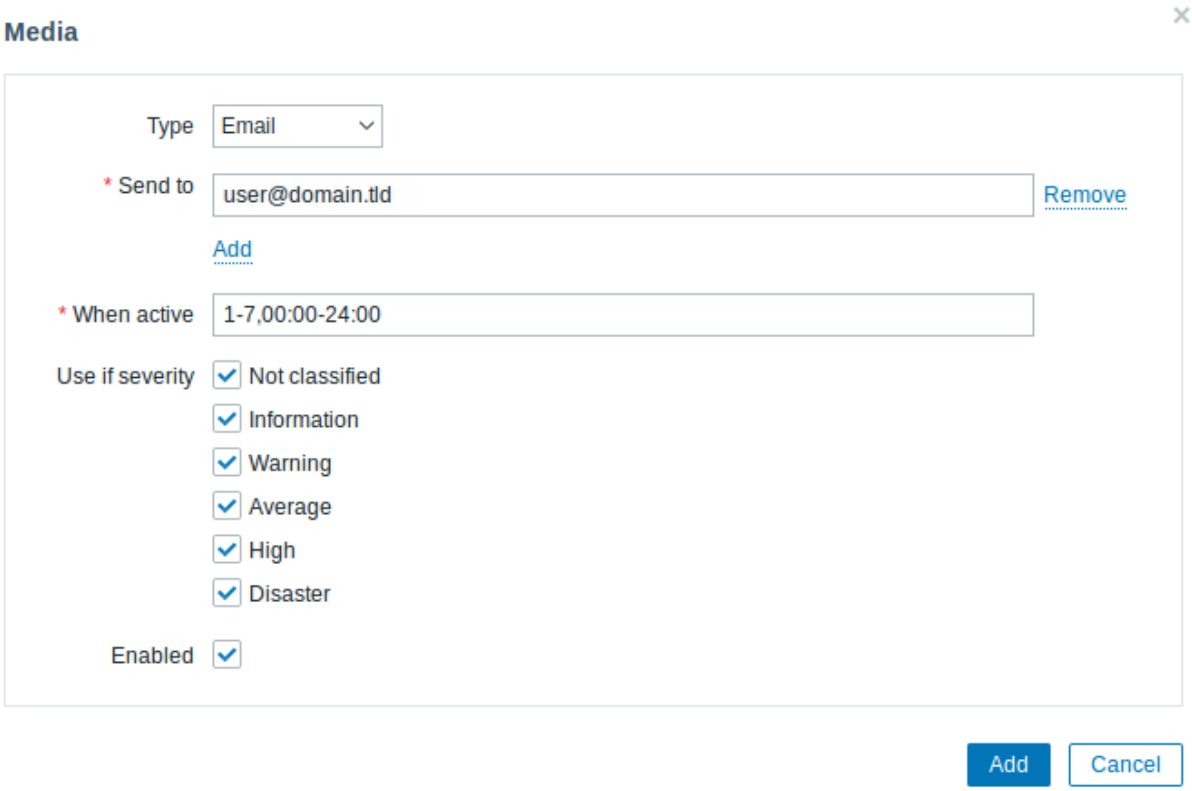


Beachten Sie, dass einige Datenpunkte keine Daten haben und sich im Status *Nicht unterstützt* befinden. Der Grund dafür ist, dass Zabbix auf dem jeweiligen Datenspeicher keine gültigen Leistungsindikatoren finden kann, da diese in der überwachten VMware-Umgebung nicht aktiviert sind.

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 für 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.
- Problemeskalationen – wie Sie mehrstufige Warnszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten gelöst wird, eine Nachricht an den Rechenzentrumsleiter senden).
- Überwachung virtueller Maschinen – zusätzliche Informationen zur VMware-Überwachung (Datenerfassungsprozess, Server-Konfigurationsoptionen, Hinweise zur Fehlerbehebung usw.).
- Datenpunktschlüssel für die VMware-Überwachung – eine vollständige Liste der VMware-Metriken, die mit Zabbix überwacht werden können.
- Vorlage *VMware* – zusätzliche Informationen über die Vorlage *VMware*.
- Vorlage *VMware FQDN* – zusätzliche Informationen über die Vorlage *VMware FQDN*.

6 Netzwerkverkehr mit Zabbix überwachen

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

Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihres Netzwerkverkehrs zu aktivieren. Wenn Sie nach umfassenden Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Configuration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie Zabbix Server, Zabbix Frontend und Zabbix Agent gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.
Beachten Sie, dass Sie den Zabbix Agent auf dem Rechner installieren sollten, für den eine 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 Anweisungen zur Konfiguration der Überwachung des Netzwerkverkehrs der Schnittstelle *eth0* auf einem separaten Rechner mit dem Namen *Remote host*.

Zabbix für das Monitoring konfigurieren Der Zabbix Agent kann Metriken im aktiven oder passiven Modus erfassen (gleichzeitig). Weitere Informationen finden Sie unter Passive und aktive Agent-Prüfungen. In diesem Leitfaden wird das Monitoring mittels passiver Prüfungen beschrieben.

Zabbix Agent konfigurieren

1. Öffnen Sie die Agent-Konfigurationsdatei auf dem Rechner, auf dem der Agent installiert ist (standardmäßig ist der Pfad */usr/local/etc/zabbix_agentd.conf*):

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

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

```
Server=192.0.2.22
```

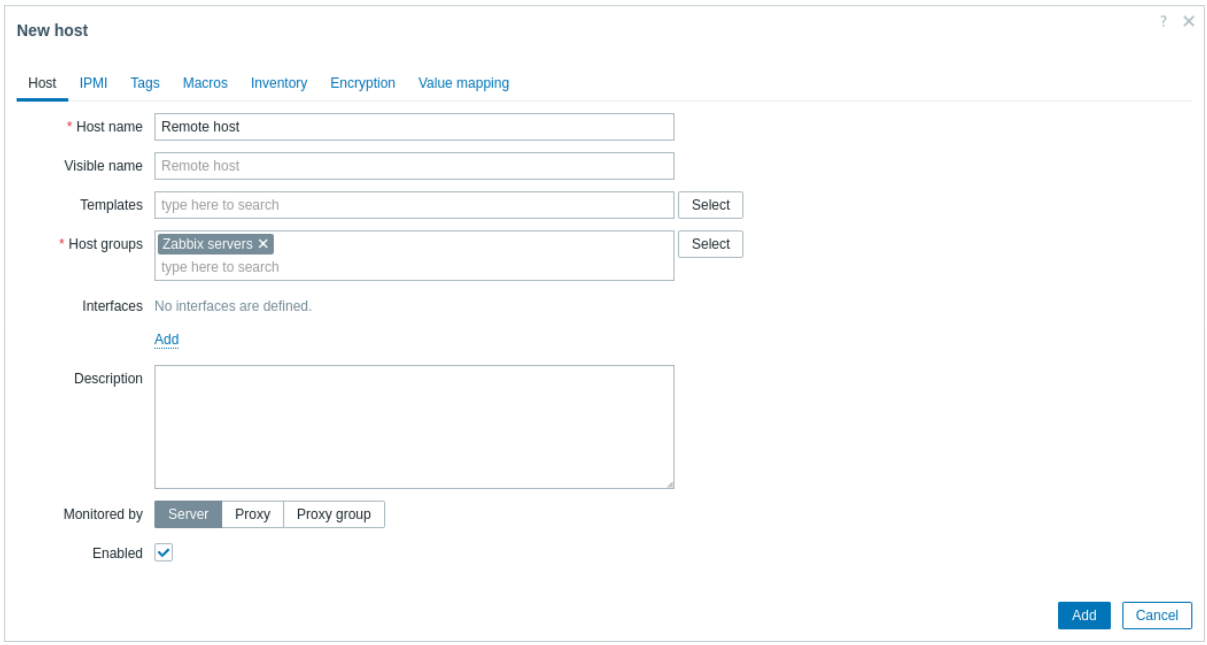
3. Starten Sie den Zabbix Agent neu:

```
systemctl restart zabbix-agent
```

Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

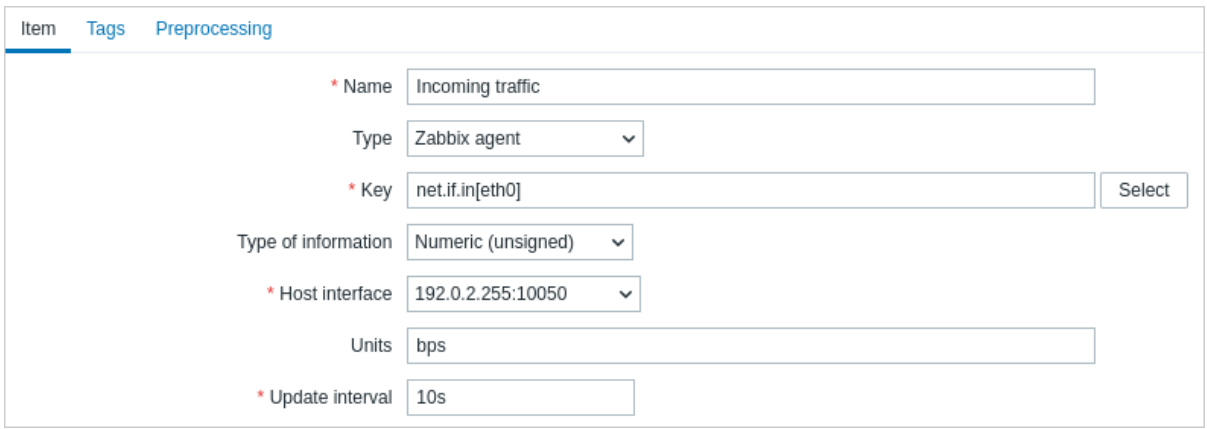
2. Erstellen Sie einen Host in der Zabbix-Weboberfläche und geben Sie die IP-Adresse oder den DNS-Namen des Rechners an, auf dem der Agent installiert ist.



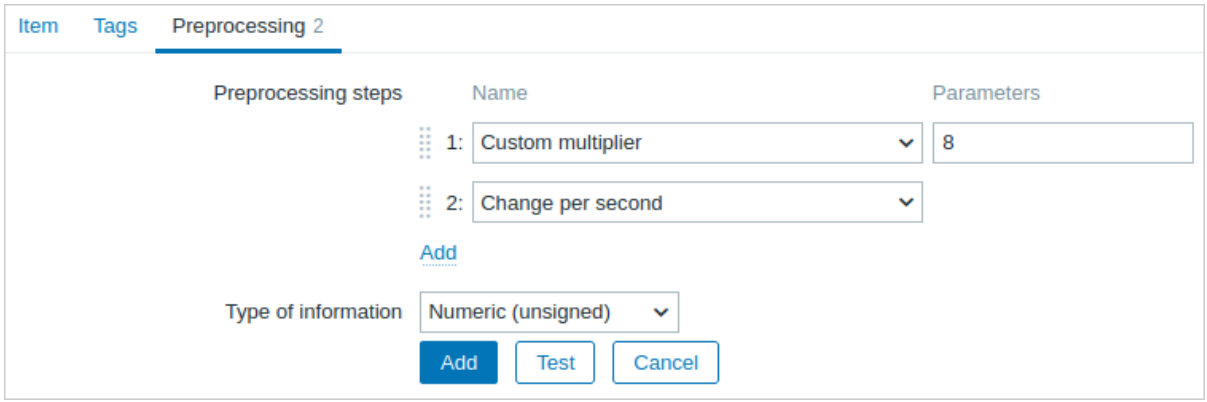
Datenpunkte erstellen Folgen Sie den Anweisungen unter Erstellen eines Datenpunkts, um die Datenpunkte für die Verkehrsüberwachung hinzuzufügen, nämlich:

- Eingehender Verkehr
- Ausgehender Verkehr
- Gesamtverkehr

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

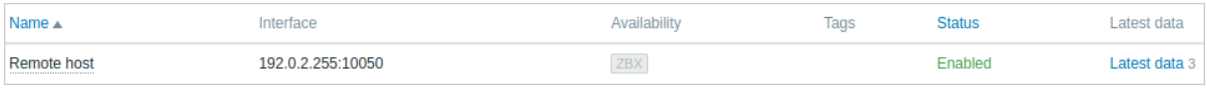


Um die erfassten Daten für die praktische Nutzung geeignet zu machen, können Sie beim Erstellen der Datenpunkte einige Schritte der Vorverarbeitung festlegen. Im vorliegenden Fall wären das die Multiplikation mit 8 (um Bytes in Bits umzuwandeln) und die Darstellung als Änderung pro Sekunde.

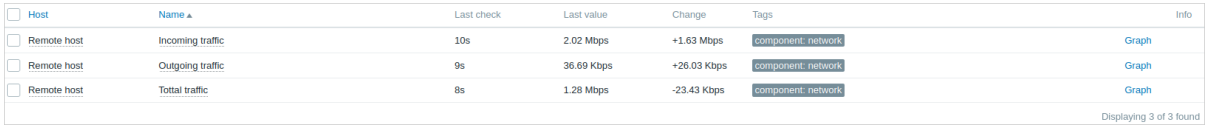


Erfasste Daten anzeigen Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren Netzwerkverkehr.

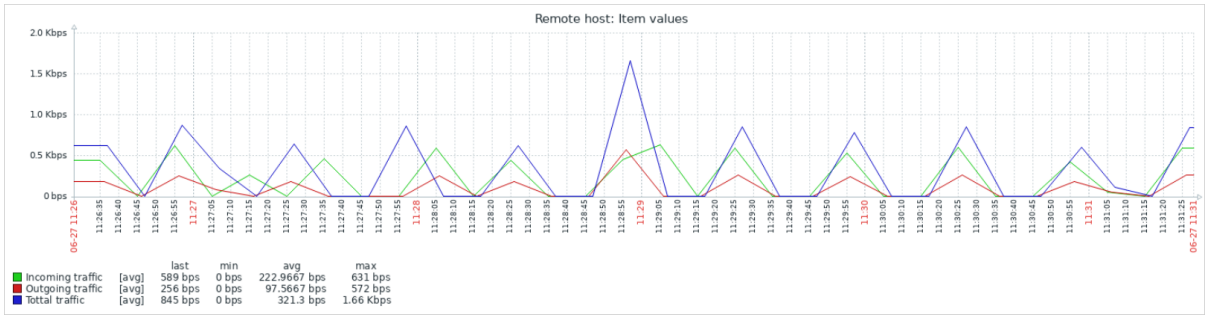
Um die erfassten Metriken anzuzeigen, öffnen Sie den Menüabschnitt *Monitoring → Hosts* und klicken Sie in der Zeile des Hosts auf *Latest data*.



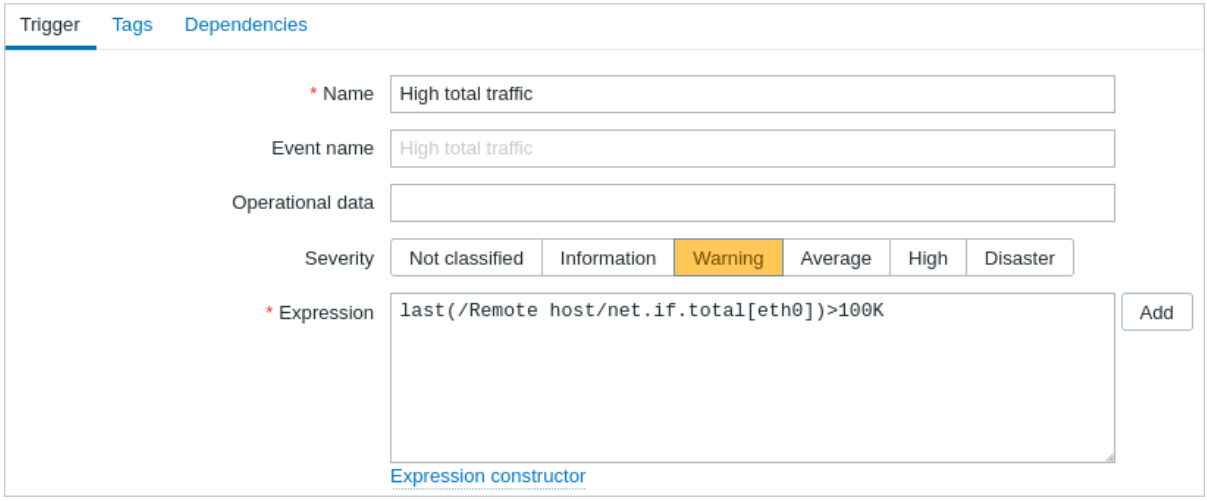
Sie sehen die Verkehrsdaten.



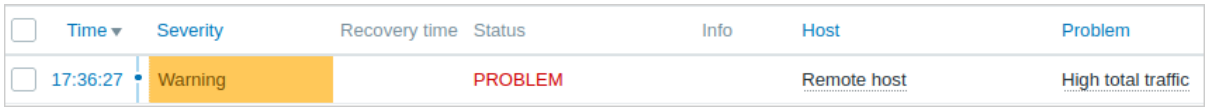
Graphen anzeigen Die gesammelten Daten können als Graphen angezeigt werden. Um diese anzuzeigen, klicken Sie im Abschnitt *Letzte Daten* in der Zeile des Datenpunkts auf *Graph* oder wählen Sie die gewünschten Datenpunkte aus und klicken Sie unten auf *Graph anzeigen*.



Auslöser konfigurieren Sie können Auslöser einrichten, um ungewöhnlichen Netzwerkverkehr zu erkennen. Lesen Sie die Anweisungen zum Konfigurieren eines Auslösers und fügen Sie beispielsweise einen Auslöser hinzu, der signalisiert, dass der Gesamtverkehr zu hoch ist:



Lassen Sie nun den Datenverkehr den Schwellenwert überschreiten, den Sie im Auslöser-Ausdruck festgelegt haben, und wechseln Sie zu *Monitoring* → *Probleme*, um zu prüfen, dass das Problem dort aufgeführt ist.



Problembenachrichtigungen einrichten Es gibt mehrere Möglichkeiten, Benachrichtigungen über das Problem zu erhalten. E-Mail ist die gängigste davon; folgen Sie den Anweisungen zum Einrichten einer Problembenachrichtigung per E-Mail. Sie können auch andere Medientypen für die Zustellung von Benachrichtigungen verwenden.

Siehe auch:

- Problemeskalationen – wie mehrstufige Warnszenarien erstellt werden (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten behoben wird, eine Nachricht an den Rechenzentrumsleiter senden).
- Problembestätigung – wie angegeben wird, dass das Problem bekannt ist, wie Kommentare zu seiner Behebung hinzugefügt und das Problem unterdrückt oder geschlossen werden kann.
- Linux mit Zabbix Agent überwachen – wie die grundlegende Überwachung der wichtigsten Datenpunkte durch Verknüpfen einer vorkonfigurierten Vorlage gestartet wird.

7 Netzwerkverkehr mit aktiven Prüfungen überwachen

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

Für wen diese Anleitung gedacht ist

Diese Anleitung 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 umfassenden Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Configuration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie Zabbix Server, Zabbix Frontend und 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 Datenverkehrs erforderlich ist. Dies kann entweder derselbe Host sein, auf dem Zabbix Server installiert ist, oder ein anderer Host.

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

Zabbix für das Monitoring konfigurieren Der Zabbix Agent kann Metriken im aktiven oder passiven Modus erfassen (gleichzeitig). Weitere Informationen finden Sie unter Passive und aktive Agent-Prüfungen. In diesem Leitfaden wird das Monitoring mittels **aktiver Prüfungen** beschrieben.

Zabbix Agent konfigurieren

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

Wenn Sie 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 Server zum Parameter ServerActive hinzu. Zum Beispiel:

```
ServerActive=192.0.2.0:10051
```

Zabbix Agent verwendet diese Adresse, um sich mit dem Trapper-Port des Zabbix Server (Standard: 10051) zu verbinden und die Konfigurationsdaten für aktive Prüfungen anzufordern.

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

```
Hostname=Remote host
```

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

4. Starten Sie Zabbix Agent neu.

Wenn Sie Zabbix agent verwenden:

```
systemctl restart zabbix-agent
```

Wenn Sie Zabbix agent 2 verwenden:

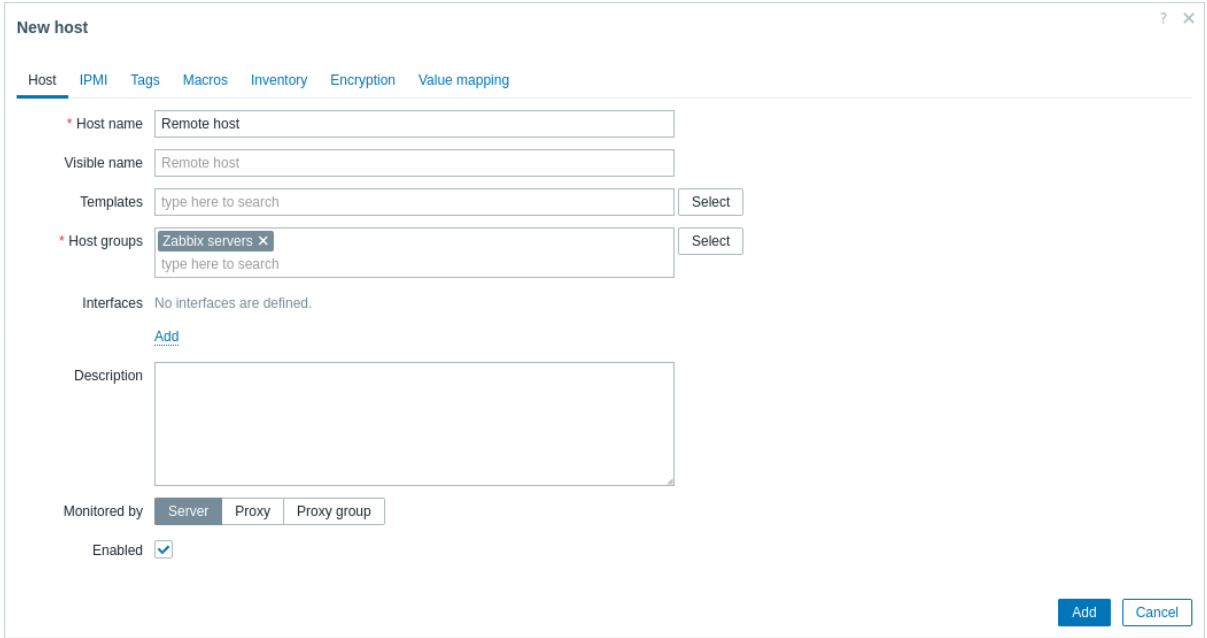
```
systemctl restart zabbix-agent2
```

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 Host-Namen 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 Host-Gruppe ein oder wählen Sie eine aus (z. B. „Zabbix servers“).
- Lassen Sie *Interfaces* undefiniert; eine Schnittstelle ist für aktive Prüfungen nicht erforderlich, da der Agent die Verbindung zum Server initiiert, anstatt auf eine Verbindung vom Server zu warten.

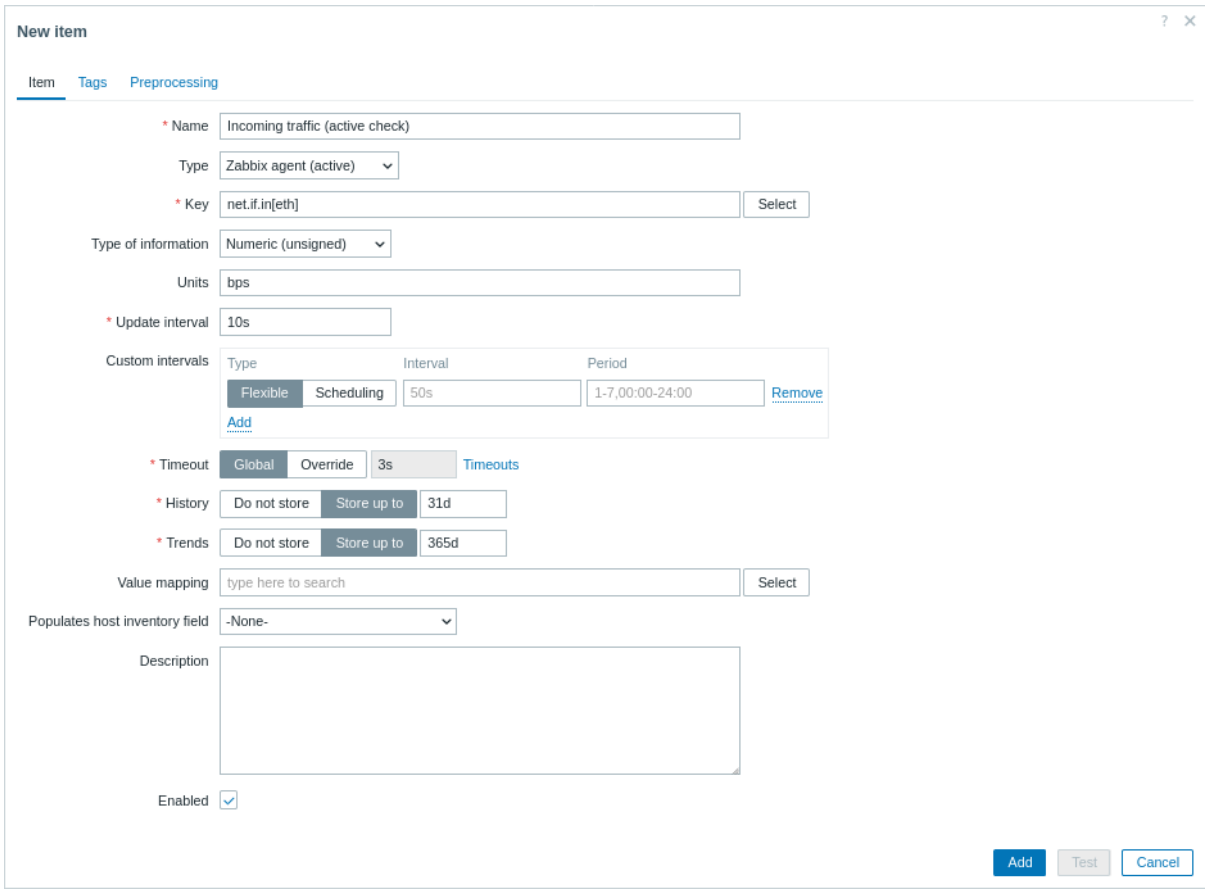


3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host wird den überwachten Linux-Rechner repräsentieren.

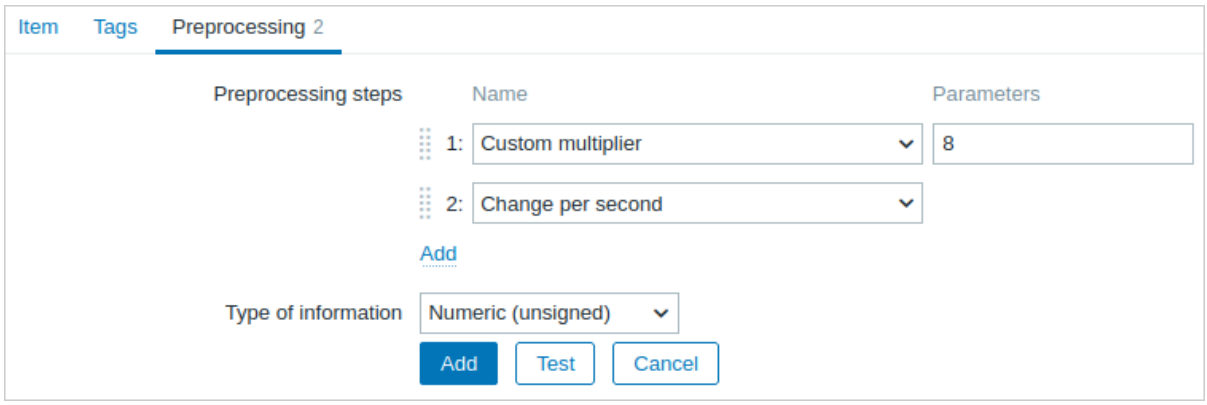
Datenpunkte erstellen Folgen Sie den Anweisungen unter Erstellen eines Datenpunkts, um die Datenpunkte für die Verkehrsüberwachung hinzuzufügen, nämlich:

- Eingehender Datenverkehr
- Ausgehender Datenverkehr
- Gesamter Datenverkehr

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

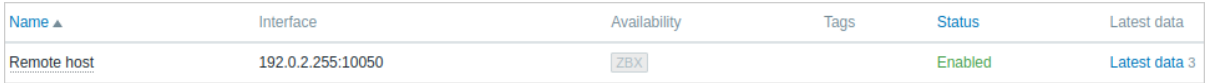


Um die erfassten Daten für die praktische Nutzung geeignet zu machen, können Sie beim Erstellen der Datenpunkte einige Schritte der Vorverarbeitung festlegen. Im vorliegenden Fall können dies die Multiplikation mit 8 (um Bytes in Bits umzuwandeln) und die Darstellung als Änderung pro Sekunde sein.

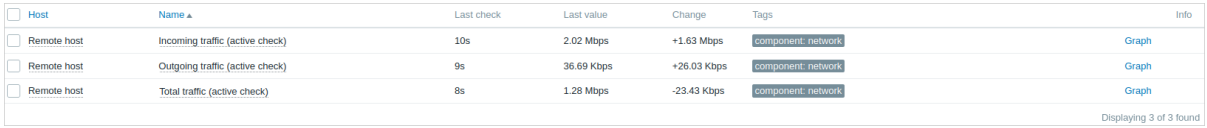


Erfasste Daten anzeigen Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren Netzwerkverkehr.

Um die erfassten Metriken anzuzeigen, öffnen Sie den Menüabschnitt *Monitoring → Hosts* und klicken Sie in der Zeile des Hosts auf *Neueste Daten*.



Sie sehen die Verkehrsdaten.



Host	Name ▲	Last check	Last value	Change	Tags	Info
Remote host	Incoming traffic (active check)	10s	2.02 Mbps	+1.63 Mbps	component: network	Graph
Remote host	Outgoing traffic (active check)	9s	36.69 Kbps	+26.03 Kbps	component: network	Graph
Remote host	Total traffic (active check)	8s	1.28 Mbps	-23.43 Kbps	component: network	Graph

Displaying 3 of 3 found

Siehe auch:

- Graphen anzeigen - wie die erfassten Daten als Graphen angezeigt werden.
- Auslöser konfigurieren - wie Auslöser eingerichtet werden, um ungewöhnlichen Netzwerkverkehr zu erkennen.
- Problembenachrichtigungen einrichten - wie Benachrichtigungen über Problemsituationen eingerichtet werden.
- Problemeskalationen - wie mehrstufige Warnszenarien erstellt werden (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten behoben wird, eine Nachricht an den Rechenzentrumsleiter senden).
- Problembestätigung - wie angegeben wird, dass das Problem bekannt ist, wie Kommentare zu seiner Behebung hinzugefügt und das Problem unterdrückt oder geschlossen werden kann.
- Linux mit Zabbix Agent überwachen - wie die grundlegende Überwachung der wichtigsten Datenpunkte durch Verknüpfen einer vorkonfigurierten Vorlage gestartet wird.
- Installation aus Paketen - wie Zabbix-Komponenten mit offiziellen RPM- und DEB-Paketen für verschiedene Linux-Distributionen installiert werden, wodurch der Zugriff auf die neuesten Funktionen und Fehlerbehebungen sichergestellt wird.

8 Websites mit Browser-Datenpunkten überwachen

Einführung

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

Für wen diese Anleitung gedacht ist

Diese Anleitung 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 umfassenden Anpassungsoptionen suchen oder eine erweiterte 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.

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

WebDriver konfigurieren

Browser-Datenpunkte benötigen ein Automatisierungs-Framework (entweder Selenium Server oder einen einfachen WebDriver, zum Beispiel ChromeDriver) als Endpunkt für Webtests, der einen Browser steuert und mit ihm interagiert und dabei Testbefehle wie das Klicken auf Schaltflächen oder die Eingabe von Text ausführt. Als Beispiel wird in dieser Anleitung Selenium Server mit Chrome in einem Docker-Container verwendet.

Es wird vorausgesetzt, dass Docker bereits konfiguriert ist. Diese Anleitung behandelt die Konfiguration von Docker nicht. Installationsanweisungen 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-Rechner Port 4444 im Container zu (dies ist der Port, den Selenium Server zum Entgegennehmen von Befehlen verwendet);
- **-p 7900:7900** - ordnet Port 7900 auf Ihrem Host-Rechner Port 7900 im Container zu (dies ist der Port, den der Virtual Network Computing (VNC)-Server verwendet, sodass Sie die Browser-GUI aus der Ferne anzeigen können; ein VNC-Client ist erforderlich);
- **--shm-size=”2g”** - weist dem Container 2 GB Shared Memory zu (dies ist wichtig, damit Chrome ordnungsgemäß ausgeführt werden kann, da es eine erhebliche Menge an Shared Memory benötigen kann, um Abstürze zu vermeiden);
- **-d** - führt den Container im Detached-Modus aus, das heißt, er läuft im Hintergrund;
- **selenium/standalone-chrome:latest** - gibt das zu verwendende Docker-Image an; in diesem Fall die neueste Version von Selenium Server with 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` läuft und erreichbar ist.

- Ermitteln Sie die IP-Adresse des Containers (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 Inhalt der Webseite 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 siehe die Docker documentation.

Zabbix Server konfigurieren

Browser-Datenpunkte werden von *browser poller*-Zabbix-Prozessen ausgeführt und verarbeitet, die durch Anpassen des Server-Konfigurationsparameters StartBrowserPollers aktiviert werden müssen. Zusätzlich sollte der Parameter WebDriverURL den zuvor konfigurierten Endpunkt für Webtests angeben.

Standardmäßig ist der Parameter StartBrowserPollers auf 1 gesetzt, daher müssen Sie nur den Endpunkt für Webtests angeben.

1. Öffnen Sie die Zabbix-Server-Konfigurationsdatei.

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

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

```
##### Option: WebDriverURL
###WebDriver-Schnittstellen-HTTP[S]-URL. Zum Beispiel http://localhost:4444, verwendet mit dem eigenständi
###
### Mandatory: no
### Default:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Starten Sie den Zabbix Server neu.

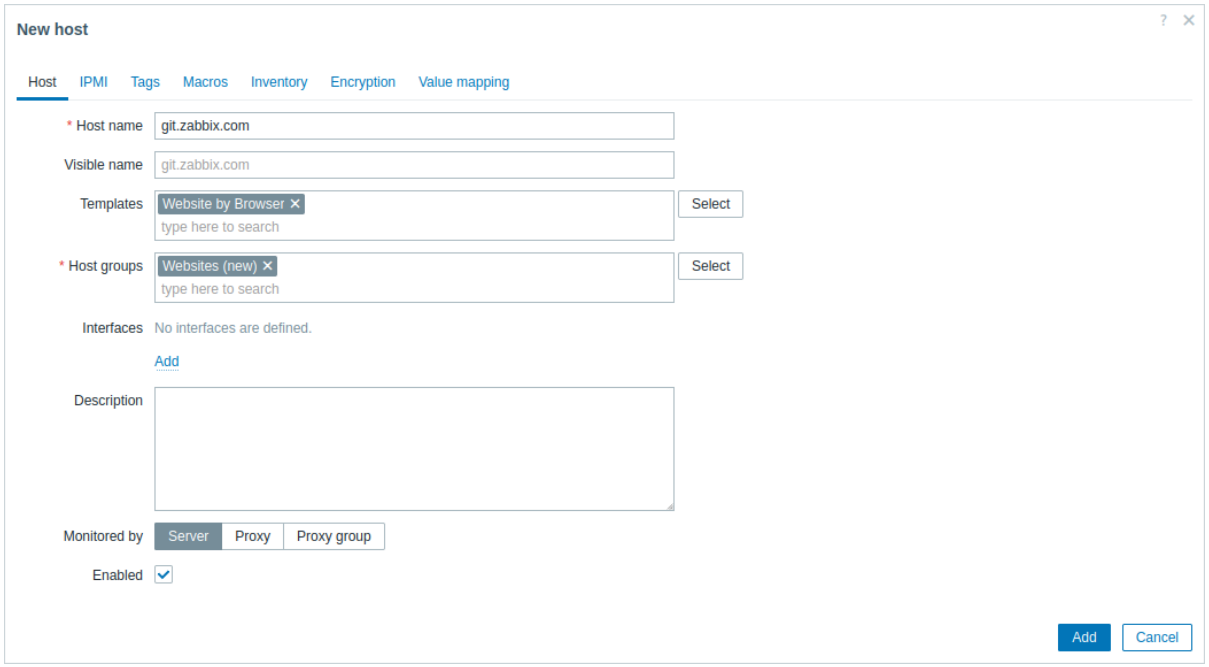
```
systemctl restart zabbix-server
```

Zabbix Frontend konfigurieren

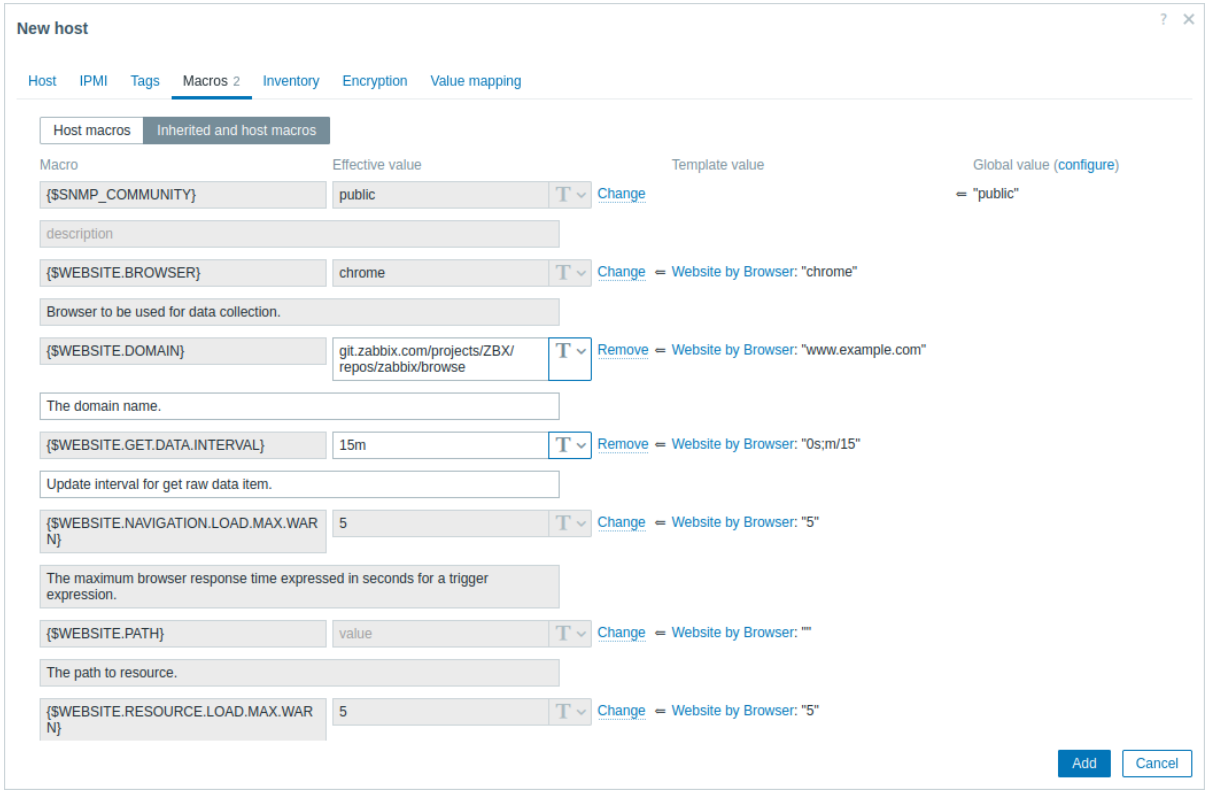
1. Melden Sie sich im Zabbix Frontend an.

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

- Geben Sie im Feld *Host name* einen Host-Namen 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 Host-Gruppe ein oder wählen Sie sie aus (zum Beispiel eine neue Host-Gruppe „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 der Datenpunkte (zum Beispiel 15m)

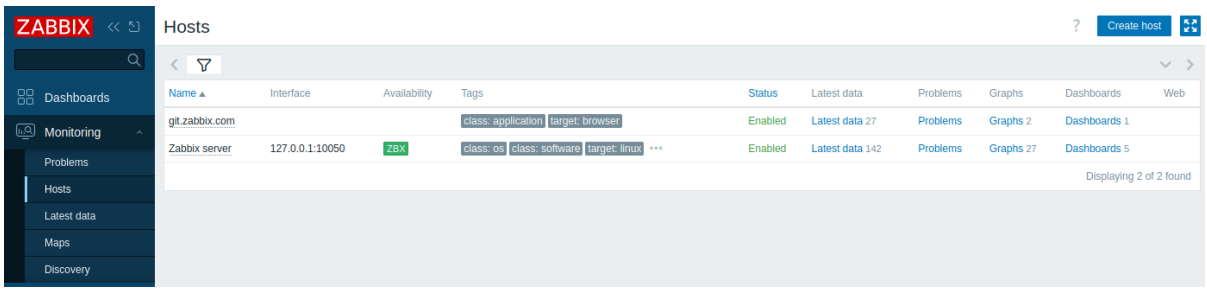


3. Klicken Sie auf die Schaltfläche *Add*, um den Host zu erstellen. Dieser Host repräsentiert die Website, die Sie überwachen möchten.

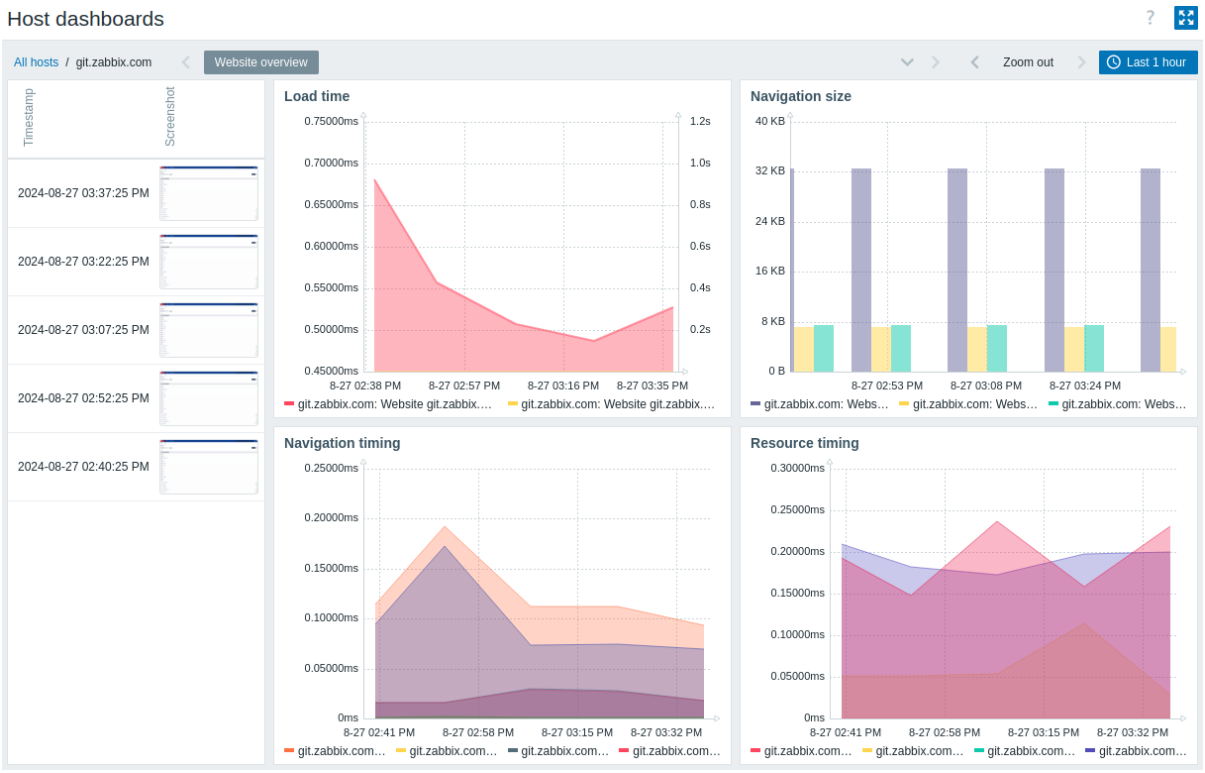
Erfasste Metriken anzeigen

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

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



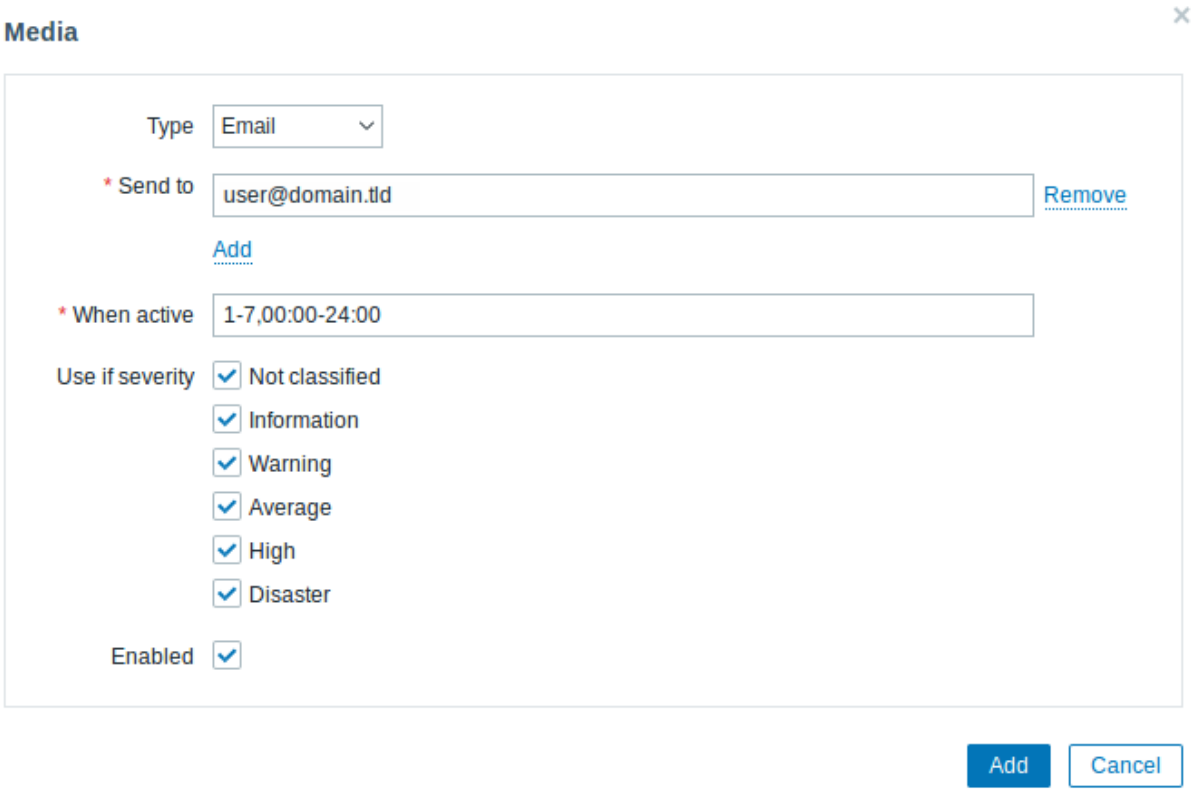
Dadurch gelangen Sie zum Host-Dashboard (auf Vorlagenebene konfiguriert) mit den wichtigsten von der Website erfassten Metriken.



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 für 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 Ihre Website-Host-Konfiguration in Zabbix.

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

3. Klicken Sie neben dem zuvor konfigurierten Makrowert {$WEBSITE.DOMAIN} 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. In wenigen Augenblicken wird Zabbix das Problem „JSON der angeforderten Website konnte nicht abgerufen werden“ erkennen, da keine Verbindung zur angegebenen Website hergestellt werden kann. Das Problem wird unter *Monitoring → Problems* angezeigt.

Wenn Benachrichtigungen konfiguriert sind, erhalten Sie außerdem eine Problembenachrichtigung.

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

Siehe auch

- Erstellen eines Datenpunkts – wie Sie mit der Überwachung zusätzlicher Metriken beginnen.
- Problemeskalationen – wie Sie mehrstufige Warnszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nicht innerhalb von 45 Minuten behoben wird, eine Nachricht an den Rechenzentrumsleiter senden).
- Browser-Datenpunkte – wie Sie Browser-Datenpunkte konfigurieren.
- Vorlage Website by Browser – zusätzliche Informationen über die Vorlage *Website by Browser*.

9 Website-Zertifikate mit Zabbix Agent 2 (passiv) überwachen

Einführung

Diese Anleitung bietet einen umfassenden Überblick darüber, wie SSL/TLS-Zertifikate mit dem Schlüssel `web.certificate.get` in Zabbix Agent 2 eingerichtet und überwacht werden. Sie wurde entwickelt, um die Überwachung von Zertifikaten für einzelne oder mehrere Websites zu vereinfachen, sodass Administratoren potenzielle Probleme wie abgelaufene oder ungültige Zertifikate schnell erkennen können.

Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung von Website-Zertifikaten zu aktivieren. Wenn Sie nach umfassenden Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Configuration 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 derselben Maschine installiert sind; daher wird `127.0.0.1` in der Konfiguration verwendet.

Zabbix Agent 2 konfigurieren

1. Öffnen Sie die Zabbix-Agent-Konfigurationsdatei (Standardpfad: `/etc/zabbix/zabbix_agent2.conf`):

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

2. Setzen Sie den Parameter Server auf 127.0.0.1, da Agent und Server auf derselben Maschine ausgeführt werden:

```
Server=127.0.0.1
```

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

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

4. Testen Sie nach der Einrichtung und Konfiguration von Zabbix agent 2 seine Verfügbarkeit mit:

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

Zabbix agent 2 enthält standardmäßig das Plugin WebCertificate, daher ist keine separate Installation oder Konfiguration erforderlich.

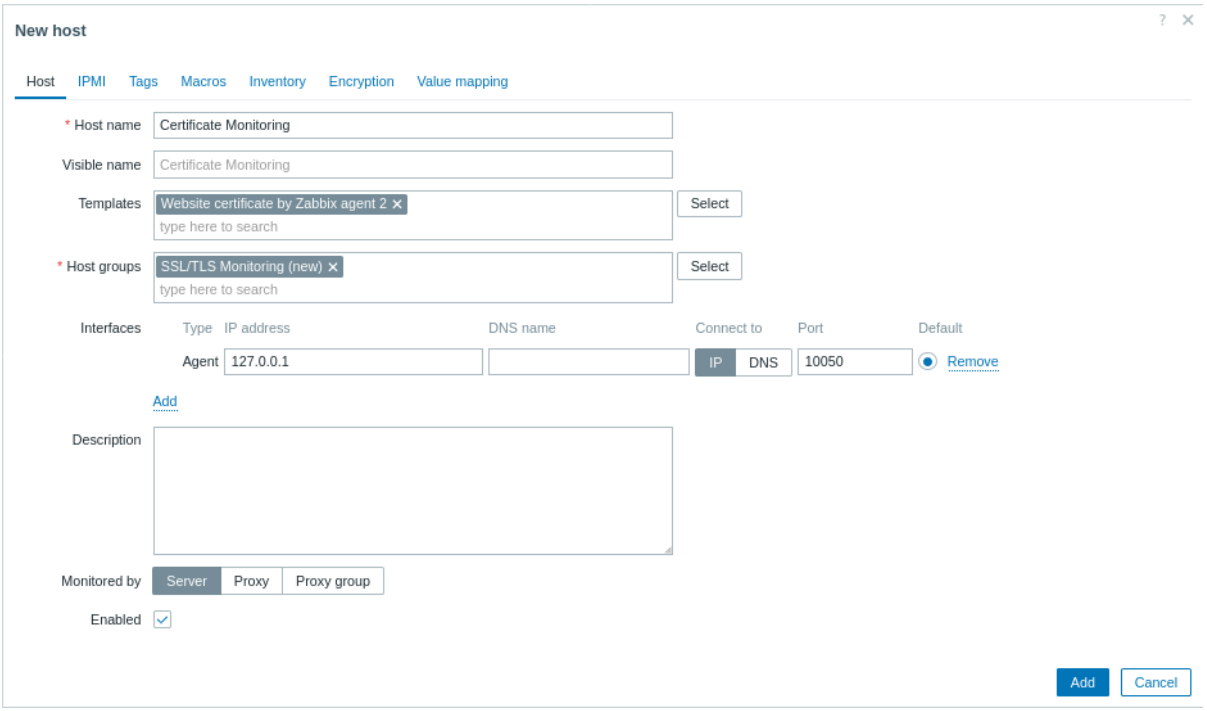
Zabbix Frontend konfigurieren

1. Melden Sie sich im Zabbix Frontend an.

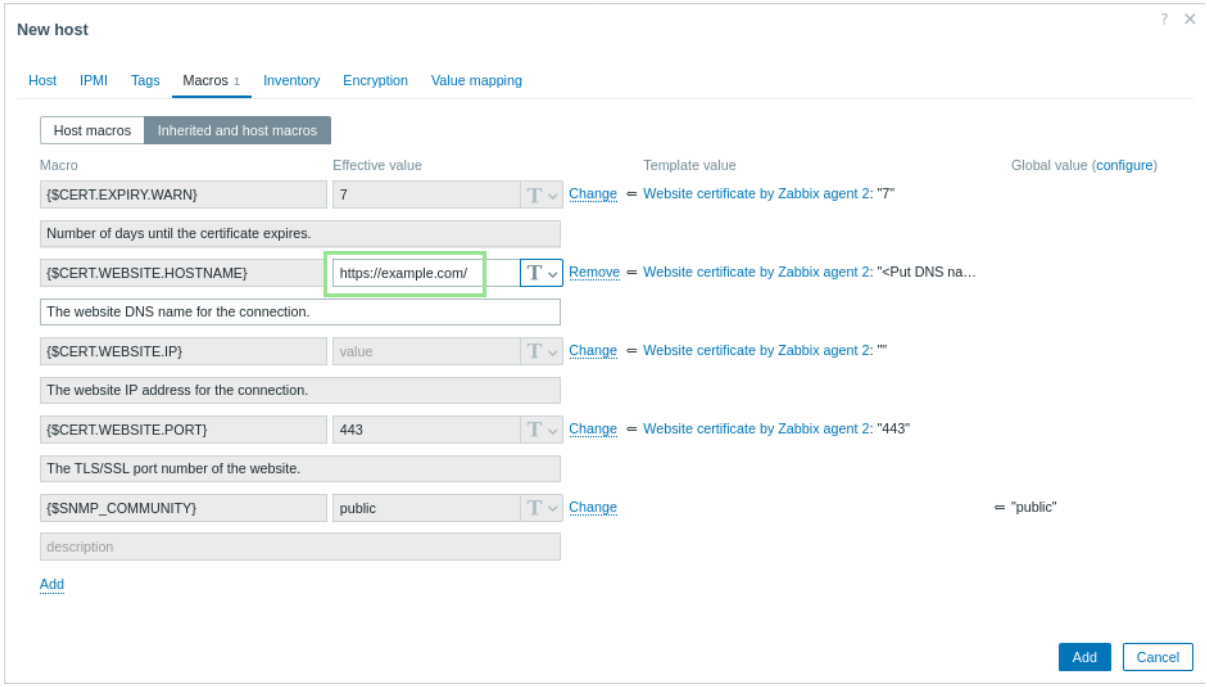
2. Navigieren Sie zu *Monitoring* > *Hosts*.

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

- Geben Sie im Feld *Host name* einen Host-Namen 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 wird.
- Geben Sie im Feld *Host groups* eine Host-Gruppe 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 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:
  - `{$CERT.WEBSITE.HOSTNAME}` - Geben Sie als Wert den gewünschten DNS-Namen der Website ein.



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

Um mehrere Websites zu überwachen, können Sie in dem 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 entsprechenden Reihenfolge der Hostnamen zugeordnet wird. Um beispielsweise example.com und example.org auf Port 8443 zu überwachen:

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

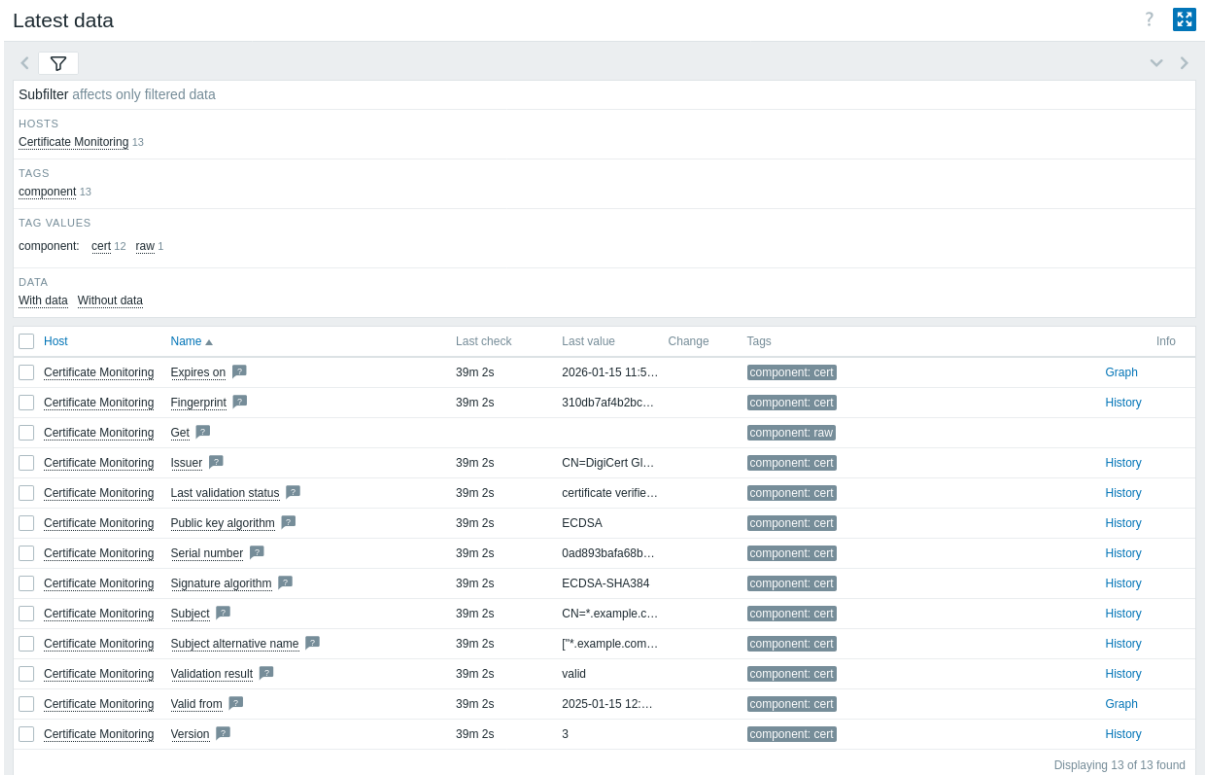
Für jede im Makro `{$CERT.WEBSITE.HOSTNAME}` angegebene Website erstellt Zabbix einen entsprechenden Satz von Datenpunkten und Auslösern. Dies ermöglicht eine individuelle Überwachung und Benachrichtigung für das SSL-Zertifikat jeder Website.

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:…

Erfasste Metriken anzeigen

Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits das gewünschte Webzertifikat.

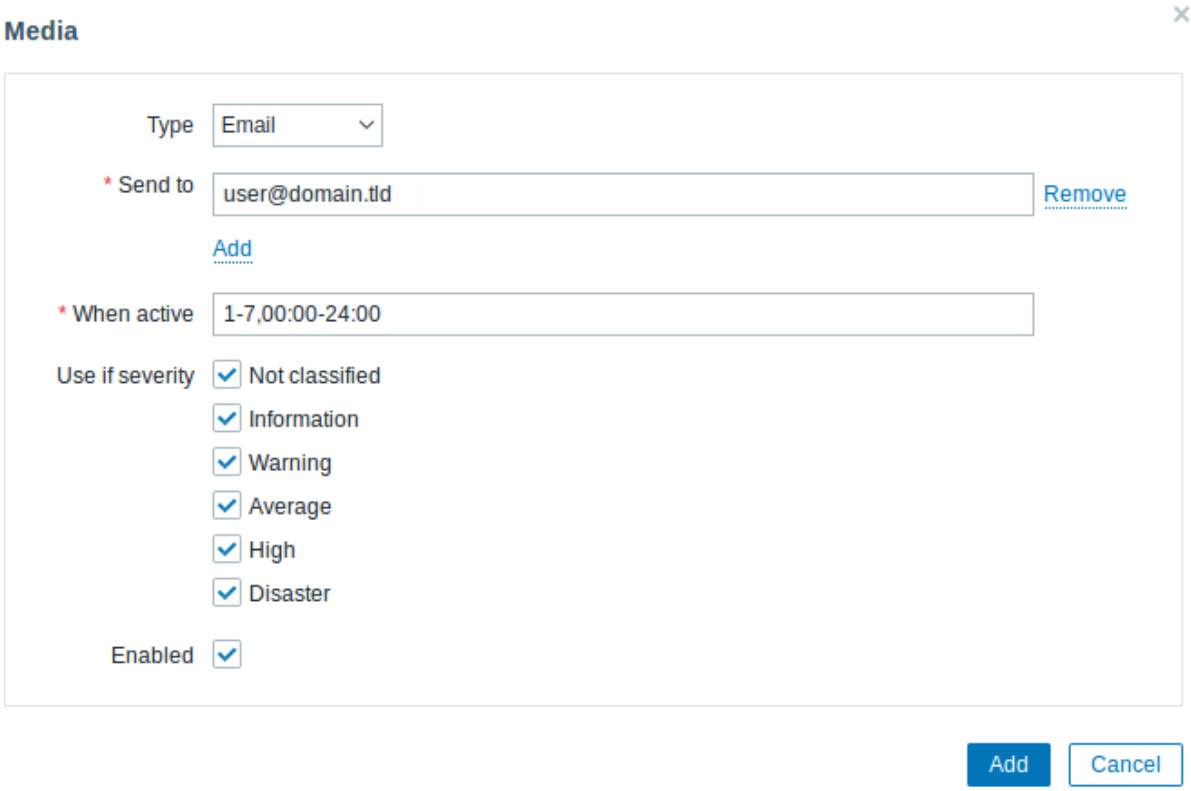
Um die erfassten Metriken anzuzeigen, wechseln Sie zum Menüabschnitt *Monitoring > Hosts* und klicken Sie neben dem Host auf *Neueste Daten*, um alle zuletzt erfassten Metriken in einer Liste anzuzeigen, z. B. Ablaufdatum, Aussteller und Betreff.



Problembenachrichtigungen einrichten

Zabbix kann Sie über Probleme in der 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 für 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.

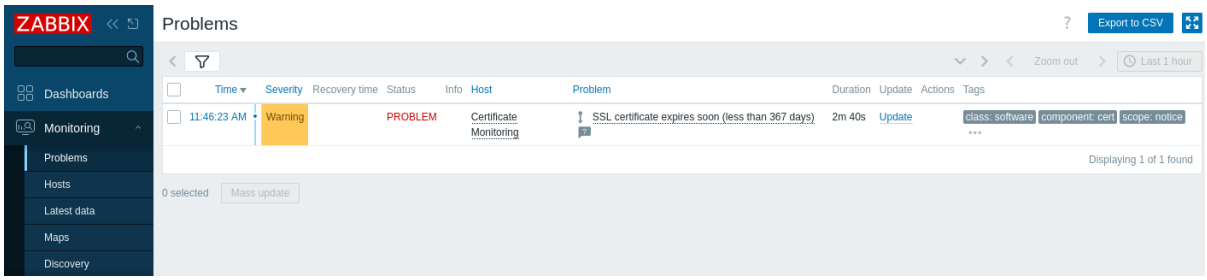
1. Öffnen Sie Ihre Host-Konfiguration „Certificate Monitoring“ in Zabbix.

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

3. Klicken Sie neben dem zuvor konfigurierten Makrowert `{$CERT.EXPIRY.WARN}` auf *Change* und setzen Sie eine sehr hohe Anzahl von Tagen (mehr als 365 Tage sollten ausreichend sein), um vor dem Ablauf des Zertifikats eine Warnung zu erhalten.

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

5. In wenigen Augenblicken wird Zabbix das Problem „SSL certificate expires soon“ mit der Anzahl der verbleibenden Tage bis zum Ablauf erkennen. Das Problem wird unter *Monitoring > Problems* angezeigt.



Wenn Warnmeldungen konfiguriert sind, erhalten Sie außerdem die Problembenachrichtigung.

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

Siehe auch

- Zabbix Agent 2 - listet Datenpunktschlüssel auf.
- Vorlage *Website-Zertifikat durch Zabbix Agent 2* - zusätzliche Informationen über die Vorlage *Website-Zertifikat durch Zabbix Agent 2*.
- Vorlage *Website-Zertifikat durch Zabbix Agent 2 aktiv* - zusätzliche Informationen über die Vorlage *Website-Zertifikat durch Zabbix Agent 2 aktiv*.
- Websites mit Browser-Datenpunkten überwachen - wie Sie die grundlegende Überwachung von Websites mit Browser-Datenpunkten starten.

10 Überwachen eines Netzwerk-Switches oder Routers mit Zabbix

Einführung Diese Anleitung führt Sie durch die Schritte, die erforderlich sind, um mit Zabbix eine grundlegende Überwachung Ihres Netzwerk-Switches oder Routers zu starten. Als Beispiel wird ein Cisco-Router 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 schnell eine grundlegende Überwachung für Netzwerkgeräte aktivieren möchten. Wenn Sie eine umfassende Anpassung oder erweiterte Konfigurationsoptionen benötigen, lesen Sie bitte die Seite SNMP agent oder den Abschnitt Configuration im Zabbix-Handbuch.

Voraussetzungen

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

- Zabbix Server und Zabbix Frontend sind installiert: Installieren Sie diese gemäß den Anweisungen für Ihr Betriebssystem (siehe Installation from packages und Web interface installation).
- Zabbix Agent ist installiert, wenn lokale Netzwerkmetriken überwacht werden sollen.
- SNMP-fähiges Gerät: ein Netzwerk-Switch oder Router (zum Beispiel ein Cisco-Router) mit aktiviertem SNMP.
- Installierte MIB-Dateien: Durch die Installation von MIB-Dateien kann Zabbix numerische OIDs in menschenlesbare Namen und Beschreibungen übersetzen. Ohne ordnungsgemäße MIB-Unterstützung werden möglicherweise nur numerische Werte angezeigt, 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, Juniper), legen Sie diese im entsprechenden MIB-Verzeichnis ab:

- Bei Linux-basierten Systemen sind gängige Speicherorte /usr/share/snmp/mibs/ oder /usr/local/share/snmp/mibs/.
- 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 korrekten 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, indem Sie ein `snmpwalk` ausführen (zum Beispiel `snmpwalk -v 2c -c <your_community_string> <device_IP>`), und kontrollieren Sie, dass OIDs mit beschreibenden Namen angezeigt werden.

Abhängig von Ihrer Umgebung können einige Schritte in dieser Anleitung leicht abweichen. 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 angeschlossen ist.

Netzwerkgerät konfigurieren (Beispiel: Cisco-Router) 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. Vorsicht: Das Anwenden dieser Befehle kann aktuelle SNMP-Konfigurationen überschreiben.

Bei einem Cisco-Router umfasst die Konfiguration in der Regel die unten beschriebenen Schritte.

SNMPv2-Beispiel 1. Aktivieren Sie SNMP und legen Sie den 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 schreibgeschützten Community-String 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) erlaubt es SNMP, Daten vom Gerät abzurufen, verhindert jedoch jegliche Konfigurationsänderungen.

Aus Sicherheitsgründen wird empfohlen, den SNMP-Zugriff nur auf die erforderlichen Geräte zu beschränken. Weitere Hinweise zur Konfiguration von Access Control Lists (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. Seine Konfiguration ist sicherer als SNMPv2 und sollte anhand der gerätespezifischen Dokumentation überprüft werden.

1. Erstellen Sie eine SNMP-Gruppe.

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

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

2. Erstellen Sie einen SNMP-Benutzer.

Fügen Sie einen SNMPv3-Benutzer mit Authentifizierung und Privacy 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-Konfigurationsanleitungen. Diese Anleitung beschreibt 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.

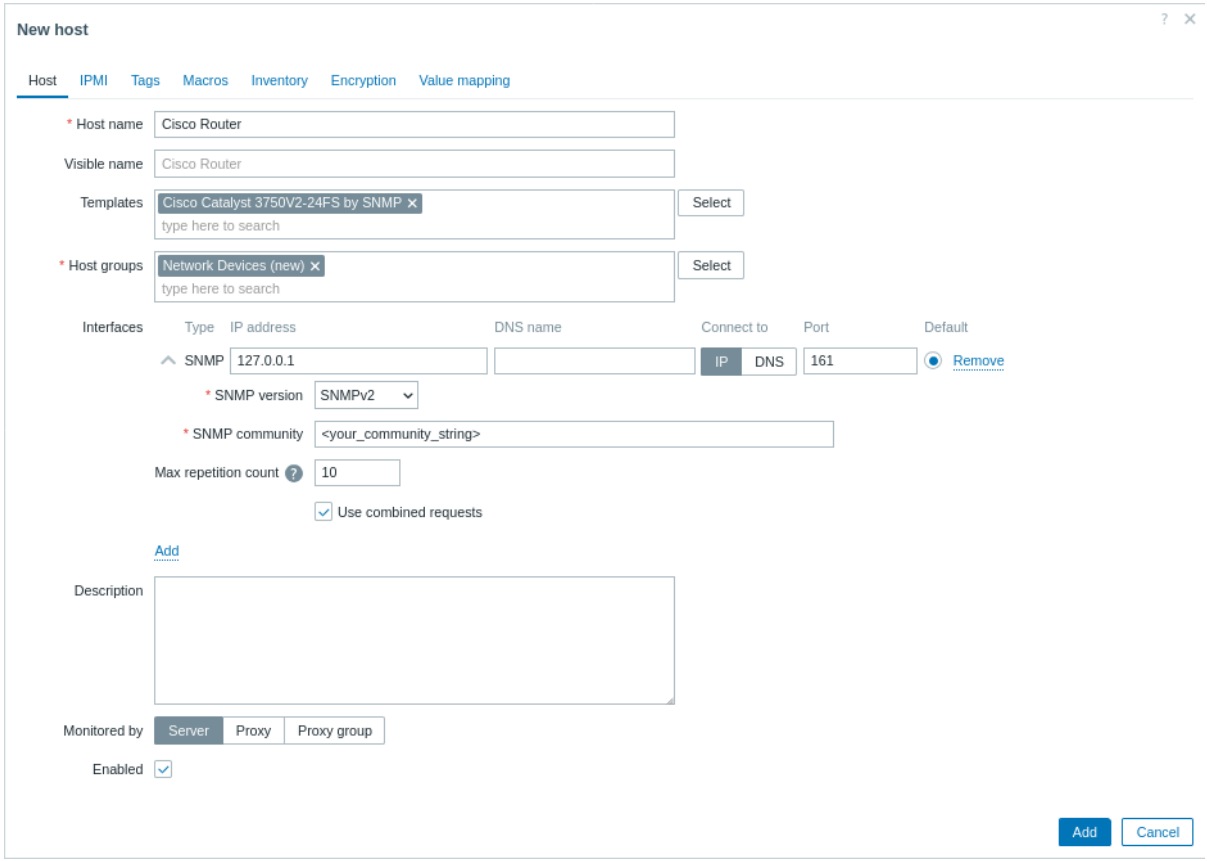
Navigieren Sie zu *Datenerfassung* > *Hosts* und klicken Sie auf *Host erstellen*.

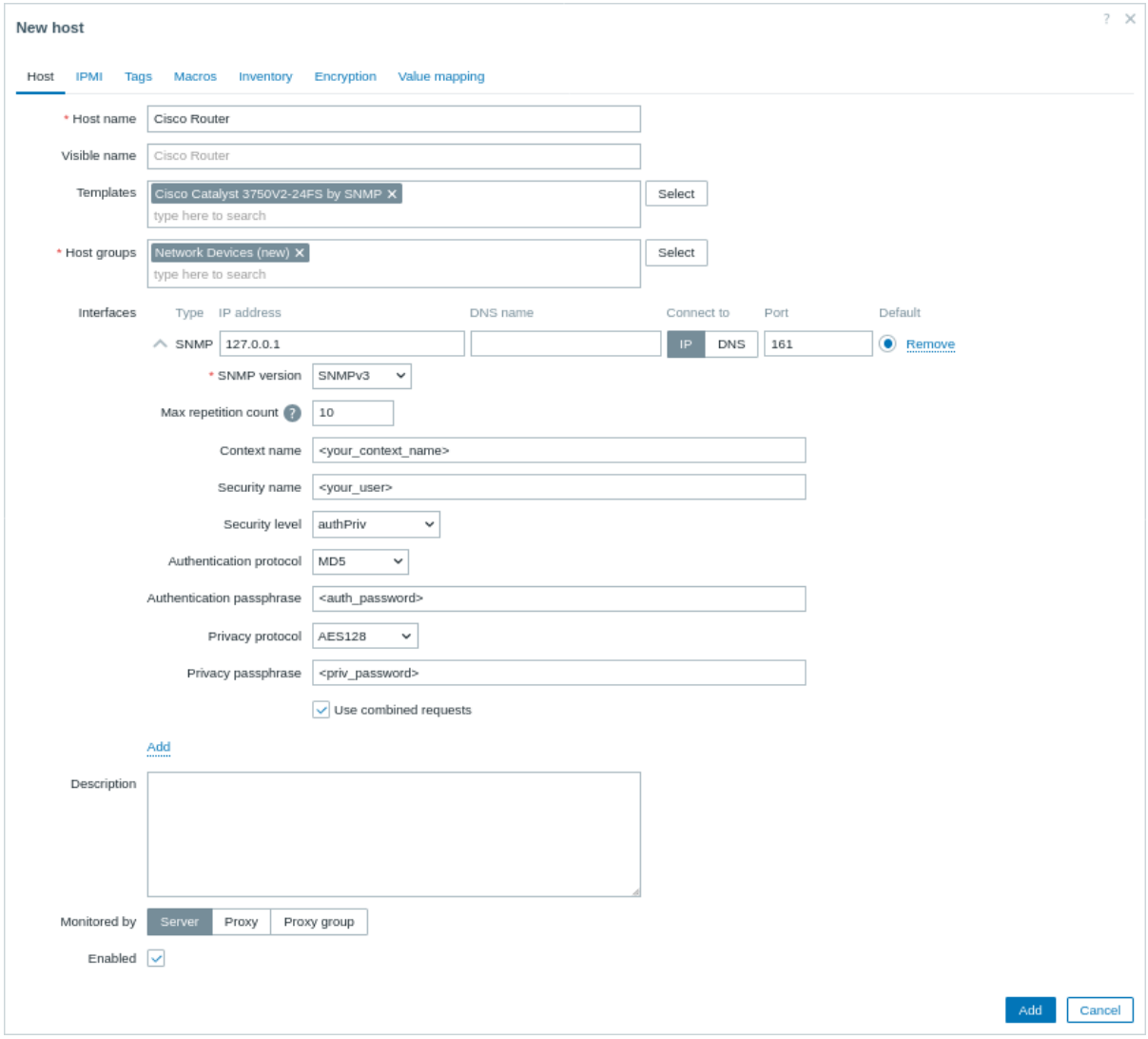
- *Host-Name*: Geben Sie einen Namen für Ihr Gerät ein (z. B. „Cisco-Router“).
- Host-Gruppen: Wählen Sie eine vorhandene Gruppe aus oder erstellen Sie eine neue Gruppe, z. B. „Netzwerkgeräte“.
- Schnittstellen:
  - Klicken Sie unter Schnittstellen auf *Hinzufügen*.
  - Wählen Sie *SNMP* als Schnittstellentyp.
  - Geben Sie die IP-Adresse oder den DNS-Namen Ihres Cisco-Routers ein.
  - Legen Sie den Standard-SNMP-Port fest (normalerweise 161).
  - Verwenden Sie das Dropdown-Menü, um die passende SNMP-Version auszuwählen (z. B. SNMPv2).
  - Geben Sie für SNMPv1/v2 den Community-String im Feld *SNMP-Community* ein. Bei SNMPv3 werden zusätzliche Anmeldedaten abgefragt (*Kontextname*, *Sicherheitsname* und *Sicherheitsstufe* usw.).

3. Vorlagen verknüpfen

Wählen Sie im Feld *Vorlagen* die SNMP-Vorlage aus, die am besten zu Ihrem Gerät passt. Zabbix bietet eine Reihe vorgefertigter SNMP-Vorlagen für viele Gerätefamilien. 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<Gerätemodell> SNMP).

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

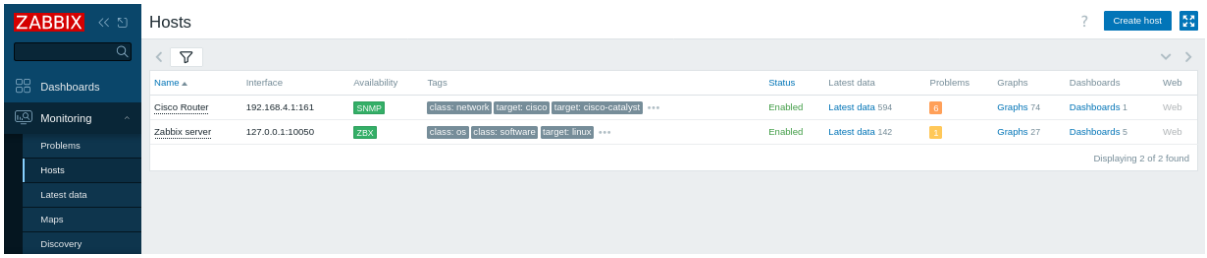




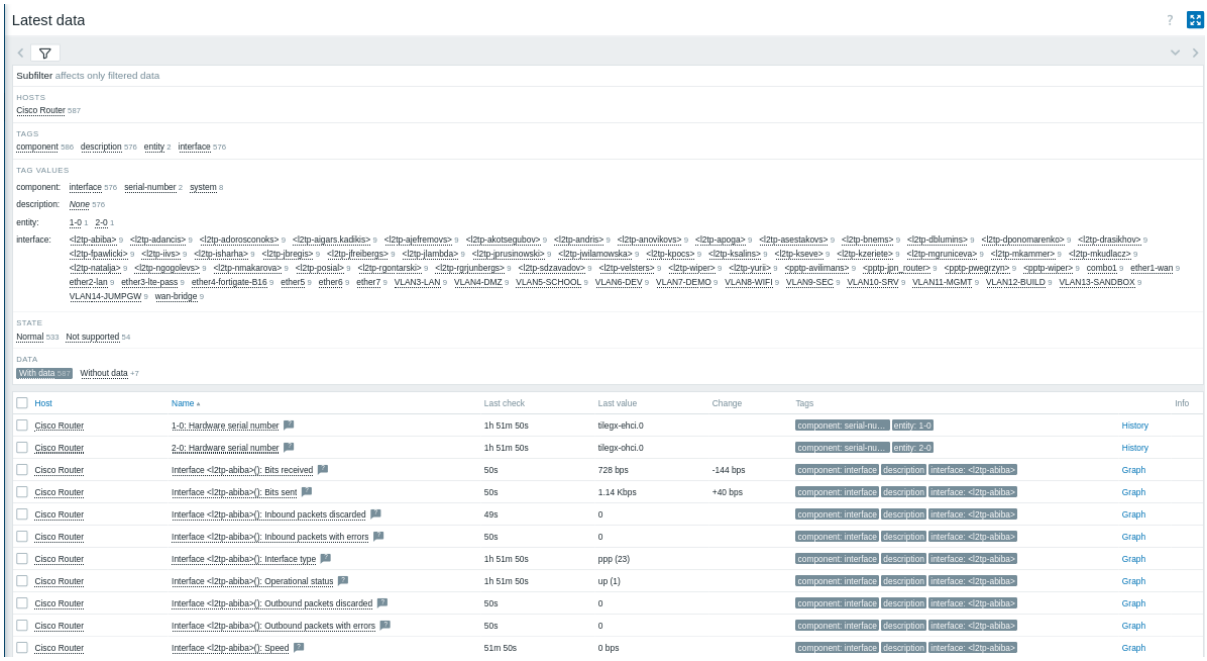
Gesammelte Metriken anzeigen Glückwunsch! Zabbix ist jetzt so eingerichtet, dass Ihr Netzwerkgerät überwacht wird.

Aktuelle Daten:

- Navigieren Sie im Zabbix Frontend zu Überwachung > Aktuelle 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 Übersichten:

Um die Performancedaten 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.
- Problembenachrichtigungen einrichten, um Benachrichtigungen über potenzielle 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 Standardmengen von Datenpunkten enthalten.

1. Die SNMP-OID ermitteln:

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

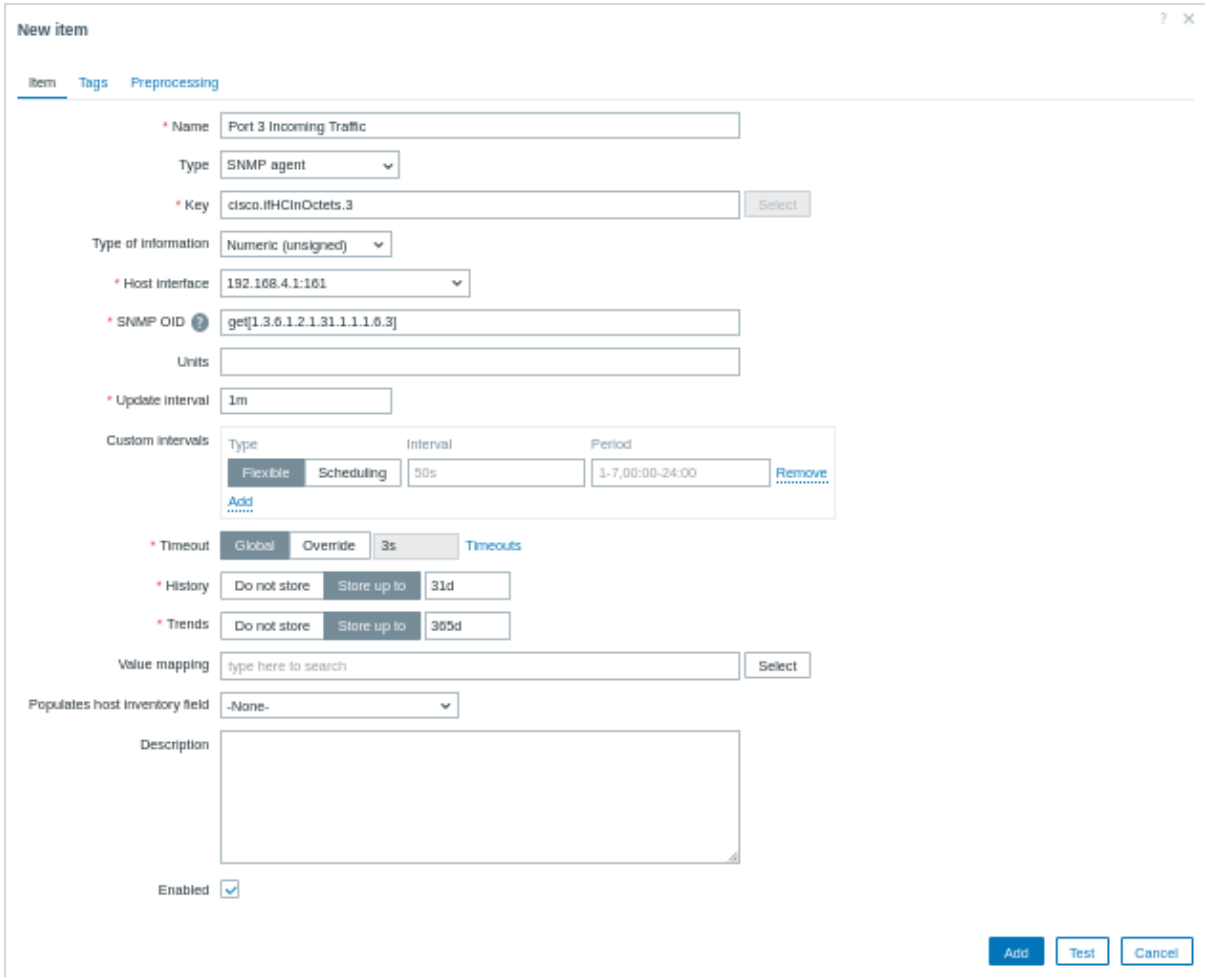
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

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

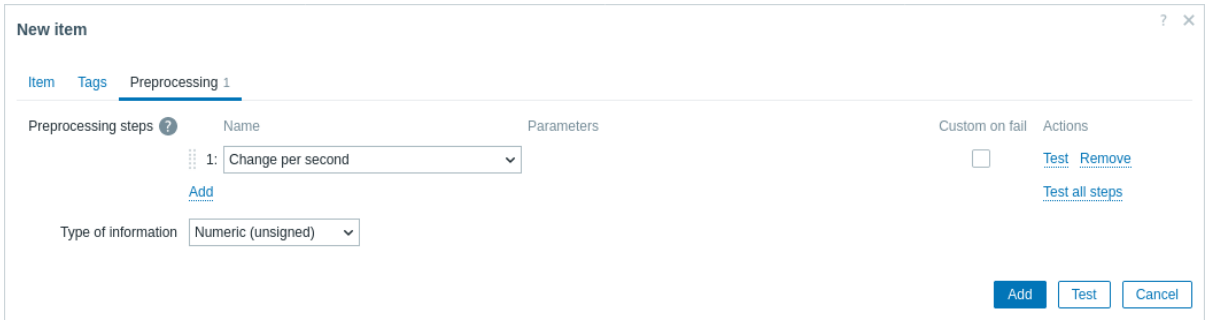
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Einen SNMP-Datenpunkt erstellen:

- Navigieren Sie zu *Datensammlung* > *Hosts* und klicken Sie beim SNMP-Host auf den Reiter *Datenpunkte* und dann auf *Datenpunkt erstellen*.
- *Name*: Geben Sie einen aussagekräftigen Namen ein (z. B. „Eingehender Datenverkehr auf Port 3“).
- *Typ*: Wählen Sie *SNMP agent* aus.
- *Schlüssel*: Geben Sie einen aussagekräftigen Schlüssel an (z. B. `cisco.ifHCInOctets.3`).
- *Host-Schnittstelle*: 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 asynchron einen Teilbaum von Werten abzurufen.



- *Vorverarbeitung* (falls erforderlich): Wenn der Datenpunkt einen kumulativen Zähler zurückgibt (z. B. Schnittstellenverkehr), wechseln Sie zum Reiter *Vorverarbeitung* und fügen Sie einen Vorverarbeitungsschritt wie „Änderung pro Sekunde“ 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 Bei der Arbeit mit SNMP müssen Sie möglicherweise zwischen numerischen OIDs und den entsprechenden MIB-Namen umwandeln. 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 kann folgende Ausgabe liefern:

```
.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` zurück in ihren MIB-Namen zu übersetzen, führen Sie Folgendes aus:

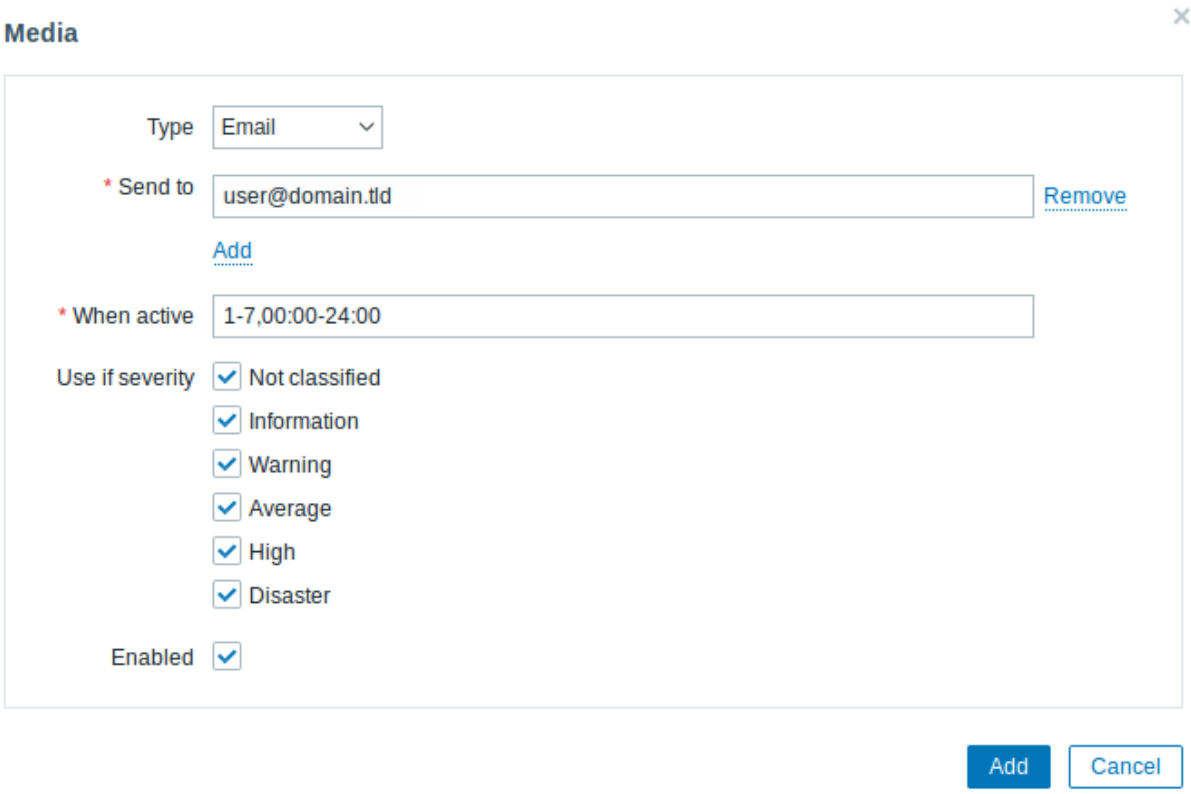
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

Dieser Befehl kann folgende Ausgabe liefern:

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

Problembenachrichtigungen 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.

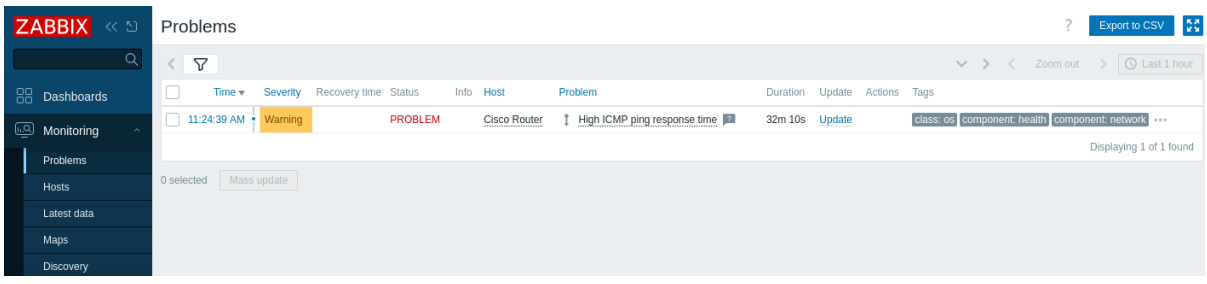


2. Folgen Sie der Anleitung für Empfangen einer Problembenachrichtigung.

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

Testen Sie Ihre Konfiguration Um sicherzustellen, dass Zabbix Probleme mit der Netzwerkleistung korrekt erkennt, simulieren Sie ein echtes Problem, indem Sie den Schwellenwert für die ICMP-Ping-Antwortzeit erhöhen.

1. Öffnen Sie die Konfiguration Ihres Hosts „Cisco Router“ in Zabbix.
2. Wechseln Sie zur Registerkarte *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. Legen Sie einen sehr niedrigen Wert fest (z. B. 0.001), damit ein Alarm ausgelöst wird, wenn die Ping-Antwort diesen Wert überschreitet.
5. Klicken Sie auf *Update*, um die Änderungen anzuwenden.
6. Warten Sie einen Moment, bis Zabbix das simulierte Problem erkennt.
7. Wechseln Sie zu *Monitoring > Problems*, um zu prüfen, dass ein Alarm angezeigt wird (z. B. „High ICMP ping response time“).



Wenn Benachrichtigungen konfiguriert sind, sollten Sie außerdem eine Problembenachrichtigung erhalten.

8. Setzen Sie den Makrowert 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 der SNMP-Überwachung Wenn Sie feststellen, dass das SNMP-Symbol im Zabbix Frontend ROT angezeigt wird oder keine Daten erfasst werden, versuchen Sie die folgenden Schritte:

1. Überprüfen Sie die SNMP-Konnektivität.

Führen Sie für SNMPv2 den folgenden Befehl auf Ihrem Zabbix Server aus:

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

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

Geben Sie für SNMPv3 die entsprechenden SNMPv3-Zugangsdaten an:

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

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

2. Stellen Sie sicher, dass MIB-Dateien installiert und wie in den Voraussetzungen beschrieben aktiviert sind. Um dies sicherzustellen, darf der folgende Befehl bei der Abfrage eines Netzwerkgeräts keinen Fehler zurückgeben:

```
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 den auf Ihrem Gerät eingestellten Werten übereinstimmen. Prüfen 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 der Community-String (für SNMPv2) oder die SNMPv3-Benutzerdetails korrekt sind.

4. Vergewissern Sie sich, dass SNMP auf Ihrem Netzwerkgerät korrekt aktiviert ist. Melden Sie sich bei 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 korrekt konfiguriert ist.

5. Stellen Sie sicher, dass keine Firewalls oder Netzwerkprobleme den SNMP-Datenverkehr (normalerweise über Port 161) zwischen dem Zabbix Server und dem Gerät blockieren. 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 auf Verbindungen wartet.

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

```
sudo ufw status
```

Oder bei iptables:

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

6. Prüfen Sie die Zabbix-Server-Logdateien auf SNMP-bezogene Fehler, um das Problem genauer einzugrenzen:

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

Mit `tail -f` können Sie Log-Aktualisierungen in Echtzeit 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.

11 Windows-Ereignisprotokoll mit aktiven Prüfungen überwachen

Einführung

Diese Anleitung erklärt, wie Sie Windows-Ereignisprotokolle mit Zabbix mithilfe aktiver Prüfungen überwachen. Mit den Windows-spezifischen Datenpunktschlüsseln von Zabbix können Sie kritische Ereignisse (wie fehlgeschlagene Anmeldeversuche, Systemfehler usw.) in Echtzeit erfassen und analysieren.

Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und Netzwerkadministratoren, die Windows-Ereignisprotokolle überwachen möchten. Informationen zu erweiterten Konfigurationsoptionen finden Sie in der Dokumentation zu Windows-spezifischen Datenpunktschlüsseln.

Voraussetzungen

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

Zabbix Agent für die Überwachung des Windows-Ereignisprotokolls konfigurieren

1. Öffnen Sie `zabbix_agentd.conf` (Standardpfad `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`) auf Ihrem Windows-Host und stellen Sie sicher, dass der Parameter ServerActive auf die IP-Adresse Ihres Zabbix Server 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. Zum Beispiel:

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

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

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

3. Prüfen Sie, dass der Windows-Host läuft:

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

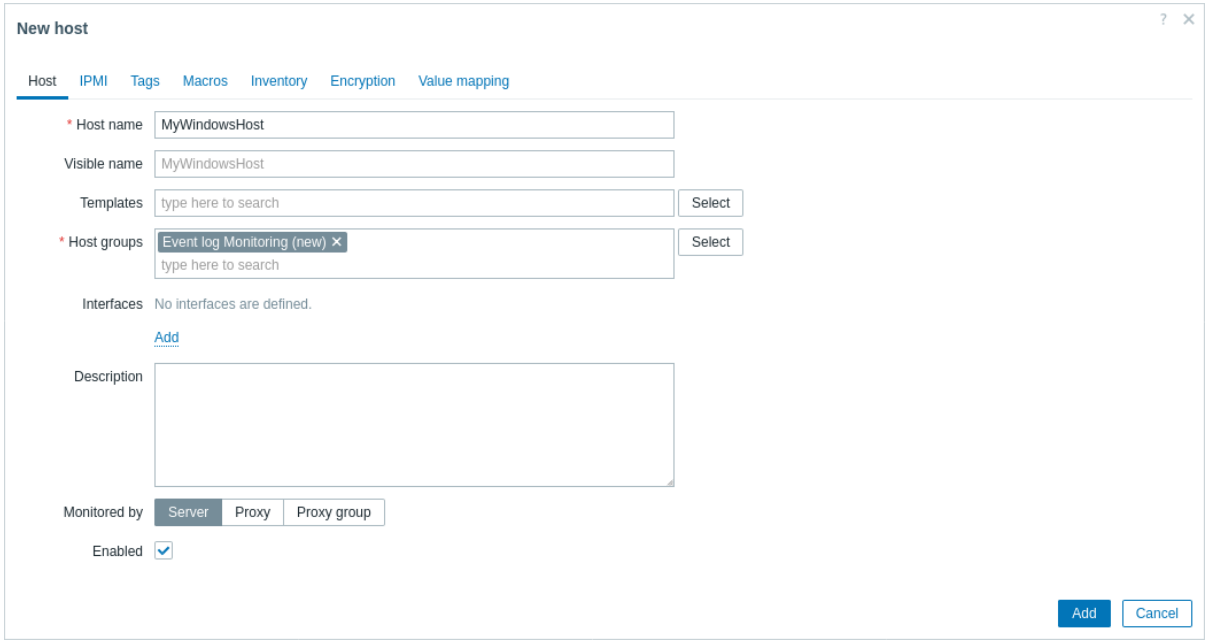
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Zabbix Frontend konfigurieren

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

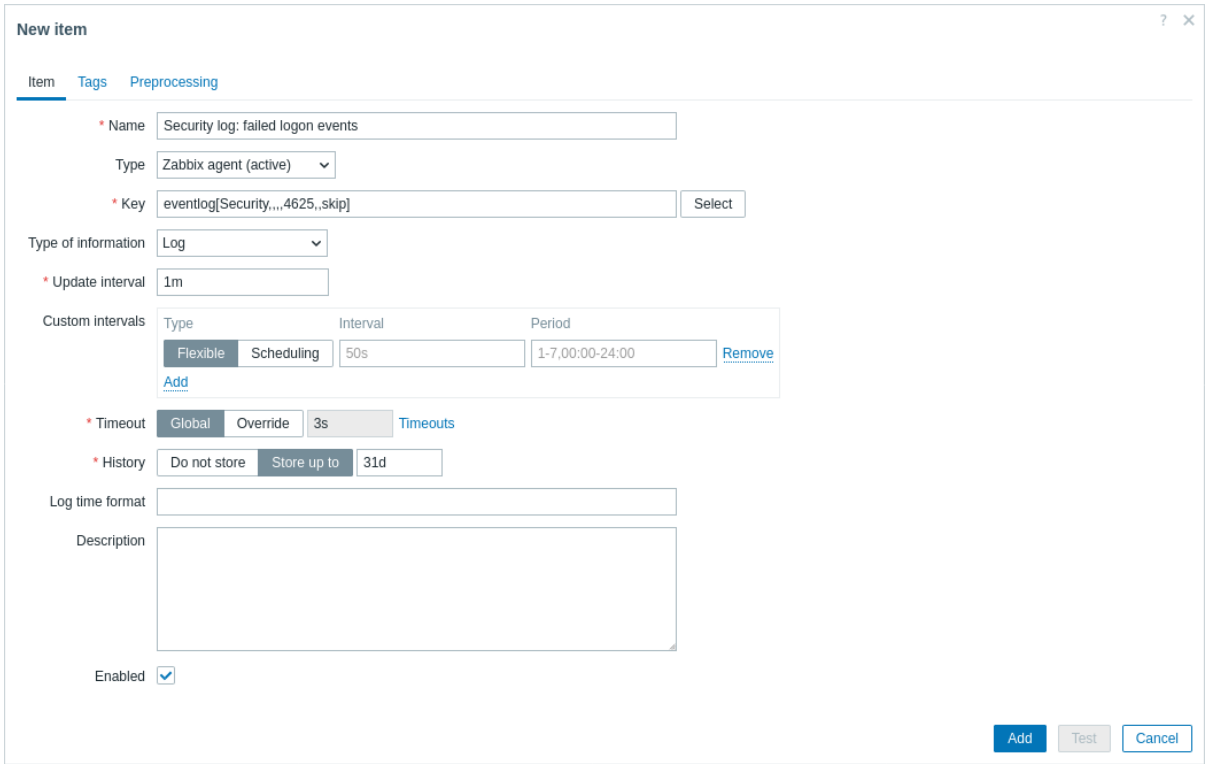
- Geben Sie im Feld *Host name* einen Host-Namen ein (z. B. „MyWindowsHost“).
- Geben Sie im Feld *Host groups* eine Host-Gruppe ein oder wählen Sie eine aus (z. B. „Event log Monitoring“).
- Klicken Sie auf *Add*, um den konfigurierten Host zu speichern.

> **Note:**
> Im Feld *Templates* können Sie die Vorlage „Windows by Zabbix Agent active“ hinzufügen, um die Fehlerbehebung 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 des Ereignisprotokolls).
- Verwenden Sie im Feld *Key* den eventlog-Datenpunktschlüssel. Um beispielsweise fehlgeschlagene Anmeldeversuche (Ereignis-ID: 4625) im Sicherheitsprotokoll zu überwachen und Einträge zu ignorieren, die älter als die letzte Prüfung des Datenpunkts sind (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.



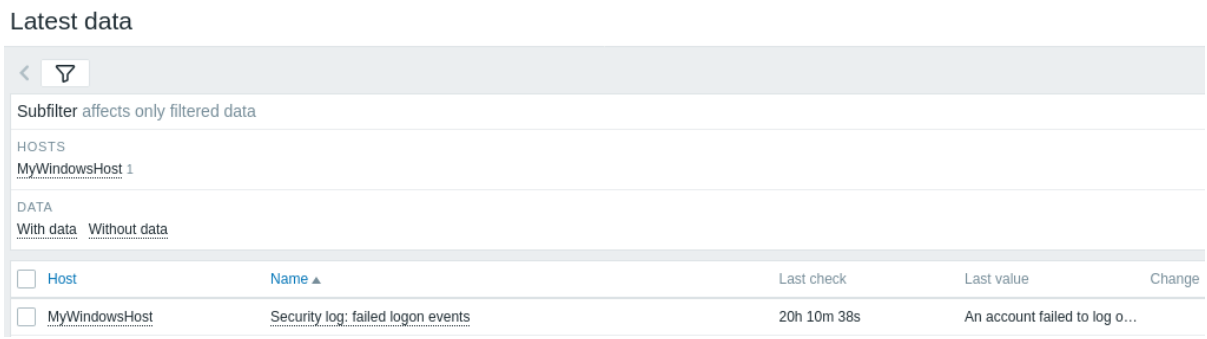
3. Klicken Sie auf *Add*, um den Datenpunkt zu speichern.

Gesammelte Metriken testen und anzeigen

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

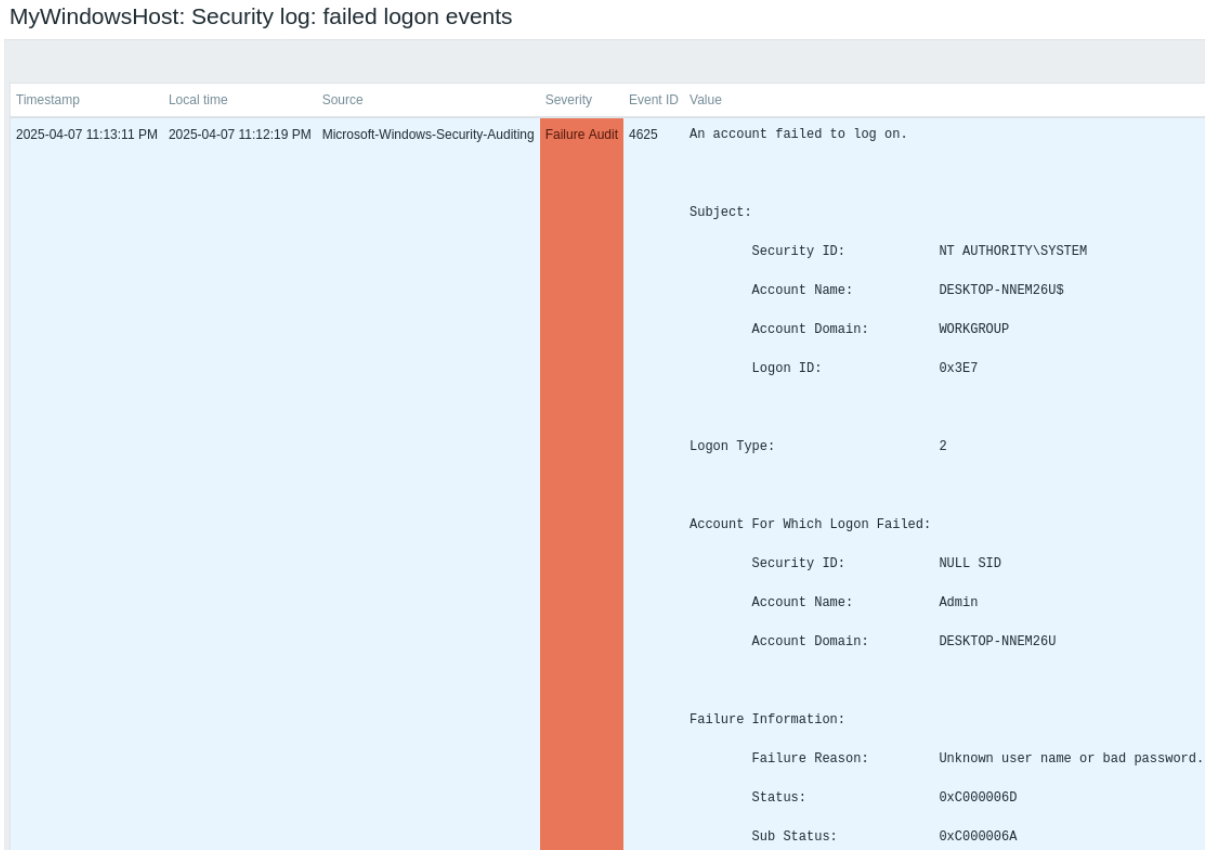
Anschließend können Sie die gesammelten Protokolle im Zabbix Frontend anzeigen:

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 Fehlerbehebung in dieser Anleitung fort.

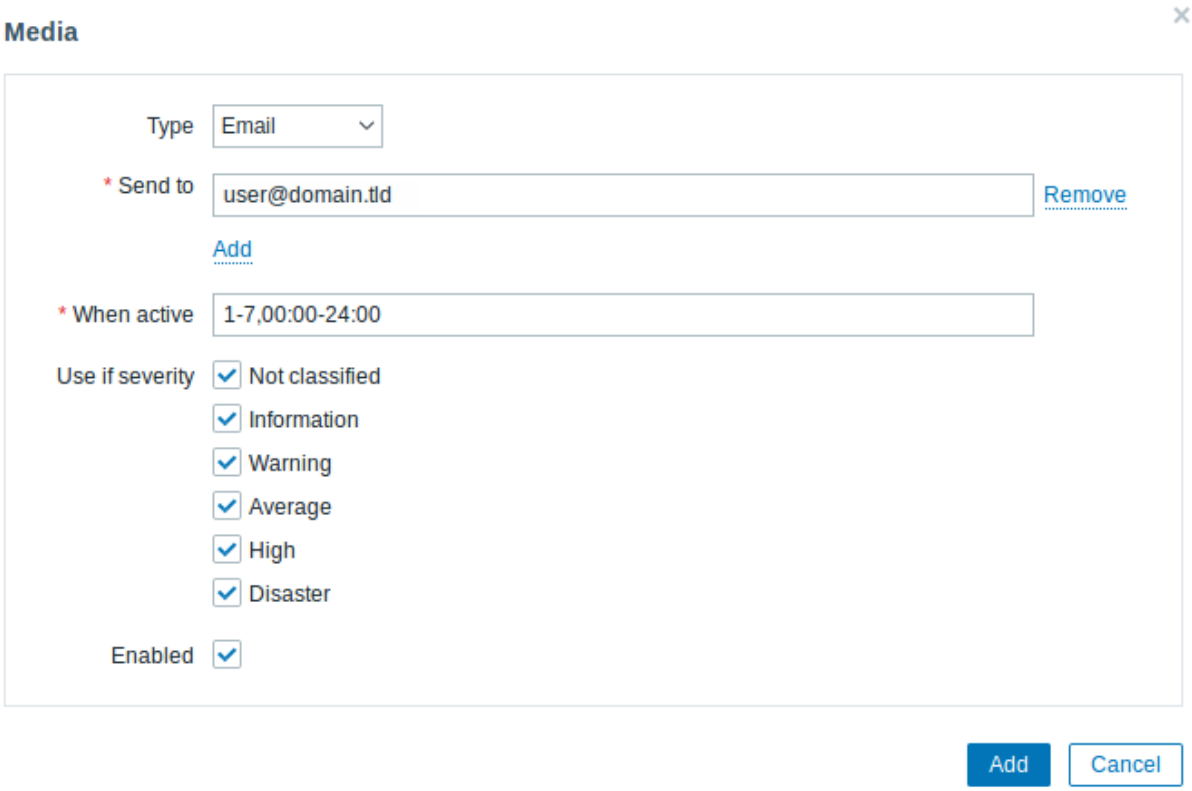
Problembenachrichtigungen einrichten

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

1. Navigieren Sie zu Datensammlung > 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 für Empfangen einer Problembenachrichtigung.

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

Fehlerbehebung

Wenn bei der Erfassung oder Anzeige 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 vergewissern Sie sich, dass es eine ACCEPT-Regel für den TCP-Port 10051 gibt.

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

Siehe auch:

- Erstellen eines Datenpunkts – erfahren Sie, wie Sie zusätzliche Metriken hinzufügen.
- Zabbix Agent unter Microsoft Windows – detaillierte Installationsanweisungen.
- Windows mit Zabbix Agent überwachen – eine umfassende Anleitung zum Einrichten einer grundlegenden Überwachung für Windows-Rechner mit Zabbix Agent.
- Windows-spezifische Datenpunktschlüssel – detaillierte Informationen zu Windows-spezifischen Datenpunktschlüsseln, die von Zabbix Agents unterstützt werden, einschließlich solcher 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.

Was ist Zabbix

Zabbix wurde von Alexei Vladishev entwickelt und wird derzeit aktiv von Zabbix SIA weiterentwickelt und unterstützt.

Zabbix ist eine Open-Source-Lösung für verteiltes Monitoring auf Enterprise-Niveau.

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 mehr überwacht. Zabbix verwendet einen flexiblen

Benachrichtigungsmechanismus, mit dem Benutzer E-Mail-basierte Warnmeldungen für praktisch jedes Ereignis konfigurieren können. Dies ermöglicht eine schnelle Reaktion auf Serverprobleme. Zabbix bietet hervorragende Funktionen für Berichterstellung und Datenvisualisierung auf Grundlage der gespeicherten Daten. Dadurch ist Zabbix ideal für die Kapazitätsplanung.

Zabbix unterstützt sowohl Polling als auch Trapping. Auf alle Zabbix-Berichte und -Statistiken sowie auf 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 wie auch für große Unternehmen mit einer Vielzahl von Servern.

Zabbix ist kostenlos. Zabbix wird unter der Lizenz AGPL-3.0 geschrieben und verbreitet. Das bedeutet, dass der Quellcode frei verbreitet wird und der Allgemeinheit zur Verfügung steht.

Kommerzieller Support ist verfügbar und wird von Zabbix Company und ihren Partnern weltweit bereitgestellt.

Erfahren Sie mehr über die 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 Aufgaben sind unten beschrieben.

Server

Der Zabbix Server ist die zentrale Komponente, an die Agenten Verfügbarkeits- und Integritätsinformationen sowie Statistiken melden. Der Server ist das zentrale Repository, in dem alle Konfigurations-, Statistik- und Betriebsdaten gespeichert werden.

Datenbankspeicherung

Alle Konfigurationsinformationen sowie die von Zabbix erfassten Daten werden in einer Datenbank gespeichert.

Weboberfläche

Für einen einfachen Zugriff auf Zabbix von überall und von jeder Plattform aus wird eine webbasierte Oberfläche bereitgestellt. Die Oberfläche ist Teil des Zabbix-Servers und läuft in der Regel (aber nicht zwingend) auf derselben physischen Maschine wie der Server.

Proxy

Zabbix Proxy kann Performance- und Verfügbarkeitsdaten im Auftrag des Zabbix Server erfassen. Ein Proxy ist ein optionaler Bestandteil einer Zabbix-Bereitstellung; er kann jedoch sehr nützlich sein, um die Last eines einzelnen Zabbix Server zu verteilen.

Agent

Zabbix-Agents werden auf Überwachungszielen eingesetzt, um lokale Ressourcen und Anwendungen aktiv zu überwachen und die erfassten Daten an den Zabbix-Server zu melden. Seit Zabbix 4.4 sind zwei Arten von Agents verfügbar: der Zabbix Agent (leichtgewichtig, auf vielen Plattformen unterstützt, in C geschrieben) und Zabbix Agent 2 (besonders flexibel, mit Plugins einfach erweiterbar, in Go geschrieben).

Datenfluss Außerdem 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 zuerst einen Host erstellen. Am anderen Ende des Zabbix-Spektrums müssen Sie zuerst einen Datenpunkt haben, um einen Auslöser zu erstellen. Sie müssen einen Auslöser haben, um eine Aktion zu erstellen. Wenn Sie also eine Warnung erhalten möchten, dass Ihre CPU-Last auf *Server X* zu hoch ist, müssen Sie zuerst einen Host-Eintrag für *Server X* erstellen, gefolgt von einem Datenpunkt zur Überwachung seiner CPU, dann einem Auslöser, der aktiviert wird, wenn die CPU-Auslastung zu hoch ist, gefolgt von einer Aktion, die Ihnen eine E-Mail sendet. Auch wenn das nach vielen Schritten aussehen mag, ist es mit der Verwendung von Vorlagen in Wirklichkeit nicht so. Aufgrund dieses Designs ist es jedoch möglich, eine sehr flexible Einrichtung zu erstellen.

Funktionen Zabbix ist eine hochintegrierte Lösung zur Netzwerküberwachung und bietet eine Vielzahl von Funktionen in einem einzigen Paket.

Datenerfassung

- Verfügbarkeits- und Leistungsprüfungen
- Unterstützung für SNMP (sowohl Traps als auch Polling), IPMI, JMX, VMware-Monitoring
- benutzerdefinierte Prüfungen
- Erfassung gewünschter Daten in benutzerdefinierten Intervallen
- durchgeführt durch Server/Proxy und durch Agenten

Flexible Schwellenwertdefinitionen

- Sie können sehr flexible Problemschwellenwerte definieren, sogenannte Auslöser, die auf Werte aus der Backend-Datenbank verweisen

Hochgradig konfigurierbare Benachrichtigungen

- das Senden von Benachrichtigungen kann für Eskalationsplan, Empfänger und Medientyp angepasst werden
- Benachrichtigungen können mithilfe von Makrovariablen aussagekräftig und hilfreich gestaltet werden
- automatische Aktionen umfassen Remote-Befehle

Echtzeit-Diagramme

- überwachte Datenpunkte werden mithilfe der integrierten Diagrammfunktion sofort grafisch dargestellt

Funktionen zur Webüberwachung

- 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
- übergeordnete (geschäftsbezogene) Ansicht überwachter Ressourcen

Speicherung historischer Daten

- Daten werden in einer Datenbank gespeichert
- konfigurierbare Historie
- integriertes Housekeeping-Verfahren

Einfache Konfiguration

- überwachte Geräte als Hosts hinzufügen
- Hosts werden zur Überwachung herangezogen, sobald sie sich in der Datenbank befinden
- Vorlagen auf überwachte Geräte anwenden

Verwendung von Vorlagen

- Gruppierung von Prüfungen in Vorlagen
- Vorlagen können andere Vorlagen erben

Netzwerkerkennung

- automatische Erkennung von Netzwerkgeräten
- automatische Agent-Registrierung
- Erkennung von Dateisystemen, Netzwerkschnittstellen und SNMP-OIDs

Schnelles Web-Interface

- ein webbasiertes Frontend in PHP
- von überall 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.

Umfangreich ausgestatteter und leicht erweiterbarer Agent

- auf Überwachungszielen bereitgestellt
- kann sowohl unter Linux als auch unter Windows bereitgestellt werden

Binäre Daemons

- in C geschrieben, für hohe Leistung und geringen Speicherbedarf
- leicht portierbar

Bereit für komplexe Umgebungen

- Remote-Monitoring wird durch die Verwendung eines Zabbix Proxy vereinfacht

Glossar In diesem Abschnitt können Sie die Bedeutung einiger Begriffe kennenlernen, die in Zabbix häufig verwendet werden.

Host

- jedes physische oder virtuelle Gerät, jede Anwendung, jeder Dienst oder jede andere logisch zusammenhängende Sammlung überwachter Parameter.

Host-Gruppe

- eine logische Gruppierung von Hosts. Host-Gruppen werden verwendet, wenn Zugriffsrechte auf Hosts für verschiedene Benutzergruppen zugewiesen werden.

Datenpunkt

- ein bestimmter Datenteil, den Sie von einem Host empfangen möchten, eine Datenmetrik.

Wert- Vorverarbeitung

- eine Umwandlung des empfangenen Metrikwerts vor dem Speichern in der Datenbank.

Auslöser

- ein logischer Ausdruck, der einen Problemschwellenwert definiert und verwendet wird, um in Datenpunkten empfangene Daten zu "bewerten".

Wenn empfangene Daten über dem Schwellenwert liegen, wechseln Auslöser vom Status 'Ok' in den Status 'Problem'. Wenn empfangene Daten unter dem Schwellenwert liegen, bleiben Auslöser im Status 'Ok' oder kehren in ihn zurück.

Vorlage

- eine Menge von Entitäten (Datenpunkte, Auslöser, Diagramme, Low-Level-Discovery- Regeln, Webszenarien), die bereit sind, auf einen oder mehrere Hosts angewendet zu werden.

Die Aufgabe von Vorlagen besteht darin, die Bereitstellung von Überwachungsaufgaben auf einem Host zu beschleunigen und außerdem Massenänderungen an Überwachungsaufgaben zu erleichtern. Vorlagen werden direkt mit einzelnen Hosts verknüpft.

Vorlagengruppe

- eine logische Gruppierung von Vorlagen. Vorlagengruppen werden verwendet, wenn Zugriffsrechte auf Vorlagen für verschiedene Benutzergruppen zugewiesen werden.

Ereignis

- ein einzelnes Vorkommnis von etwas, das Aufmerksamkeit verdient, wie etwa ein Auslöser, der seinen Status ändert, oder eine Discovery-/Agent-Autoregistrierung, die stattfindet.

Ereignis-Tag

- eine vordefinierte Kennzeichnung für das Ereignis. Sie kann bei der Ereignis- Korrelation, der Granularität von Berechtigungen usw. verwendet werden.

Ereigniskorrelation

- eine Methode, Probleme flexibel und präzise mit ihrer Behebung zu korrelieren.

Sie können zum Beispiel festlegen, dass ein von einem Auslöser gemeldetes Problem durch einen anderen Auslöser behoben werden kann, der sogar eine andere Methode der Datenerfassung verwenden kann.

Problem

- ein Auslöser, der sich im Status "Problem" befindet.

Problemaktualisierung

- von Zabbix bereitgestellte Optionen zur Problemverwaltung, wie das Hinzufügen eines Kommentars, das Bestätigen, das Ändern des Schweregrads oder das manuelle Schließen.

Aktion

- ein vordefiniertes Mittel, auf ein Ereignis zu reagieren.

Eine Aktion besteht aus Operationen (z. B. dem Senden einer Benachrichtigung) und Bedingungen (*wann* die Operation ausgeführt wird)

Eskalation

- ein benutzerdefiniertes Szenario zur Ausführung von Operationen innerhalb einer Aktion; eine Abfolge des Sendens von Benachrichtigungen/Ausführens von Remote-Befehlen.

Makro

- ein Platzhalter für eine Variable, der je nach Kontext zu einem bestimmten Wert aufgelöst wird.

Medien

- ein Mittel zur Zustellung von Benachrichtigungen; Zustellungskanal.

Benachrichtigung

- eine Nachricht über ein Ereignis, die über den gewählten Medienkanal an einen Benutzer gesendet wird.

Remote- Befehl

- ein vordefinierter Befehl, der unter bestimmten Bedingungen automatisch auf einem überwachten Host ausgeführt wird.

Webszenario

- eine oder mehrere HTTP-Anfragen zur Überprüfung der Verfügbarkeit einer Web- Site.

Frontend

- die mit Zabbix bereitgestellte Weboberfläche.

Dashboard

- ein anpassbarer Bereich der Weboberfläche, der Zusammenfassungen und Visualisierungen wichtiger Informationen in visuellen Einheiten namens 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

- Mit der Zabbix API können Sie das JSON RPC-Protokoll verwenden, um Zabbix-Objekte (wie Hosts, Datenpunkte, Diagramme und andere) zu erstellen, zu aktualisieren und abzurufen oder andere benutzerdefinierte Aufgaben auszuführen.

Zabbix Server

- ein zentraler Prozess der Zabbix-Software, der die Überwachung durchführt, mit Zabbix Proxys und Agenten interagiert, Auslöser berechnet, Benachrichtigungen sendet; ein zentrales Daten-Repository.

Zabbix Proxy

- ein Prozess, der Daten im Auftrag des Zabbix Server sammeln kann und dem Server einen Teil der Verarbeitungslast abnimmt.

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 Agent zur aktiven Überwachung lokaler Ressourcen und Anwendungen, die die Verwendung benutzerdefinierter Plugins für die Überwachung ermöglicht.

> **Attention:**
> Da Zabbix Agent 2 viele Funktionen mit Zabbix Agent gemeinsam hat, steht der Begriff ”Zabbix Agent” in der Dokumentation für beide – Zabbix Agent und Zabbix Agent 2 –, wenn das funktionale Verhalten identisch ist. 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.

Agent-Autoregistrierung

- ein automatisierter Prozess, bei dem ein Zabbix Agent selbst als Host registriert und die Überwachung gestartet wird.

Netzwerkerkennung

- automatisierte Erkennung von Netzwerkgeräten.

Low-Level-Discovery

- automatisierte Erkennung von Low-Level-Entitäten auf einem bestimmten Gerät (z. B. Dateisysteme, Netzwerkschnittstellen usw.).

Low-Level-Discovery- Regel

- *eine Menge von Definitionen für die automatisierte Erkennung von Low-Level-Entitäten* auf einem Gerät.

Datenpunkt- Prototyp

- *eine Metrik mit bestimmten Parametern als Variablen, bereit für Low-Level-Discovery*. 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, bereit für Low-Level-Discovery*. 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 in der Low-Level-Discovery verwendet – Diagrammprototypen, Host-Prototypen, Host-Gruppen-Prototypen.

Zabbix Cloud

Zabbix Cloud

Entdecken Sie die verfügbaren Konfigurationsoptionen und Zugriffskontrollen für Organisationen und Knoten in Zabbix Cloud.

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.

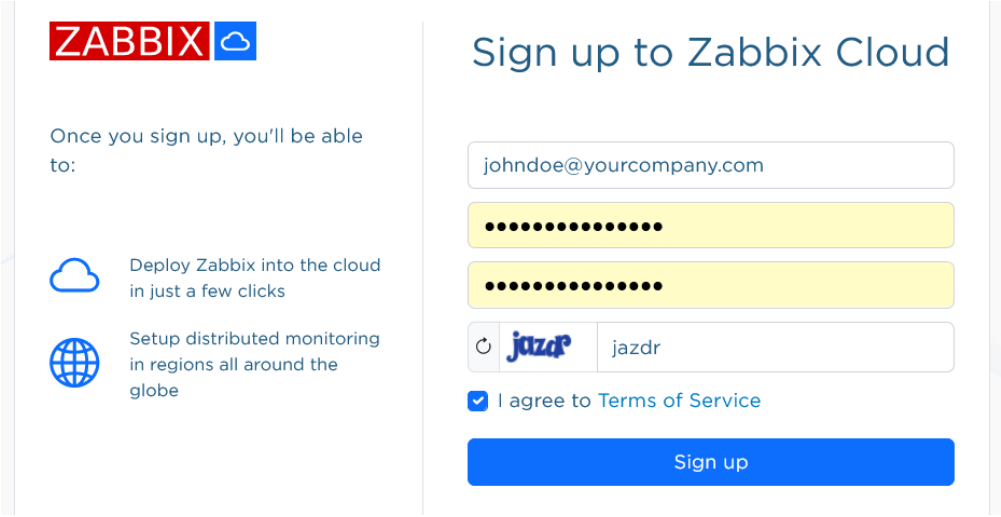
Zabbix in der Cloud bereitstellen

Um Zabbix in der Cloud zu starten, folgen Sie diesem Ablauf:

- Melden Sie sich bei cloud.zabbix.com an oder registrieren Sie sich
- Erstellen Sie einen Cloud-Knoten
- Beginnen Sie mit der Nutzung von Zabbix in der Cloud

Anmelden oder registrieren Sie können sich unter cloud.zabbix.com mit einem bestehenden Google-, GitHub- oder Microsoft-Konto anmelden.

Alternativ können Sie die Seite sign up aufrufen, um sich zu registrieren.



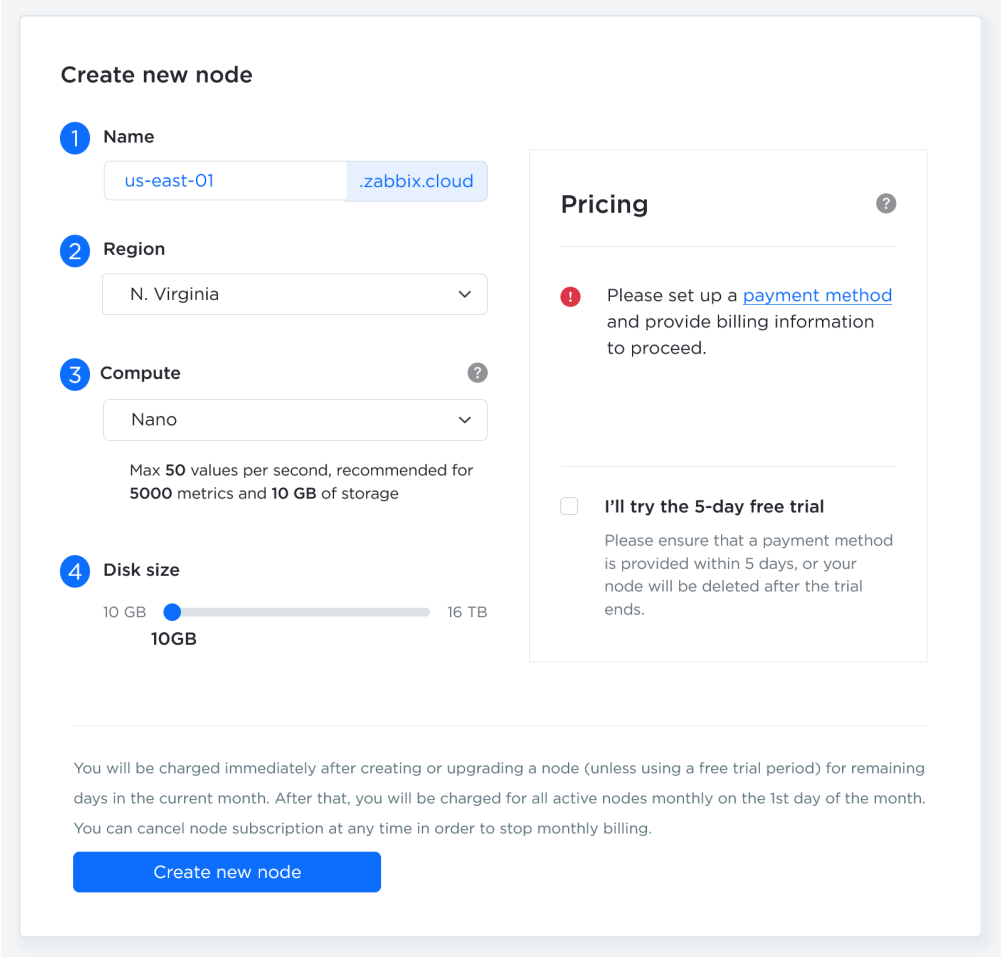
Nach der Registrierung erhalten Sie eine Bestätigung Ihres neuen Kontos per E-Mail.

Gehen Sie zu cloud.zabbix.com, um sich anzumelden. Bei der Anmeldung werden Sie nach einem Einmalpasswort gefragt. Dieses Passwort wird Ihnen per E-Mail zugesendet.

Cloud-Knoten erstellen Ein Cloud-Knoten ist eine Zabbix-Instanz, die in der Cloud betrieben wird. Sie umfasst den Zabbix Server, die Datenbank und das Web-Frontend.

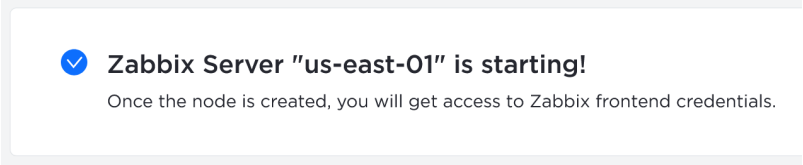
Nach der Anmeldung sehen Sie, dass standardmäßig bereits eine *Private*- Organisation für Ihr Konto vorhanden ist. Sie können die Standardorganisation später umbenennen.

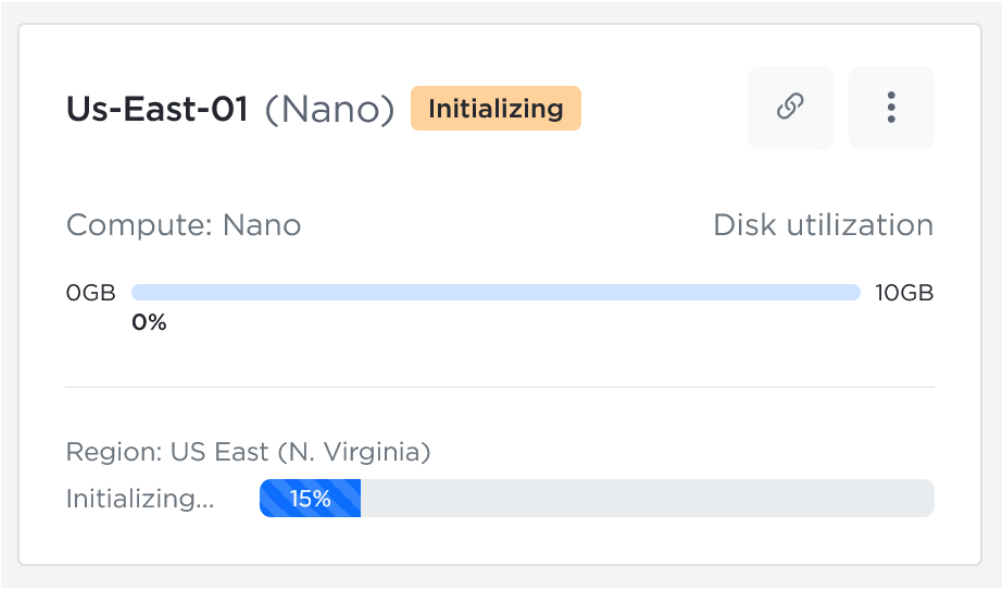
Klicken Sie auf *Create New Node*. Ein Formular zur Knotenkonfiguration wird geöffnet.

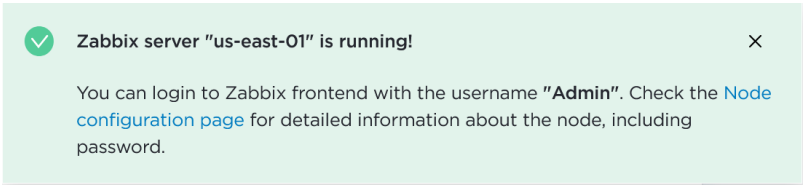


Parameter	Beschreibung
Name	Wählen Sie einen eindeutigen Knotennamen aus. Der Knotenname bildet eine eindeutige URL für den Cloud-Server/das Frontend im Format `<your-name>.zabbix.cloud`.
Region	Wählen Sie den nächstgelegenen Rechenzentrumsstandort, um die Netzwerklatenz zu minimieren.
Compute	Wählen Sie die Rechenleistung für den Knoten aus. Um zu bestimmen, welche Ressourcenstufe Sie benötigen, siehe pricing details. Sie können später jederzeit ein Upgrade durchführen, wenn Sie mehr Leistung benötigen. Vor dem Erstellen eines Nicht-Test-Knotens ist eine Abrechnungsmethode erforderlich. Fügen Sie die Abrechnungsinformationen über das Menü *Billing* hinzu. Eine 5-tägige **kostenlose Testversion** steht zum Ausprobieren eines Knotens der Stufe *Nano* zur Verfügung. Während des Testzeitraums werden Ihnen keine Kosten berechnet. Nach dessen Ende wird Ihr Knoten jedoch entfernt, wenn er nicht in einen kostenpflichtigen Knoten umgewandelt wird.
Disk size	Der Schieberegler ist standardmäßig auf ein je nach Stufe empfohlenes Minimum eingestellt. Erhöhen Sie ihn, wenn Sie große Mengen an Verlauf oder Trends speichern. Beachten Sie, dass größere Datenträger höhere monatliche Kosten verursachen und dass Sie einen Datenträger nicht selbst verkleinern können — Sie müssen den Support kontaktieren, um ihn zu reduzieren.

Nach dem Ausfüllen des Formulars klicken Sie auf *Create New Node*. Während der Knoten initialisiert wird, wird eine entsprechende Meldung angezeigt, und der Knoten erscheint bereits unter Organizations.





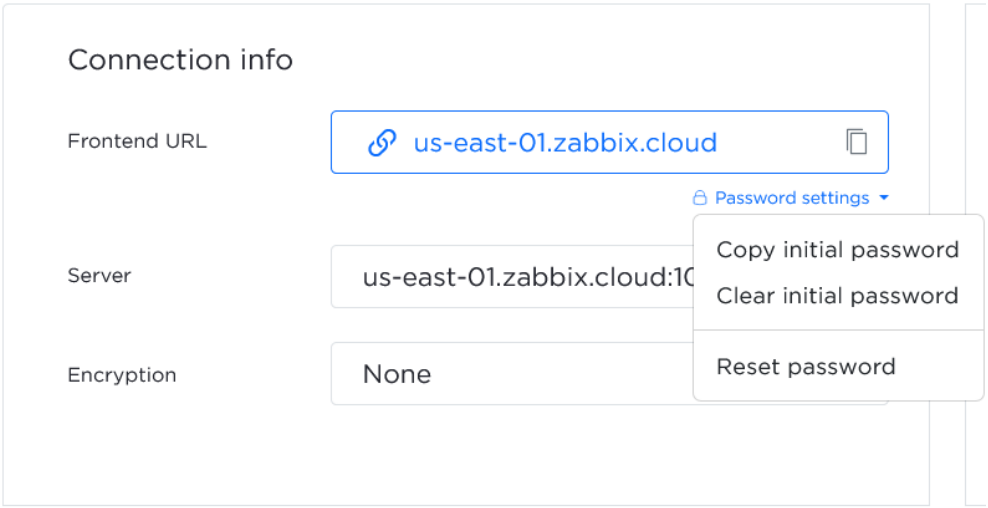


Zabbix verwenden

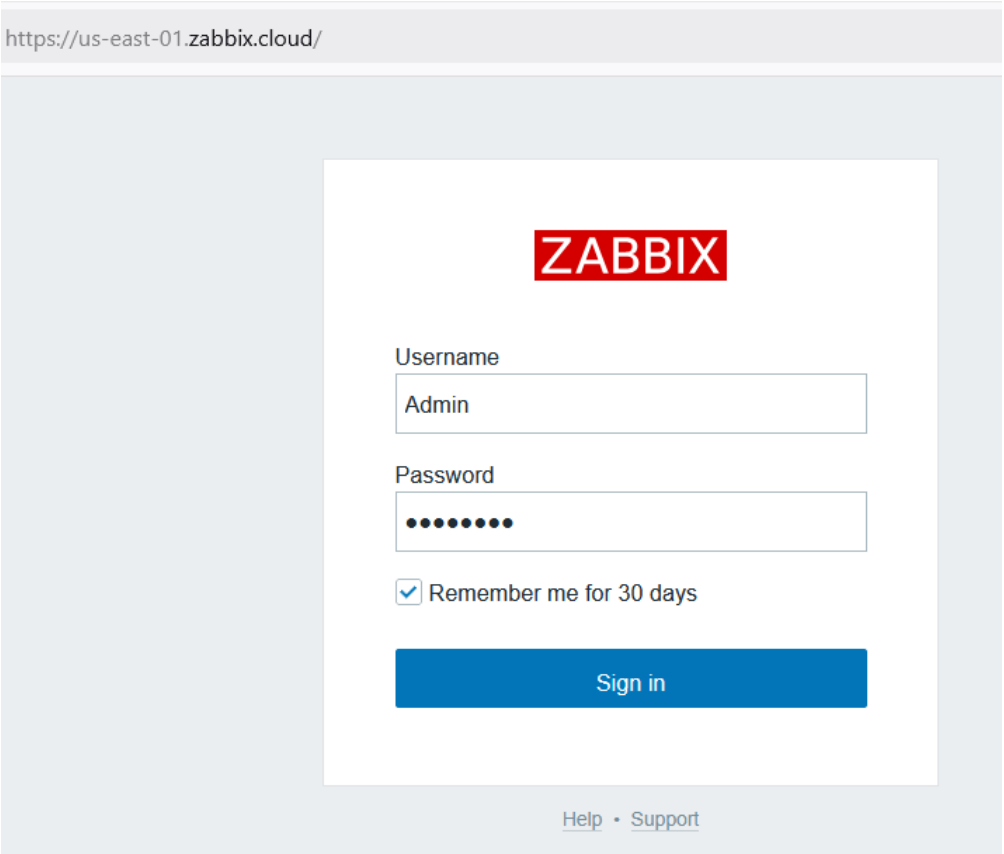
Wenn sich der Status des Knotens von *Initializing* zu *Running* ändert, können Sie sich am Zabbix Frontend anmelden.

Um auf das ursprüngliche Frontend-Passwort zuzugreifen, klicken Sie auf den erstellten Knoten.

Sie werden zu den Einstellungen des Knotens weitergeleitet (Übersicht, Zugriffsfilter, Verschlüsselung, Backups usw.). Klicken Sie unter **Connection info** auf *Password settings* und wählen Sie im Dropdown-Menü *Copy initial password* aus.



Klicken Sie auf die Frontend-URL, um das Zabbix Frontend zu öffnen (`us-east-01.zabbix.cloud`).



Melden Sie sich mit den folgenden Zugangsdaten an:

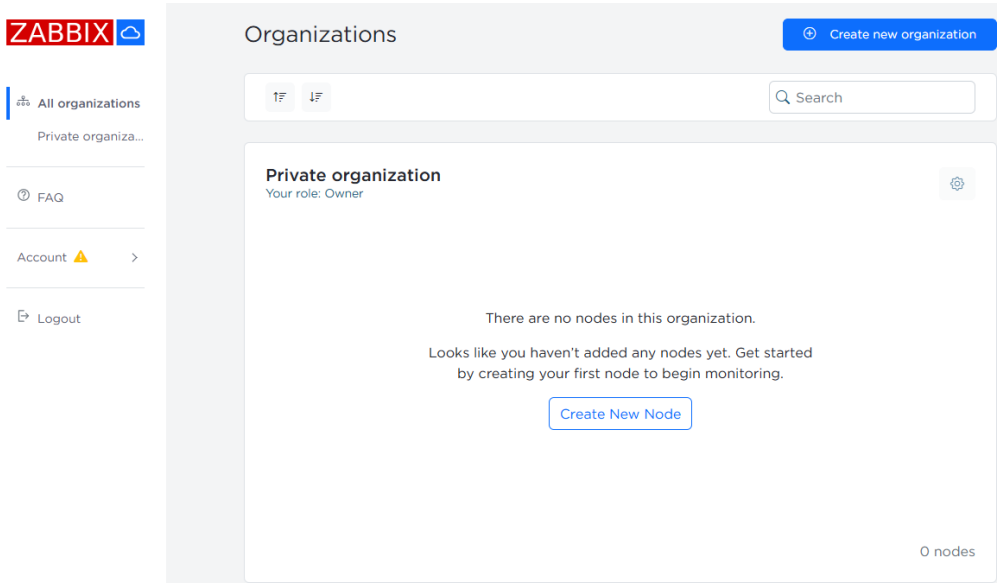
- *Username* - `Admin`
- *Password* - geben Sie das ursprüngliche Passwort ein (z. B. HzTG9t7Y)

Um mit der Überwachung zu beginnen, müssen Sie Zugriffsfilter definieren – IP-Adresse oder IP-Adressen, die eine Verbindung zum Zabbix-Cloud-Server herstellen dürfen. Mindestens ein Filter 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  Einstellungen, um auf die Verwaltungsoptionen zuzugreifen.

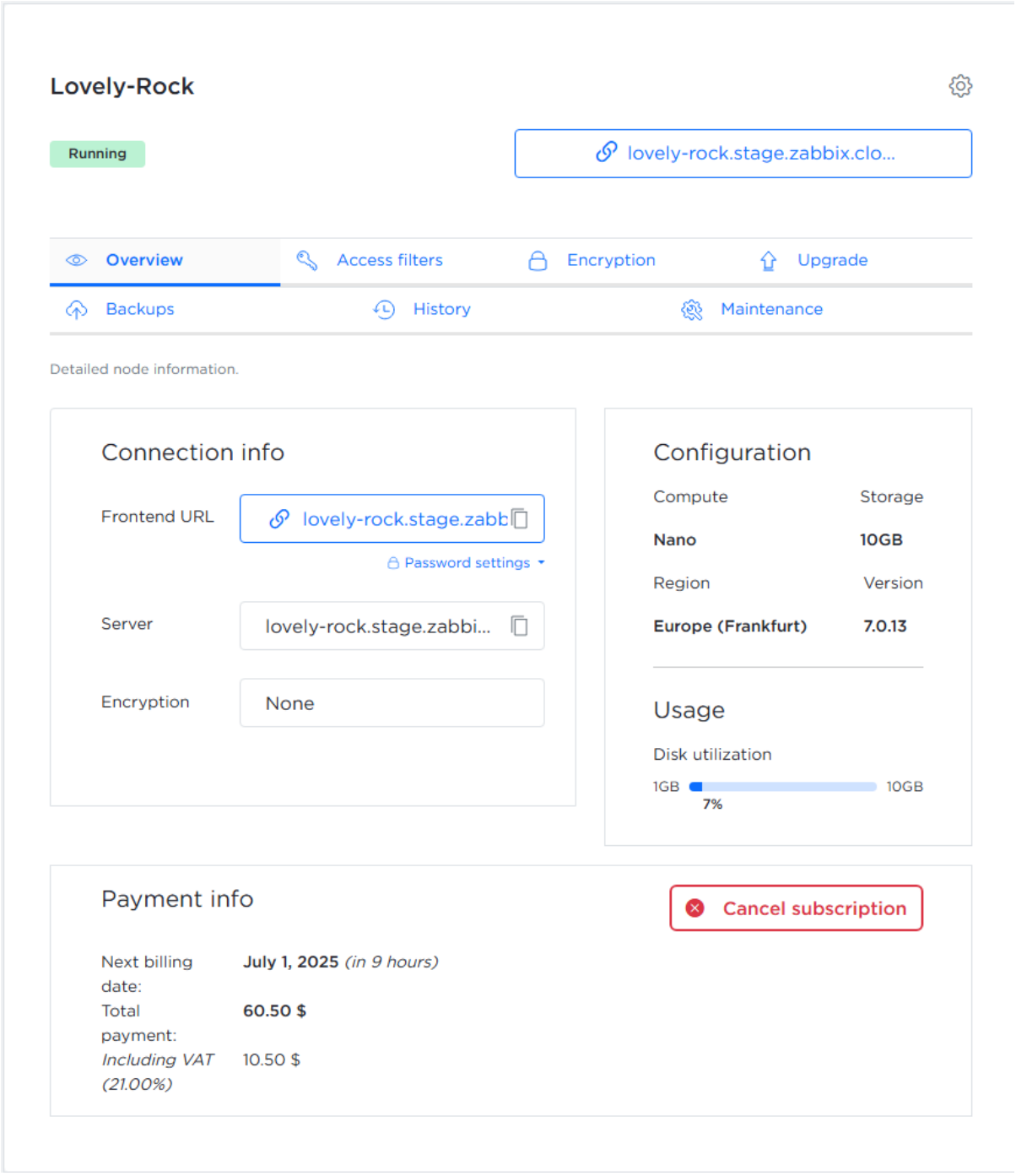
Node-Konfiguration

Übersicht

Um auf die Konfigurationsseite des Node zuzugreifen, klicken Sie auf den Node entweder unter *Alle Organisationen*, in Ihrem Organisationsmenü oder über den Link *Node configuration page*, der angezeigt wird, nachdem ein Node gestartet wurde.

In der Node-Konfigurationsansicht können Sie über jede Registerkarte einen bestimmten Aspekt Ihres Node verwalten:

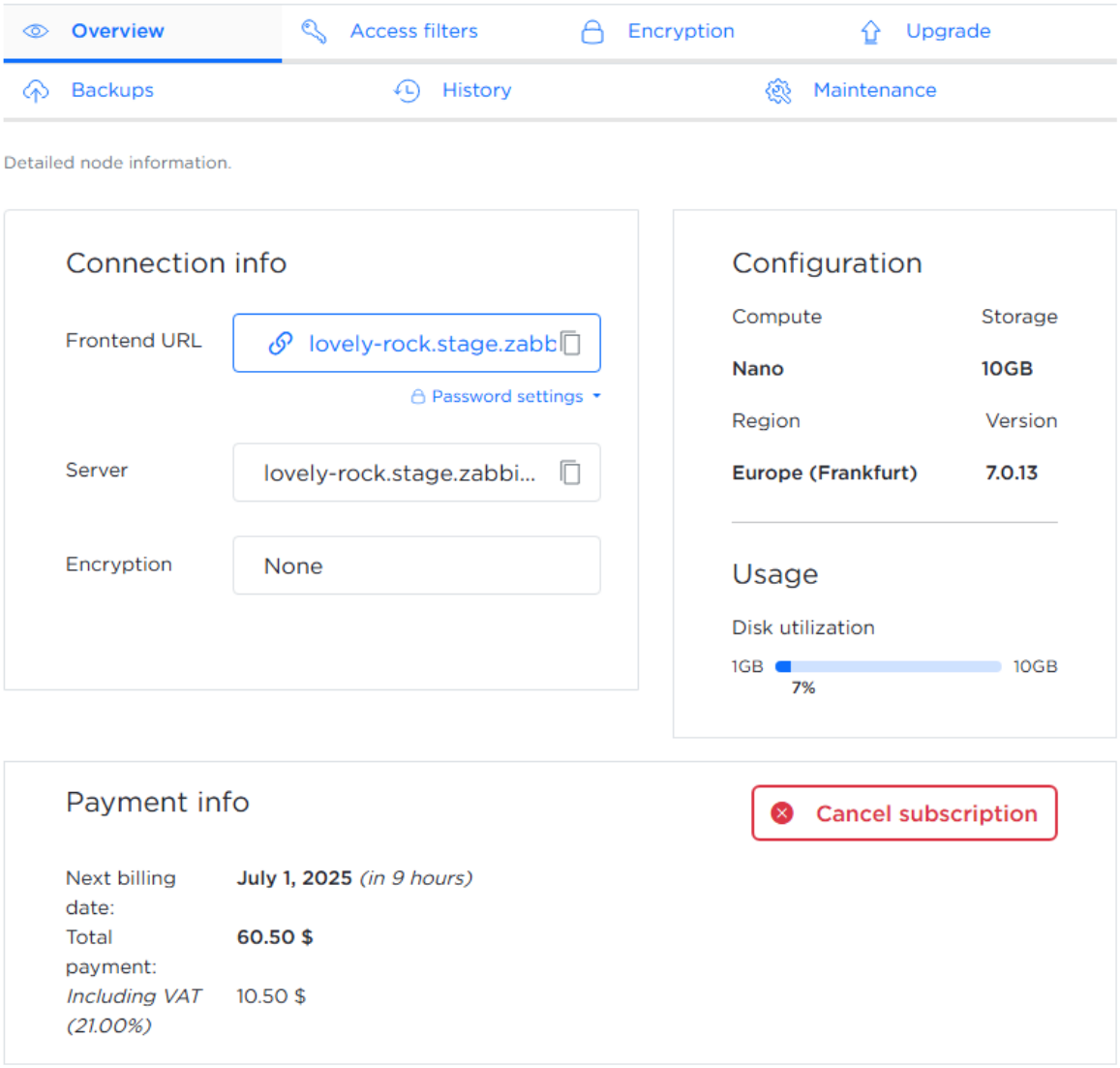
- Übersicht - Node-Details anzeigen, sein Frontend öffnen und Node-Benutzer verwalten.
- Zugriffsfilter - den Zugriff nach IP-Adresse oder Subnetz steuern.
- Verschlüsselung - TLS durch Hochladen von Zertifikaten, Schlüsseln und optionalen CRLs konfigurieren.
- Backups - Node-Snapshots erstellen, wiederherstellen und löschen; das nächste geplante Backup anzeigen.
- Verlauf - Aufbewahrungszeiträume für Audit-Protokolle, Verlauf und Trends festlegen, mit voreingestellten Einheiten und einer Aktualisierungsbegrenzung auf 3x pro Tag.
- Wartung - wöchentliche Wartungsfenster nach Zeitzone, Wochentag und Stunde definieren.
- Upgrade - Compute-/Storage-Ressourcen skalieren oder Ihren Test-/Zahlungsstatus verwalten.



Note:
Einige Registerkarten sind möglicherweise nicht verfügbar, abhängig vom Benutzertyp, der die Seite anzeigt, und vom Node-Status (Testversion/Nicht-Testversion).

Registerkarte „Übersicht“

Die Registerkarte *Übersicht* bietet detaillierte Informationen über den Knoten.



Sie ist in drei Abschnitte unterteilt:

Verbindungsinformationen:

- Anklickbare Frontend-URL mit einer Option zum Kopieren in die Zwischenablage
- Passworteinstellungen
- Aktuelle Verschlüsselungsmethode

Konfiguration und Nutzung:

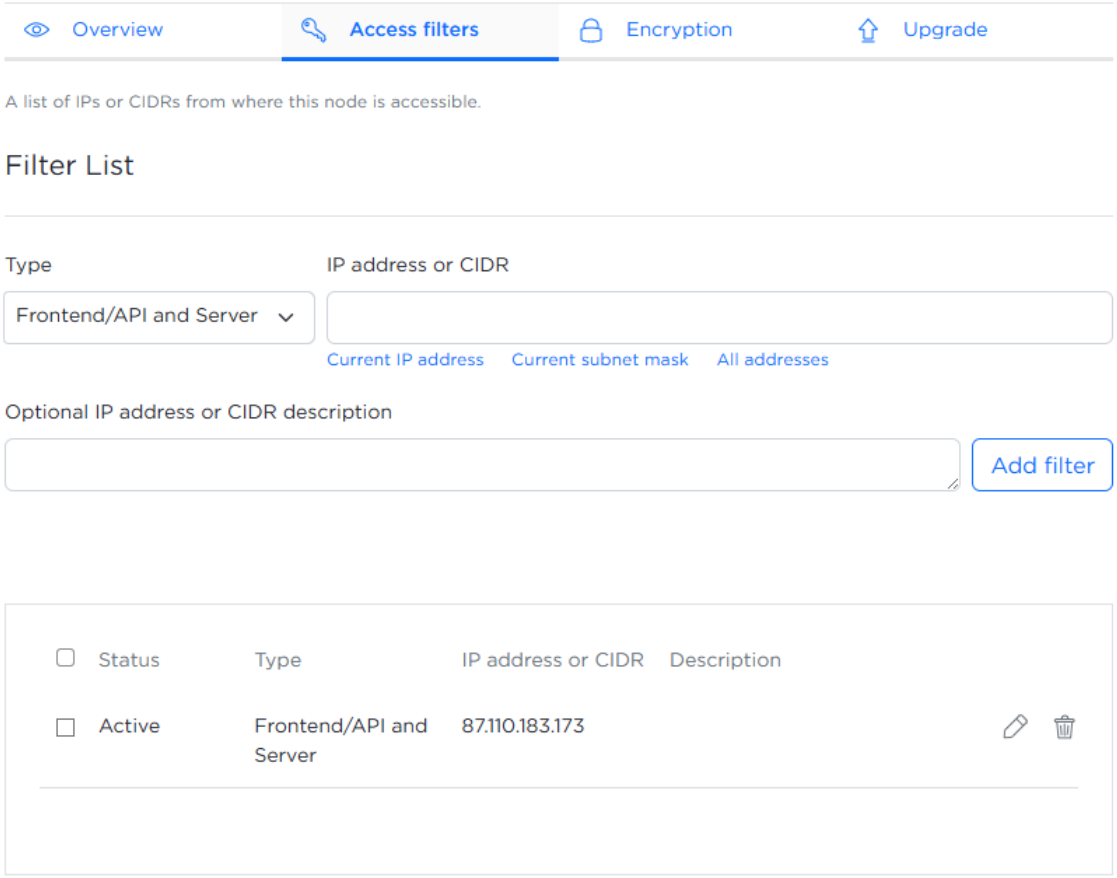
- Compute-Tier, maximaler Speicher, Region
- Version der Zabbix-Komponenten
- Festplattenauslastung

Knoteninformationen/Zahlungsinformationen:

- Ablaufdatum (für Test- und Prepaid-Knoten)
- Nach Abschluss der Initialisierung des Knotens wird abhängig vom Knotentyp eine kontextspezifische Schaltfläche angezeigt: - *Auf 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 sie eine lange Gültigkeitsdauer haben und nicht gekündigt werden können

Registerkarte „Zugriffsfilter“

Auf der Registerkarte *Zugriffsfilter* können Sie den Zugriff auf Ihren Zabbix-Knoten anhand von IP-Adresse oder Subnetz einschränken.



Um einen Zugriffsfilter zu konfigurieren, füllen Sie die folgenden Felder aus:

- *Typ* - verwenden Sie die Dropdown-Liste, um auszuwählen, welche Zabbix-Komponente eingeschränkt werden soll: nur Frontend/API, nur Server oder beides.
- *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 CIDR* - fügen Sie für jede IP oder jedes Subnetz eine optionale Beschreibung hinzu.

Klicken Sie nach dem Ausfüllen der Felder auf *Filter hinzufügen*.
Hinzugefügte Filter werden in der Tabelle darunter angezeigt, einschließlich Status, Typ, IP-Adresse oder Subnetz sowie Beschreibung.

Warning:
Während der Initialisierung des Knotens wird automatisch ein Zugriffsfilter für die von Ihnen verwendete IP-Adresse erstellt. Wenn Sie sich später von einer anderen IP-Adresse oder aus einem anderen Netzwerk verbinden, müssen Sie diese Adresse hier hinzufügen, andernfalls wird der Zugriff blockiert.

Sie können:

- - einen Filter bearbeiten.
- - einen Filter oder mithilfe der Kontrollkästchen mehrere Filter löschen.

Registerkarte „Verschlüsselung“

Auf der Registerkarte *Verschlüsselung* können Sie die TLS-Verschlüsselung 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:

- *Root-CA-Zertifikate* – werden verwendet, um die Authentizität von Zertifikaten der Gegenstelle zu überprüfen.
  Klicken Sie auf *Datei auswählen* und wählen Sie die Datei mit vertrauenswürdigen Root-CA-Zertifikaten aus (in der Regel

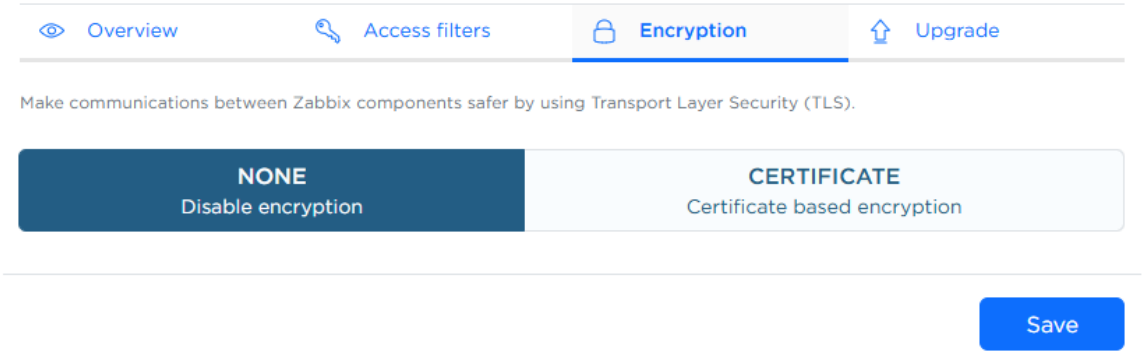
eine PEM-kodierte Datei mit der Endung `.crt` oder `.pem`).

- *Zertifikatskette* – stellt die Vertrauenskette für Ihr eigenes Zertifikat dar.
  Klicken Sie auf *Datei auswählen* und wählen Sie die PEM-kodierte Datei aus (z. B. `.crt` oder `.pem`), die Ihr Zertifikat und alle Zwischenzertifikate der CA enthält.

- *Privater Schlüssel* – der private Schlüssel, der zu Ihrem Zertifikat gehört.
  Klicken Sie auf *Datei auswählen* und wählen Sie die PEM-kodierte `.key`-Datei aus.

- *Widerrufene Zertifikate* (optional) – wenn Sie eine Certificate Revocation List (CRL) pflegen, klicken Sie auf *Datei auswählen* und wählen Sie die CRL-Datei aus, um Zertifikate anzugeben, denen nicht länger vertraut werden soll.

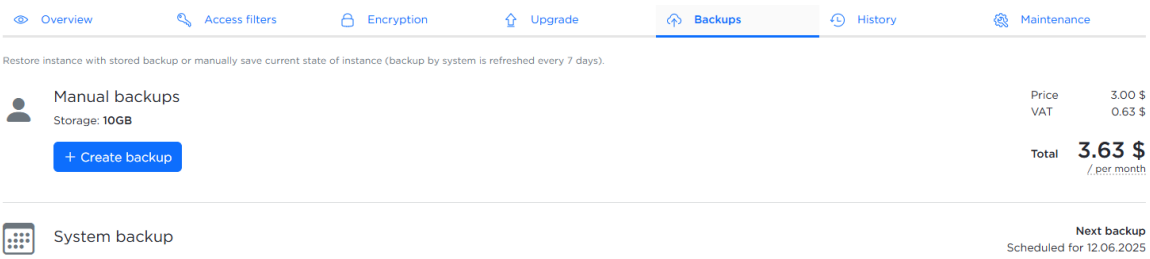
Nachdem Sie die Dateien ausgewählt haben, klicken Sie auf *Speichern*. Die Dateien werden hochgeladen, validiert und angewendet, um die TLS-Verschlüsselung zwischen Zabbix-Komponenten zu konfigurieren.



Registerkarte „Backups“

Auf der Registerkarte *Backups* können Sie Backups Ihres Zabbix-Knotens anzeigen, erstellen, wiederherstellen und löschen.

Sie können Ihre Instanz aus jedem gespeicherten Backup wiederherstellen oder jederzeit ein neues manuelles Backup erstellen. Die Registerkarte zeigt außerdem das Datum des nächsten geplanten Backups an.



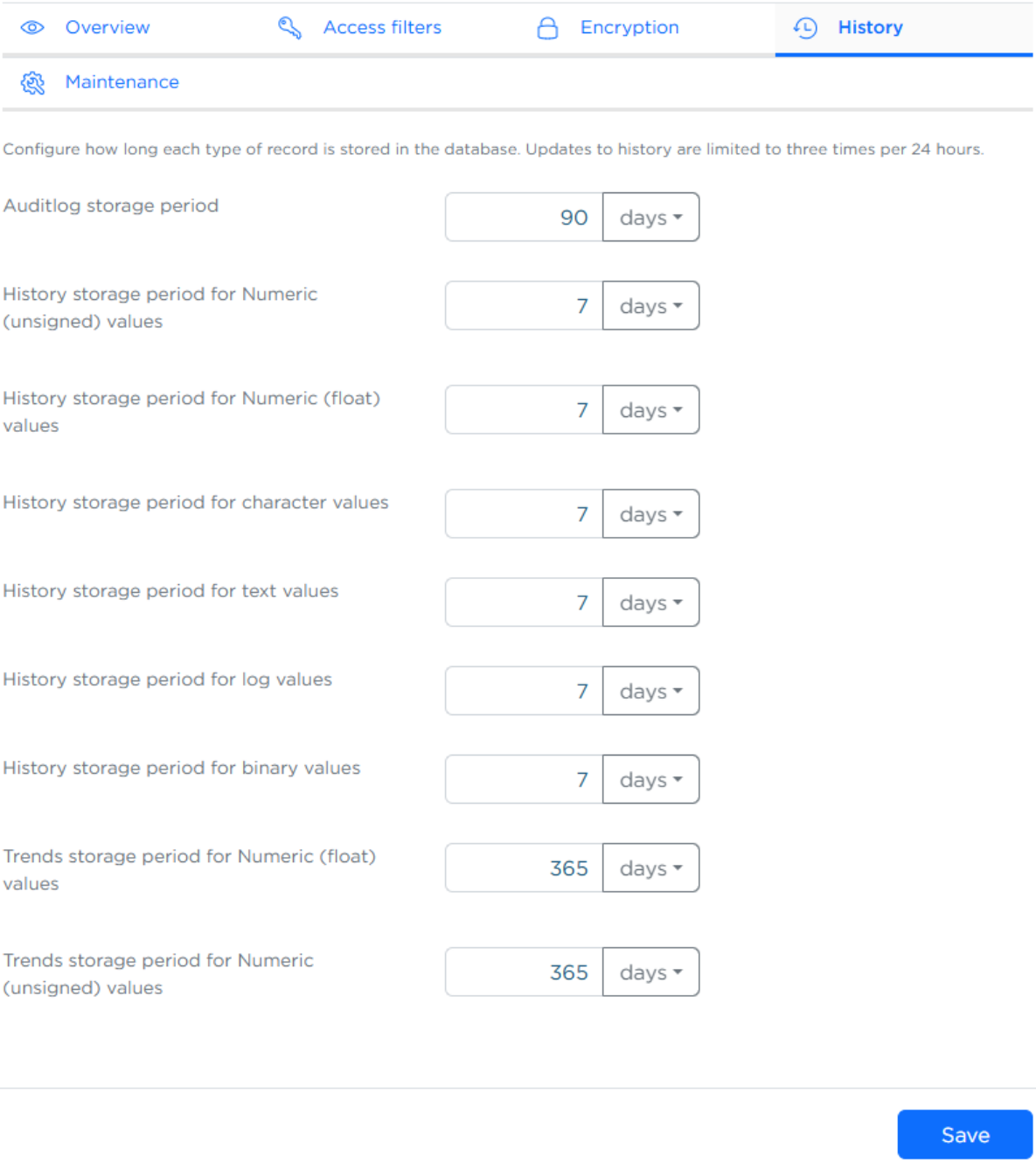
So erstellen Sie ein Backup:

1. Klicken Sie auf die Schaltfläche *Create backup*.
2. Geben Sie im Popup-Fenster optional eine Beschreibung ein.
3. Klicken Sie zur Bestätigung auf *Create*.

Warning:
Zusätzliche Backups werden monatlich auf Grundlage der Speichernutzung berechnet.

Registerkarte „Verlauf“

Auf der Registerkarte *Verlauf* können Sie konfigurieren, wie lange verschiedene Arten von Verlaufsdaten gespeichert werden.



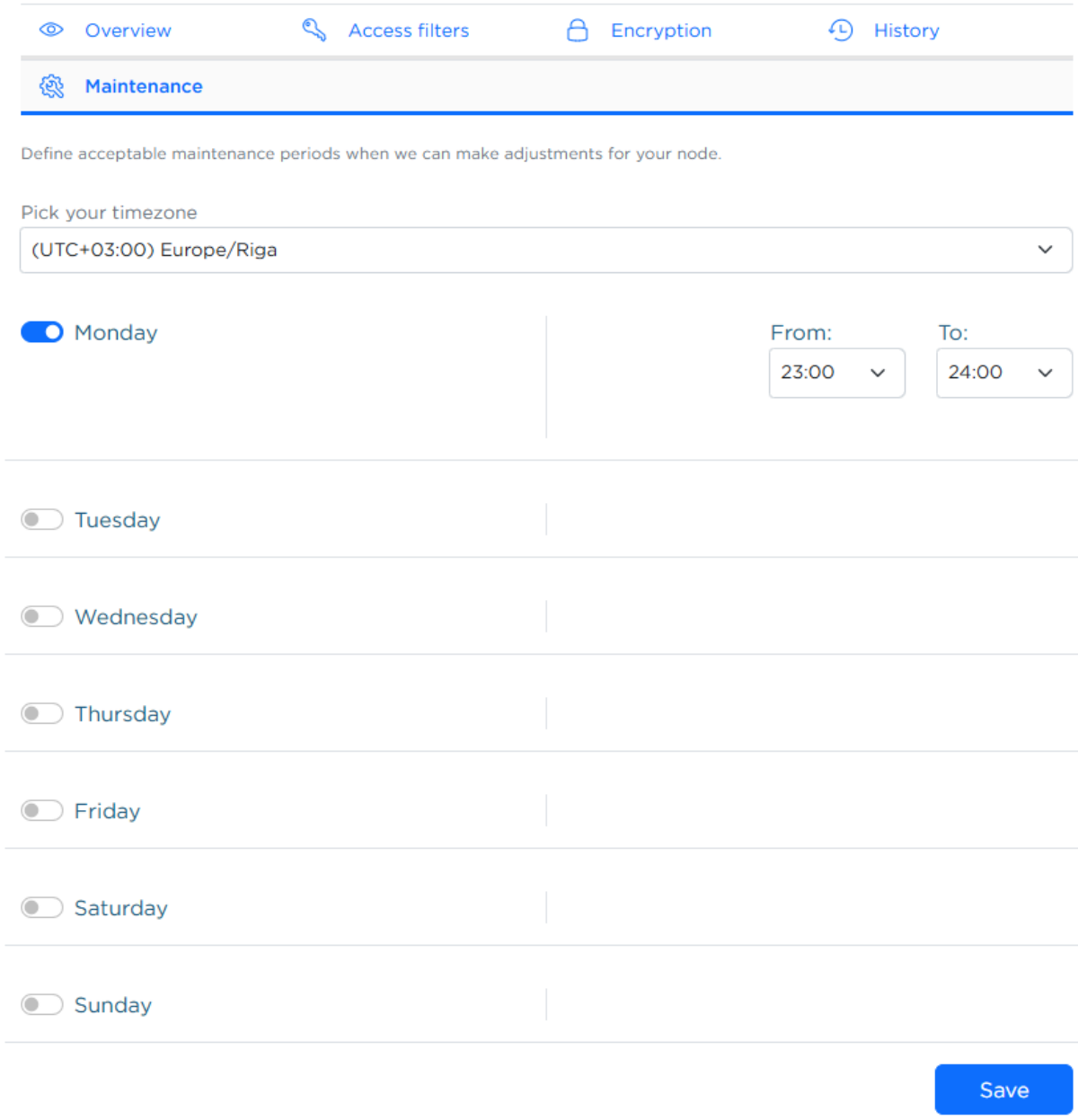
Sie können den Speicherzeitraum für jeden Datentyp festlegen (z. B. numerische Werte, Text, Protokolle) und die Einheit auswählen (Tage, Monate, Jahre). Änderungen können innerhalb von 24 Stunden bis zu dreimal vorgenommen werden.

Nachdem Sie die Werte aktualisiert haben, klicken Sie auf *Speichern*, um die Änderungen anzuwenden.

Warning:
Längere Aufbewahrungszeiträume können den Festplattenspeicherverbrauch erhöhen und sich auf die Performance auswirken. Es wird empfohlen, die Aufbewahrungszeiträume entsprechend Ihren tatsächlichen Daten- und Analyseanforderungen zu konfigurieren.
 Zabbix Cloud verwendet zur Verwaltung der Datenspeicherung eine Datenbankpartitionierung anstelle des herkömmlichen Zabbix-Housekeepings. Dies ermöglicht eine schnellere Performance und eine besser vorhersehbare Handhabung der Aufbewahrung.

Registerkarte „Wartung“

Auf der Registerkarte *Wartung* können Sie zulässige Wartungsfenster festlegen, in denen Aktualisierungen oder Anpassungen an Ihrem Knoten durchgeführt werden können.

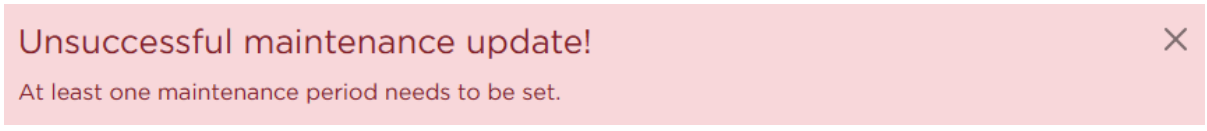


Sie müssen:

- Ihre Zeitzone aus der Dropdown-Liste auswählen.
- Mindestens einen Tag und ein stündliches Wartungsfenster auswählen.

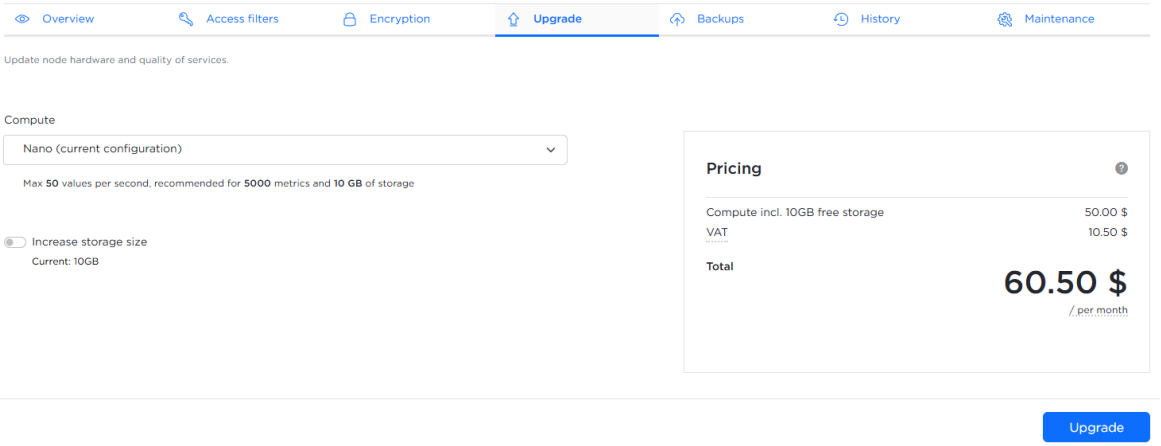
Klicken Sie auf *Speichern*, um die Änderungen anzuwenden.

Wenn kein Tag oder kein Zeitfenster ausgewählt ist, wird eine Fehlermeldung angezeigt.



Registerkarte „Upgrade“

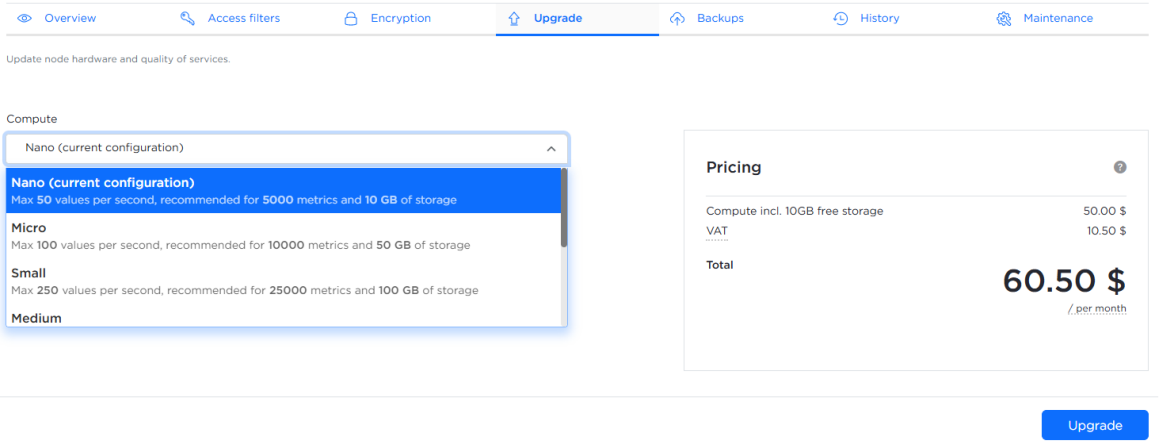
Die Registerkarte *Upgrade* ermöglicht es Ihnen, die Ressourcen Ihres Node zu skalieren oder sein Abonnement zu verwalten.



Sie können die Compute-Stufe oder die Datenträgergröße erhöhen — beachten Sie, dass die Datenträgergröße nur einmal innerhalb von 24 Stunden erhöht werden kann und über die Benutzeroberfläche nicht reduziert werden kann (wenden Sie sich für Verkleinerungsanfragen an den Zabbix-Support).
Nodes mit kostenloser Testversion können in kostenpflichtige Abonnements umgewandelt werden.

So erhöhen Sie die Compute-Stufe:

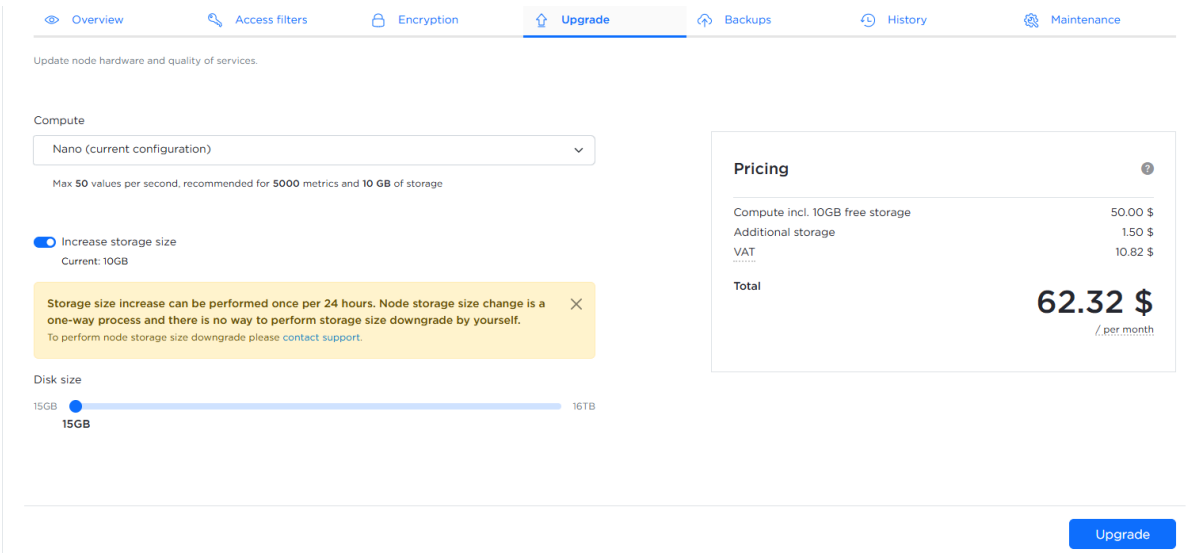
1. Klicken Sie auf das Dropdown-Menü *Compute*.



2. Wählen Sie die gewünschte Konfigurationsstufe aus.
3. Klicken Sie auf die Schaltfläche *Upgrade*.

So erhöhen Sie die Datenträgergröße:

1. Klicken Sie auf das Optionsfeld *Increase storage size*.
2. Stellen Sie den Schieberegler *Disk size* auf den gewünschten Wert ein.



3. Klicken Sie auf die Schaltfläche *Upgrade*.

Note:
Der aktualisierte monatliche Gesamtpreis wird vor dem Bestätigen 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.

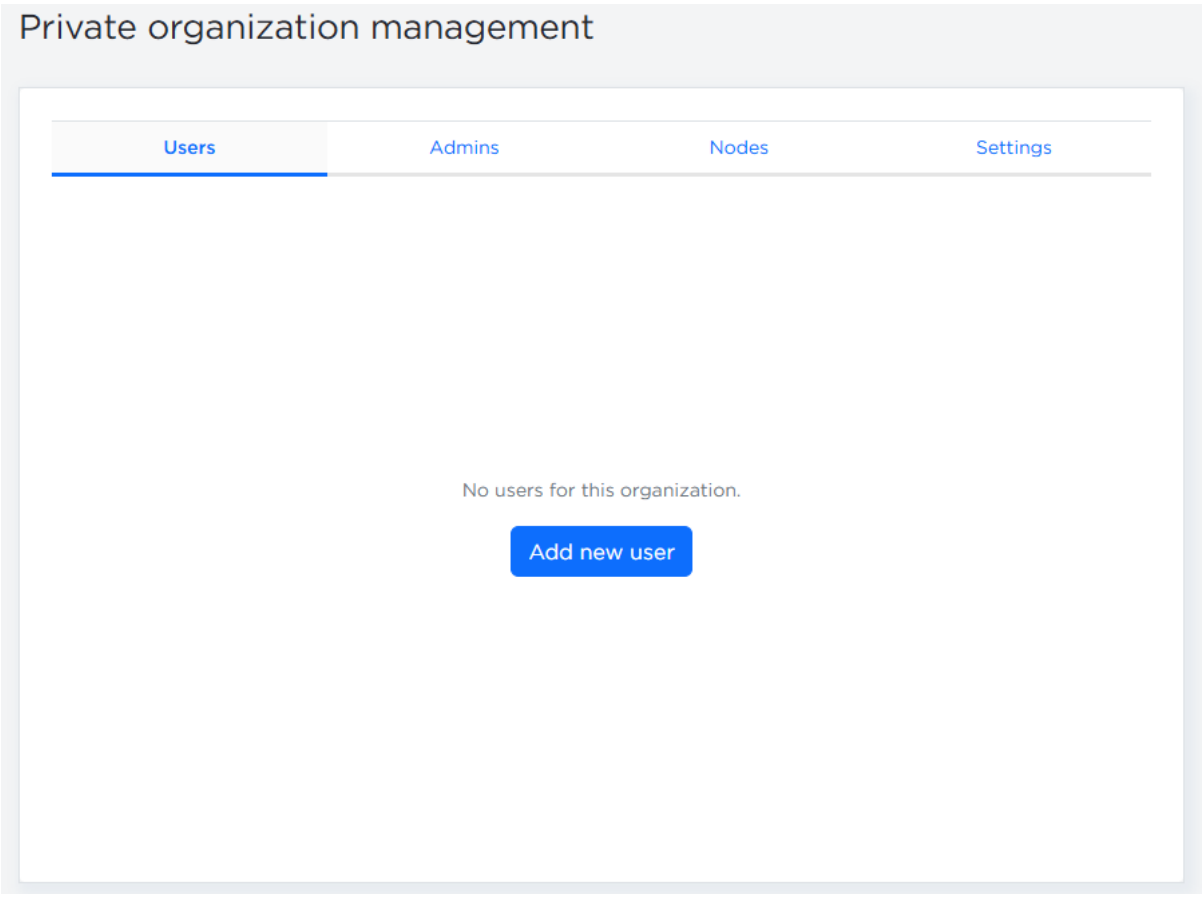
Benutzer zu Organisationen hinzufügen

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*.

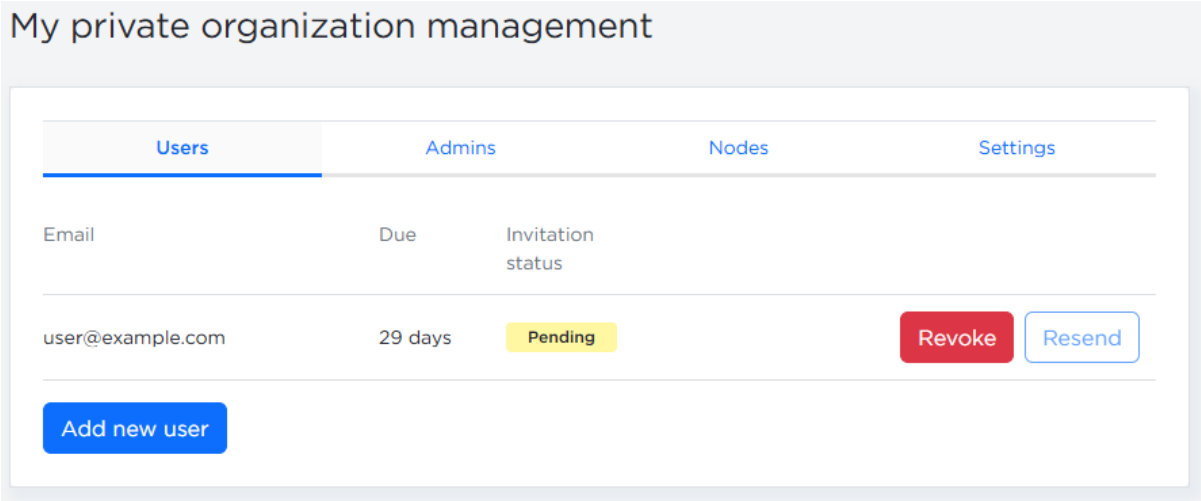
Beachten Sie, dass die Auswahl der Rolle *User* den Benutzer zur Organisation hinzufügt, ihm jedoch keinen Zugriff auf Nodes gewährt.

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.

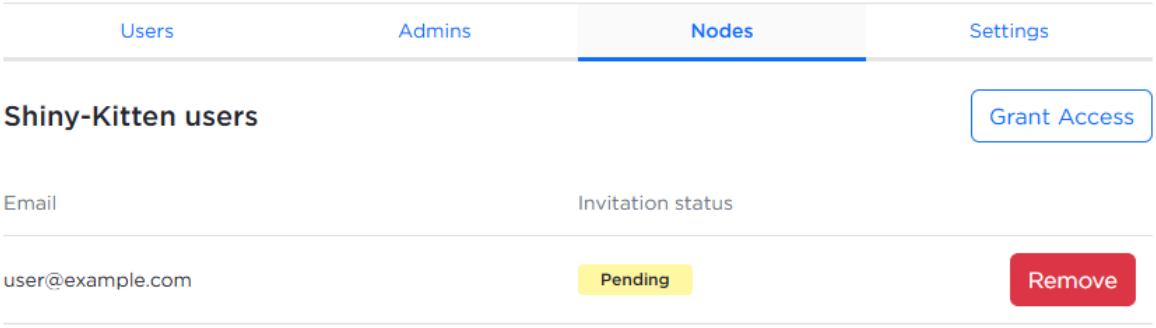
Der eingeladene Benutzer wird mit dem Status *Pending* hinzugefügt.
Sie können die Einladung widerrufen oder erneut senden (ein erneutes Senden ist nur einmal alle 10 Minuten möglich – die Schaltfläche wird deaktiviert, wenn dieses Limit erreicht ist).



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

Vergleichstabelle Während sich die meisten Funktionen zwischen lokalen Zabbix-Installationen und Cloud-Knoten überschneiden, gibt es auch Unterschiede. Nachfolgend finden Sie eine zusammenfassende Vergleichstabelle. In den folgenden Abschnitten werden ausgewählte Datenpunkte im Detail beschrieben.

Funktion / Aspekt	Zabbix Cloud	Lokales Zabbix
Release-Zyklus	Im Einklang mit LTS-Releases	Sie wählen Ihre Zabbix-Version (stabil, Beta, benutzerdefinierte Builds)
Benutzerfreundlichkeit	Einsatzbereite Plattform, vollständig von Zabbix verwaltet	Erfordert Installation, Konfiguration und Wartung
Patchen von Sicherheitslücken	Automatisches Patchen von Betriebssystem und Zabbix	Manuelles Patchen durch den Benutzer
HTTPS-Zertifikat	Gültiges Zertifikat sofort einsatzbereit (kein Let's Encrypt)	Standardmäßig selbstsigniert (Let's Encrypt oder andere CA kann konfiguriert werden)
DB-Leistung (INSERT/UPDATE/SELECT)	Automatische Partitionierung für maximale Geschwindigkeit; keine lang laufenden DELETEs	Manuelle Partitionierung und Bereinigung (lange DELETE-Transaktionen möglich)
Leistungsoptimierung	Vorab optimiert durch das Zabbix-Cloud-Team (keine Änderungsprotokolle)	Volle Transparenz und Kontrolle über Optimierungsparameter
Firewall-Kontrolle	Vom Zabbix-Cloud-Team verwaltet; GUI-/Trapper-Ports sind nicht öffentlich zugänglich	Volle Kontrolle über Firewall-Regeln und Portfreigabe
DB-Isolierung und -Schutz	Stark isoliert, kein direkter DB-/SSH-Zugriff	Hängt von Ihrem Netzwerk und der Sicherheit des Hosts ab
Hochverfügbarkeit (HA)	Läuft auf einer hochverfügbaren Cloud-Plattform, Zabbix-Proxys können für eine höhere Ausfallsicherheit bei der Datenerfassung verwendet werden	Vom Benutzer verwaltete HA
SNMP-Traps	Nur über dedizierten Proxy (kein HA-/lastverteilter Trap-Empfänger)	Direkt oder über Proxy, HA/Lastverteilung möglich
SNMP-Abfrage	Schwierig ohne Proxys (jedes Gerät benötigt NAT/benutzerdefinierten Port)	Native SNMP-Abfrage; Proxy optional

Funktion / Aspekt	Zabbix Cloud	Lokales Zabbix
Aufbewahrungseinstellungen	Verlauf/Trends/Audit nur über die UI (keine API oder Konfigurationsdatei)	Konfigurierbar in der Zabbix-Server-Konfigurationsdatei oder über die API
Benutzerdefinierte Skripte	`AlertScriptsPath`, `ExternalScripts`, Frontend- und Community-Module werden nicht unterstützt	Vollständig unterstützt (Skripte-Pfad, Module, Integrationen)
ODBC-Monitoring	Nur PostgreSQL (Treiber `{postgresql}`); MariaDB-Plugin als Platzhalter vorhanden	ODBC für PostgreSQL, MySQL, Oracle usw. – vollständig konfigurierbar
Begrenzung von ODBC-Aufrufen	Nicht möglich (`StartODBCPollers=1` nur; umfangreiche synchrone Abfragen blockieren andere Aufgaben)	Vollständig abstimmbare Anzahl von Pollern und Zeitplanung
SAML-Zertifikat-Upload	Nicht unterstützt	Über UI oder API unterstützt
Geplante Berichte	Eigener E-Mail-Medientyp muss erstellt werden (Skript-Medientyp nicht unterstützt)	Unterstützt standardmäßig sowohl Skript- als auch E-Mail-Medientyp
Host-Schnittstelle für aktive Prüfungen	Von der Plattform vergebene IP-Adresse (automatisch erstellt)	Schnittstellen werden von Ihnen verwaltet; IP-Adressen 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.

Schnittstelle für aktive Prüfungen

Aktive Prüfungen in der Cloud erstellen automatisch eine Host-Schnittstelle mit einer IP-Adresse, die nicht mit Ihrem lokalen Netzwerk zusammenhängt. Standardmäßig kann diese IP-Adresse extern sein. Um die Konsistenz zu wahren, 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 das E-Mail-Medium Ihrem Benutzer unter *Benutzereinstellungen* > *Medien* zu.
3. Wählen Sie diesen Medientyp aus, wenn Sie geplante Berichte konfigurieren.

Entwicklerzentrum

Entwicklerzentrum

Beginnen Sie mit der Entwicklung benutzerdefinierter Erweiterungen für Zabbix mit Anleitungen zu Modulen, Widgets und Plugins.

Auf Zabbix aufbauen

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.

Plugins

Übersicht Benutzerdefinierte ladbare Plugins erweitern die Funktionalität von Zabbix Agent 2. Sie werden separat kompiliert, verwenden jedoch ein Paket, das mit Zabbix Agent 2 gemeinsam genutzt wird.

Jedes Plugin ist ein *Go*-Paket, das die Struktur definiert und eine oder mehrere Plugin-Schnittstellen implementiert (*Exporter*, *Configurator*, *Runner*).

Springen zu:

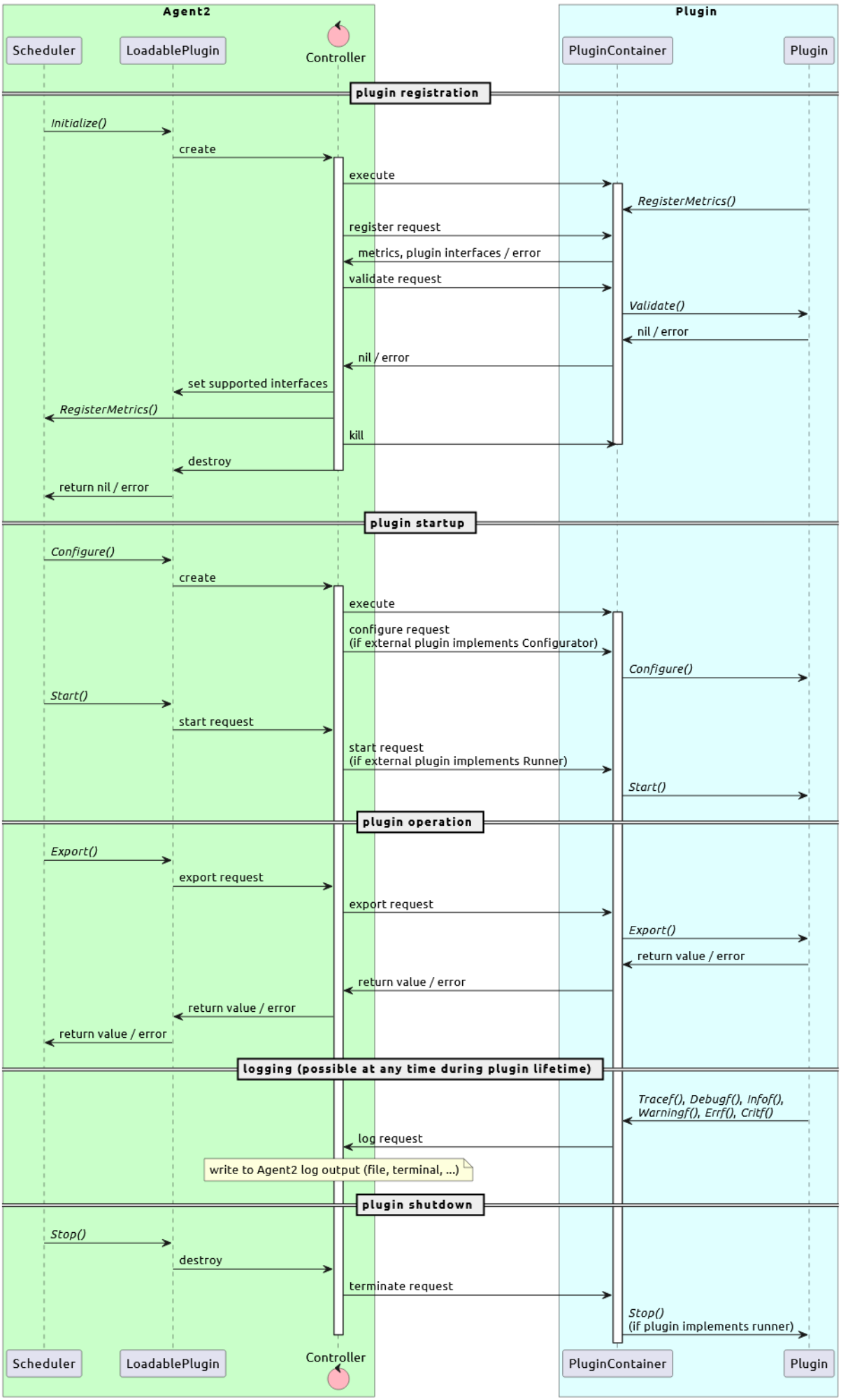
- Schreiben Sie Ihr erstes Plugin
- Plugin-Schnittstellen

Siehe auch:

- Beispiel-Plugin für Zabbix Agent 2

Verbindungsdiagramm Zabbix Agent 2 verbindet sich bidirektional mit den Plugins über UNIX-Sockets unter Linux und Named Pipes unter Windows.

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

What you'll create This is ste-by-step tutorial for creating a simple loadable plugin named **MyIP**. This plugin will implement a single item key (`myip`) returning the external IP address of the Zabbix agent host.

Step 1: Setup

1. A plugin is a standard Go module. Start by initializing the `go.mod` file in the plugin directory to track plugin dependencies:

```
cd path/to/plugins/myip # Switch to your plugin directory
go mod init myip
```

2. Install the mandatory dependency Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Replace `$LATEST_COMMIT_HASH` with the latest `HEAD` commit hash from the `golang.zabbix.com/sdk` repository in the appropriate release branch. For example:

```
go get golang.zabbix.com/sdk@af85407
```

Note that `golang.zabbix.com/sdk` versioning is currently not supported, but this may change in the future.

Additional dependencies can be installed as needed using `go get`.

3. Create an empty `main.go` file for the plugin source code:

```
touch main.go
```

Now the initial setup is complete, and the plugin is ready for development.

Step 2: Plugin structure The `golang.zabbix.com/sdk` module, installed in the previous step, provides the necessary framework for plugin development and ensures all plugins have a consistent structure.

1. Set up basic execution flow.

Start by defining the main execution flow of the plugin. Add the following code to `main.go`:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

This establishes the basic execution flow for the plugin. The run function will later contain the core logic of the plugin.

2. Explore the plugin interfaces.

A Zabbix agent 2 plugin shall be represented by a struct that implements interfaces from the `golang.zabbix.com/sdk/plugin` package:

- *Accessor* - defines essential methods all plugins must implement, such as setting the plugin name and handling item key timeouts.
- One or more of the following functional plugin interfaces:
  - *Exporter* - performs a poll and returns a value (or values), nothing, or an error; often used alongside the *Collector* interface.
  - *Collector* - manages the periodic collection of data.
  - *Runner* - defines plugin startup and shutdown procedures.
  - *Watcher* - allows to implement independent metric polling, bypassing the agent's internal scheduler; useful for trap-based or event-driven monitoring.
  - *Configurator* - defines how the plugin reads and applies its configuration settings.

You can either implement these interfaces yourself or use the default ones provided by the Zabbix Go SDK, modifying them as needed. This tutorial uses the default implementations.

3. Create the plugin struct.

Now, import the *plugin* package and create a `myIP` struct that embeds the `plugin.Base` struct:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

The myIP struct currently satisfies the Accessor interface. A method for implementing one of the functional plugin interfaces, the `Exporter`, will be added later in the tutorial.

Step 3: Define item keys Your plugin needs the item keys to collect data and provide it to Zabbix server or proxy.

1. Import *errs* package for error handling:

```
import "golang.zabbix.com/sdk/errs"
```

2. Register item keys using the `plugin.RegisterMetrics()` function within the `run()` function:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

To register several item keys, repeat the parameters *metric name* and *description* for each metric. For example:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Step 4: Set up the handler The handler facilitates communication between the agent and the plugin.

1. Import the *container* package:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Indise the `run()` function add code to create and set up a handler:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }
```

```
    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Step 5: Implement data collection Data collection is done via the Exporter interface, which describes the `Export` method:

```
func Export(
  key string,             // The item key to collect.
  params []string,        // Arguments passed to the item key (`myip[arg1, arg2]`).
  context ContextProvider // Metadata about the item key data collection.
) (any, error)
```

1. Import the required packages for HTTP requests and response reading:

```
import (
    "io"
    "net/http"
)
```

2. Implement the `Export` method for the `myIP` struct:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Step 6: Build and configure the plugin

1. To build the plugin, run:

```
go mod tidy
go build
```

This should create an executable `myip` in the current directory.

2. Configure Zabbix agent 2 to use the plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

Replace `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` with the path to the `myip` created in step 5.

The plugin name in the configuration parameter name (*MyIP* in this tutorial) must match the plugin name defined in the *plugin.RegisterMetrics()* function.

3. To test the plugin and its `myip` item, run:

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

The output should contain an external IP address of your host and look similar to this:

```
myip                                          [s|192.0.2.0]
```

With that, you have created a simple loadable plugin for Zabbix agent 2. Congrats!

```go
package main

import (
    "io"
    "net/http"

    "golang.zabbix.com/sdk/errs"
    "golang.zabbix.com/sdk/plugin"
    "golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
    plugin.Base
}

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // Plugin name
        "myip",                             // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
```

```
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request from the agent is received.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Complete source code

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

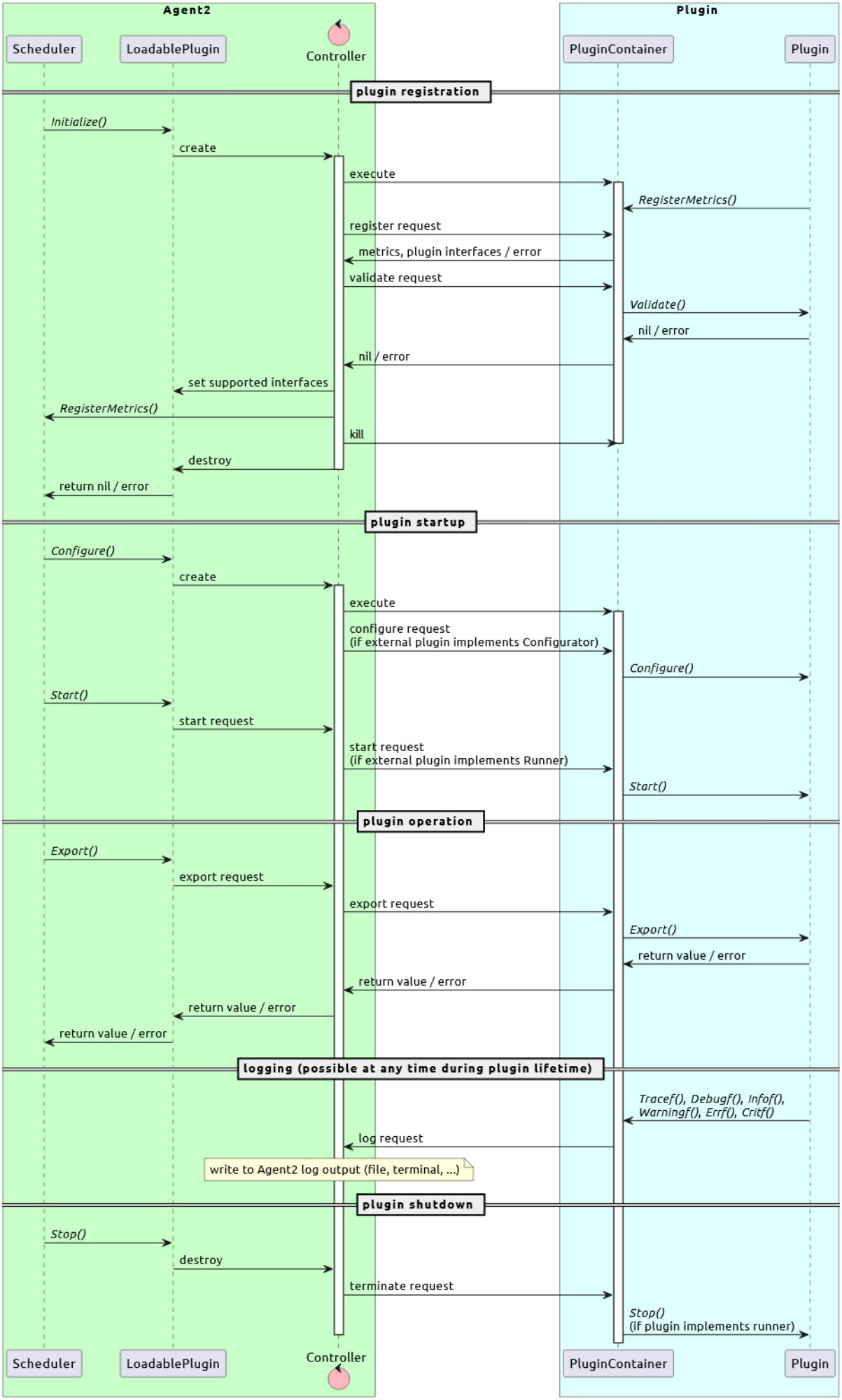
gibt es eine Grenze von maximal 100 gleichzeitigen *Export()* Aufrufen pro Plugin, die je nach den Anforderungen für jedes Plugin reduziert werden kann.

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-linterfaces starten/stoppen.

Connection diagram Zabbix agent 2 connects bidirectionally to the plugins using UNIX sockets on Linux and Named Pipes on Windows.

The connection diagram below illustrates the communication process between Zabbix agent 2 and a loadable plugin and the metrics collection process.


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

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

So fordern Sie einen Datenpunktwert an:

1. Erstellen Sie eine `Getter`-Instanz und geben Sie dabei 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='192.0.2.0', port=10050)
response = agent.get('system.uname')
```

Verwenden 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 `Getter` beim Herstellen der Verbindung zum Zabbix Agent verwenden soll:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

Timeout verwenden

Sie können für den `Getter` einen Antwort-`timeout` festlegen, um zu steuern, wie lange Ihr Skript auf eine Antwort vom Zabbix Agent warten soll, bevor es aufgibt:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    timeout=30
)
```

Verwendung von Verschlüsselung

Der `Getter` 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
    # ...

agent = Getter(
    host='192.0.2.0',
    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 zu warten. Dadurch kann Ihr Skript effizienter werden, wenn es viele Werte erfassen muss oder wenn die Erfassung einiger Werte lange dauert.

Bei der Verwendung des asynchronen Modus gibt es im Vergleich zum synchronen Modus wichtige Unterschiede:

- Importieren Sie das Python-Modul `asyncio` (Sie müssen zunächst 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 mit dem asynchronen Modus einen einzelnen Wert 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='192.0.2.0', port=10050)

    # 3. Fetch the system.uname value from Zabbix agent (must await):
    response = await agent.get('system.uname')

    # 4. Print the value returned by Zabbix agent:
    print(response.value)

### 5. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

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
)
```

Verwenden der Agent-Konfigurationsdatei

Sie können zabbix_utils die Parameter `Server` oder `ServerActive` aus einer lokalen Zabbix-Agent- oder Agent-2-Konfigurationsdatei lesen lassen. In solchen Fällen 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 Server-Instanzen enthält, sendet `Sender` Daten an den ersten verfügbaren Server in jedem Cluster. Wenn `ServerActive` nicht gesetzt ist, wird die Adresse aus `Server` mit dem Standard-Port (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

Ein Modul schreiben Ein Beispiel für das Schreiben eines Moduls besteht aus den folgenden Schritten (klicken Sie auf den Datei- oder Ordnernamen, um weitere Details zu dem jeweiligen 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 die Modulaktion(en).
5. Erstellen Sie die Datei Module.php (oder Widget.php für Dashboard-Widgets) und definieren Sie Regeln für die Initialisierung und Ereignisbehandlung.
6. Erstellen Sie den Ordner *assets* für JavaScript-Dateien (in *assets/js*), CSS-Stile (in *assets/css*) oder andere zusätzliche Dateien.
7. Stellen Sie sicher, dass Sie die erforderlichen Ansichten, Aktionen und Asset-Dateien in der manifest.json angeben.
8. Registrieren Sie das Modul im Zabbix-Frontend und verwenden Sie es.

> **Note:**
> Sie können ein Modul registrieren und aktivieren, sobald Sie die Datei manifest.json erstellt haben. Sobald das Modul aktiviert ist, können Sie alle Änderungen, die Sie an den Moduldateien vorgenommen haben, sofort sehen, indem Sie das Zabbix-Frontend aktualisieren.

manifest.json

Jedes Modul benötigt die Datei manifest.json. Die Datei sollte sich im primären Verzeichnis des Moduls befinden (zum Beispiel *zabbix/ui/modules/module_name/manifest.json*).

Als Minimum müssen folgende Felder in der Datei manifest.json enthalten sein:

```
{
"manifest_version": 2.0,
"id": "meine_ip_addresse",
"name": "Meine IP Addresse",
"namespace": "MeineIPAddresse",
"version": "1.0"
}
```

Die Datei **manifest.json** unterstützt folgende Parameter (Parameternamen für eine detaillierte Beschreibung anklicken):

Parameter	Beschreibung	Erforderlich
manifest_version	Manifest-Version des Moduls.	Ja
id	Eindeutige Modul-ID.	
name	Modulname, der im Abschnitt „Administration“ angezeigt wird.	
Namespace	PHP-Namespace für Modulklassen.	
Version	Modulversion.	
Typ	Modultyp. Für Widgets muss er auf *widget* gesetzt werden	Ja für Widgets, sonst nein
widget	Widget-Konfiguration. Wird nur für Widgets verwendet.	
Aktionen	Aktionen die im Modul registriert werden.	
Styles	Einzuschließende CSS-Stile und JavaScript-Dateien.	Nein
Autor	Modulautor.	
Konfiguration	Standardwerte für benutzerdefinierte Moduloptionen.	
Beschreibung	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 Klassenquelldatei 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. Für Widgets wird dieser Link geöffnet, wenn Sie auf das Hilfesymbol ? im Fenster *Widget hinzufügen* oder *Widget bearbeiten* klicken.

Wenn **url** nicht angegeben ist, öffnet ein Klick auf das Hilfesymbol die allgemeine Seite Dashboard-Widgets.

Typ: Zeichenfolge

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 dann deren 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 formatiert.

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 styles

CSS-Dateien können ein benutzerdefiniertes Attribut `theme` enthalten, um einen anderen Stil für ein bestimmtes Frontend-Theme zu definieren.

Verfügbare Themen und ihre Attributwerte:

- **Blue** - [theme='blue-theme']
- **Dark** - [theme='dark-theme']
- **High-contrast light** - [theme='hc-light']
- **High-contrast dark** - [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 *Disabled* - der Status des Moduls ändert sich in *Enabled*. Klicken Sie auf den Modulnamen, um zusätzliche Informationen über das Modul anzuzeigen, z. B. den Autor, die Version oder die Kurzbeschreibung (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
<?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 für die in der Klasse *WidgetForm* definierten Felder festzulegen, welche das Aussehen und Verhalten bei der Darstellung in der Konfigurationsansicht definieren.

Die Klasse *CWidgetFormView* unterstützt die folgenden Methoden:

- *addField()* - erhält eine Instanz der Klasse *CWidgetFieldView* als Parameter; jede *CWidgetField* Klasse hat eine entsprechende *CWidgetFieldView* Klasse zur Verwendung in der Widget-Konfigurationsansicht.
- *addFieldset()* - empfängt eine Instanz der Klasse *CWidgetFieldsGroupView*, die Felder in einem zusammenklappbaren Container kombiniert.
- *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 Konfigurationsansicht des Widgets geladen wird.

Die Klasse *CWidgetFormView* sollte sich im Verzeichnis *views* befinden.

views/widget.edit.php Beispiel

```
<?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

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.) geerbt werden. Klassen, die *CWidgetField* erweitern, sind für das Empfangen, Speichern und Validieren von Widget-Konfigurationswerten verantwortlich.

Die folgenden *CWidgetField*-Klassen sind verfügbar.

CWidgetField-Klasse	Datenbankfeldtyp	Beschreibung
CWidgetFieldCheckBox	int32	Einzelnes Kontrollkästchen.
CWidgetFieldCheckBoxList	Array von int32	Mehrere Kontrollkästchen unter einem einzigen Konfigurationsfeld.
CWidgetFieldColor	string	Farbauswahlfeld.
CWidgetFieldDatePicker	string	Datumsauswahlfeld.
CWidgetFieldHostPatternSelect	string	Multiselect-Feld, das die Auswahl eines oder mehrerer Hosts ermöglicht. Unterstützt die Definition von Hostnamenmustern (alle passenden Hosts werden ausgewählt).
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 Breiten- und Längengraden sowie der Kartenzoomstufe.
CWidgetFieldMultiSelectAction	ID	Multiselect-Feld zur Auswahl von Aktionen (aus der Liste der Aktionen, die unter *Alarme* → *Aktionen* definiert sind).
CWidgetFieldMultiSelectGraph	ID	Multiselect-Feld zur Auswahl benutzerdefinierter Diagramme.
CWidgetFieldMultiSelectGraphPrototype	ID	Multiselect-Feld zur Auswahl benutzerdefinierter Diagrammprototypen.
CWidgetFieldMultiSelectGroup	ID	Multiselect-Feld zum Auswählen von Hostgruppen.
CWidgetFieldMultiSelectHost	ID	Multiselect-Feld zum Auswählen von Hosts.
CWidgetFieldMultiSelectItem	ID	Multiselect-Feld zum Auswählen von Elementen.
CWidgetFieldMultiSelectItemPattern	ID	Multiselect-Feld zum Auswählen von Elementmustern.
CWidgetFieldMultiSelectItemPrototype	ID	Multiselect-Feld zum Auswählen von Elementprototypen.
CWidgetFieldMultiSelectMap	ID	Multiselect-Feld zum Auswählen von Karten.
CWidgetFieldMultiSelectMediaType	ID	Multiselect-Feld zum Auswählen von Medientypen.
CWidgetFieldMultiSelectOverrideHost	ID	Multiselect-Feld zum Auswählen einer Datenquelle (Dashboard oder anderes Widget), die einen Host enthält, für den das Widget Daten anzeigen kann.
CWidgetFieldMultiSelectService	ID	Multiselect-Feld zum Auswählen von Diensten.
CWidgetFieldMultiSelectSla	ID	Multiselect-Feld zum Auswählen von SLAs.
CWidgetFieldMultiSelectUser	ID	Multiselect-Feld zum Auswählen von Benutzern.
CWidgetFieldNumericBox	string	Feld zum Eingeben einer Gleitkommazahl.
CWidgetFieldRadioButtonList	int32	Radiobox-Gruppe, die aus einer oder mehreren Radioboxen besteht.
CWidgetFieldRangeControl	int32	Schieberegler zum Auswählen eines Werts vom Typ Integer.
CWidgetFieldReference	string	Erstellt eine eindeutige Kennung für dieses Widget auf dem Dashboard. Wird verwendet, um von anderen Widgets auf dieses Widget zu verweisen.
CWidgetFieldSelect	int32	Dropdown-Auswahlfeld.
CWidgetFieldSeverities	Array von int32	*CWidgetFieldCheckBoxList* voreingestellt mit Trigger-Schweregraden.
CWidgetFieldTags	Array von (String, Int32, String)	Ermöglicht die Konfiguration einer oder mehrerer Tag-Filterzeilen.
CWidgetFieldTextArea	String	Textbereich zur Eingabe von mehrzeiligem Text.
CWidgetFieldTextBox	String	Textfeld zur Eingabe von einzeiligem Text.
CWidgetFieldTimePeriod	Array von String	Feld zur Auswahl des Zeitraums.
CWidgetFieldTimeZone	String	Dropdown mit Zeitzonen.

CWidgetField-Klasse	Datenbankfeldtyp	Beschreibung
CWidgetFieldThresholds	Array von (String, String)	Ermöglicht die Konfiguration von Farb- und Zahlenpaaren.
CWidgetFieldUrl	String	Textfeld zur Eingabe von URLs.

Die folgenden *CWidgetField*-Klassen wurden für bestimmte Widgets erstellt.

Diese Klassen haben sehr spezifische Anwendungsfälle, können aber bei Bedarf auch wiederverwendet werden.

|CWidgetField-Klasse|Datenbankfeldtyp|Beschreibung|

|----|----|----------| |*CWidgetFieldColumnsList*|Array von (mehrere gemischt)|Für das Widget *Top-Hosts*. Erstellen Sie eine Tabelle mit benutzerdefinierten Spalten zulässiger Typen.| |*CWidgetFieldNavTree*|Zeichenfolge|Für das Widget *Kartennavigationsbaum*. Ersetzt die Widgetansicht im Bearbeitungsmodus durch den Kartenauswahlbaum.|

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

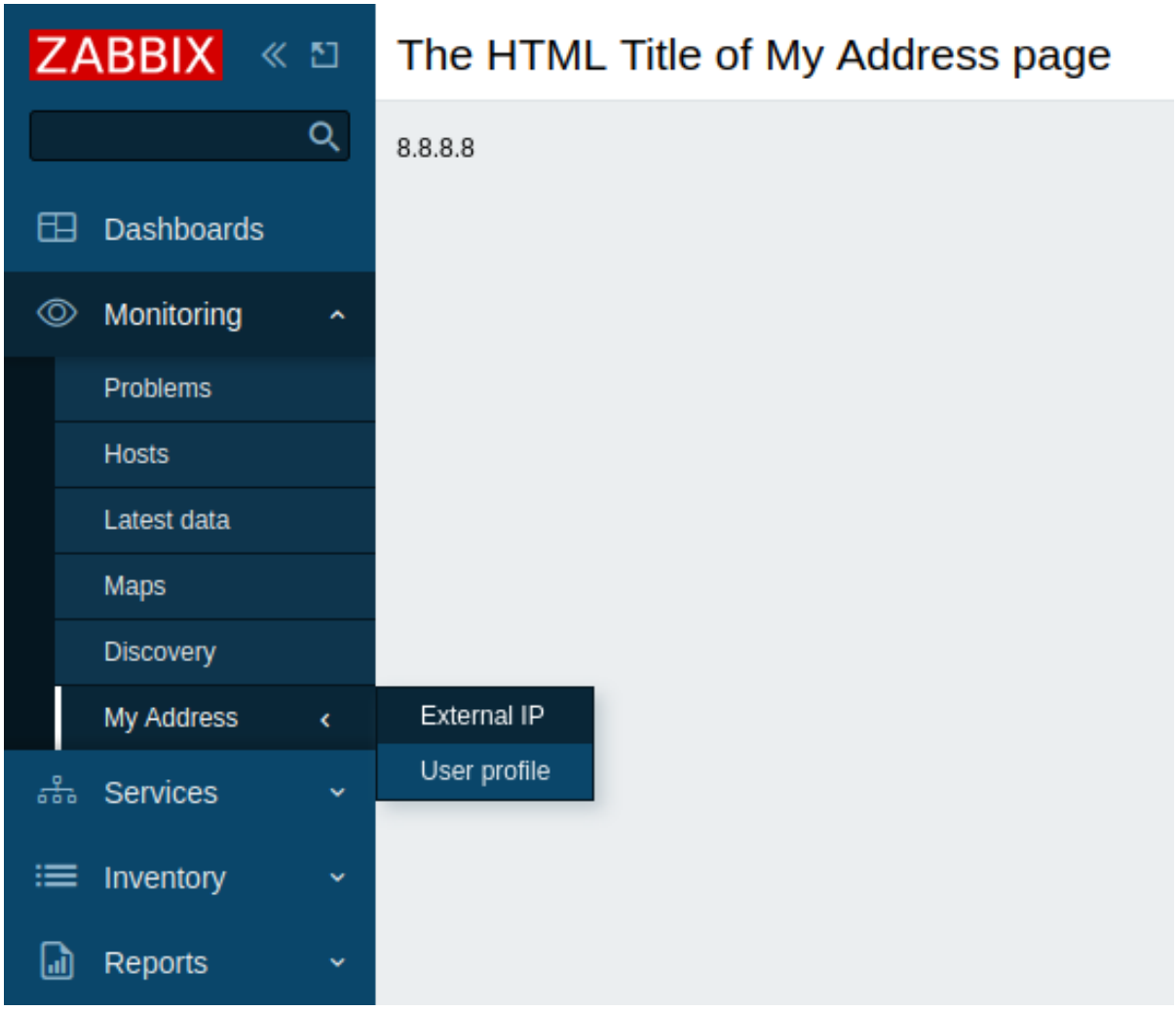
Dieser Abschnitt enthält praktische Schritt-für-Schritt-Tutorials, die veranschaulichen, wie man in Zabbix ein benutzerdefiniertes Modul und ein Widget erstellt.

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

Während dieses Tutorials erstellen Sie zunächst ein Frontend-Modul, das einen neuen Menüabschnitt *Meine Adresse* hinzufügt und konvertieren es dann in ein fortgeschritteneres Frontend-Modul, 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 *Meine Adresse* anzeigt. So sieht das fertige Modul aus:



Teil I - Neuer Menübereich

Ein leeres Modul zum Zabbix Frontend hinzufügen

1. Erstellen Sie im Verzeichnis *modules* Ihrer Zabbix-Frontend-Installation ein Verzeichnis *MyAddress* (zum Beispiel *zabbix/ui/modules*).
2. Erstellen Sie eine Datei *manifest.json* mit grundlegenden Modul-Metadaten (siehe die Beschreibung der unterstützten Parameter).

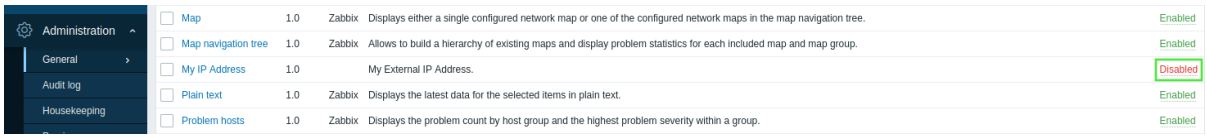
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
```

```
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Gehen Sie im Zabbix Frontend zum Abschnitt *Administration* → *General* → *Modules* und klicken Sie auf die Schaltfläche *Scan directory*.

4. Suchen Sie das neue Modul *My IP Address* in der Liste 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 jetzt im Frontend registriert. Es ist jedoch noch nicht sichtbar, da Sie die Funktionalität des Moduls noch definieren müssen. Sobald Sie dem Modulverzeichnis Inhalte hinzufügen, sehen Sie die Änderungen im Zabbix Frontend sofort nach dem Aktualisieren der Seite.

Erstellen Sie einen Menüabschnitt

1. Erstellen Sie eine *Module.php*-Datei im *MyAddress*-Verzeichnis.

Diese Datei implementiert eine neue *Module*-Klasse, die die Standardklasse *CModule* erweitert.

Die *Module*-Klasse fügt einen neuen *My Address*-Menüabschnitt in das Hauptmenü ein.

Die *setAction()*-Methode gibt eine Aktion an, die beim Klicken auf den Menüabschnitt ausgeführt werden soll.

Zu Beginn können Sie die vordefinierte Aktion *userprofile.edit* verwenden, welche die Seite *Benutzerprofil* öffnet.

In Teil III dieses Tutorials erfahren Sie, wie Sie eine benutzerdefinierte Aktion erstellen.

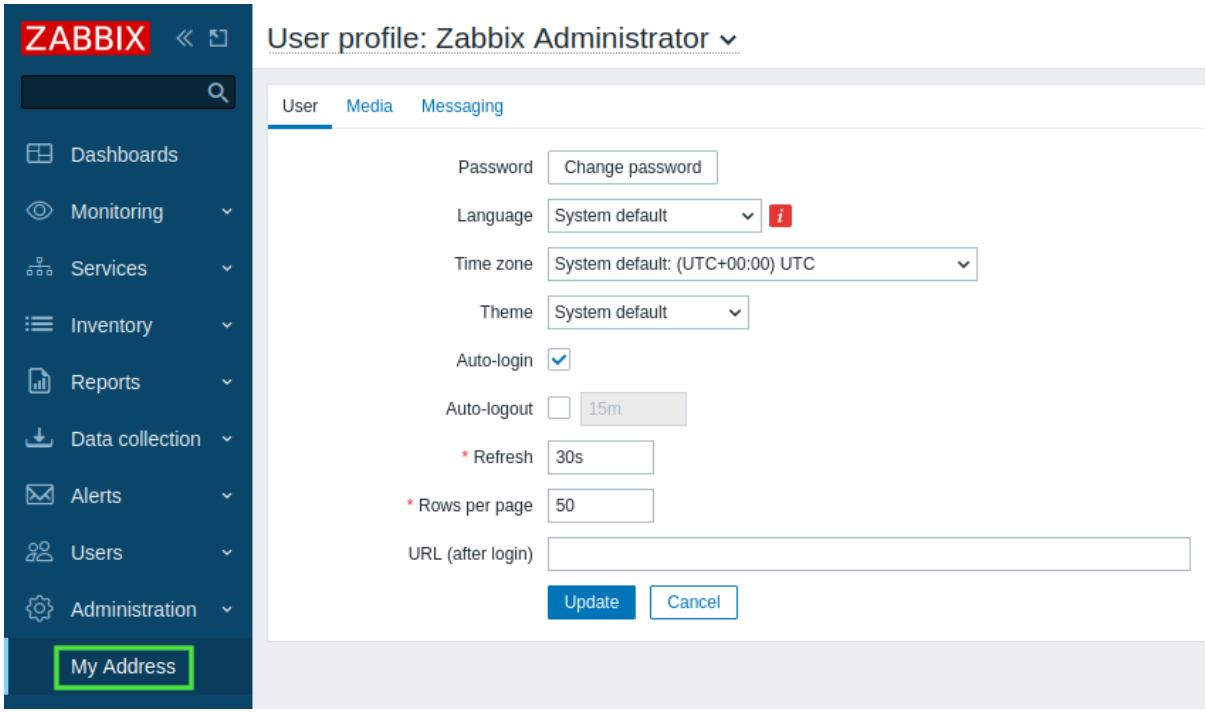
ui/modules/MyAddress/Module.php

```
<?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(_('Meine Adresse')))
->setAction('userprofile.edit'));
}
}
```

Note:
Sie können `'userprofile.edit'` durch andere Aktionen ersetzen, z. B. `'charts.view'` (öffnet benutzerdefinierte Diagramme), `'problems.view'` (öffnet *Überwachung* → *Probleme*) oder `'report.status'` (öffnet *Systeminformationen*-Bericht).

3. Aktualisieren Sie das Zabbix-Frontend. Unten im Zabbix-Hauptmenü gibt es jetzt einen neuen Abschnitt *Meine Adresse*.

Klicken Sie auf *Meine Adresse*, um die Seite *Benutzerprofil* zu öffnen.



Teil II – Änderung der Position des Menüabschnitts

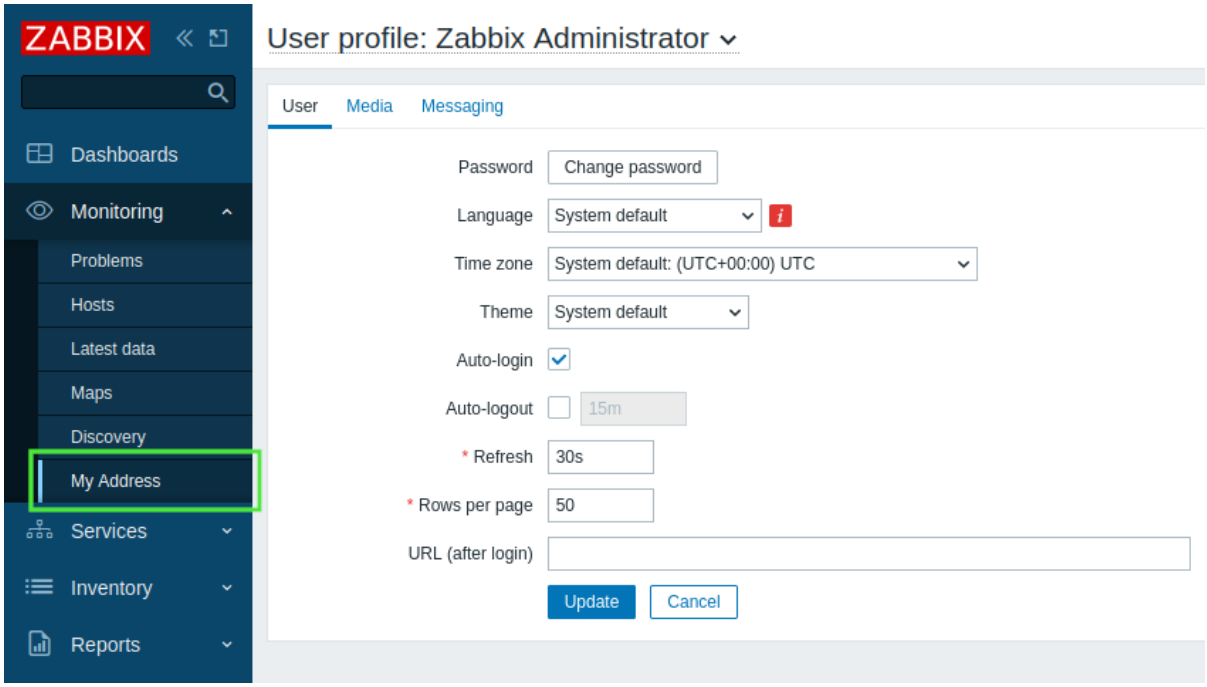
In diesem Teil verschieben Sie den Menüabschnitt *Meine Adresse* in den Abschnitt *Überwachung* und fügen ihm dann ein verschachteltes Menü hinzu. Dadurch können Benutzer vom Menüabschnitt *Überwachung* → *Meine Adresse* auf zwei Untermenüseiten zugreifen.

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 beachten Sie, dass sich der Abschnitt *My address* jetzt unter dem Abschnitt *Discovery* befindet.



3. Um verschachtelte Seiten zum Menüabschnitt *My Address* hinzuzufügen, öffnen und bearbeiten Sie die Datei *Module.php* erneut.

Dieser Schritt erstellt zwei Unterabschnitte:

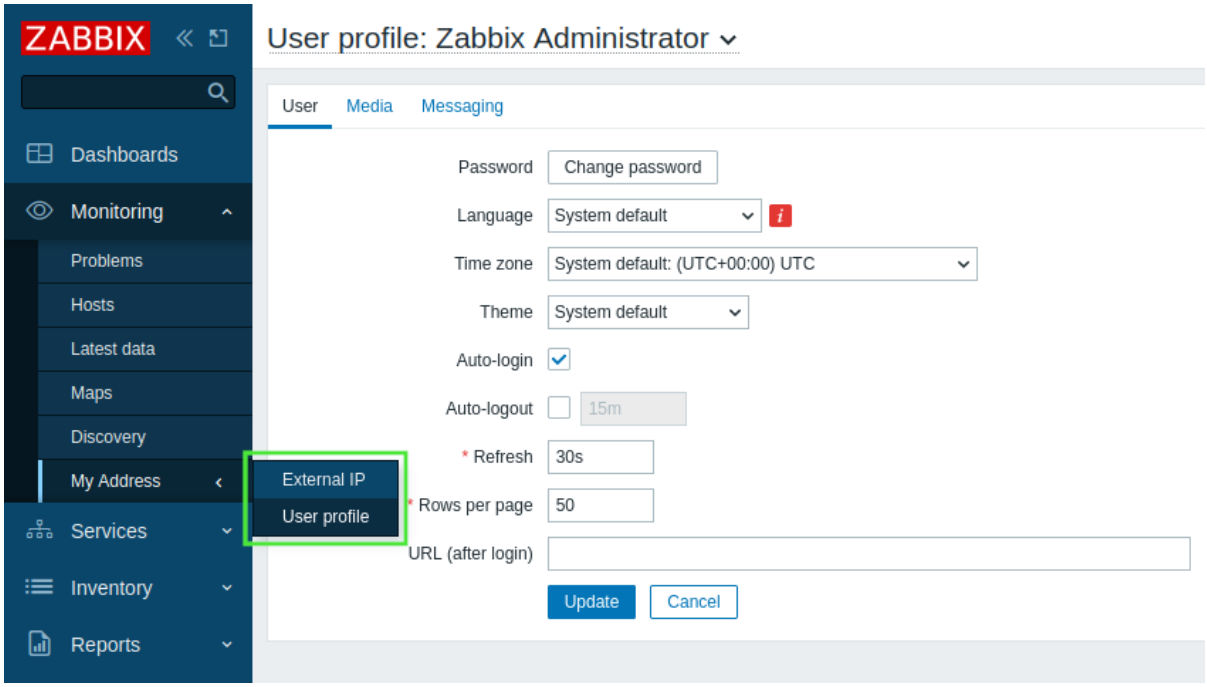
- *Externe IP*, die eine neue „my.address“-Aktion ausführt, die in den nächsten Schritten definiert wird;
- *Benutzerprofil*, das die vordefinierte „userprofile.edit“-Aktion ausführt, um die Seite „Benutzerprofil*“ zu öffnen.

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. Beachten Sie, dass der Menüabschnitt *My address* jetzt 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*** kümmert sich um die Implementierung der Geschäftslogik, während die Datei ***views/my.address.php*** für die Ansicht zuständig ist.

1. Erstellen Sie ein Verzeichnis *actions* im Verzeichnis *MyAddress*.
2. Erstellen Sie die Datei *MyAddress.php* im Verzeichnis *actions*.

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 zuständig.

> **Attention:**
> Die Daten müssen als assoziatives Array organisiert sein. Das Array kann mehrdimensional sein und alle von der Ansicht erwarteten Daten enthalten.

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 ein neues Verzeichnis *views* im Verzeichnis *MyAddress*.
4. Erstellen Sie eine Datei *my.address.php* im Verzeichnis *views* 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(_('Der HTML-Titel der Seite „Meine Adresse“))
->addItem(new CDiv($data['my-ip']))
->show();
```

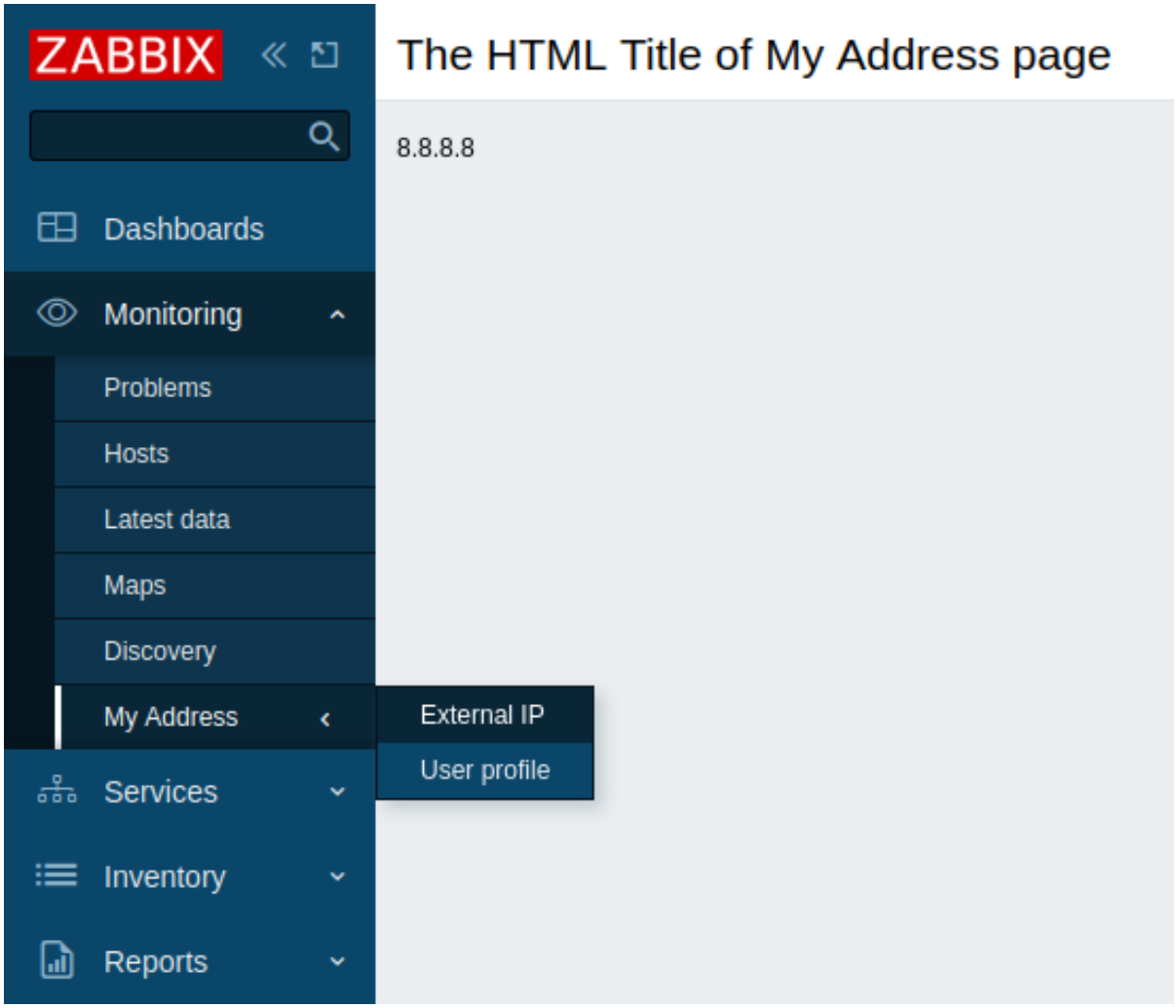
5. Die Modulaktion muss in der Datei *manifest.json* registriert werden. Öffnen Sie *manifest.json* und fügen Sie ein neues Objekt „actions“ hinzu, das Folgendes enthält:

- den Aktionsschlüssel mit dem Aktionsnamen in Kleinbuchstaben (a-z) und mit durch Punkte getrennten Wörtern (z. B. „my.address“);
- den Aktionsklassennamen („MyAddress“) als Wert für den Schlüssel „class“ des Objekts „my.address“;
- den Aktionsansichtsnamen („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": "Meine IP-Adresse",
"version": "1.0",
"namespace": "MyAddress",
"description": "Meine externe IP-Adresse.",
"actions": {
"my.address": {
"class": "MyAddress",
"view": "my.address"
}
}
}
```

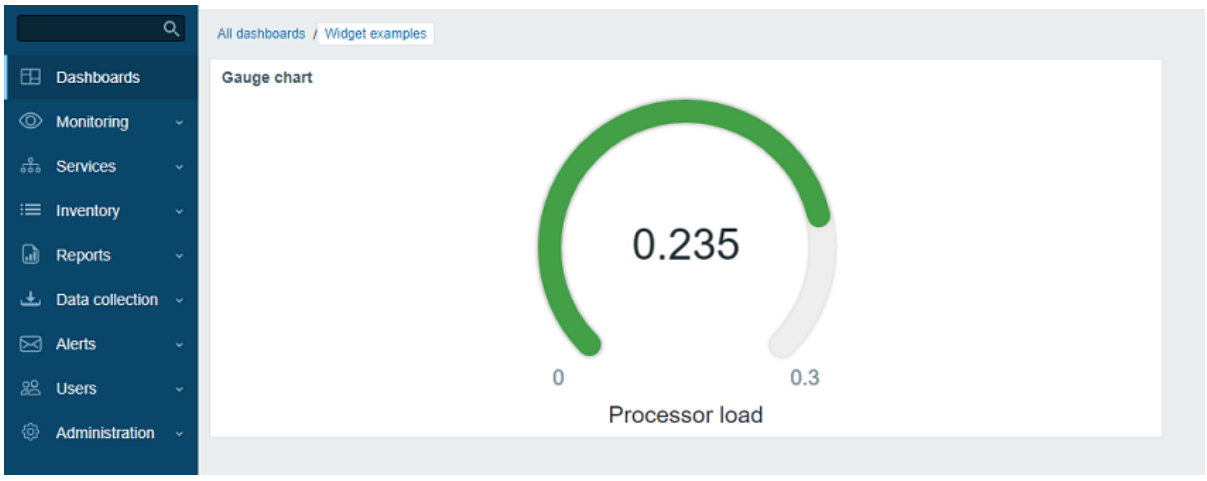
6. Aktualisieren Sie das Zabbix-Frontend. Klicken Sie auf *Meine Adresse* → *Externe IP*, um die IP-Adresse Ihres Computers anzuzeigen.



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 bauen werden In diesem Tutorial erstellen Sie zunächst ein Basic-Widget Basic „Hello, world!“ und konvertieren Sie es dann in ein Erweitertes Widget, das einen Inhalt als Gauge-Diagramm 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 im Verzeichnis *modules* Ihrer Zabbix-Frontend-Installation ein Verzeichnis *lesson_gauge_chart* (zum Beispiel *zabbix/ui/modules*).

Note:
Alle benutzerdefinierten Widgets werden als externe Module behandelt und müssen zum Verzeichnis *modules* Ihrer Zabbix-Frontend-Installation hinzugefügt werden (zum Beispiel *zabbix/ui/modules*). Das Verzeichnis *zabbix/ui/widgets* ist für integrierte Zabbix-Widgets reserviert und wird zusammen mit der Zabbix-Benutzeroberfläche aktualisiert.

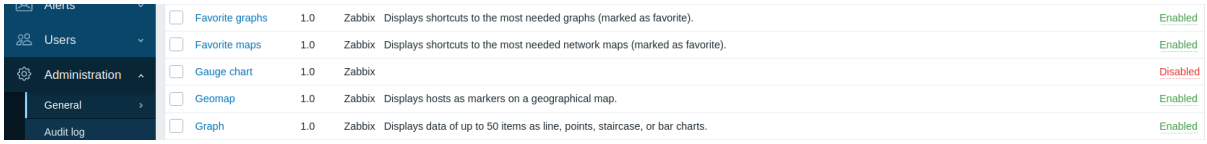
2. Erstellen Sie eine Datei *manifest.json* mit grundlegenden Widget-Metadaten (siehe die Beschreibung der unterstützten Parameter).

ui/modules/lesson_gauge_chart/manifest.json

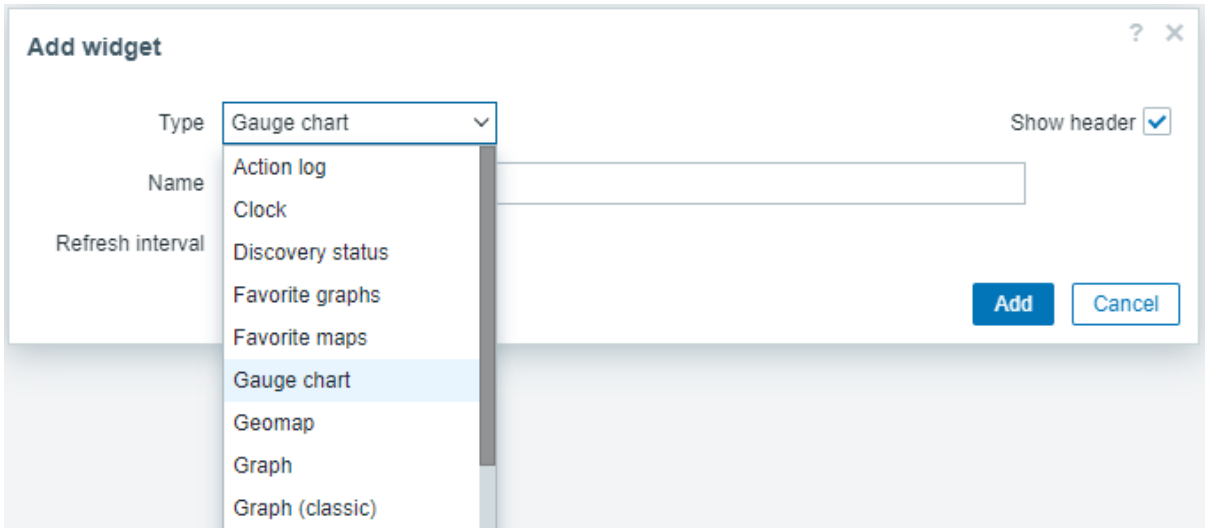
```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

3. Gehen Sie im Zabbix Frontend zum Abschnitt *Administration* → *General* → *Modules* und klicken Sie auf die Schaltfläche *Scan directory*.

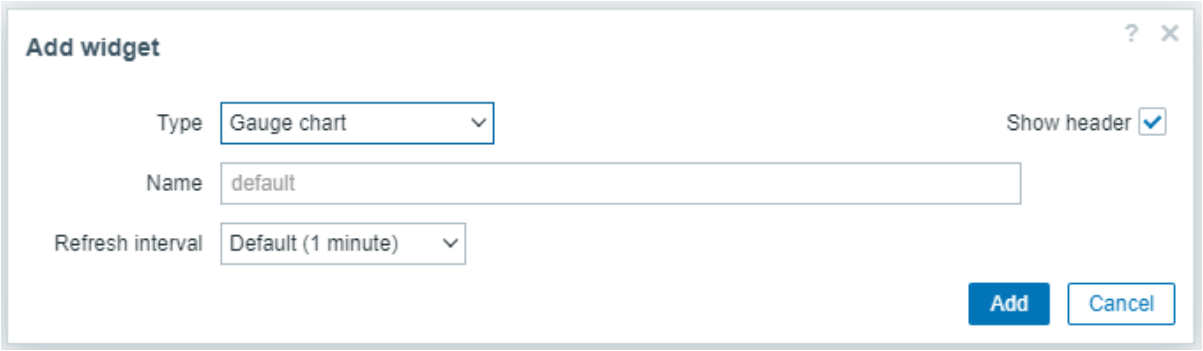
4. Suchen Sie das neue Modul *Gauge chart* in der Liste und klicken Sie auf den Hyperlink "Disabled", um den Modulstatus von "Disabled" auf "Enabled" zu ändern (falls das Modul nicht aufgeführt ist, siehe den Abschnitt Fehlerbehebung).



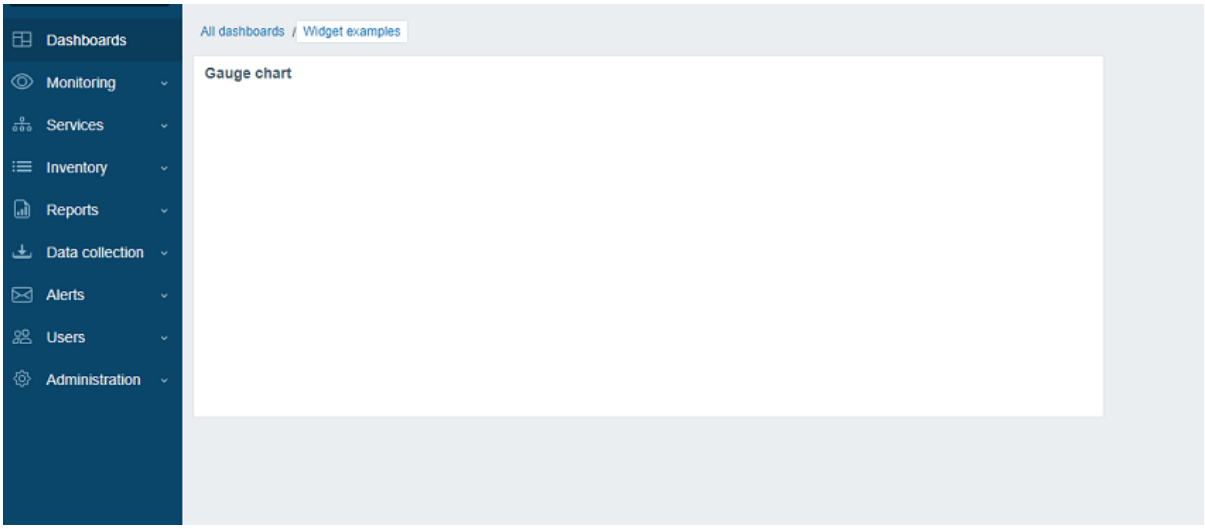
5. Öffnen Sie ein Dashboard, wechseln Sie in den Bearbeitungsmodus und fügen Sie ein neues Widget hinzu. Wählen Sie im Feld *Type* die Option "Gauge chart" aus.



6. An diesem Punkt enthält die Widget-Konfiguration von *Gauge chart* nur die allgemeinen Widget-Felder *Name* und *Refresh interval*. Klicken Sie auf *Add*, um das Widget zum Dashboard hinzuzufügen.



7. Auf dem Dashboard sollte ein leeres Widget erscheinen. Klicken Sie oben rechts auf *Save changes*, um das Dashboard zu speichern.



Eine widget-Ansicht hinzufügen

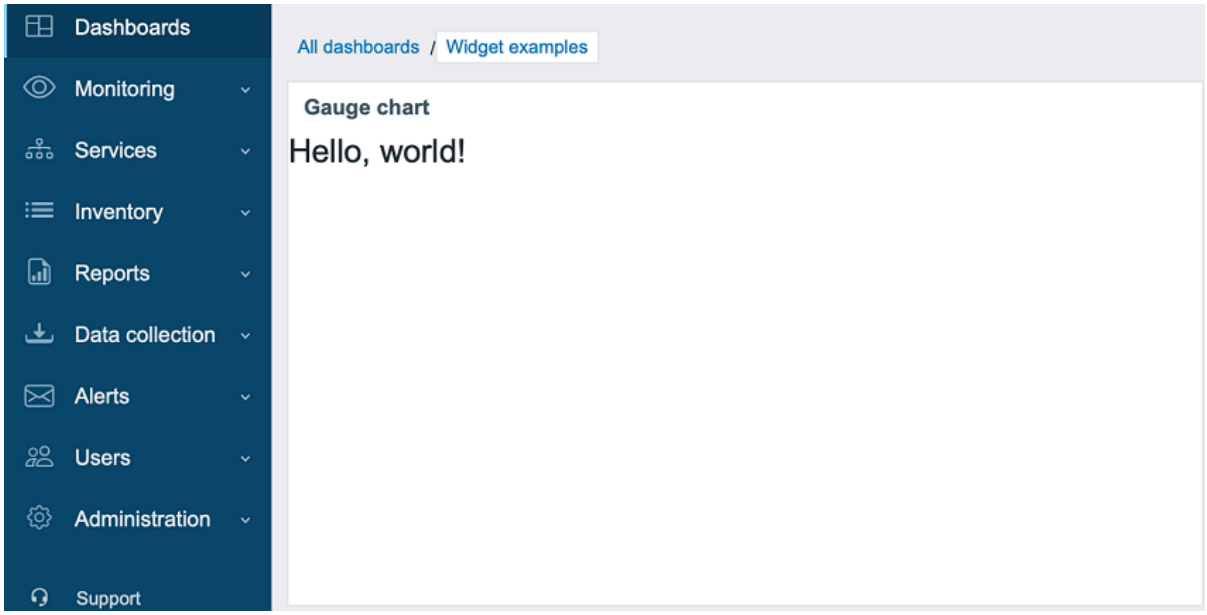
Note:
Die **Ansichts**-Datei des Widgets sollte sich im Verzeichnis *views* befinden (für dieses Tutorial: *ui/modules/lesson_gauge_chart/views/*). Wenn die Datei den Standardnamen *widget.view.php* hat, müssen Sie sie nicht in der Datei *manifest.json* registrieren. Wenn die Datei einen anderen Namen hat, geben Sie ihn 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-Ansicht.
 *
 * @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 sie 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 Klasse *WidgetForm* erstellen, die *Zabbix\Widgets\CWidgetForm* erweitert.

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 Eingabeelementtyp konvertiert den Wert in ein einheitliches Format, um ihn in der Datenbank zu speichern.

> **Note:**
> Die **form**-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 ihn im Abschnitt *widget/form_class* der Datei manifest.json an.

1. Erstellen Sie ein neues Verzeichnis *includes* im Verzeichnis *lesson_gauge_chart*.
2. Erstellen Sie eine Datei *WidgetForm.php* im Verzeichnis *includes*.

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 gewöhnliches Textfeld, in das ein Benutzer beliebige Zeichen eingeben kann. Sie können dafür 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;
```

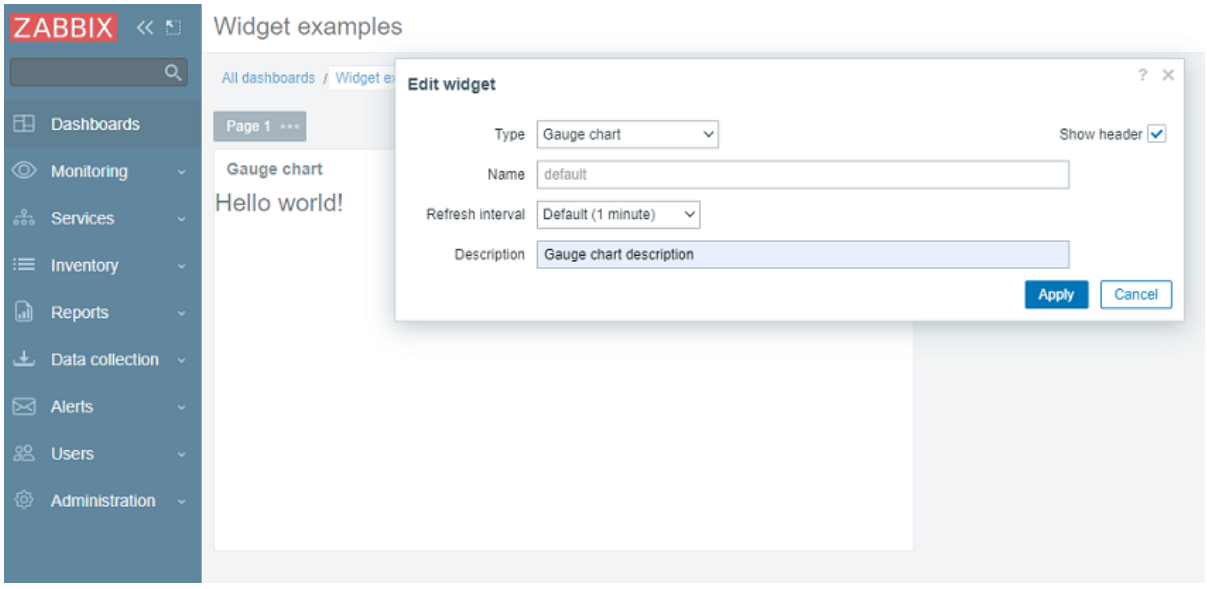
```
class WidgetForm extends CWidgetForm {

    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* ist 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 im Widget auf das Zahnradsymbol, 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 dann 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 erscheint, muss der Wert des Feldes *Description* aus der Datenbank abgerufen und an die Widget-Ansicht übergeben werden. Dazu müssen Sie eine Aktionsklasse erstellen.

8. Erstellen Sie ein neues Verzeichnis *actions* im Verzeichnis *lesson_gauge_chart*.
9. Erstellen Sie eine Datei *WidgetView.php* im Verzeichnis *actions*. Die Aktionsklasse *WidgetView* erweitert die 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_chart.view*.

ui/modules/lesson_gauge_chart/manifest.json

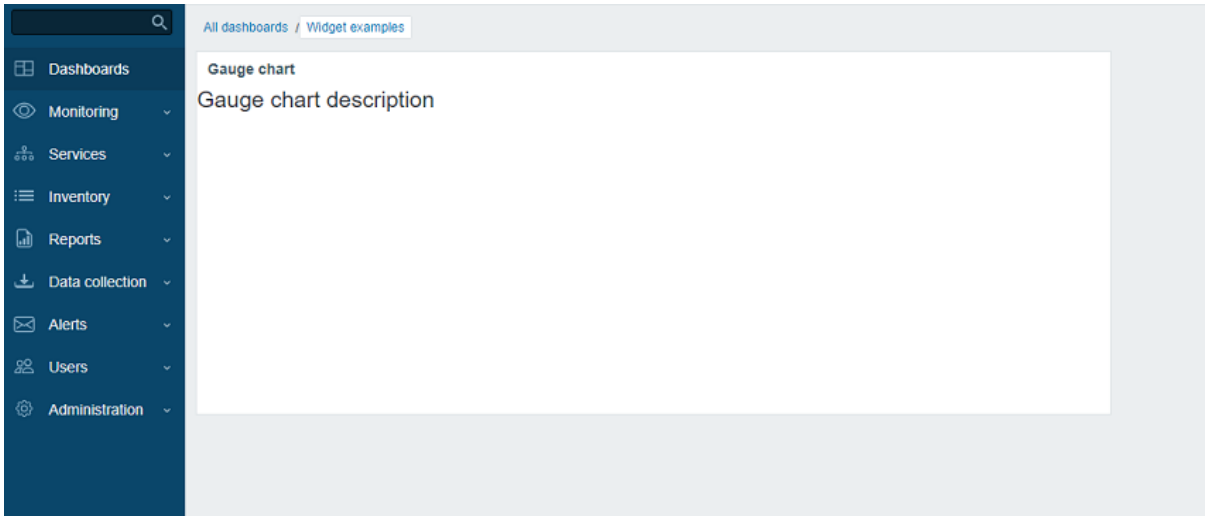
```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. Jetzt können Sie den Wert des Beschreibungsfeldes, 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 jetzt den Beschreibungstext des Widgets anstelle von „Hello, world!“ sehen.



Einen Datenpunktwert über die API abrufen

Das Widget soll den letzten Wert eines Datenpunkts nach Wahl des Benutzers anzeigen. Dazu müssen Sie die Möglichkeit hinzufügen, Datenpunkte in der Widget-Konfiguration auszuwählen.

In diesem Abschnitt erfahren Sie, wie Sie dem Widget-Formular ein Auswahlfeld für Datenpunkte hinzufügen und wie Sie den visuellen Teil dieses Feldes zur Konfigurationsansicht hinzufügen. Anschließend kann der Widget-Controller Daten des Datenpunkts und dessen Wert über eine API-Anfrage abrufen. Sobald der Wert empfangen wurde, kann er in der Widget-Ansicht angezeigt werden.

1. Öffnen Sie *includes/WidgetForm.php* und fügen Sie das Feld *CWidgetFieldMultiSelectItem* hinzu. Damit kann im Konfigurationsformular ein Datenpunkt ausgewählt werden.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

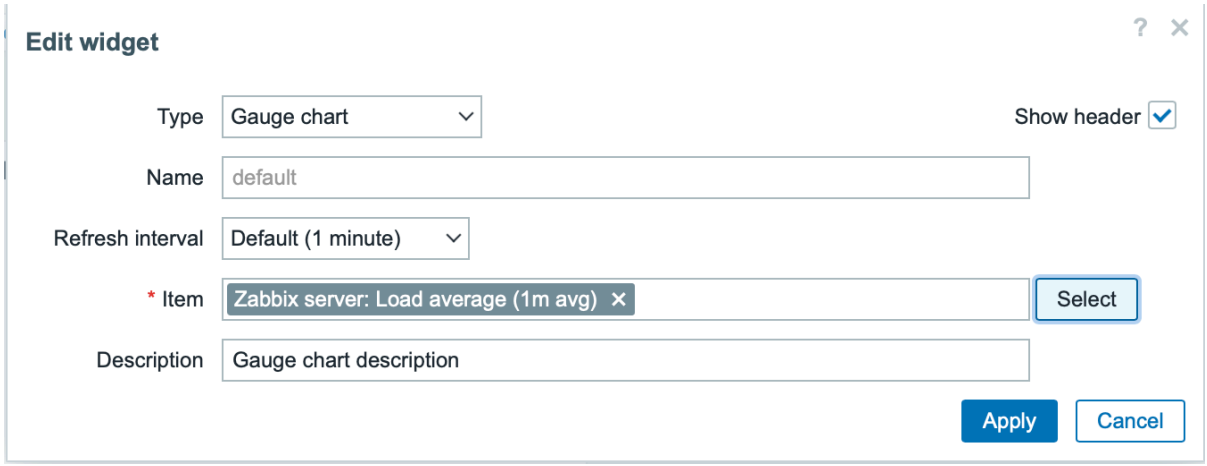
```
<?php

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 Feldes zur Konfigurationsansicht hinzu.

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 im Widget auf das Zahnradsymbol, um das Widget-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 dann 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()* sammelt die Daten des Datenpunkts (Name, Werttyp, Einheiten) mit der API-Methode *item.get* und den letzten Wert des Datenpunkts mit 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' => [
```

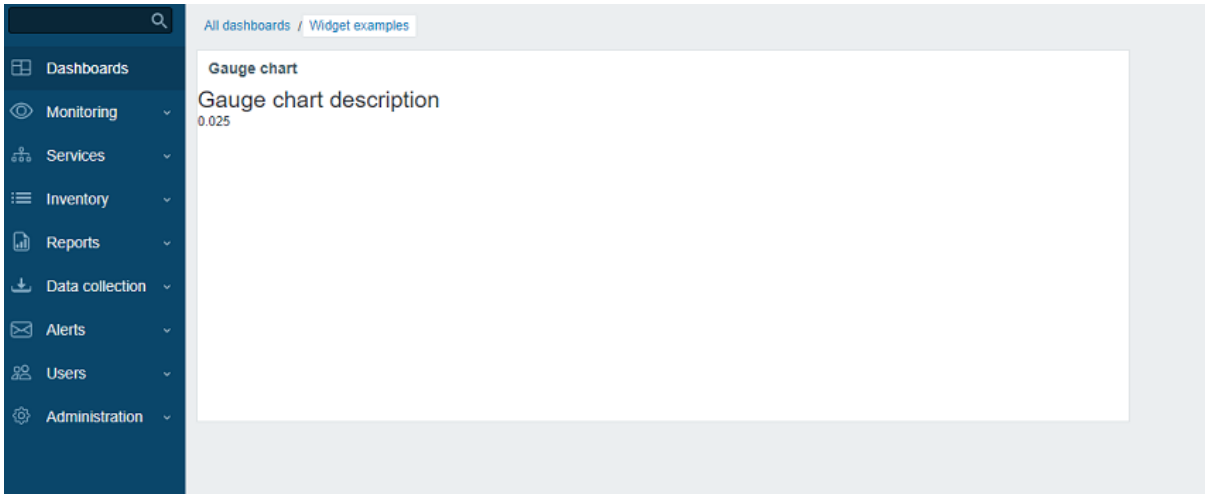
```
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $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 den neuesten Datenpunktwert an.



Erweiterte Konfigurationseinstellungen zu einer Konfigurationsansicht hinzufügen

In diesem Abschnitt erfahren Sie, wie Sie einen ausklappbaren/einklappbaren Abschnitt *Erweiterte Konfiguration* mit optionalen Parametern hinzufügen, z. B. Farbe, Minimal- und Maximalwerte, Einheiten sowie das zuvor erstellte Feld *Beschreibung*.

1. Erstellen Sie im Haupt-Widget-Verzeichnis *lesson_gauge_chart* eine Datei *Widget.php*, um eine neue Klasse *Widget* zu erstellen.

Die Klasse *Widget* erweitert die Basisklasse *CWidget*, um die Standard-Widget-Einstellungen zu ergänzen/zu überschreiben (in diesem Fall Übersetzungen). Das unten bereitgestellte JavaScript zeigt die Zeichenfolge „No data“ an, falls Daten fehlen. Die Zeichenfolge „No data“ ist in den Übersetzungsdateien der Zabbix-Benutzeroberfläche vorhanden.

Falls 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* (Farbwähler), *Min* (numerisches Feld), *Max* (numerisches Feld) und *Units* (Auswahlfeld) hinzu. Definieren Sie außerdem die Standard-Farbpalette für den Farbwähler, 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 Feldkomponenten zur Konfigurationsansicht hinzu.

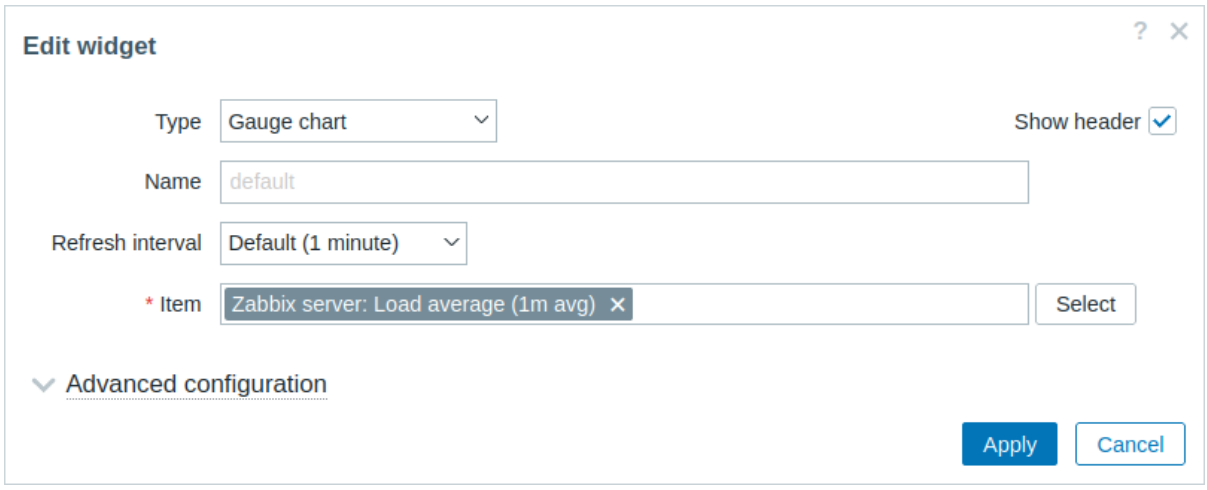
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?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();
```

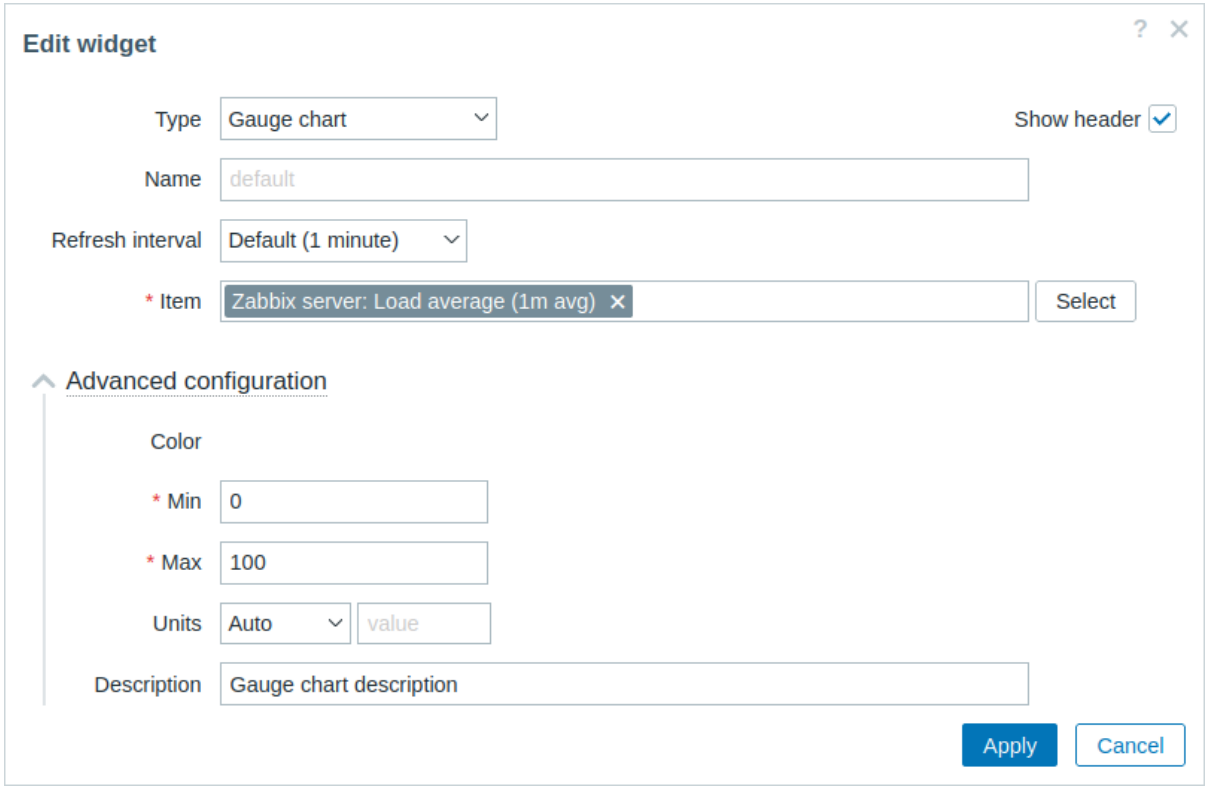
Note:
Die Methode *addField()* der Klasse *CWidgetFormView* akzeptiert als zweiten Parameter eine CSS-Klassenzeichenfolge.

4. Kehren Sie zum Dashboard zurück, wechseln Sie in den Bearbeitungsmodus und klicken Sie im Widget auf das Zahnradsymbol, um das Widget-Konfigurationsformular zu öffnen. Das Widget-Konfigurationsformular enthält jetzt einen neuen ausklappbaren/einklappbaren Abschnitt *Erweiterte Konfiguration*.



5. Klappen Sie den Abschnitt *Erweiterte Konfiguration* aus, um zusätzliche Widget-Konfigurationsfelder anzuzeigen. Beachten Sie, dass das Feld *Color* noch keinen Farbwähler hat. Das liegt daran, dass der Farbwähler mit JavaScript initialisiert werden

muss, das im nächsten Abschnitt hinzugefügt wird – *JavaScript zum Widget hinzufügen*.



JavaScript zum Widget hinzufügen

In diesem Abschnitt erfahren Sie, wie Sie ein Tachodiagramm hinzufügen – erstellt mit JavaScript –, das anzeigt, ob der letzte Wert normal oder zu hoch/zu niedrig ist.

1. Erstellen Sie im Verzeichnis *views* eine Datei *widget.edit.js.php*.

JavaScript ist für die Initialisierung der Farbauswahl 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();
    }
```

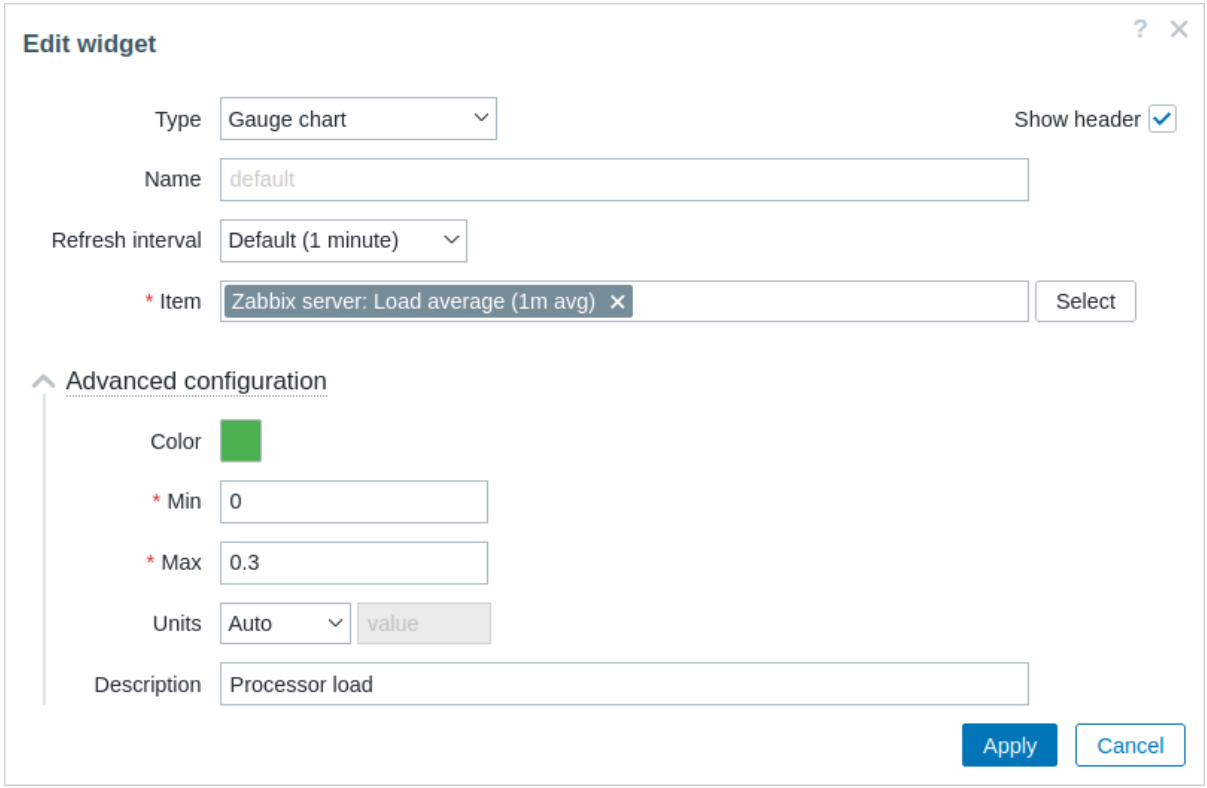
```
    updateForm() {
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

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 im Widget auf das Zahnradsymbol, um das Widget-Konfigurationsformular zu öffnen. Erweitern Sie nun den Abschnitt *Advanced configuration*, um die initialisierte Farbauswahl zu sehen. Füllen Sie die Felder mit Werten aus und wählen Sie eine Farbe für das Tachodiagramm aus.



4. Klicken Sie im Widget-Konfigurationsformular auf *Apply*. Klicken Sie dann 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 jetzt die Werte aller Felder aus *Advanced configuration*. Vervollständigen Sie den Controller, damit die Konfiguration und der Wert des ausgewählten Datenpunkts 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 Tachodiagramm 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 gegebenenfalls weiteren Assets wie Schriftarten oder Bildern verwendet.
8. Erstellen Sie für das 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 grundlegende JavaScript-Klasse aller Dashboard-Widgets – *CWidget*.

Das Dashboard ist auf eine korrekte Implementierung eines Widgets angewiesen und übermittelt dem Widget alle relevanten Informationen durch Aufruf der entsprechenden JavaScript-Methoden. Das Dashboard erwartet außerdem, 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 Erweiterung der Klasse angepasst werden kann.

In diesem Fall ist eine gewisse Anpassung erforderlich, daher wird benutzerdefinierte Logik für das folgende Widget-Verhalten implementiert:

- Widget-Initialisierung, die für die Definition des Anfangszustands des Widgets verantwortlich ist (siehe Methode *onInitialize()*);
- Anzeige des Widget-Inhalts (d. h. Zeichnen des Tachodiagramms), wenn der Widget-Aktualisierungsprozess erfolgreich und ohne Fehler abgeschlossen wurde (siehe Methode *processUpdateResponse(response)* sowie die zugehörigen Methoden *_resizeChart()* und *_updatedChart()*)
- Größenänderung des Widgets (siehe Methode *onResize()* sowie die zugehörige Methode *_resizeChart()*)

Für andere Aspekte des Tachodiagramm-Widgets wird die Standardimplementierung des Widget-Verhaltens 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 zusammen 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*** aktualisieren, wie in Schritt 10 gezeigt.

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 Folgendes 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 zusammen 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 können, um das Widget ansprechender zu gestalten.

1. Für Widget-Stile erstellen Sie ein neues Verzeichnis *css* im Verzeichnis *assets*..
2. Erstellen Sie eine Datei *widget.css* im Verzeichnis *assets/css*. Zur Gestaltung von Widget-Elementen 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;
}
```

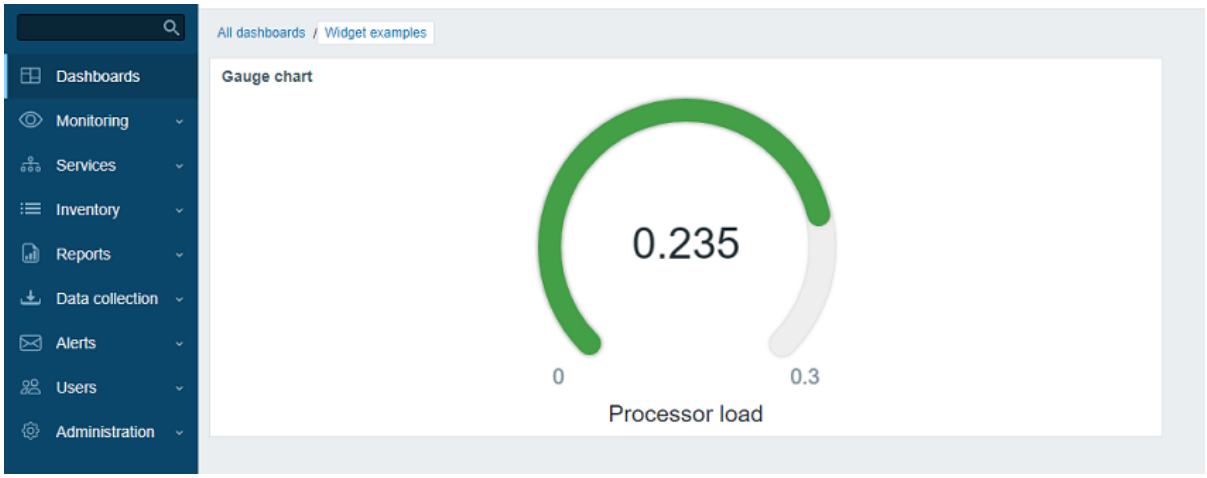
3. Öffnen Sie *manifest.json* und fügen Sie den Namen der CSS-Datei (*widget.css*) in das Array im Abschnitt *assets/css* ein. Dadurch können die in *widget.css* definierten CSS-Stile mit der Dashboard-Seite geladen werden.

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 zu sehen.



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 Entwicklung von Erweiterungen

Diese Seite listet alle Änderungen (falls vorhanden) bei der Entwicklung benutzerdefinierter Zabbix-Erweiterungen auf.

Zabbix-Manpages

Referenzdokumentation für Zabbix-Befehlszeilenwerkzeuge und -Dienste mit einer Auflistung der verfügbaren Befehle, Optionen und Anwendungsbeispiele.

##zabbix_agent2 {#manpages-zabbix_agent2}

Abschnitt: Wartungsbefehle (8)
Aktualisiert: 2019-01-29
Index Zurück zum Hauptinhalt

NAME

zabbix_agent2 - Zabbix-Agent 2

ZUSAMMENFASSUNG

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

BEZEICHNUNG

zabbix_agent2 ist eine Anwendung zur Überwachung von Parametern von verschiedenen Diensten.

OPTIONEN

-c, **--config** *Konfigurationsdatei* Verwenden Sie die alternative *Konfigurationsdatei* anstelle der Standarddatei.

-R, **--runtime-control** *runtime-option* Führen Sie Verwaltungsfunktionen gemäß *runtime-option* aus.

Laufzeitsteuerungsoptionen: userparameter_reload
User-Parameter 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 generische 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 gesetzt und mit dem Typ des zurückgegebenen Werts vorangestellt, getrennt durch ein Pipe-Zeichen. Bei User-Parametern ist der Typ immer **t**, da der Agent nicht alle möglichen Rückgabewerte bestimmen kann. Als funktionsfähig angezeigte Datenpunkte funktionieren beim Abfragen eines laufenden Agent-Daemons vom Zabbix Server oder mit zabbix_get möglicherweise nicht, da sich Berechtigungen oder die Umgebung unterscheiden können. 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 Log-Überwachungs-Datenpunkt, oder durch einen Datenpunkt, der mehrere gesammelte Werte erfordert. Berechtigungsprobleme oder fehlerhafte User-Parameter können ebenfalls zum Status „nicht unterstützt“ führen.

s
Text. Maximale Länge nicht begrenzt.

t
Text. 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 beim Testen eines Datenpunkts oder beim Ausgeben bekannter Datenpunkte aktivieren.

-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 dieser nicht während der Kompilierung geändert wurde).

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

Index

Dieses Dokument wurde erstellt am: 14:07:57 GMT, 22. November 2021

##zabbix_agent2 {#manpages-zabbix_agentd}

Abschnitt: Wartungsbefehle (8)
Aktualisiert: 2019-01-29
Index Zurück zum Hauptinhalt

NAME

zabbix_agentd – Zabbix-Agent-Daemon

ZUSAMMENFASSUNG

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 Dämon zur Überwachung verschiedener Serverparameter.

OPTIONEN

-c, **--config** *Konfigurationsdatei* Verwenden Sie die alternative *Konfigurationsdatei* anstelle der Standarddatei.

-f, **--Vordergrund** Führen Sie den Zabbix-Agenten im Vordergrund aus.

-R, **--runtime-control** *runtime-option* Führen Sie Verwaltungsfunktionen gemäß *runtime-option* aus.

Laufzeitsteuerungsoptionen userparameter_reload
Benutzerparameter aus der Konfigurationsdatei neu laden

log_level_increase[=*target*]
Log-Level erhöhen; betrifft alle Prozesse, wenn kein target angegeben ist

log_level_decrease[=*target*]
Log-Level verringern; betrifft alle Prozesse, wenn kein target angegeben ist

Ziele für die Steuerung der Protokollierungsstufe *process-type*
Alle Prozesse des angegebenen Typs (active checks, collector, listener)

process-type,N
Prozesstyp und Nummer (z. B. listener,3)

pid
Prozesskennung, bis zu 65535. Für größere Werte geben Sie das Ziel als ”process-type,N” an

-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 gesetzt 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 funktionieren nicht zwangsläufig vom Zabbix Server oder von zabbix_get aus, wenn ein laufender Agent-Daemon abgefragt wird, da Berechtigungen oder Umgebung unterschiedlich sein können. Typen zurückgegebener 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 erfasste Werte benötigt. Berechtigungsprobleme oder falsche Benutzerparameter können ebenfalls zum Status „nicht unterstützt“ führen.

s
Text. Maximale Länge nicht begrenzt.

t
Text. Wie **s**.

u
Vorzeichenlose Ganzzahl.

-t, **--test** *item-key*
Einzelnen Datenpunkt testen und beenden. Siehe **--print** für die Beschreibung der Ausgabe.

-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 dieser beim Kompilieren nicht geändert wurde).

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Index

Dieses Dokument wurde erstellt am: 20:50:13 GMT, 22. November 2021

zabbix_get

Abschnitt: Benutzerbefehle (1)
Aktualisiert: 2022-01-06
Index Zurück zum Hauptinhalt

NAME

zabbix_get - Zabbix get utility

ZUSAMMENFASSUNG

zabbix_get -s *Hostname oder IP* [**-p** *Portnummer*] [**-I** *IP-Adresse*] [**-t** *Timeout*] **-k** *Elementschlüssel*
zabbix_get -s *Hostname oder IP* [**-p** *Portnummer*] [**-I** *IP-Adresse*] [**-t** *Timeout*] **--tls-connect Zertifikat --tls-ca-file** *CA-Datei* [**--tls-crl-file** *CRL-Datei*] [**--tls-agent-cert-issuer** *Zertifikatausssteller*] [**--tls-agent-cert-subject** *Zertifikatssubjekt*] **--tls-cert-file** *Zertifikatsdatei* **--tls-key-file** *Schlüsseldatei* [**--tls-cipher13** *Chiffre-Zeichenfolge*] [**--tls-cipher** *Chiffre-Zeichenfolge*] **-k** *Elementschlüssel*
zabbix_get -s *Hostname oder IP* [**-p** *Portnummer*] [**-I** *IP-Adresse*] [**-t** *Timeout*] **--tls-connect psk --tls-psk-identity** *PSK-Identität* **--tls-psk-file** *PSK-Datei* [**--tls-cipher13** *Chiffre-Zeichenfolge*] [**--tls-cipher** *Chiffre-Zeichenfolge*] **-k** *item-key*
zabbix_get -h
zabbix_get -V

BESCHREIBUNG

zabbix_get ist ein Befehlszeilenprogramm zum Abrufen von Daten von einem Zabbix Agent.

OPTIONEN

-s, **--host** *host-name-or-IP*
Host-Namen oder IP-Adresse eines Hosts angeben.

-p, **--port** *port-number*
Portnummer des auf dem Host laufenden Agent angeben. Standard ist 10050.

-I, **--source-address** *IP-address*
Quell-IP-Adresse angeben.

-t, **--timeout** *seconds*
Zeitüberschreitung angeben. Gültiger Bereich: 1–600 Sekunden (Standard: 30)

-k, **--key** *item-key*
Schlüssel des Datenpunkts angeben, für den der Wert abgerufen werden soll.

-P, **--protocol** *value*
Für die Kommunikation mit dem Agent verwendetes Protokoll. Werte:

auto Verbindung über das JSON-Protokoll herstellen, bei Bedarf 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 (6.4.x und ältere Versionen)

--tls-connect *value*
Festlegen, wie die Verbindung zum Agent hergestellt wird. Werte:

unverschlüsselt
ohne Verschlüsselung verbinden (Standard)

psk
unter Verwendung von TLS und einem vorinstallierten Schlüssel verbinden

cert
unter Verwendung von TLS und einem Zertifikat verbinden

--tls-ca-file *CA-file*
Vollständiger Pfadname einer Datei, die die Zertifikate der übergeordneten CA(s) für die Verifizierung des Zertifikats der Gegenstelle enthält.

--tls-crl-file *CRL-file*
Vollständiger Pfadname einer Datei, die widerrufene Zertifikate enthält.

--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 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 vorinstallierten Schlüssel enthält.

--tls-cipher13 *cipher-string*
Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer für TLS 1.3.
Überschreibt die standardmäßigen Auswahlkriterien für Cipher-Suites.
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-Cipher-Zeichenfolge
(nur für TLS 1.2).
Überschreibt die standardmäßigen Auswahlkriterien für Cipher-Suites.

-h, **--help**
Diese Hilfe anzeigen und beenden.

-V, **--version**
Versionsinformationen ausgeben und beenden.

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 operations,O=Example Corp,DC=example,DC=com” --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]” --tls-connect psk --tls-psk-identity ”PSK ID Zabbix agentd” --tls-psk-file /home/zabbix/zabbix_agentd.psk

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Index

Dieses Dokument wurde erstellt am: 08:42:29 GMT, 11. Juni 2021

zabbix_js

Abschnitt: Benutzerbefehle (1)
Aktualisiert: 2022-01-06
Index Zurück zum Hauptinhalt

NAME¶

¶ zabbix_js - Zabbix JS utility

SYNOPSE¶

¶ **zabbix_js -s** *Skript Datei* **-p** *Eingabe-Parameter* [**-l**¶ *log-level*] [**-t** *Laufzeit*]¶ **zabbix_js -s** *Skript Datei* **-i** *Eingabe-Datei* [**-l**¶ *log-level*] [**-t** *Laufzeit*]¶ **zabbix_js -h**¶ **zabbix_js -V**

BESCHREIBUNG

zabbix_js ist ein Befehlszeilenprogramm, 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 Eingabeparameter an.

-i, **--input** *input-file*
Geben Sie den Dateinamen des Eingabeparameters an. Wenn als Dateiname '-' angegeben wird, wird die Eingabe von stdin gelesen.

-w, **--webdriver** *url*
Geben Sie die webdriver-URL an.

-l, **--loglevel** *log-level*
Geben Sie die Protokollebene an.

-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.

Beispiel

¶ **zabbix_js -s Skript.js -p Beispiel**

Siehe auch

¶ Dokumentation https://www.zabbix.com/manuals¶ ¶ **zabbix_agent2**(8),¶ **zabbix_agentd**(8),¶ **zabbix_get**(1), **zabbix_proxy**(8),¶ **zabbix_sender**(1),¶ **zabbix_server**(8)¶

Index

Dieses Dokument wurde erstellt am: 21:23:35 GMT, März 18, 2020

zabbix_proxy

Abschnitt: Wartungsbefehle (8)
Aktualisiert: 2020-09-04
Index Zurück zum Hauptinhalt

###NAME

zabbix_proxy - Zabbix proxy Dämon

ZUSAMMENFASSUNG

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

BESCHREIBUNG

zabbix_proxy ist ein Daemon, der Monitoring-Daten von Geräten sammelt und sie an den Zabbix Server sendet.

OPTIONEN

-c, **--config** *config-datei*
Nutze die alternative *config-datei* anstatt der Standard-Datei.

-f, **--foreground**
Führe Zabbix Proxy im Vordergrund aus.

-R, **--runtime-control** *runtime-option*
Führe administrative Funktionen laut *runtime-option* aus.

Laufzeitsteuerungsoptionen config_cache_reload
Konfigurations-Cache neu laden.
Wird ignoriert, wenn der Cache gerade geladen wird.
Ein aktiver Zabbix Proxy verbindet sich mit dem Zabbix Server und fordert Konfigurationsdaten an.
Die Standardkonfigurationsdatei wird verwendet (sofern die Option **-c** nicht angegeben ist), um die PID-Datei zu finden, und das Signal wird an den in der PID-Datei aufgeführten Prozess gesendet.

snmp_cache_reload
SNMP-Cache neu laden.
Beachten Sie, dass Zabbix SNMPv3-EngineID→IP-Zuordnungen zwischenspeichert und EngineIDs möglicherweise automatisch wiederverwendet, um den Polling-Overhead zu reduzieren.

housekeeper_execute
Den Housekeeper ausführen.
Wird ignoriert, wenn der Housekeeper gerade ausgeführt wird.

diaginfo[=*section*]
Interne Diagnoseinformationen des angegebenen Abschnitts protokollieren.

Der Abschnitt kann *historycache*, *preprocessing* oder *locks* sein.
Standardmäßig werden Diagnoseinformationen aller Abschnitte protokolliert.

log_level_increase[=*target*]
Log-Level erhöhen; betrifft alle Prozesse, wenn kein Ziel angegeben ist.

log_level_decrease[=*target*]
Log-Level verringern; betrifft alle Prozesse, wenn kein Ziel angegeben ist.

Ziele für die Steuerung der Protokollierungsstufe *process-type*
Alle Prozesse des angegebenen Typs (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector)

process-type,N
Prozesstyp und Nummer (z. B. poller,3)

pid
Prozesskennung, bis zu 65535. Für größere Werte geben Sie das Ziel als ”process-type,N” an.

Ziele für die Steuerung des Profilings

process-type
Alle Prozesse des angegebenen Typs (configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, preprocessing manager, icmp pinger, ipmi manager, ipmi poller, java poller, poller, agent poller, http agent poller, snmp poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector, history poller, availability manager, odbc poller)

process-type,N
Prozesstyp und Nummer (z. B. history syncer,1)

pid
Prozesskennung, bis zu 65535. Für größere Werte geben Sie das Ziel als ”process-type,N” an.

scope
Profiling-Bereich (rwlock, mutex, processing) kann zusammen mit dem Prozesstyp verwendet werden (z. B. history syncer,1,processing)

-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_proxy.conf
Standardpfad der Zabbix Proxy-Konfigurationsdatei (falls dieser nicht während der Kompilierung geändert wurde).

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Index

Dieses Dokument wurde erstellt am: 16:12:22 GMT, 04. September 2020

zabbix_sender

Abschnitt: Benutzerbefehle (1)
Aktualisiert: 2021-06-01
Index Zurück zum Hauptinhalt

NAME

zabbix_sender – Zabbix-Sender-Dienstprogramm

ÜBERSICHT

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**] [**-g**] **-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**] [**-g**] **-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**] [**-g**] **-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**] [**-g**] **-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**] [**-g**] **-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-identity* **--tls-psk-file** *PSK-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 psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender -h
zabbix_sender -V

BESCHREIBUNG

zabbix_sender ist ein Befehlszeilenprogramm zum Senden von Überwachungsdaten an den Zabbix-Server oder -Proxy. Auf dem Zabbix-Server sollte ein Element vom Typ **Zabbix-trapper** mit entsprechendem Schlüssel erstellt werden. Beachten Sie, dass eingehende Werte nur von Hosts akzeptiert werden, die im Feld **Zulässige Hosts** für dieses Element angegeben sind.

OPTIONEN

-c, **--config** *config-file*
Verwende *config-file*. **Zabbix sender** liest Server-Details aus der agentd-Konfigurationsdatei. Standardmäßig liest **Zabbix sender** keine Konfigurationsdatei. Nur die Parameter **Hostname**, **ServerActive**, **SourceIP**, **TLSConnect**, **TLSCAFile**, **TLSCRLFile**,

TLSServerCertIssuer, **TLSServerCertSubject**, **TLSCertFile**, **TLSKeyFile**, **TLSPSKIdentity** und **TLSPSKFile** werden unterstützt.
Ein über den Parameter **HostnameItem** definierter Hostname wird nicht übernommen; in diesem Fall sollte der Hostname über die Befehlszeile angegeben werden (siehe Option -s).
Alle im Agent-Konfigurationsparameter **ServerActive** definierten Adressen werden zum Senden von Daten verwendet. Wenn das Senden von Stapeldaten an eine Adresse fehlschlägt, werden die folgenden Stapel 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 Proxys verwendet werden. Bei gemeinsamer Verwendung mit **--config** überschreibt diese Option die im Parameter **ServerActive** der agentd-Konfigurationsdatei angegebenen Einträge.

-p, **--port** *port*
Gibt die Portnummer des auf dem Server laufenden Zabbix-Server-Trappers an. Standard ist 10051. Bei gemeinsamer Verwendung mit **--config** überschreibt diese Option die Port-Einträge des Parameters **ServerActive** in der agentd-Konfigurationsdatei.

-l, **--source-address** *IP-address*
Gibt die Quell-IP-Adresse an. Bei gemeinsamer Verwendung mit **--config** überschreibt diese Option den in der agentd-Konfigurationsdatei angegebenen Parameter **SourceIP**.

-t, **--timeout** *seconds*
Gibt das Timeout an. Gültiger Bereich: 1–300 Sekunden (Standard: 60)

-s, **--host** *host*
Gibt den Hostnamen an, zu dem der Datenpunkt gehört (wie im Zabbix Frontend registriert). Die Host-IP-Adresse und der DNS-Name funktionieren nicht. Bei gemeinsamer Verwendung mit **--config** überschreibt diese Option den in der agentd-Konfigurationsdatei angegebenen Parameter **Hostname**.

-k, **--key** *key*
Gibt den Schlüssel des Datenpunkts an, an den der Wert gesendet werden soll.

-o, **--value** *value*
Gibt den Wert des Datenpunkts an.

-i, **--input-file** *input-file*
Lädt Werte aus einer Eingabedatei. Gib **-** als **<input-file>** an, um Werte von der Standardeingabe zu lesen. Jede Zeile der Datei enthält durch Leerraum getrennt: **<hostname> <key> <value>**. Jeder Wert muss in einer eigenen Zeile angegeben werden. Jede Zeile muss 3 durch Leerraum getrennte Einträge enthalten: **<hostname> <key> <value>**, wobei „hostname“ der Name des überwachten Hosts ist, wie er im Zabbix Frontend registriert ist, „key“ der Schlüssel des Zieldatenpunkts und „value“ der zu sendende Wert. Gib **-** als **<hostname>** an, um den Hostnamen aus der Agent- Konfigurationsdatei oder aus dem Argument **--host** zu verwenden.

Ein Beispiel für eine Zeile in einer Eingabedatei:

”Linux DB3” db.connections 43

Der Werttyp muss in der Datenpunkt-Konfiguration des Zabbix Frontends korrekt gesetzt sein. Zabbix sender sendet bis zu 250 Werte in einer Verbindung. Das Größenlimit für das Senden von Werten aus einer Eingabedatei hängt von der im Zabbix-Kommunikationsprotokoll beschriebenen Größe ab. Der Inhalt der Eingabedatei muss in UTF-8-Kodierung vorliegen. Alle Werte aus der Eingabedatei werden in sequentieller Reihenfolge von oben nach unten gesendet. Einträge müssen nach den folgenden Regeln formatiert werden:

- Einträge in Anführungszeichen und ohne Anführungszeichen werden unterstützt.
- Das doppelte Anführungszeichen ist das Zeichen zum Maskieren.
- Einträge mit Leerraum müssen in Anführungszeichen gesetzt werden.
- Doppelte Anführungszeichen und Backslash-Zeichen innerhalb eines in Anführungszeichen gesetzten Eintrags müssen mit einem Backslash maskiert werden.
- Escaping wird in nicht in Anführungszeichen gesetzten Einträgen nicht unterstützt.
- Escape-Sequenzen für Zeilenumbrüche (\n) werden in Zeichenketten in Anführungszeichen unterstützt.
- Escape-Sequenzen für Zeilenumbrüche werden am Ende eines Eintrags abgeschnitten.

-T, **--with-timestamps**
Diese Option kann nur zusammen mit der Option **--input-file** verwendet werden.

Jede Zeile der Eingabedatei muss 4 durch Leerraum getrennte Einträge enthalten: **<hostname> <key> <timestamp> <value>**. Der Zeitstempel sollte im Unix-Zeitstempelformat angegeben werden. Wenn der Zieldatenpunkt Auslöser hat, die auf ihn verweisen, müssen alle Zeitstempel in aufsteigender Reihenfolge vorliegen, andernfalls ist die Ereignisberechnung nicht korrekt.

Ein Beispiel für eine Zeile in der Eingabedatei:

”Linux DB3” db.connections 1429533600 43

Weitere Details siehe Option **--input-file**.

Wenn ein Wert mit Zeitstempel für einen Host gesendet wird, der sich in einer Wartung vom Typ „keine Daten“ befindet, wird dieser Wert verworfen; es ist jedoch möglich, einen Wert mit Zeitstempel für einen abgelaufenen Wartungszeitraum zu senden, und dieser wird akzeptiert.

-N, **--with-ns**
Diese Option kann nur zusammen mit der Option **--with-timestamps** verwendet werden.

Jede Zeile der Eingabedatei muss 5 durch Leerraum getrennte Einträge enthalten: **<hostname> <key> <timestamp> <ns> <value>**.

Ein Beispiel für eine Zeile in der Eingabedatei:

”Linux DB3” db.connections 1429533600 7402561 43

Weitere Details siehe Option **--input-file**.

-r, **--real-time**
Sendet Werte einzeln, sobald sie empfangen werden. Dies kann beim Lesen von der Standardeingabe verwendet werden.

-g, **--group**
Gruppiert Werte nach Hosts und sendet sie für jeden Host in einem separaten Stapel.

--tls-connect *value*
Wie die Verbindung zum Server oder Proxy hergestellt wird. Werte:

unverschlüsselt
connect without encryption (default)

psk
eine Verbindung mit TLS und einem vorab freigegebenen Schlüssel herstellen

cert
eine Verbindung mit TLS und einem Zertifikat herstellen

--tls-ca-file *CA-file*
Vollständiger Pfadname einer Datei, welche die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält.

--tls-crl-file *CRL-file*
Vollständiger Pfadname einer Datei, die widerrufene Zertifikate enthält.

--tls-server-cert-issuer *cert-issuer*
Erlaubter Aussteller des Serverzertifikats.

--tls-server-cert-subject *cert-subject*
Erlaubter 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-Kennzeichenfolge.

--tls-psk-file *PSK-file*
Vollständiger Pfadname einer Datei, die den Pre-Shared Key enthält.

--tls-cipher13 *cipher-string*
Chiffrierzeichenfolge für OpenSSL 1.1.1 oder neuer für TLS 1.3. Überschreibt die Standardkriterien für die Auswahl der Ciphersuite. 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-Verschlüsselungszeichenfolge (nur für TLS 1.2). Überschreiben Sie die Standardkriterien für die Auswahl der Chiffriersuite.

-v, **--verbose**
Verbose Modus, **-vv** für weitere Einzelheiten.

-h, **--help**
Diese Hilfe anzeigen und beenden.

-V, **--version**
Versionsinformationen ausgeben und beenden.

EXIT-STATUS

Der Exit-Status ist 0, wenn die Werte erfolgreich gesendet und vom Server vollständig verarbeitet wurden. Wurden Daten gesendet, aber mindestens einer der Werte konnte nicht verarbeitet werden, ist der Exit-Status 2. Wenn das Senden der Daten fehlschlägt, beträgt der Exit-Status 1.

BEISPIELE

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

Sendet **342.45** als Wert für das Element **mysql.queries** des überwachten Host. Verwenden Sie den überwachten Host und den Zabbix-Server, die in der Agenten Konfigurationsdatei definiert sind.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s „Überwachter Host“ -k mysql.queries -o 342.45

Senden Sie **342.45** als Wert für den Eintrag **mysql.queries** des **Überwachten Host** über den in der Konfigurationsdatei des Agenten definierten Zabbix-Server.

zabbix_sender -z 192.168.1.113 -i data_values.txt

Werte aus der Datei **data_values.txt** an den Zabbix-Server mit der IP senden **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 von der Kommandozeile an den Zabbix-Server, der in der Konfigurationsdatei des Agenten angegeben ist. Bindestrich in den Eingabedaten zeigt an, dass auch der Hostname aus derselben Konfigurationsdatei verwendet werden soll. Datei stammt.

echo '„Zabbix server“ trapper.item „“ ' | zabbix_sender -z 192.168.1.113 -p 10000 -i -

Leeren Wert eines Items an den Zabbix-Server mit IP-Adresse senden **192.168.1.113** an Port **10000** über die Kommandozeile. Leere Werte müssen durch leere doppelte Anführungszeichen gekennzeichnet werden.

zabbix_sender -z 192.168.1.113 -s „Überwachter Host“ -k mysql.queries -o 342.45 --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-cert-datei /home/zabbix/zabbix_agentd.crt --tls-key-datei /home/zabbix/zabbix_agentd.key

Senden Sie **342.45** als Wert für den Eintrag **mysql.queries** in **Monitored Host** an den Server mit der IP **192.168.1.113** unter Verwendung von TLS mit Zertifikat.

zabbix_sender -z 192.168.1.113 -s „Überwachter Host“ -k mysql.queries -o 342.45 --tls-connect psk --tls-psk-identity „PSK ID Zabbix agentd“ --tls-psk-datei /home/zabbix/zabbix_agentd.psk

Senden Sie **342.45** als Wert für den Eintrag **mysql.queries** in **Monitored Host** an den Server mit der IP **192.168.1.113** unter Verwendung von TLS mit Pre-Shared Key (PSK).

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Index

Dieses Dokument wurde erstellt am: 08:42:39 GMT, Juni 11, 2021

##zabbix_server {#manpages-zabbix_server}

Abschnitt: Wartungsbefehle (8)
Aktualisiert: 2020-09-04
Index Zurück zum Hauptinhalt

NAME

zabbix_server – Zabbix-Server-Daemon

ZUSAMMENFASSUNG

zabbix_server [**-c** *Konfigurationsdatei*]
zabbix_server [**-c** *Konfigurationsdatei*] **-R** *Laufzeitoption*
zabbix_server [**-c** *Konfigurationsdatei*] **-T**
zabbix_server -h
zabbix_server -V

BESCHREIBUNG

zabbix_server ist der zentrale Daemon der Zabbix-Software.

OPTIONEN

-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 Verwaltungsfunktionen gemäß *runtime-option* aus.

-T, **--test-config** Validiert die Konfigurationsdatei und beendet das Programm.

-h, **--help** Zeigt diese Hilfe an und beendet das Programm.

-V, **--version** Gibt Versionsinformationen aus und beendet das Programm.

Beispiele für das Ausführen des Zabbix-Servers mit Befehlszeilenparametern:

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

RUNTIME CONTROL

Laufzeitsteuerungsoptionen:

config_cache_reload
Konfigurations-Cache neu laden.
Wird ignoriert, wenn der Cache gerade geladen wird.

Die Standardkonfigurationsdatei (sofern nicht die Option **-c** 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
SNMP-Cache neu laden, die SNMP-Eigenschaften (Engine-Zeit, Engine-Boots, Engine-ID, Anmeldedaten) für alle Hosts löschen. Beachten Sie, dass Zabbix SNMPv3-EngineID→IP-Zuordnungen zwischenspeichert und EngineIDs möglicherweise automatisch wiederverwendet, um den Abfrage-Overhead zu reduzieren.

housekeeper_execute
Den Housekeeper ausführen.
Wird ignoriert, wenn der Housekeeper gerade ausgeführt wird.

trigger_housekeeper_execute
Den Auslöser-Housekeeper ausführen (Probleme für gelöschte Auslöser entfernen).
Wird ignoriert, wenn der Auslöser-Housekeeper für Services gerade ausgeführt wird.

diaginfo[=*section*]
Interne Diagnoseinformationen des angegebenen Abschnitts protokollieren.
Der Abschnitt kann *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*, *locks* sein.
Standardmäßig werden Diagnoseinformationen aller Abschnitte protokolliert.

ha_status
Den Status des Hochverfügbarkeitsclusters (HA) protokollieren.

ha_remove_node[=*target*]
Entfernt den High-Availability-(HA)-Knoten, der durch seinen Namen oder seine ID angegeben ist.
Beachten Sie, dass aktive/Standby-Knoten nicht entfernt werden können.

ha_set_failover_delay[=*delay*]
Legt die High-Availability-(HA)-Failover-Verzögerung fest.
Zeitsuffixe werden unterstützt, z. B. 10s, 1m.

proxy_config_cache_reload[=*target*]
Lädt den Proxy-Konfigurations-Cache neu.

secrets_reload
Lädt Secrets aus Vault neu.

service_cache_reload
Lädt den Service-Manager-Cache neu.

prof_enable[=*target*]
Aktiviert Profiling.
Wirkt sich auf alle Prozesse aus, wenn *target* nicht angegeben ist.
Aktiviertes Profiling liefert Details zu allen rwlocks/Mutexen nach Funktionsnamen.

prof_disable[=*target*]
Deaktiviert Profiling.
Wirkt sich auf alle Prozesse aus, wenn *target* nicht angegeben ist.

log_level_increase[=*target*]
Erhöht die Protokollierungsstufe; wirkt sich auf alle Prozesse aus, wenn *target* nicht angegeben ist

log_level_decrease[=*target*]
Verringert die Protokollierungsstufe; wirkt sich auf alle Prozesse aus, wenn *target* nicht angegeben ist

Ziele für die Steuerung der Protokollierungsstufe *process-type*
Alle Prozesse des angegebenen Typs (alerter, alert manager, availability manager, browser poller, configuration syncer, configuration syncer worker, connector manager, connector worker, discovery manager, escalator, ha manager, history poller, 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, proxy group 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. poller,3)

pid
Prozesskennung, bis zu 65535. Für größere Werte geben Sie das Ziel als ”process-type,N” an

Ziele für die Steuerung des Profilings

process-type
Alle Prozesse des angegebenen Typs (alerter, alert manager, availability manager, browser poller, configuration syncer, configuration syncer worker, connector manager, connector worker, discovery manager, escalator, ha manager, history poller, 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, proxy group 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, bis zu 65535. Für größere Werte geben Sie das Ziel als ”process-type,N” an

scope
Profiling-Bereich (rwlock, mutex, processing) kann zusammen mit dem Prozesstyp verwendet werden (z. B. history syncer,1,processing)

DATEIEN

/usr/local/etc/zabbix_server.conf
Standardpfad der Zabbix-Server-Konfigurationsdatei (falls dieser beim Kompilieren nicht geändert wurde).

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

Index

Dieses Dokument wurde erstellt am: 16:12:14 GMT, 04. September 2020

zabbix_web_service

Abschnitt: Wartungsbefehle (8)
Updated: 2019-01-29
Index Zurück zum Hauptinhalt

NAME

zabbix_web_service - Zabbix-Webdienst

Zusammenfassung

zabbix_web_service [**-c** *Konfigurationsdatei*]
zabbix_web_service [**-c** *Konfigurationsdatei*] **-T**
zabbix_web_service -h
zabbix_web_service -V

Beschreibung

zabbix_web_service ist eine Anwendung zur Bereitstellung von Webdiensten für Zabbix-Komponenten.

OPTIONS

-c, **--config** *config-file*
Verwende die alternative *Konfigurationsdatei* anstelle der Standarddatei.

-T, **--test-config**
Validiere die Konfigurations-Datei und beende das Programm.

-h, **--help**
Zeige diese Hilfe an und beende das Programm.

-V, **--version**
Gibt Versionsinformationen aus und beende das Programm.

DATEIEN

/usr/local/etc/zabbix_web_service.conf
Standardspeicherort der Zabbix-Webdienst-Konfigurationsdatei (sofern nicht während der Kompilierung geändert).

SIEHE AUCH

Dokumentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Index

Dieses Dokument wurde erstellt am: 12:58:30 GMT, Juni 11, 2021

Urheberrechtshinweis