Documentation 8.2

ZABBIX

05.11.2026

Contents

Manuale utente **5**

Benvenuti nella documentazione di Zabbix **5**

Inizia a usare Zabbix . . . 5
Zabbix Cloud . . . 5
App mobile . . . 5
Centro sviluppatori . . . 5
Community e altre risorse . . . 5
Installazione e primi passi . . . 6
Installazione dell’interfaccia web . . . 6
Guide rapide . . . 11
Installazione: guide aggiuntive . . . 85
Aggiornamento . . . 166
Problemi noti . . . 182
Requisiti . . . 191
Novità e note sull’aggiornamento: 8.0 . . . 201
Note di aggiornamento . . . 201
Modifiche incompatibili . . . 202
Altro . . . 202
Novità e note sull’aggiornamento: 8.0.x . . . 202
Appliance Zabbix . . . 202
Processi . . . 206
server . . . 208
proxy . . . 235
agent . . . 280
Agent 2 . . . 311
Java gateway . . . 374
Mittente . . . 378
Ottieni . . . 381
JS . . . 382
Servizio web . . . 383
Adattatore bridge . . . 385
server MCP . . . 389
Variabili di ambiente . . . 396
Configurazione . . . 398
Host e gruppi di host . . . 409
Item . . . 427
Rilevamento dei problemi con i trigger . . . 653
Visualizzazione . . . 705
Template e gruppi di template . . . 731
Template predefiniti pronti all’uso . . . 731
Notifiche sugli eventi . . . 745
Macro . . . 797
Utenti e gruppi di utenti . . . 812
Report pianificati . . . 823
Esportazione dei dati . . . 827
Esportazione/importazione della configurazione . . . 836
Espressioni regolari . . . 902
Eventi e risoluzione . . . 908
Riconoscimento e soppressione dei problemi . . . 910
Chiusura manuale dei problemi . . . 914

Periodi di manutenzione . . . 915
Etichettatura . . . 921
Complex event processing . . . 923
Correlazione degli eventi . . . 927
Scoperta . . . 934
Rilevamento della rete . . . 934
Autoregistrazione dell'agent attivo . . . 943
Scoperta a basso livello . . . 947
Monitoraggio delle macchine virtuali . . . 970
Chiavi degli item per il monitoraggio VMware . . . 972
Campi chiave di discovery delle macchine virtuali . . . 990
Esempi JSON per gli item VMware . . . 994
Esempio di configurazione del monitoraggio VMware . . . 999
Monitoraggio web . . . 1004
Elementi di monitoraggio web . . . 1013
Scenario reale . . . 1014
Monitoraggio dei servizi . . . 1021
Albero dei servizi . . . 1022
SLA . . . 1026
Esempio di configurazione . . . 1027
Aggiornamento del monitoraggio dei servizi . . . 1032
Crittografia e sicurezza . . . 1032
Crittografia tramite certificati . . . 1041
Crittografia tramite chiavi pre-condivise . . . 1049
Connessione sicura al database . . . 1052
Connessione sicura tra frontend e server . . . 1064
Controllo degli accessi . . . 1068
Crittografia . . . 1072
Proteggere il server web . . . 1073
Archiviazione dei segreti . . . 1076
Fiducia del certificato TLS di Google Chrome . . . 1085
Best practice di sicurezza . . . 1086
Interfaccia web . . . 1087
Menu . . . 1087
Sezioni del frontend . . . 1097
Impostazioni utente . . . 1372
Ricerca globale . . . 1377
Modalità di manutenzione del frontend . . . 1379
Parametri della pagina . . . 1380
Definizioni . . . 1383
Lingue aggiuntive del frontend . . . 1383
Creare il proprio tema . . . 1384
Modalità di debug . . . 1385
Cookie utilizzati da Zabbix . . . 1386
Fusi orari . . . 1387
Reimpostazione della password . . . 1388
Periodo di tempo e selettori host . . . 1389
Parametri di configurazione: frontend Zabbix . . . 1390
API . . . 1397
Riferimento ai metodi . . . 1402
Appendice 1. Commento di riferimento . . . 2284
Appendice 2. Modifiche dalla 7.4 alla 8.0 . . . 2291
Appendice 3. Modifiche nella versione 8.0 . . . 2291
Estensioni . . . 2291
Moduli caricabili . . . 2293
Moduli del frontend . . . 2302
Appendici . . . 2303
Installazione e configurazione . . . 2303
Item . . . 2350
Funzioni supportate . . . 2391
Macro . . . 2391
Compatibilità delle versioni . . . 2429
Sintassi del periodo di tempo . . . 2430

Simboli delle unità . . . 2431
Esecuzione dei comandi . . . 2432
Esempi di escaping . . . 2433
Cos'è Zabbix . . . 2434

Zabbix Cloud **2439**

Zabbix Cloud **2439**
Configurare e gestire Zabbix Cloud . . . 2439
Scopri Zabbix Cloud . . . 2439
Distribuire Zabbix nel cloud . . . 2439
Configurazione del nodo . . . 2449
Panoramica . . . 2450
Scheda Panoramica . . . 2451
Scheda Filtri di accesso . . . 2454
Scheda Crittografia . . . 2455
Scheda Backup . . . 2455
Scheda Cronologia . . . 2456
Scheda Manutenzione . . . 2457
Scheda Upgrade . . . 2457
Aggiunta di utenti . . . 2459
Introduzione . . . 2459
Aggiunta di utenti alle organizzazioni . . . 2459
Aggiunta di utenti ai nodi . . . 2460
Differenze principali di Zabbix Cloud . . . 2461
Introduzione . . . 2461
Registro di audit . . . 2463
Configurare un proxy Zabbix Cloud . . . 2465

App mobile **2473**

App mobile **2473**
Configurare Zabbix per l'app . . . 2473
Scopri l'app . . . 2473
Configurare Zabbix per l'app . . . 2474
Panoramica dell'app . . . 2475
Accesso mobile sicuro . . . 2483

Centro sviluppatori **2484**

Centro sviluppatori **2484**
Sviluppare su Zabbix . . . 2484
Creazione di un plugin . . . 2484
Interfacce plugin . . . 2490
Libreria Python per Zabbix . . . 2492
Installazione . . . 2492
Guida rapida . . . 2492
Utilizzo della API di Zabbix . . . 2496
Raccogliere dati da Zabbix agent . . . 2499
Inviare dati al server o al proxy Zabbix . . . 2500
Logging di debug . . . 2504
Moduli . . . 2505
Struttura del file del modulo . . . 2505
Widget . . . 2513
Tutorial . . . 2521
Esempi . . . 2548
Modifiche allo sviluppo delle estensioni . . . 2548

Pagine man di Zabbix **2549**
zabbix_agent2 . . . 2549
zabbix_agentd . . . 2550
zabbix_get . . . 2552
zabbix_js . . . 2554
zabbix_proxy . . . 2555
zabbix_sender . . . 2557

zabbix_server 2562
zabbix_web_service 2564

Avviso sul copyright **2565**

Manuale utente

Benvenuti nella documentazione di Zabbix

La risorsa di riferimento per lavorare con il monitoraggio Zabbix, dalle configurazioni di base a quelle avanzate. Questo manuale copre tutto ciò che serve per installare, configurare e utilizzare Zabbix.

Novità in Zabbix 8.0

Inizia a usare Zabbix

Installazione Istruzioni dettagliate per installare Zabbix sulla piattaforma preferita, con supporto per vari sistemi operativi e configurazioni di database.

Requisiti Un elenco delle piattaforme supportate e dei prerequisiti software per aiutarti a preparare il tuo ambiente a una distribuzione di Zabbix di successo.

Guide introduttive Guide pratiche e concise, orientate alle attività, che ti accompagnano attraverso le basi: dai primi passaggi di configurazione fino alla ricezione del primo avviso di problema.

Zabbix Cloud

Inizia a usare Zabbix Cloud Istruzioni guidate per mettere rapidamente in funzione la tua istanza Zabbix Cloud, inclusa la configurazione iniziale e la connessione dei dispositivi monitorati.

Zabbix Cloud vs. on-premises Confronta le differenze in termini di gestione, scalabilità e manutenzione tra i due tipi di deployment e scopri quale può supportare il tuo flusso di lavoro nel modo più fluido.

Esplora Zabbix Cloud Ottieni una panoramica di Zabbix Cloud, incluse le sue funzionalità principali, i vantaggi e ciò che è incluso in un abbonamento. Per le domande più comuni su configurazione e utilizzo, consulta le FAQ.

App mobile

Scarica l'app Installa l'app mobile Zabbix (disponibile prossimamente) sul tuo dispositivo e inizia a monitorare il tuo ambiente Zabbix ovunque ti trovi.

Configura Zabbix per l'app Installa l'adattatore bridge di Zabbix per consentire ai dispositivi mobili di connettersi a Zabbix. Quindi aggiungi il tuo dispositivo e attiva le notifiche push.

Panoramica dell'app Scopri le nozioni di base sull'utilizzo dell'app mobile Zabbix.

Centro sviluppatori

Moduli del frontend Guide e riferimenti per creare moduli personalizzati che estendono o modificano il frontend di Zabbix per adattarlo a casi d'uso specifici.

Widget Una panoramica della struttura e della logica dei widget, con istruzioni per creare elementi dashboard personalizzati in base alle tue esigenze.

Plugin Una panoramica su come sviluppare e gestire i plugin di Zabbix per estendere le funzionalità o integrarsi con sistemi esterni.

Community e altre risorse

Forum di Zabbix Uno spazio per scambiare idee, trovare soluzioni e condividere esperienze a tutti i livelli di competenza su Zabbix.

Blog di Zabbix Notizie, tutorial e casi di studio curati dal team di Zabbix e dai contributori della community.

Video tutorial Una libreria video con dimostrazioni, discussioni e suggerimenti per aiutarti a sfruttare al meglio Zabbix.

Installazione e primi passi

La seguente sequenza illustra il processo di configurazione di Zabbix:

1. Verificare che il software richiesto sia installato e in esecuzione.
2. Installare Zabbix.
3. Installare l'interfaccia web di Zabbix.
4. Avviare il monitoraggio.

Di seguito sono riportati i dettagli di ciascun passaggio.

Software richiesto Zabbix funziona bene con i seguenti software (spesso chiamati stack LAMP):

- **Sistema operativo Linux:** Ubuntu, Debian, CentOS o altri
- **Web server:** Apache o NGINX
- **Database server:** MySQL/MariaDB o PostgreSQL
- **PHP**

Nella maggior parte dei sistemi Linux moderni, PHP potrebbe essere già installato. In tal caso, potrebbe essere necessario configurare solo il database server e il web server.

Per informazioni più dettagliate, ad esempio sulle versioni dei software supportate, vedere Software richiesto.

Installare Zabbix Il modo **più semplice** per installare Zabbix consiste nell'utilizzare i **pacchetti della distribuzione** (in alternativa, utilizzando i container). I pacchetti installano le dipendenze software minori.

Le istruzioni di installazione precise, adattate al sistema operativo in uso, sono disponibili sul **sito web di Zabbix**.

Selezionare la piattaforma e seguire le istruzioni fornite. Per iniziare, è consigliabile installare un Zabbix server, un Zabbix agent 2 e il frontend (interfaccia web). In seguito è possibile aggiungere proxy Zabbix e altri agent.

Parte del processo di installazione consiste nella creazione di un database (per conservare i dati di configurazione e dello storico) e nell'importazione dei dati iniziali al suo interno.

Installare l'interfaccia web di Zabbix Completare i passaggi della **procedura guidata di installazione** dell'interfaccia web.

Verrà richiesta la password del database (creata nel passaggio precedente), affinché il frontend possa comunicare con il database.

Avviare il monitoraggio Accedere all'interfaccia web di Zabbix.

Utilizzare le **guide rapide** per iniziare a monitorare risorse comuni (Linux, Apache, MySQL o altre). Queste guide rapide spiegano con quali template lavorare per iniziare a monitorare le risorse più comuni.

È possibile iniziare con la guida rapida per il monitoraggio di un server Linux.

Installazione dell'interfaccia web

Questa sezione fornisce istruzioni per installare l'interfaccia web di Zabbix. Seguire questi passaggi dopo aver installato il backend di Zabbix.

Il frontend di Zabbix è scritto in PHP e richiede un web server supportato, come Apache o NGINX, con PHP configurato. Per informazioni sul software e sulle librerie richiesti, consultare i requisiti del frontend.

Per utilizzare una lingua diversa dall'inglese nell'interfaccia web di Zabbix, è necessario installare la relativa locale sul web server, insieme all'estensione PHP gettext. Tenere presente che la modifica della lingua può modificare il formato di data e ora.

Per configurare SSL per il frontend di Zabbix, consultare Configurazione di SSL per il frontend di Zabbix.

Procedura guidata di installazione

Per installare il frontend, aprire l'URL del frontend Zabbix in un browser:

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

Schermata di benvenuto

La procedura guidata di installazione del frontend si apre sulla schermata di benvenuto.

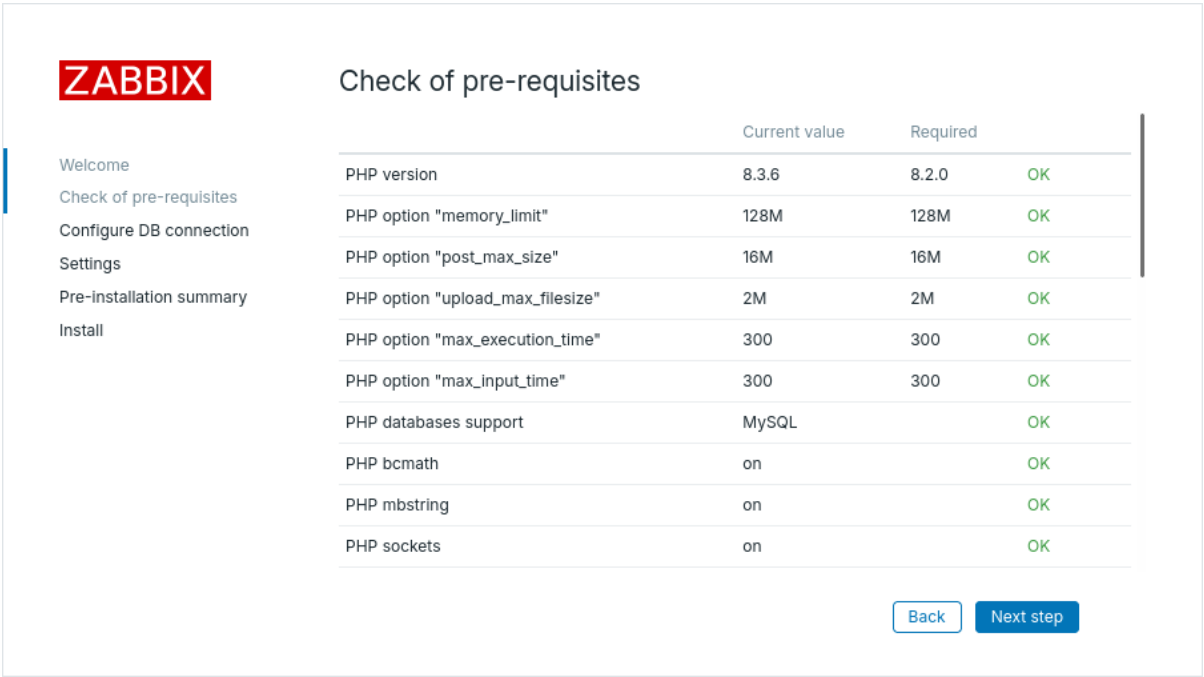


La lingua predefinita a livello di sistema è l'inglese (en_US), con il formato di ora/data statunitense. Per modificare la lingua predefinita a livello di sistema, selezionare la lingua desiderata dal menu a discesa *Lingua predefinita*.

Fare clic sul pulsante *Passaggio successivo* per continuare.

Verifica dei prerequisiti

Assicurarsi che tutti i prerequisiti obbligatori del frontend Zabbix siano soddisfatti.



Vedere Requisiti per un elenco delle estensioni PHP obbligatorie e facoltative, la versione PHP richiesta e i database supportati. I pre-requisiti facoltativi non soddisfatti vengono visualizzati con lo stato *Avviso*; il processo di configurazione può comunque proseguire anche se non sono soddisfatti.

Valori richiesti per le direttive di configurazione PHP nel file `php.ini` utilizzato dal server web:

```
memory_limit = 128M ; Valore minimo
post_max_size = 16M ; Valore minimo
upload_max_filesize = 2M ; Valore minimo
```

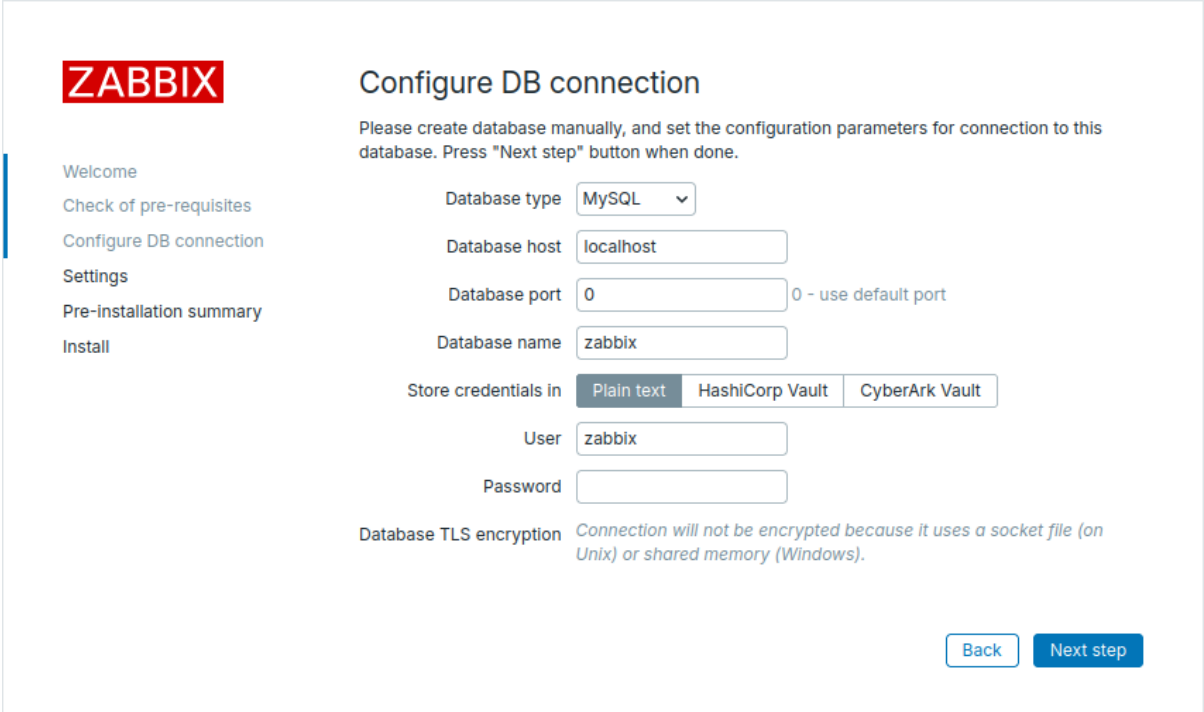
```
max_execution_time = 300 ; Valore minimo; sono consentiti anche i valori 0 e -1
max_input_time = 300 ; Valore minimo; sono consentiti anche i valori 0 e -1
session.auto_start = 0 ; Deve essere disabilitato
arg_separator.output = "&" ; È consentito anche "&amp;"
```

Attention:
Se è necessario modificare l'utente o il gruppo del server web, verificare che il nuovo utente o gruppo disponga delle autorizzazioni necessarie per la directory delle sessioni PHP; in caso contrario, la configurazione di Zabbix potrebbe non poter proseguire.

Se tutti i prerequisiti obbligatori sono soddisfatti, procedere al passaggio successivo.

Configura la connessione al DB

Inserisci i dettagli per la connessione al database.



Se il campo *Host del database* viene lasciato vuoto, viene utilizzato il socket Unix predefinito.

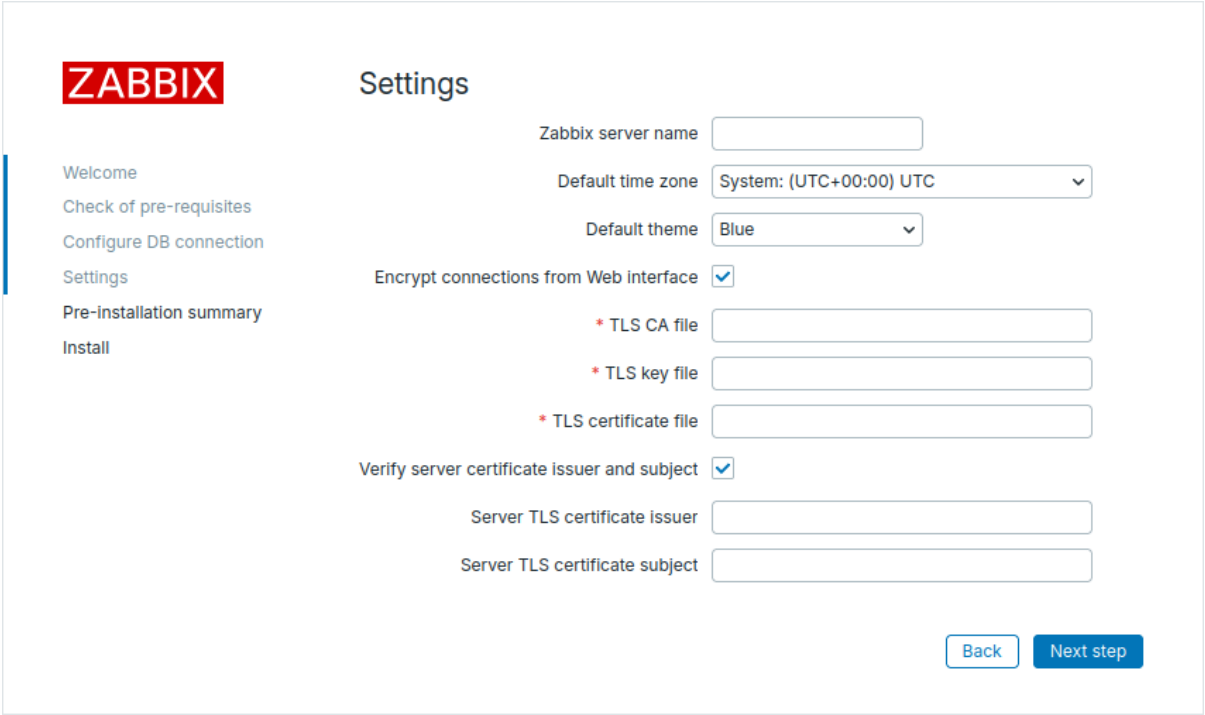
Per MySQL, inserire `localhost` nel campo *Host del database* comporta anch'esso una connessione tramite il socket Unix predefinito. La procedura guidata non fornisce un'opzione per configurare il socket del database. Per utilizzare un socket personalizzato, configuralo nelle impostazioni del server Zabbix (ad esempio, con il parametro DBSocket= nel file di configurazione del server Zabbix (`zabbix_server.conf`)). In questo modo il frontend rimane coerente con le impostazioni della connessione dal server al database.

Per PostgreSQL, il campo *Host del database* può contenere più voci `host:port` separate da virgole. La stringa viene suddivisa in parti e ogni porta viene convalidata nello stesso modo del campo *Porta del database*. Dopo la convalida, la stringa viene analizzata per ricavare host e porte e costruire la stringa di connessione. Se per un host nell'elenco separato da virgole non è definita alcuna porta, viene utilizzato come porta predefinita il valore di *Porta del database*. Se viene specificato un percorso del socket (ad esempio, `/var/run/pgbouncer`), viene utilizzato invece quel socket Unix.

Per configurare una connessione sicura al database, seleziona la casella di controllo *Crittografia TLS del database* e procedi a Frontend al database per configurare i campi. La casella di controllo *Crittografia TLS del database* è disabilitata nei seguenti casi: - Per MySQL, *Host del database* è impostato su *localhost*. - Per PostgreSQL, il valore del campo *Host del database* inizia con una barra oppure il campo è vuoto.

Per configurare Zabbix in modo che recuperi i segreti da HashiCorp Vault, seleziona *HashiCorp Vault* per *Archivia credenziali in* e procedi a Configurazione di HashiCorp per compilare i campi espansi. Per configurare Zabbix in modo che recuperi i segreti da CyberArk Vault, seleziona *CyberArk Vault* per *Archivia credenziali in* e procedi a Configurazione di CyberArk per compilare i campi espansi.

Impostazioni



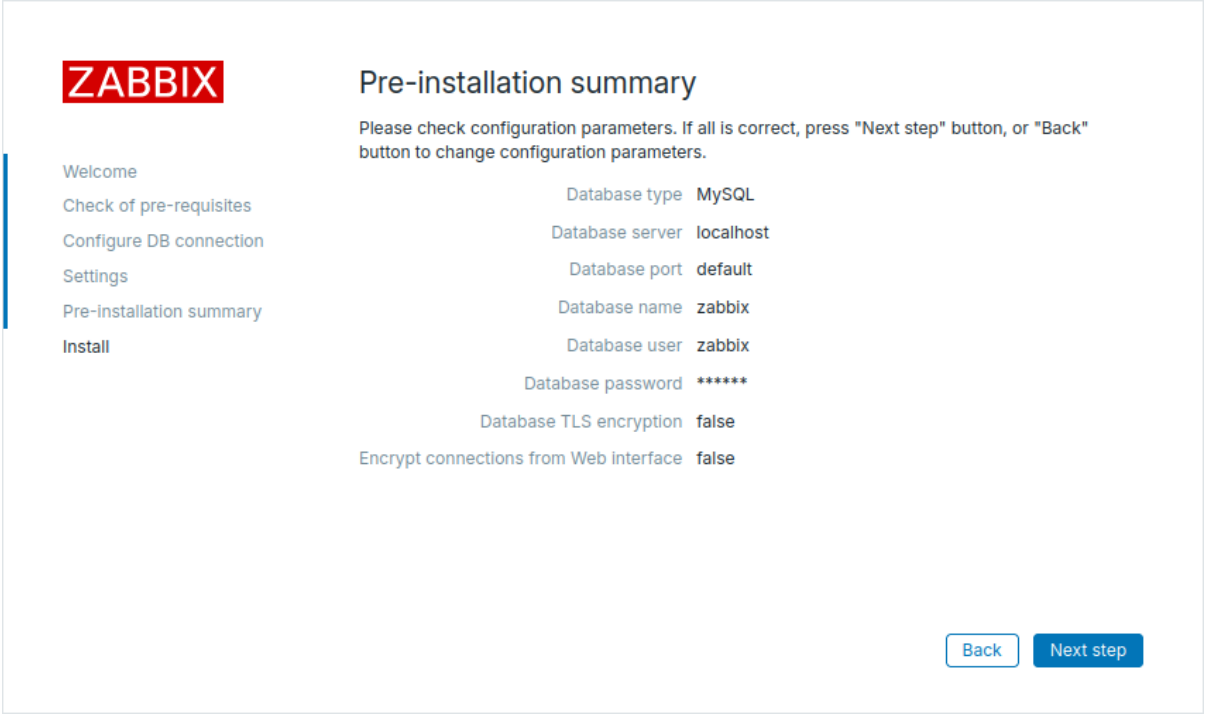
L'inserimento di un nome per il server Zabbix è facoltativo; tuttavia, se viene specificato, verrà visualizzato nella barra dei menu e nei titoli delle pagine.

Impostare il fuso orario predefinito e il tema per il frontend.

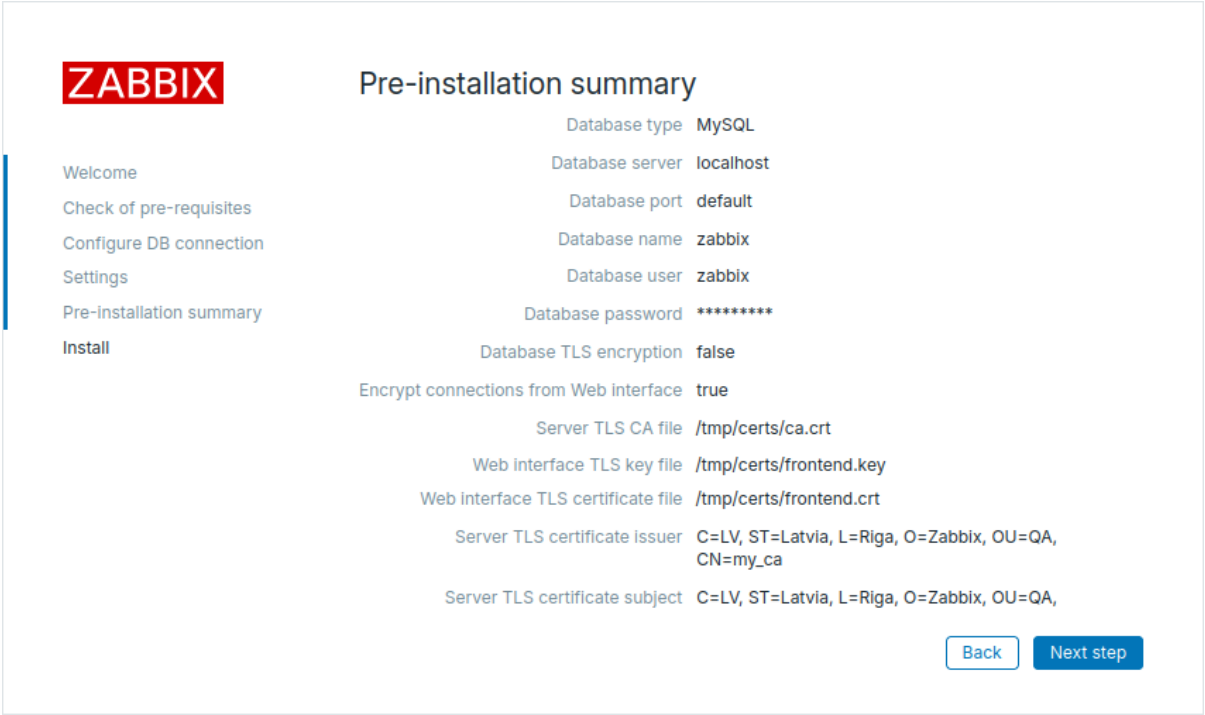
Per configurare le connessioni TLS tra il frontend Zabbix e il server Zabbix, selezionare la casella di controllo *Crittografa le connessioni dall'interfaccia Web* e procedere alla configurazione della connessione TLS per compilare i campi espansi.

Riepilogo della preinstallazione

Esaminare un riepilogo delle impostazioni.



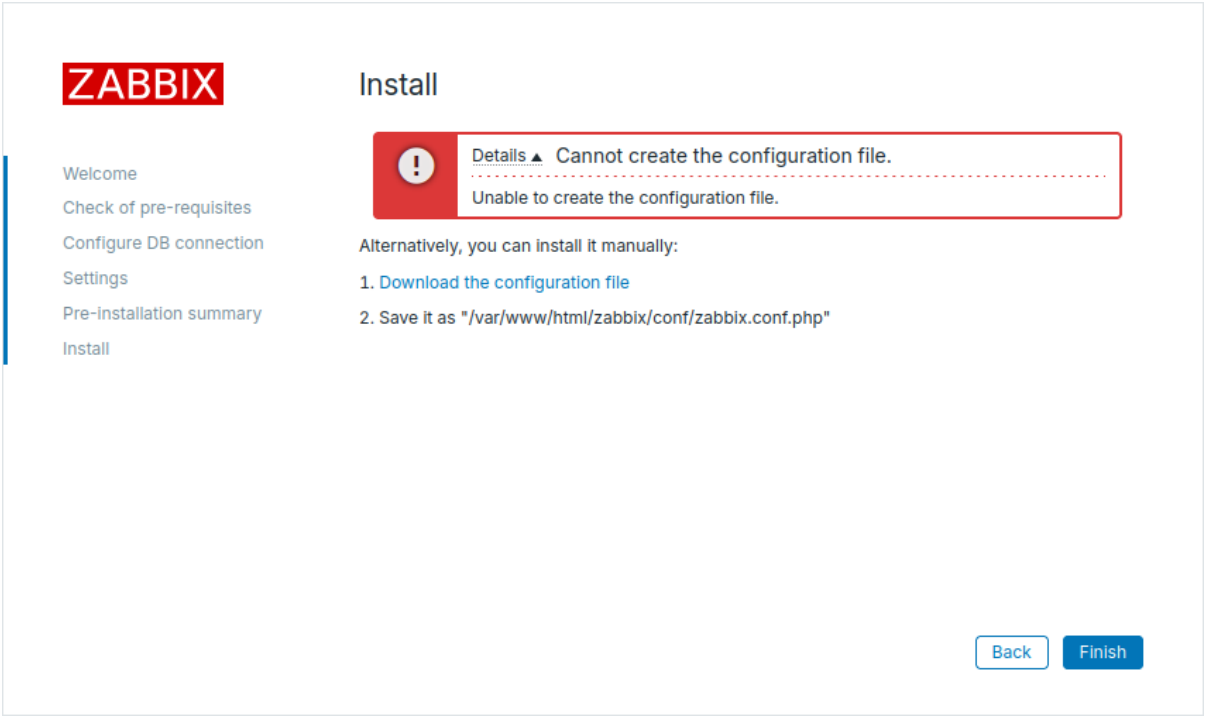
Se TLS è configurato, la procedura guidata ne visualizzerà i dettagli.



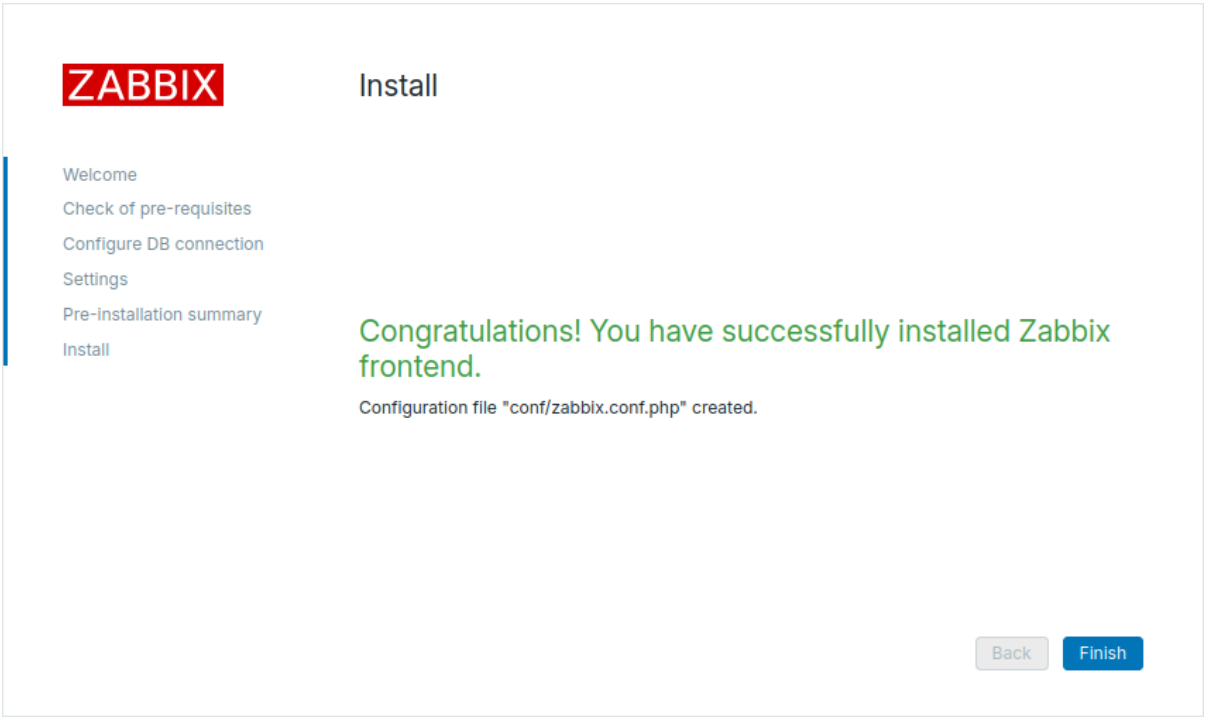
Installazione

Se l'utente del web server dispone dell'accesso in scrittura alla directory `conf/`, il file di configurazione verrà salvato automaticamente e sarà possibile procedere subito al passaggio successivo.

Se l'utente del web server non dispone dell'accesso in scrittura alla directory `conf/`, è necessario scaricare il file di configurazione e inserirlo nella directory `conf/` all'interno della directory dei documenti HTML del web server, dove sono stati copiati i file PHP di Zabbix.

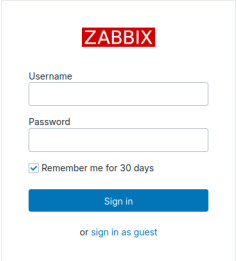


Completare l'installazione.



Accedi

Accedi con le credenziali predefinite: - Nome utente: *Admin* - Password: *zabbix*



File di configurazione del frontend

Dopo l'installazione, è possibile riconfigurare il frontend di Zabbix aggiornando i parametri nel file di configurazione del frontend `zabbix.conf.php`.

Il percorso predefinito del file di configurazione dipende dal sistema operativo e dal metodo di installazione di Zabbix:

- Per l'installazione dai pacchetti: `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- Per l'installazione dai sorgenti: `/var/www/html/zabbix/conf/zabbix.conf.php`
- Per l'appliance: `/etc/zabbix/web/zabbix.conf.php`

Per informazioni dettagliate sulla configurazione del frontend di Zabbix, vedere le opzioni del file di configurazione.

Guide rapide

Dopo aver installato Zabbix e la relativa interfaccia web, è il momento di iniziare il monitoraggio.

Utilizza queste guide rapide per iniziare a monitorare risorse comuni:

- Monitoraggio di un server Linux
- Monitoraggio di un server web Apache
- Monitoraggio di un server MySQL

Le guide rapide si concentrano sull'utilizzo dei template ufficiali di Zabbix per il monitoraggio. Un template è un insieme preconfigurato di elementi di monitoraggio. Un template consente di scalare il monitoraggio: puoi testare un template su un singolo target di monitoraggio (dispositivo, app, ecc.), modificarlo e quindi applicarlo a più target di monitoraggio simili.

Sebbene i template siano adatti al monitoraggio immediato, potresti essere interessato anche a personalizzarne gli elementi.

Per comprendere meglio gli elementi di base che compongono un template, consulta Configurazione di base.

Configurazione di base

Queste guide introducono gli elementi di base della configurazione di Zabbix:

- Accesso e configurazione dell'utente
- Nuovo host
- Nuovo item
- Nuovo trigger
- Ricezione della notifica di problema
- Nuovo template

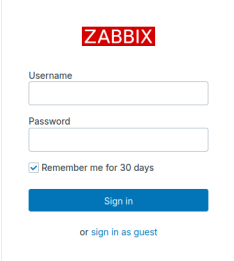
Accesso e configurazione dell'utente

Panoramica

In questa sezione, imparerai come accedere e configurare un utente di sistema in Zabbix.

Accesso

Aprire l'URL del frontend Zabbix in un browser: - Per Apache: http://<server_ip_or_name>/zabbix - Per NGINX: http://<server_ip_or_name>



Questa è la schermata di benvenuto di Zabbix. Inserire il nome utente **Admin** con la password **zabbix** per accedere come superuser Zabbix. Verrà concesso l'accesso a tutte le sezioni del menu.

Per motivi di sicurezza, si consiglia vivamente di modificare immediatamente la password predefinita dell'account Admin dopo il primo accesso.

Accesso persistente

Per rimanere connessi fino a 30 giorni, selezionare *Ricordami per 30 giorni* prima di fare clic su *Accedi*.

Ricordami per 30 giorni abilitato:

- La sessione rimane attiva per 30 giorni.
- *Disconnessione automatica* viene ignorata, mantenendo l'accesso fino alla fine del periodo.
- Alle visite future entro 30 giorni, l'accesso verrà eseguito automaticamente senza reinserire le credenziali.

Ricordami per 30 giorni disabilitato:

- Qualsiasi accesso automatico precedentemente abilitato viene cancellato.
- La sessione scadrà in base all'intervallo di *Disconnessione automatica* configurato.

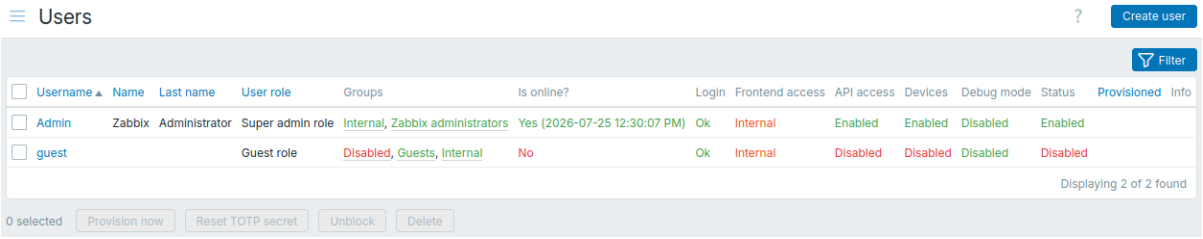
Protezione contro gli attacchi brute force

In caso di cinque tentativi di accesso falliti consecutivi, l'interfaccia Zabbix si metterà in pausa per 30 secondi per prevenire attacchi brute force e attacchi a dizionario.

L'indirizzo IP di un tentativo di accesso fallito verrà visualizzato dopo un accesso riuscito.

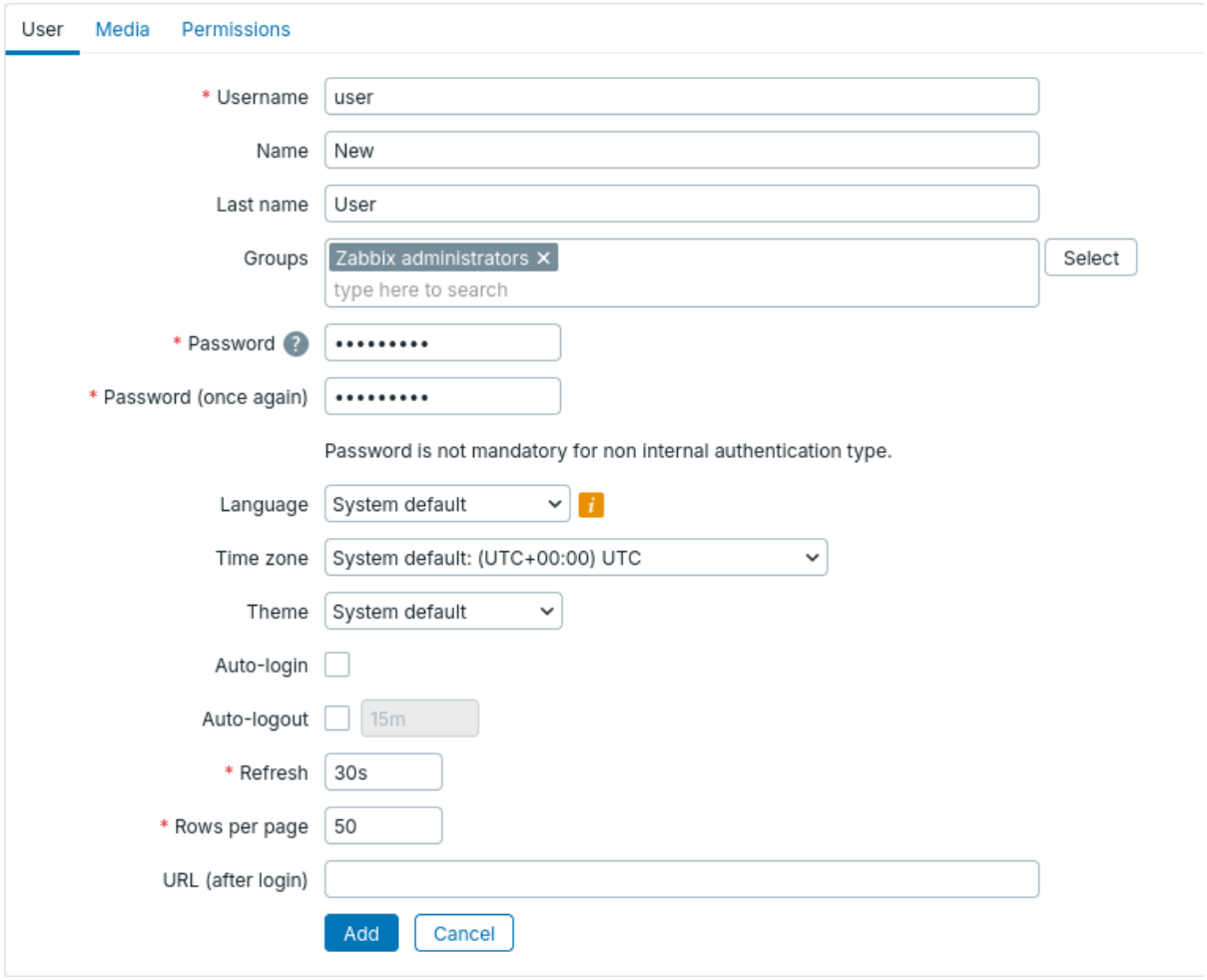
Aggiunta di un utente

Per visualizzare le informazioni sugli utenti, vai a *Users* > *Users* nel menu verticale della barra laterale.



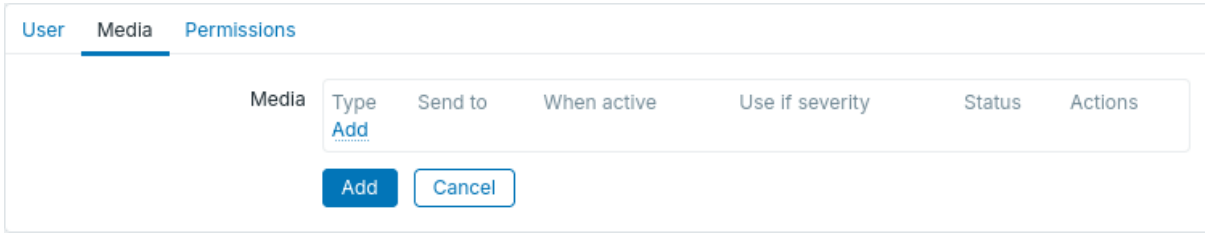
Per aggiungere un nuovo utente, seleziona *Create user* nell'angolo in alto a destra.

Nel nuovo modulo utente, assicurati di aggiungere l'utente a uno dei gruppi di utenti esistenti, ad esempio 'Zabbix administrators'.



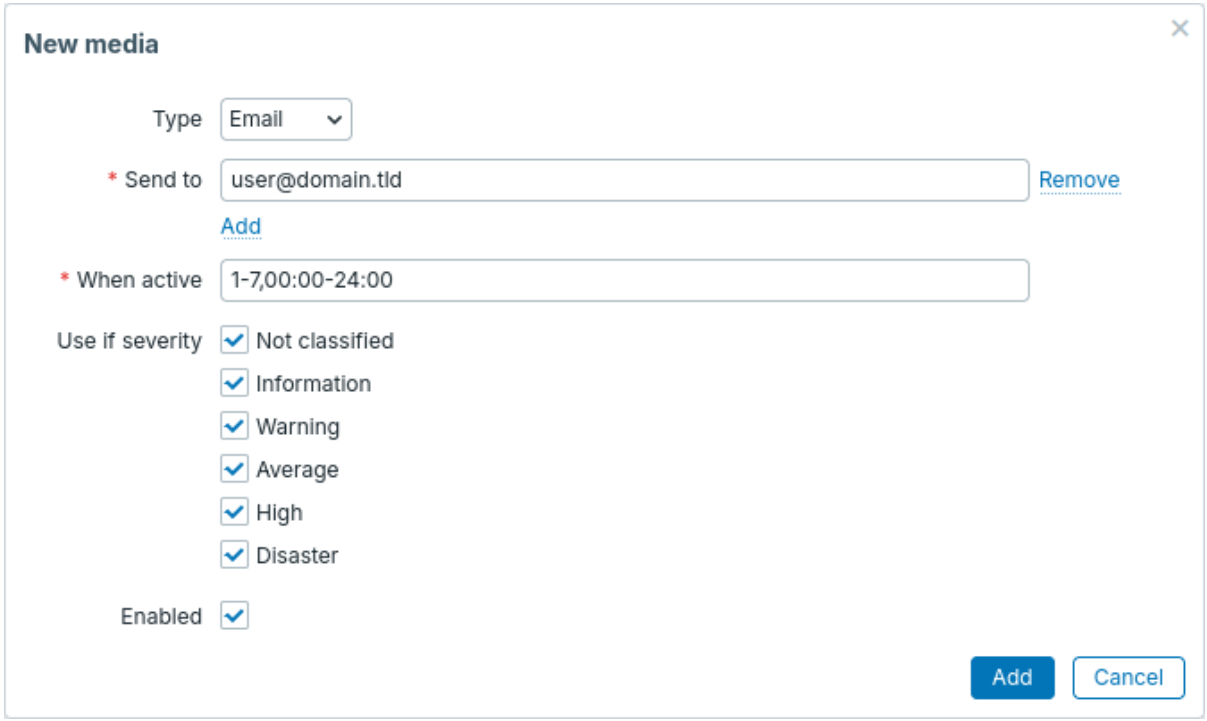
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso. Per i dettagli sui campi di input in questo modulo di configurazione, consulta la pagina User settings.

Per impostazione predefinita, i nuovi utenti non hanno alcun media (metodi di recapito delle notifiche) definito. Per crearne uno, vai alla scheda *Media* e fai clic su *Add*.



Nel pop-up, inserisci l'indirizzo email dell'utente.
 Tutti i tipi di media sono disabilitati per impostazione predefinita e possono essere abilitati in *Alerts* > *Media types*.

Puoi specificare un periodo di tempo in cui il medium sarà attivo (vedi la pagina Time period specification per una descrizione del formato). Per impostazione predefinita, un medium è sempre attivo. Puoi anche personalizzare i livelli di trigger severity per i quali il medium sarà attivo, ma per ora lasciali tutti abilitati.



Fai clic su *Add* per salvare il medium, quindi vai alla scheda *Permissions*.

La scheda Permissions ha un campo obbligatorio *Role*. Il ruolo determina quali elementi del frontend l'utente può visualizzare e quali azioni è autorizzato a eseguire. Fai clic su *Select* e seleziona uno dei ruoli dall'elenco. Ad esempio, seleziona *Admin role* per consentire l'accesso a tutte le sezioni del frontend di Zabbix, eccetto *Administration*. In seguito, potrai modificare i permessi o creare altri ruoli utente. Dopo aver selezionato un ruolo, i permessi appariranno nella stessa scheda:

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

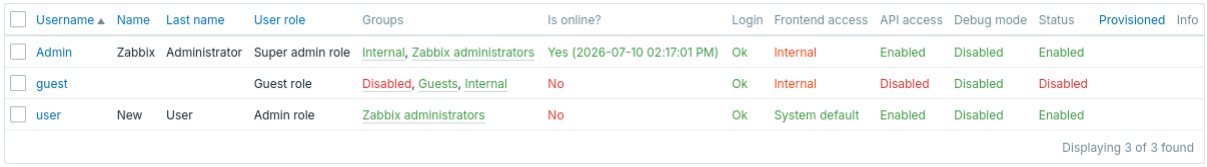
Enabled | Manage own devices

Access to actions

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

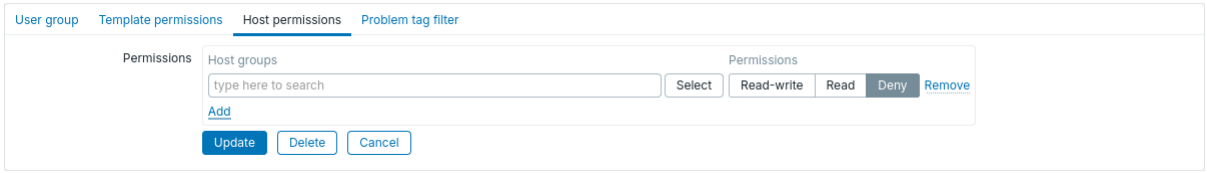
Add | Cancel

Fai clic su *Add* nel modulo delle proprietà dell'utente per salvare l'utente. Il nuovo utente appare nell'elenco utenti.

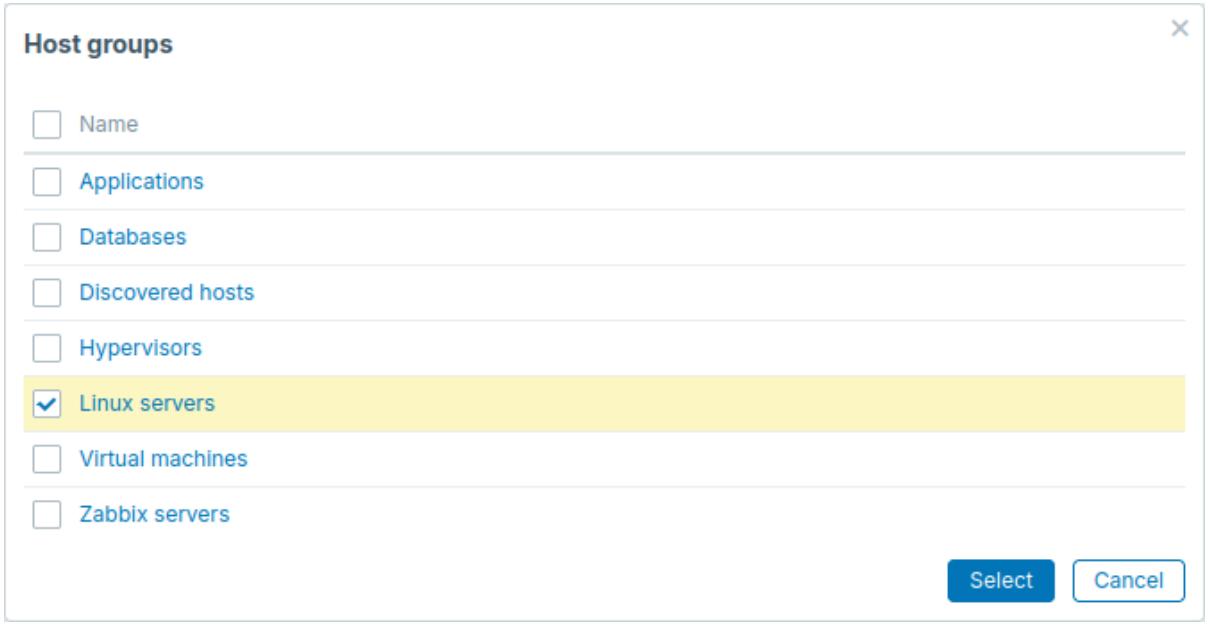


Aggiunta dei permessi

Per impostazione predefinita, un nuovo utente non ha permessi per accedere a host e template. Per concedere i diritti all'utente, fai clic sul gruppo dell'utente nella colonna *Groups* (in questo caso - *Zabbix administrators*). Nel modulo delle proprietà *User groups*, vai alla scheda *Host permissions* per assegnare i permessi ai gruppi di host. Fai clic su Add per visualizzare il campo di selezione del gruppo di host:



Quindi fai clic su *Select* accanto al campo per vedere l'elenco dei gruppi di host. A questo utente deve essere concesso l'accesso in sola lettura al gruppo *Linux servers*, quindi seleziona la casella di controllo appropriata nell'elenco e fai clic su *Select* per confermare la scelta.



Fai clic sul pulsante *Read* per impostare il livello di autorizzazione e poi su *Update* per salvare le modifiche apportate alla configurazione del gruppo di utenti.

Per concedere i permessi ai template, dovrai passare alla scheda *Template permissions* e specificare i gruppi di template. I passaggi sono identici all'assegnazione dei permessi ai gruppi di host. Una panoramica dei template è disponibile nella sezione New template di questa Quickstart.

> **Attention:**
> In Zabbix, i diritti di accesso a host e template vengono assegnati ai user groups, non ai singoli utenti.

Fatto! Puoi provare ad accedere usando le credenziali del nuovo utente.

Nuovo host

Panoramica

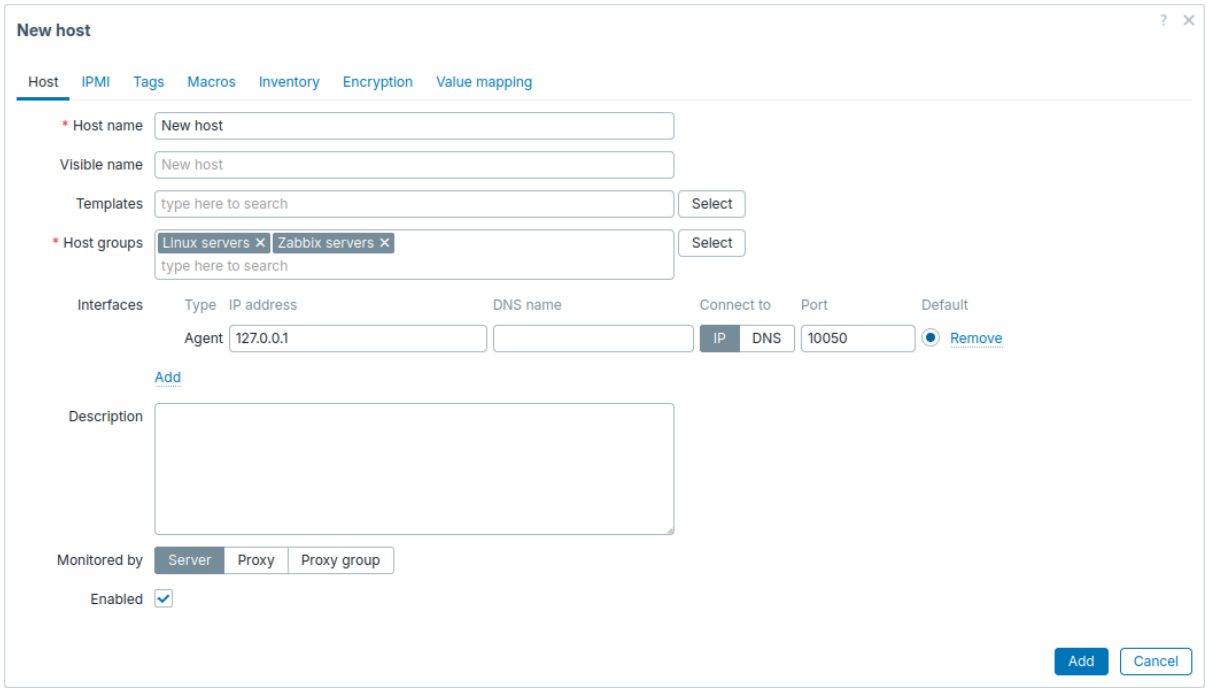
In questa sezione imparerai come configurare un nuovo host.

Un host in Zabbix è un'entità in rete (fisica, virtuale) che si desidera monitorare. La definizione di ciò che può essere un "host" in Zabbix è piuttosto flessibile. Può essere un server fisico, uno switch di rete, una macchina virtuale o un'applicazione.

Aggiunta di un host

Le informazioni sugli host configurati in Zabbix sono disponibili nella sezione di menu *Raccolta dati* > *Host*, nonché nella sezione di menu *Monitoraggio* > *Host*. È già presente un host predefinito denominato "Zabbix server", ma vogliamo imparare ad aggiungerne un altro.

Per aggiungere un nuovo host, fare clic su *Crea host*. Viene aperto un modulo di configurazione dell'host.



Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

Il minimo indispensabile da inserire è:

Nome host

- Inserire un nome host. Sono consentiti caratteri alfanumerici, spazi, punti, trattini e caratteri di sottolineatura.

Gruppi di host

- Selezionare uno o più gruppi esistenti facendo clic sul pulsante *Seleziona* oppure inserire il nome di un gruppo inesistente per crearne uno nuovo.

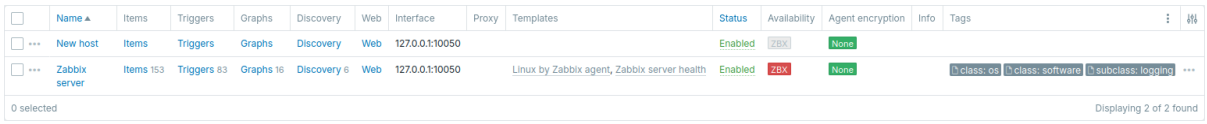
Note:
Tutte le autorizzazioni di accesso vengono assegnate ai gruppi di host, non ai singoli host. Per questo motivo, un host deve appartenere ad almeno un gruppo.

Interfacce: indirizzo IP

- Sebbene tecnicamente non sia un campo obbligatorio, un'interfaccia host è necessaria per raccogliere determinate metriche. Per utilizzare i controlli passivi tramite Zabbix agent, specificare in questo campo l'indirizzo IP o il DNS dell'agent. Tenere presente che è necessario specificare anche l'indirizzo IP o il DNS del server Zabbix nella direttiva 'Server' del file di configurazione dell'agent Zabbix. Se Zabbix agent e Zabbix server sono installati sulla stessa macchina, è necessario specificare lo stesso IP/DNS in entrambi i punti.

Le altre opzioni possono essere lasciate invariate, utilizzando i valori predefiniti.

Al termine, fare clic su *Aggiungi*. Il nuovo host dovrebbe essere visibile nell'elenco degli host.



La colonna *Disponibilità* contiene gli indicatori della disponibilità dell'host per ciascuna interfaccia. Quando è definita un'interfaccia Zabbix agent, è possibile utilizzare l'icona della disponibilità dell'agent (contrassegnata con *ZBX*) per determinare la disponibilità dell'host:

- ZBX - lo stato dell'host non è stato stabilito; non è ancora stato eseguito alcun controllo della metrica.
- ZBX - l'host è disponibile, il controllo della metrica è riuscito.

- ZBX - l'host non è disponibile, il controllo della metrica non è riuscito (spostare il cursore del mouse sull'icona per visualizzare il messaggio di errore). Potrebbe essersi verificato un errore di comunicazione, probabilmente causato da credenziali dell'interfaccia non corrette. Verificare che Zabbix server sia in esecuzione e provare ad aggiornare la pagina anche in un secondo momento.

Nuovo item

Panoramica

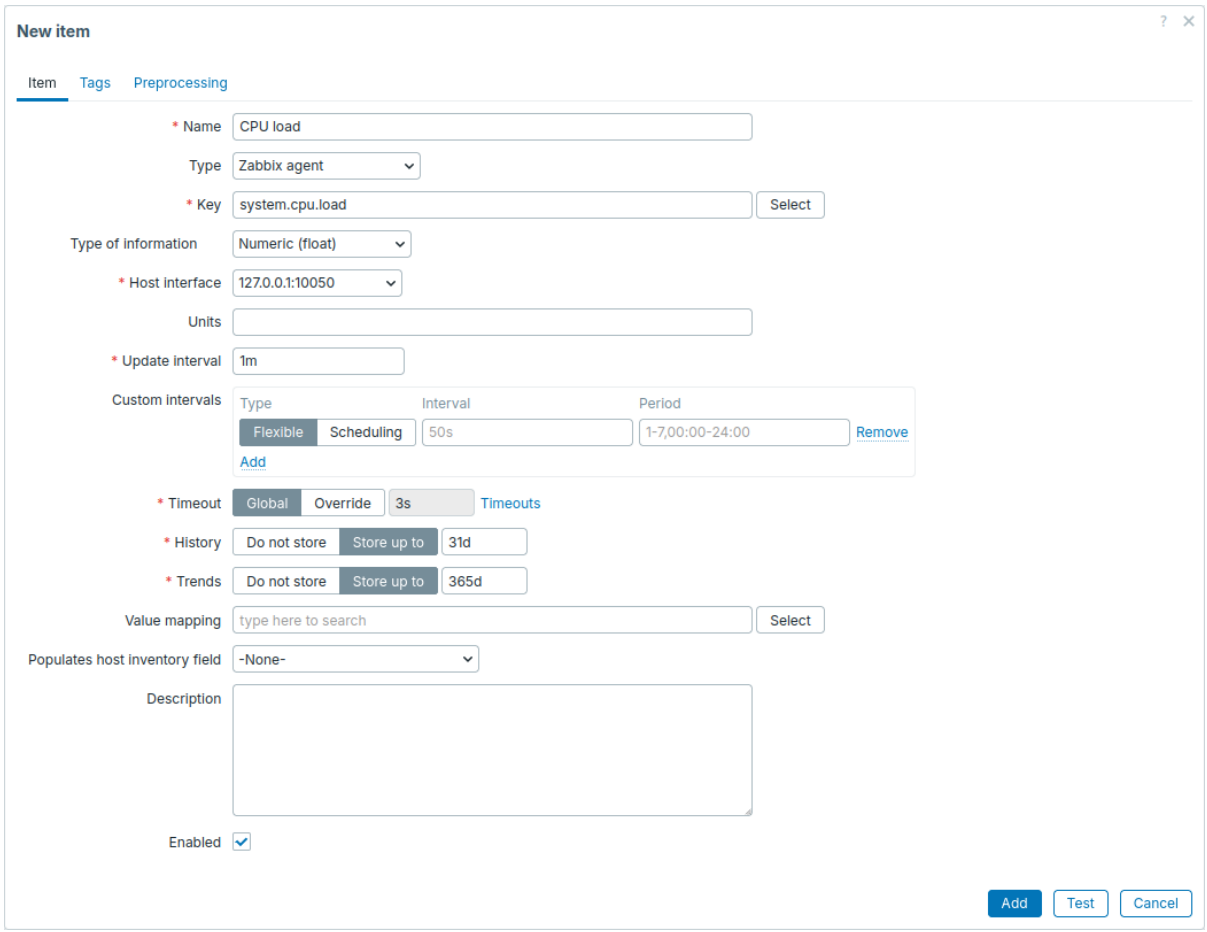
In questa sezione, imparerai come configurare un item.

Gli item sono la base per la raccolta dei dati in Zabbix. Senza gli item, non ci sono dati, perché solo un item definisce una singola metrica o il tipo di dati da raccogliere da un host.

Aggiunta di un item

Tutti gli item sono raggruppati in base agli host. Per questo motivo, per configurare un item di esempio andiamo in *Data collection* > *Hosts* e troviamo il "New host" che abbiamo creato.

Fai clic sul link *Items* nella riga di "New host", quindi fai clic su *Create item*. Si apre un modulo di configurazione dell'item.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Per il nostro item di esempio, le informazioni essenziali da inserire sono:

Name

- Inserisci *CPU load* come valore. Questo sarà il nome dell'item visualizzato negli elenchi e altrove.

Key

- Inserisci manualmente *system.cpu.load* come valore. Questo è il nome tecnico di un item che identifica il tipo di informazione che verrà raccolta. La chiave specifica è solo una delle chiavi predefinite fornite con Zabbix agent.

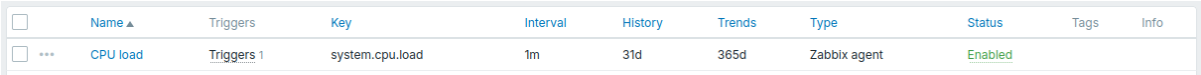
Type of information

- Questo attributo definisce il formato dei dati attesi. Per la chiave *system.cpu.load*, questo campo verrà impostato automaticamente su *Numeric (float)*.

Note:
Potresti anche voler ridurre il numero di giorni per cui verrà conservata la cronologia dell'item, portandolo a 7 o 14. Questa è una buona pratica per alleggerire il database dal mantenimento di molti valori storici.

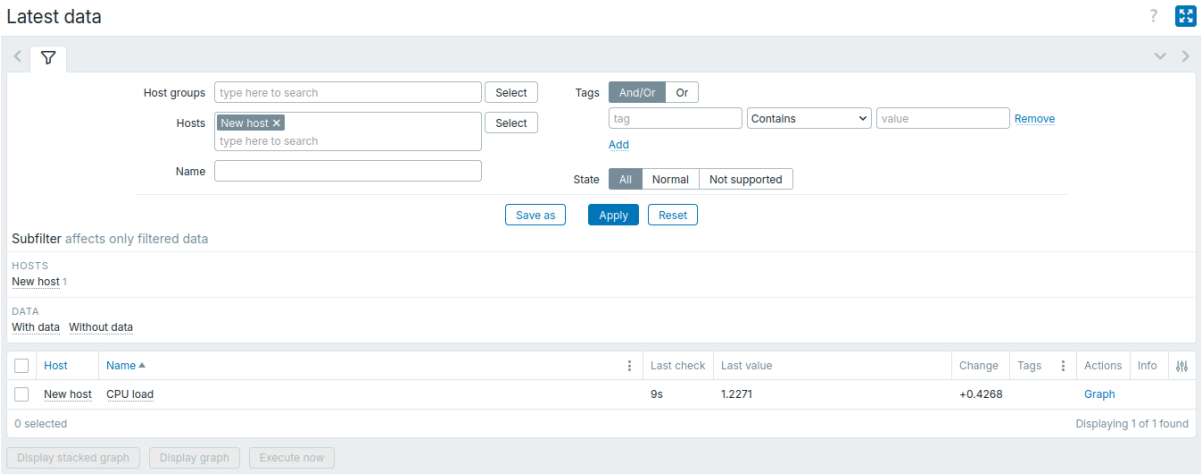
Altre opzioni possono essere lasciate invariate, usando i valori predefiniti.

Al termine, fai clic su *Add*. Il nuovo item dovrebbe apparire nell'elenco degli item.



Visualizzazione dei dati

Una volta definito un item, potresti chiederti se stia effettivamente raccogliendo dati. Per verificarlo, vai su *Monitoring* > *Latest data*, seleziona 'New host' nel filtro e fai clic su *Apply*.



Detto questo, potrebbero essere necessari fino a 60 secondi prima che arrivino i primi dati. Per impostazione predefinita, infatti, è questo l'intervallo con cui il server legge le modifiche di configurazione e rileva i nuovi item da eseguire.

Se nella colonna 'Change' non vedi alcun valore, è possibile che sia stato ricevuto finora un solo valore. Attendi 30 secondi affinché ne arrivi un altro.

Se non vedi informazioni sull'item come nello screenshot, assicurati che:

- i campi 'Key' e 'Type of information' dell'item siano compilati esattamente come nello screenshot;
- sia l'agent sia il server siano in esecuzione;
- lo stato dell'host sia 'Monitored' e l'icona di disponibilità sia verde;
- l'host selezionato nel filtro host sia corretto;
- l'item sia abilitato.

Grafici

Dopo che l'item è rimasto in funzione per un po' di tempo, potrebbe essere il momento di visualizzare qualcosa. I grafici semplici sono disponibili per qualsiasi item numerico monitorato senza alcuna configurazione aggiuntiva. Questi grafici vengono generati durante l'esecuzione.

Per visualizzare il grafico, vai a *Monitoraggio* > *Ultimi dati* e fai clic sul collegamento *Grafico* accanto all'item.



Nuovo trigger

Panoramica

In questa sezione imparerai come configurare un trigger.

Gli item raccolgono solo dati. Per valutare automaticamente i dati in arrivo, dobbiamo definire dei trigger. Un trigger contiene un'espressione che definisce una soglia di ciò che è un livello accettabile per i dati.

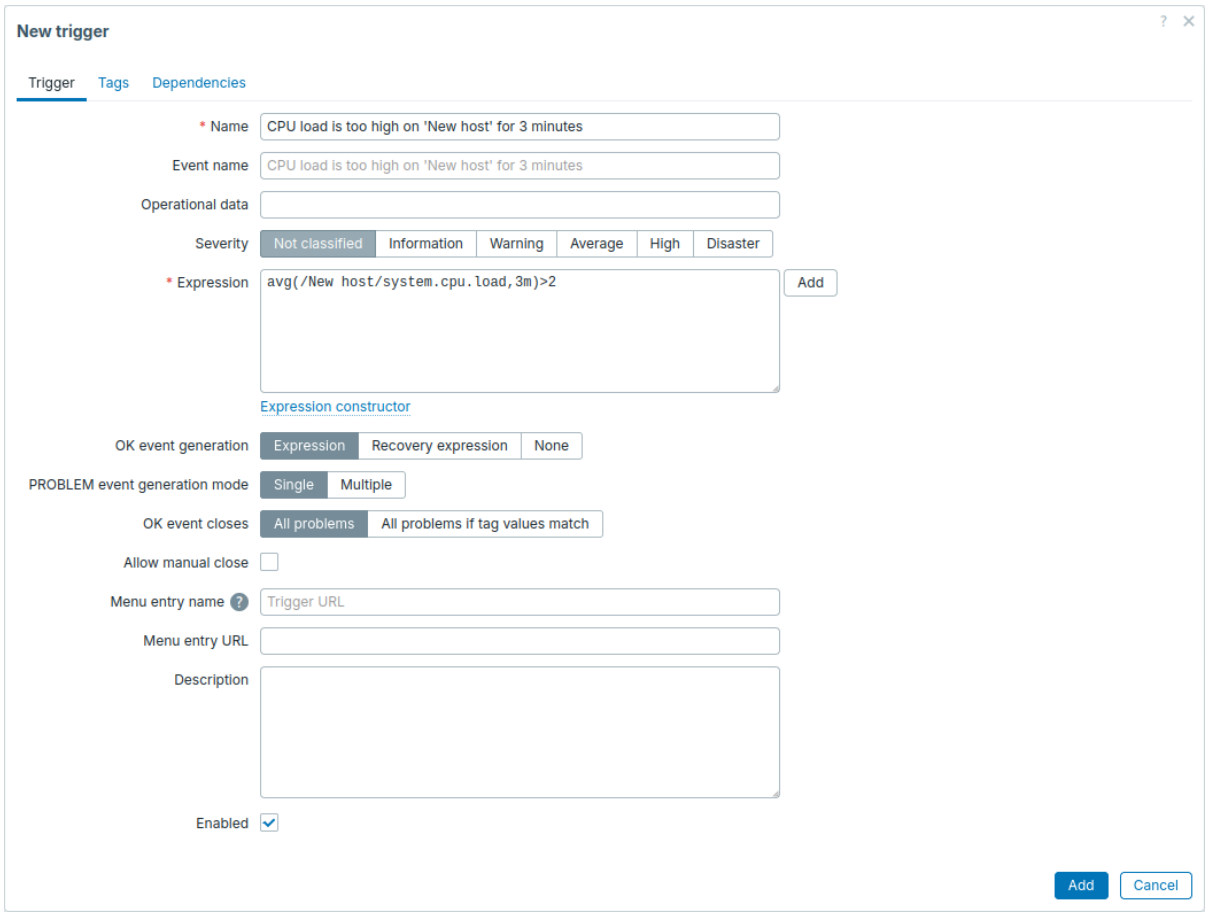
Se quel livello viene superato dai dati in arrivo, un trigger si "attiva" o passa allo stato 'Problem' - indicando che è accaduto qualcosa che potrebbe richiedere attenzione. Se il livello torna di nuovo accettabile, il trigger ritorna allo stato 'Ok'.

Aggiunta di un trigger

Per configurare un trigger per il nostro item:

1. Vai su *Data collection* > *Hosts*.
2. Fai clic sul link *Triggers* nella riga di 'New host'.
3. Fai clic su *Create trigger*.

Si apre un modulo di configurazione del trigger.



Per il nostro trigger, le informazioni essenziali da inserire qui sono:

Name

- Inserisci *CPU load is too high on 'New host' for 3 minutes* come valore. Questo sarà il nome del trigger visualizzato negli elenchi e altrove.

Expression

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

Questa è l'espressione del trigger. Assicurati che l'espressione sia inserita correttamente, fino all'ultimo simbolo. La chiave dell'item qui (`system.cpu.load`) viene usata per fare riferimento all'item. Questa particolare espressione, in sostanza, indica che la soglia di problema viene superata quando il valore medio del carico della CPU per 3 minuti è superiore a 2. Puoi saperne di più sulla sintassi delle espressioni di trigger.

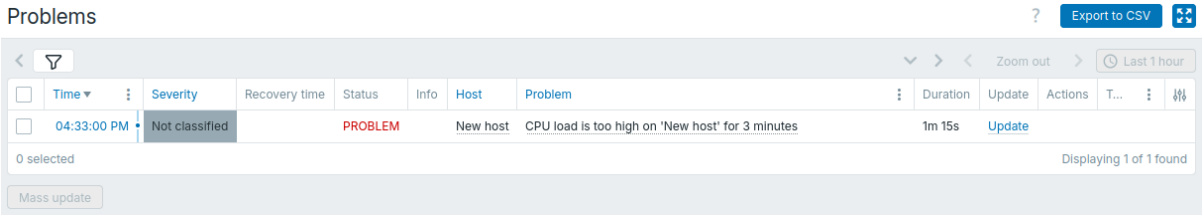
Altre opzioni possono essere lasciate invariate, usando i valori predefiniti.

Al termine, fai clic su *Add*. Il nuovo trigger dovrebbe apparire nell'elenco dei trigger.

Visualizzazione dello stato del trigger

Con un trigger definito, potresti voler vedere il suo stato.

Se il carico della CPU ha superato il livello di soglia definito nel trigger, il problema verrà visualizzato in *Monitoring* > *Problems*.



Il lampeggiamento nella colonna dello stato indica una modifica recente dello stato del trigger, avvenuta negli ultimi 30 minuti.

Ricezione della notifica di problema

Panoramica

In questa sezione imparerai come configurare gli avvisi sotto forma di notifiche in Zabbix.

Con gli item che raccolgono dati e i trigger progettati per ”attivarsi” in caso di situazioni problematiche, sarebbe utile avere anche un meccanismo di notifica per avvisare di eventi importanti anche quando il frontend di Zabbix non viene controllato attivamente.

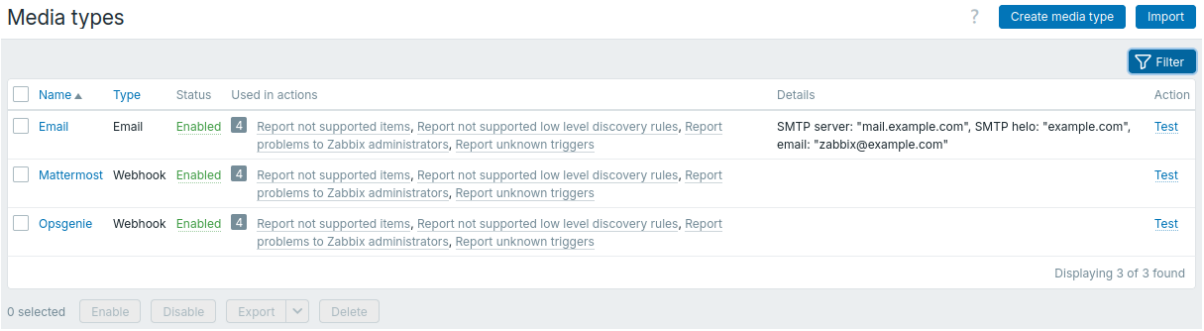
Questo è ciò che fanno le notifiche.

Poiché l’email è il metodo di recapito più diffuso per le notifiche di problema, impareremo come configurare una notifica email.

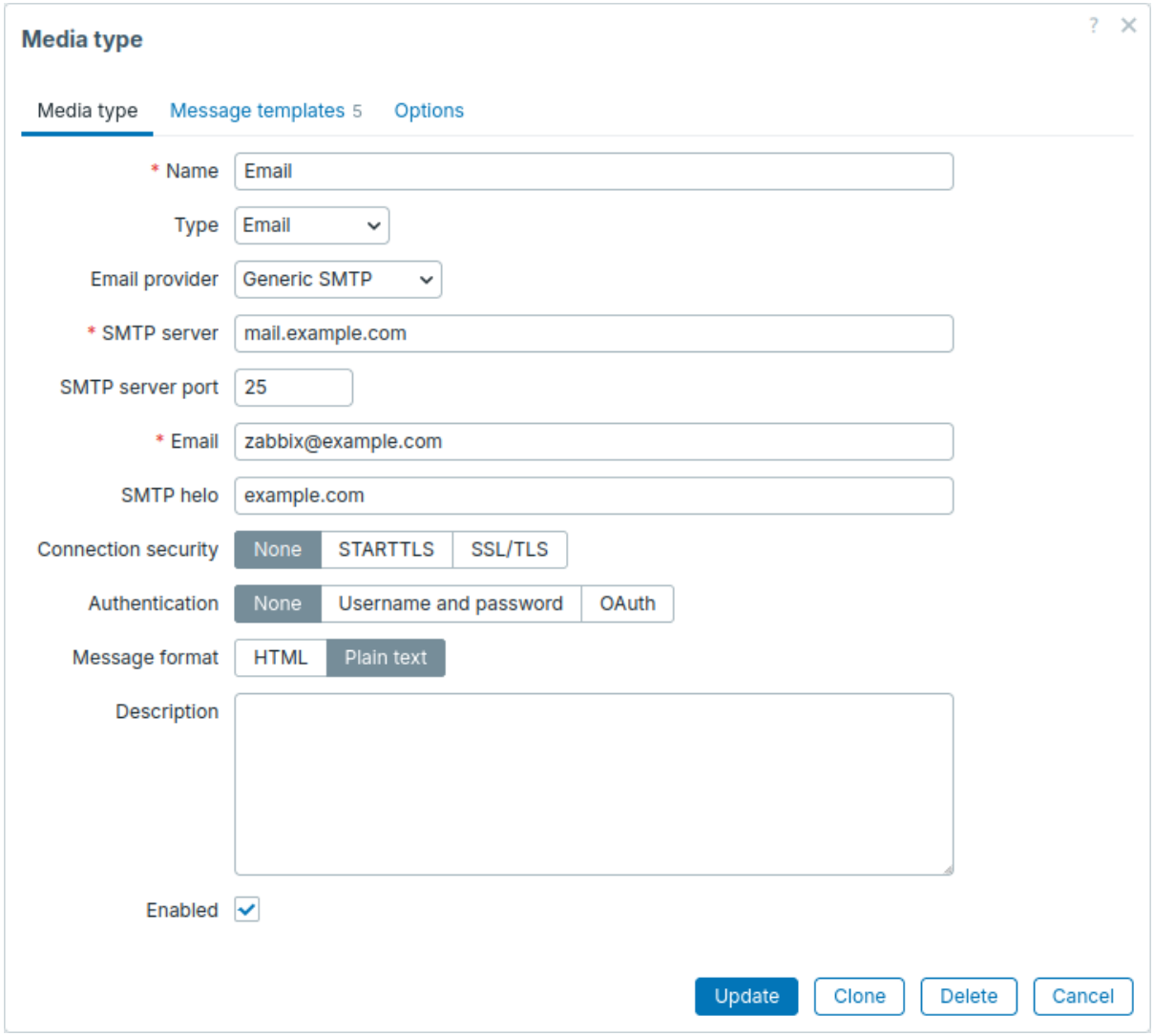
Impostazioni email

Inizialmente in Zabbix sono presenti diversi metodi di consegna di notifica predefiniti. Email è uno di questi.

Per configurare le impostazioni email, vai su *Alerts* > *Media types* e fai clic su *Email* nell’elenco dei media types predefiniti.



Si apre un modulo di configurazione delle impostazioni email.



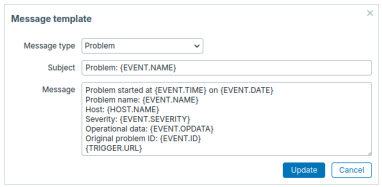
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Nella scheda *Media type*, imposta i valori di *SMTP server*, *SMTP helo* e *Email* in base all'ambiente in uso.

Note:
Il valore del campo *Email* verrà utilizzato come indirizzo 'From' per le notifiche inviate da Zabbix.

Successivamente, è necessario definire il contenuto del messaggio di problema. Il contenuto viene definito tramite un message template, configurato nella scheda *Message templates*.

Fai clic su *Add* per creare un message template e seleziona *Problem* come tipo di messaggio.

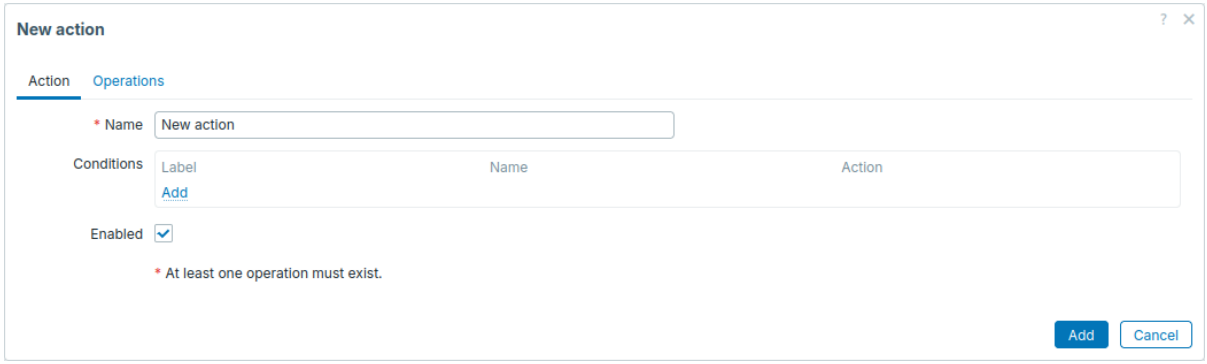


Fai clic su *Add* quando sei pronto e salva il modulo.

Ora hai configurato *Email* come media type funzionante. Il media type deve anche essere associato agli utenti definendo indirizzi di consegna specifici (come abbiamo fatto quando abbiamo configurato un nuovo utente); in caso contrario non verrà utilizzato.

Nuova action

L'invio delle notifiche è una delle funzioni svolte dalle actions in Zabbix. Pertanto, per configurare una notifica, vai su *Alerts* > *Actions* > *Trigger actions* e fai clic su *Create action*.

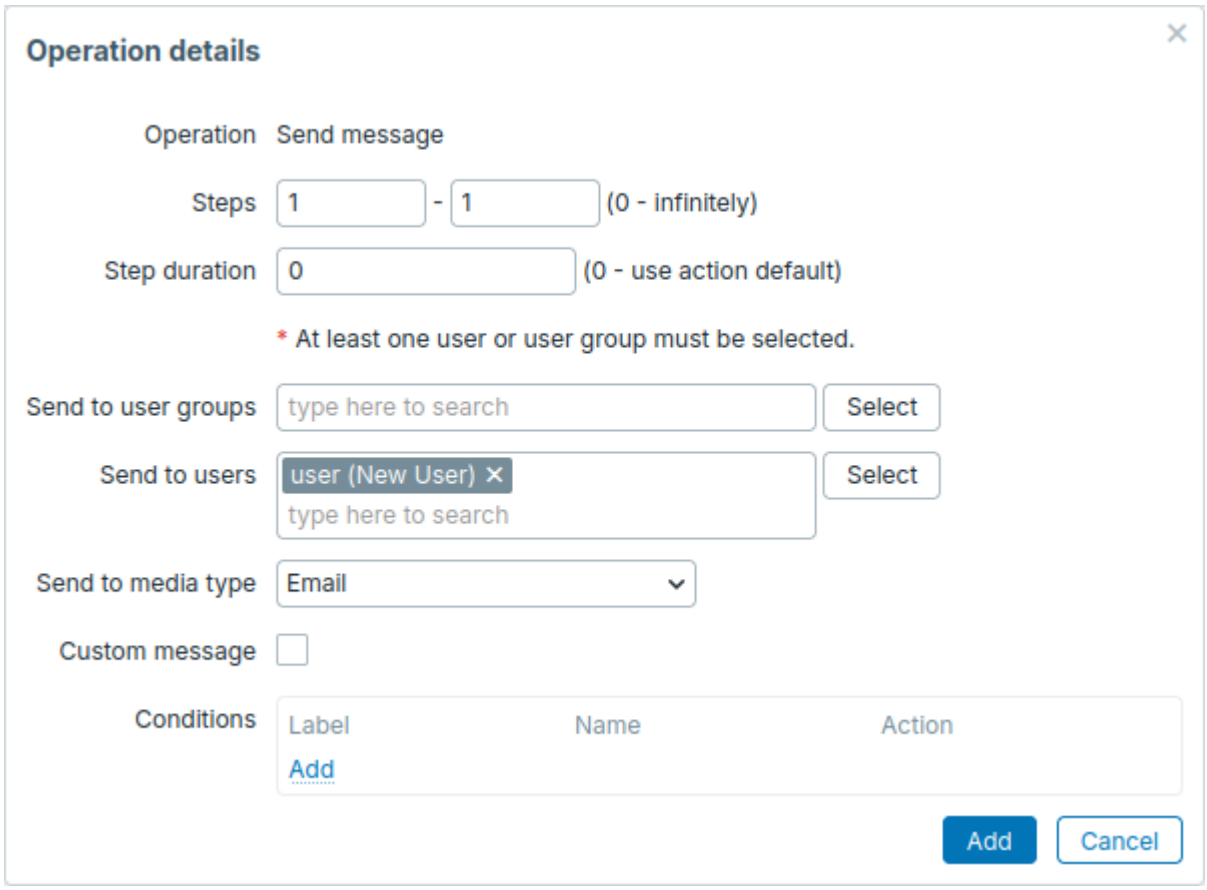


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

In questo modulo, inserisci un nome per l'action.

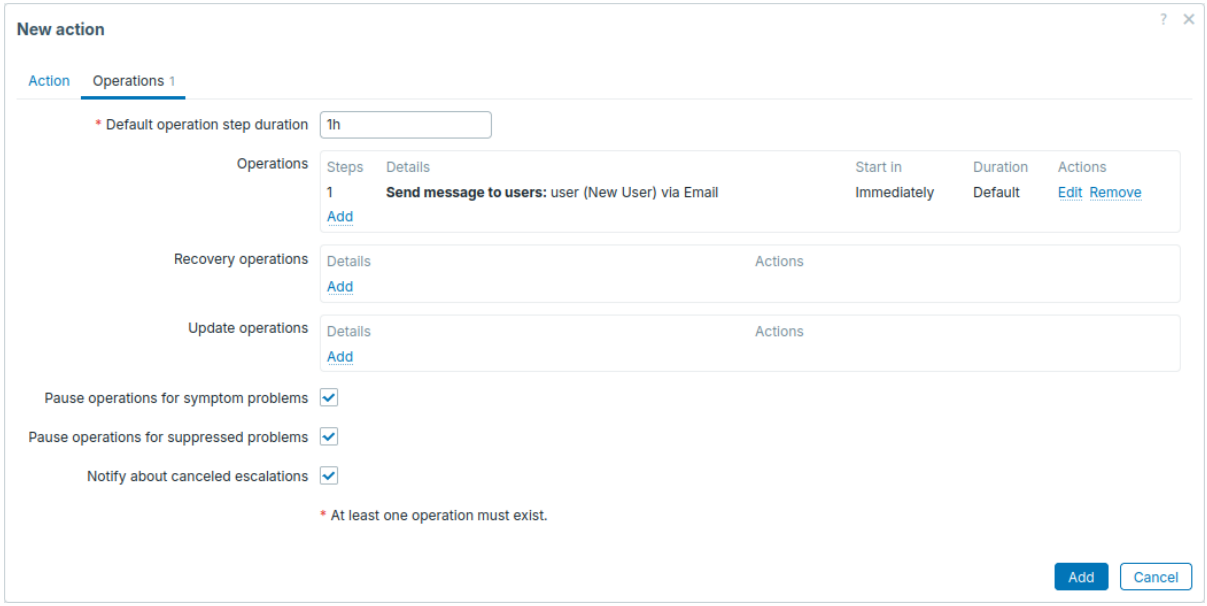
Nel caso più semplice, se non aggiungiamo altre conditions specifiche, l'action verrà eseguita a ogni cambiamento del trigger da 'Ok' a 'Problem'.

Dobbiamo ancora definire cosa deve fare l'action, e questo si configura nella scheda *Operations*. Fai clic su *Add* nel blocco *Operations*, che apre un nuovo modulo di operazione.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Qui, fai clic su *Select* nel blocco *Send to Users* e seleziona l'utente ('user') che abbiamo definito. Seleziona "Email" come valore di *Send to media type*. Una volta terminato, fai clic su *Add* e l'operazione dovrebbe essere aggiunta:



Questo è tutto per una configurazione semplice di un'action, quindi fai clic su *Add* nel modulo dell'action.

Ricezione della notifica

Ora che la consegna delle notifiche è configurata, sarebbe utile riceverne effettivamente una. Per aiutarci in questo, possiamo aumentare intenzionalmente il carico sul nostro host, in modo che il nostro trigger "scatti" e riceviamo una notifica di problema.

Apri la console sul tuo host ed esegui:

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

Puoi eseguire uno o più di questi processi.

Ora vai in *Monitoring* > *Latest data* e osserva come i valori di 'CPU Load' sono aumentati. Ricorda che, affinché il nostro trigger "scatti", il valore di 'CPU Load' deve superare '2' per 3 minuti consecutivi. Una volta che ciò accade:

- in *Monitoring* > *Problems* dovresti vedere il trigger con uno stato `Problem` lampeggiante;
- dovresti ricevere una notifica di problema nella tua email.

Attention:
Se le notifiche non funzionano:

- verifica ancora una volta che sia le impostazioni email sia l'action siano state configurate correttamente
- assicurati che l'utente che hai creato abbia almeno i permessi di lettura sull'host che ha generato l'evento, come indicato nel passaggio *Aggiunta utente*. L'utente, essendo parte del gruppo di utenti 'Zabbix administrators', deve avere almeno accesso in lettura al gruppo di host 'Linux servers' a cui appartiene il nostro host.
- Inoltre, puoi controllare il registro delle action andando su *Reports* > *Action log*.

Nuovo template

Panoramica

In questa sezione imparerai a configurare un template.

In precedenza abbiamo imparato a configurare un item, un trigger e a ricevere una notifica di problema per il host.

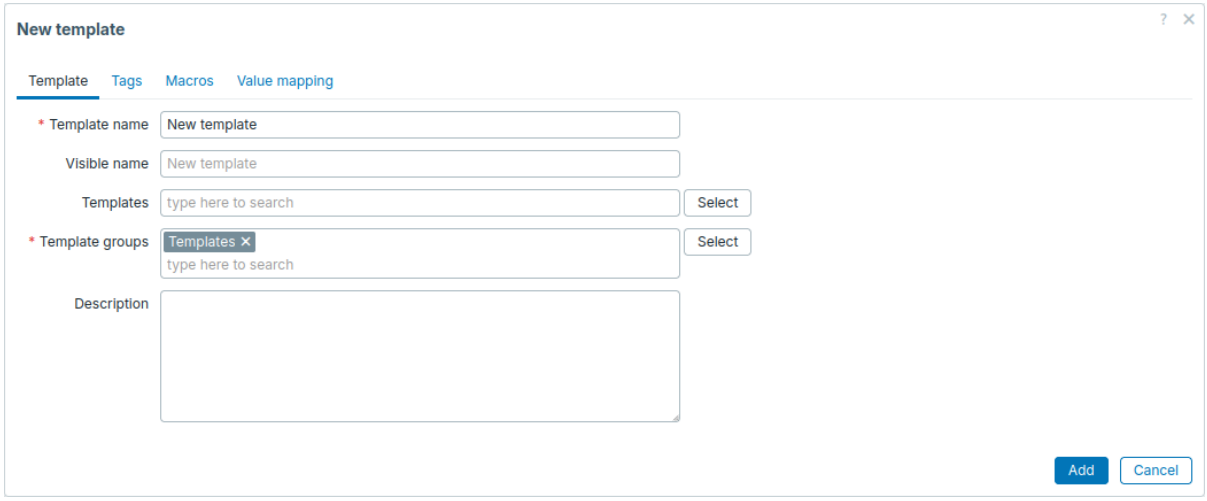
Sebbene tutti questi passaggi offrano molta flessibilità, potrebbe sembrare necessario eseguire numerose operazioni se, ad esempio, dovessero essere applicate a mille host. Sarebbe utile disporre di una certa automazione.

È qui che entrano in gioco i template. I template consentono di raggruppare item, trigger e altre entità utili, in modo da poterli riutilizzare più volte applicandoli agli host in un unico passaggio.

Quando un template viene collegato a un host, l'host eredita tutte le entità del template. In questo modo, è possibile applicare molto rapidamente un insieme di controlli già preparato.

Aggiunta di un template

Per iniziare a lavorare con i template, dobbiamo prima crearne uno. Per farlo, in *Data collection* > *Templates* fare clic su *Create template*. Verrà visualizzato un modulo di configurazione del template.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I parametri richiesti da inserire qui sono:

Template name

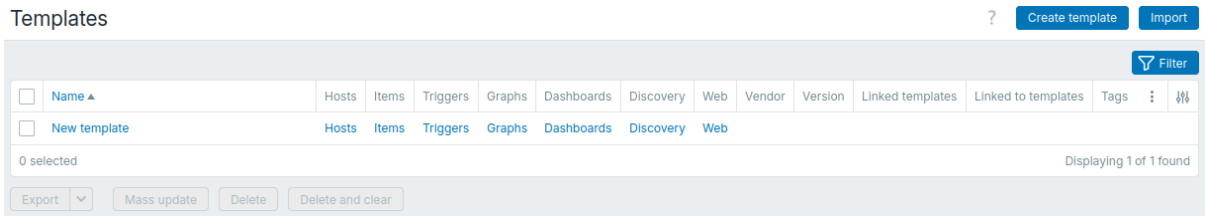
- Inserire un nome per il template. Sono consentiti caratteri alfanumerici, spazi e underscore.

Template groups

- Selezionare uno o più gruppi facendo clic sul pulsante *Select*. Il template deve appartenere a un gruppo.

Note:
I permessi di accesso ai gruppi di template vengono assegnati nella configurazione del user group nella scheda **Template permissions**, nello stesso modo dei permessi degli host. Tutti i permessi di accesso vengono assegnati ai gruppi, non ai singoli template; per questo motivo è obbligatorio includere il template in almeno un gruppo.

Al termine, fare clic su *Add*. Il nuovo template dovrebbe essere visibile nell'elenco dei template. È anche possibile usare il filter per trovare il template.

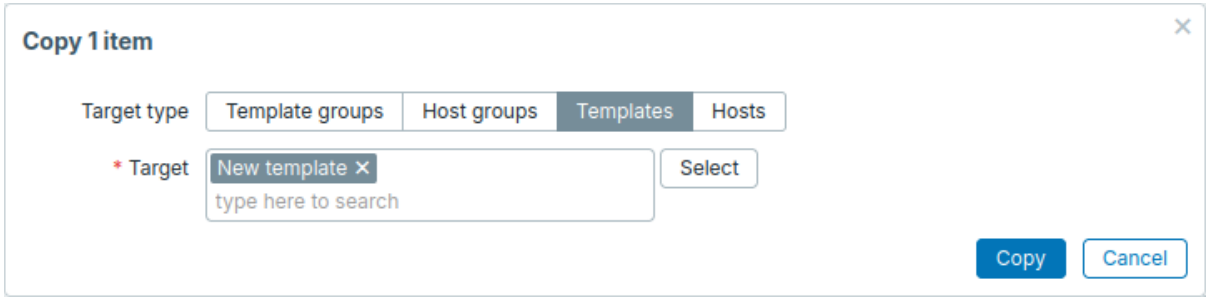


Come si può vedere, il template è presente, ma non contiene nulla: nessun item, trigger o altre entità.

Aggiunta di un item al template

Per aggiungere un item al template:

1. Vai a *Raccolta dati* > *Host*.
2. Individua "New host" nell'elenco e fai clic sul collegamento ipertestuale *Items*.
3. Seleziona la casella di controllo dell'item "CPU Load" nell'elenco.
4. Fai clic su *Copy* sotto l'elenco.
5. Seleziona la scheda *Templates*.
6. Seleziona il template in cui copiare l'item.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso. 7. Fai clic su *Copy*.

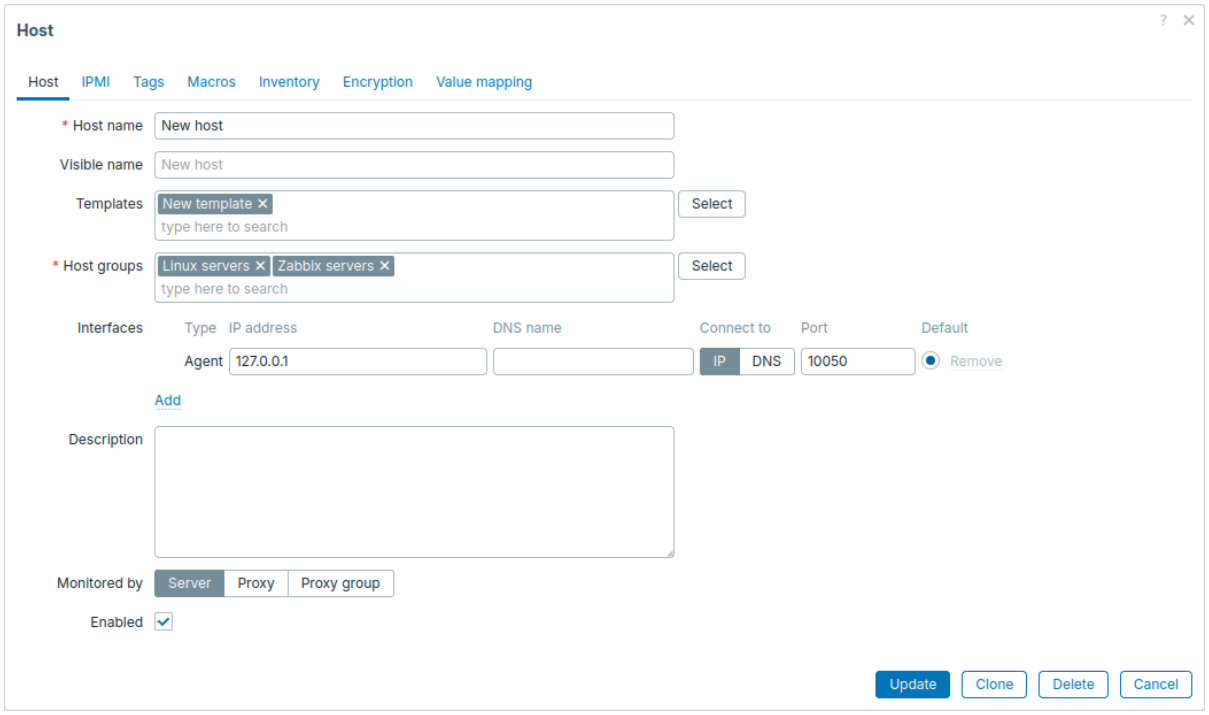
Se ora vai a *Raccolta dati* > *Templates*, "New template" dovrebbe contenere un nuovo item.

Per il momento ci fermeremo a un solo item, ma allo stesso modo puoi aggiungere al template qualsiasi altro item, trigger o altra entità, fino a ottenere un insieme di entità abbastanza completo per lo scopo previsto (monitoraggio del sistema operativo, monitoraggio di una singola applicazione).

Collegamento del template a un host

Con il template pronto, non resta che aggiungerlo a un host. Per farlo:

1. Vai a *Raccolta dati* > *Host*.
2. Fai clic su "Nuovo host" per aprire il relativo modulo di configurazione e individua il campo *Template*.
3. Inizia a digitare *Nuovo template* nel campo *Template*. Il nome del template appena creato dovrebbe apparire nell'elenco a discesa.
4. Scorri verso il basso per selezionarlo. Verifica che venga visualizzato nel campo *Template*.



5. Fai clic su *Aggiorna* nel modulo per salvare le modifiche.

Il template è ora aggiunto all'host, insieme a tutte le entità che contiene.

In questo modo può essere applicato anche a qualsiasi altro host. Eventuali modifiche agli item, ai trigger e alle altre entità a livello di template verranno propagate agli host a cui il template è collegato.

Collegamento di template predefiniti agli host

Come avrai notato, Zabbix include un insieme di template predefiniti per vari sistemi operativi, dispositivi e applicazioni. Per iniziare a monitorare molto rapidamente, puoi collegare a un host quello più adatto tra questi, ma tieni presente che questi template devono essere ottimizzati per il tuo ambiente. Alcuni controlli potrebbero non essere necessari e gli intervalli di polling potrebbero essere troppo frequenti.

Sono disponibili ulteriori informazioni sui template.

Monitoraggio del web server Apache

Per iniziare a monitorare il web server Apache, segui questa procedura:

1. Assicurati di avere accesso alle statistiche di Apache.
2. Applica il template di monitoraggio Apache.
3. Consulta i dati più recenti.

Per utilizzare questa guida devi avere Zabbix server e l'interfaccia web installati. Devi aver effettuato l'accesso a Zabbix come utente predefinito *Admin*.

Accesso alle statistiche di Apache

L'obiettivo di questo passaggio è assicurarsi che le statistiche del web server Apache possano essere lette da Zabbix.

In Apache, le informazioni in tempo reale sulle prestazioni e sull'attività del server sono fornite dal modulo integrato `mod_status`. Quando è abilitato, crea una pagina accessibile via web (di solito in `/server-status`) che mostra i dati relativi a prestazioni e attività.

Per verificare che `mod_status` sia abilitato per il web server Apache, eseguire:

Su Debian/Ubuntu	Su sistemi basati su RHEL
`apache2ctl -M | grep status`	`httpd -M | grep status`

Se nell'output compare "`status_module (shared)`", allora il modulo status è già abilitato. In caso contrario, è necessario abilitare il modulo status.

Applicare il template di monitoraggio Apache

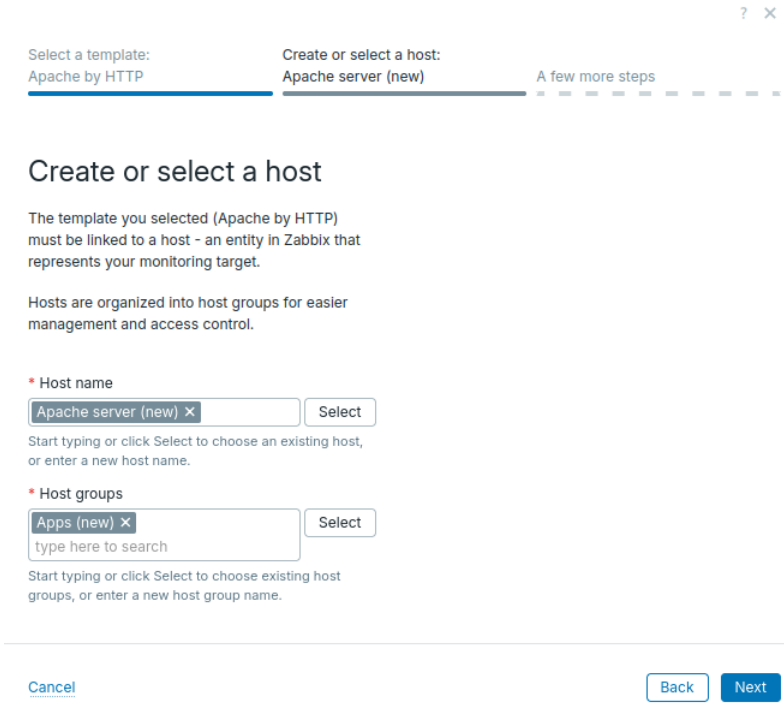
In Zabbix questo template è pronto all'uso. Contiene elementi preconfigurati pronti per la raccolta dei dati, la visualizzazione e l'analisi.

Per applicare il template:

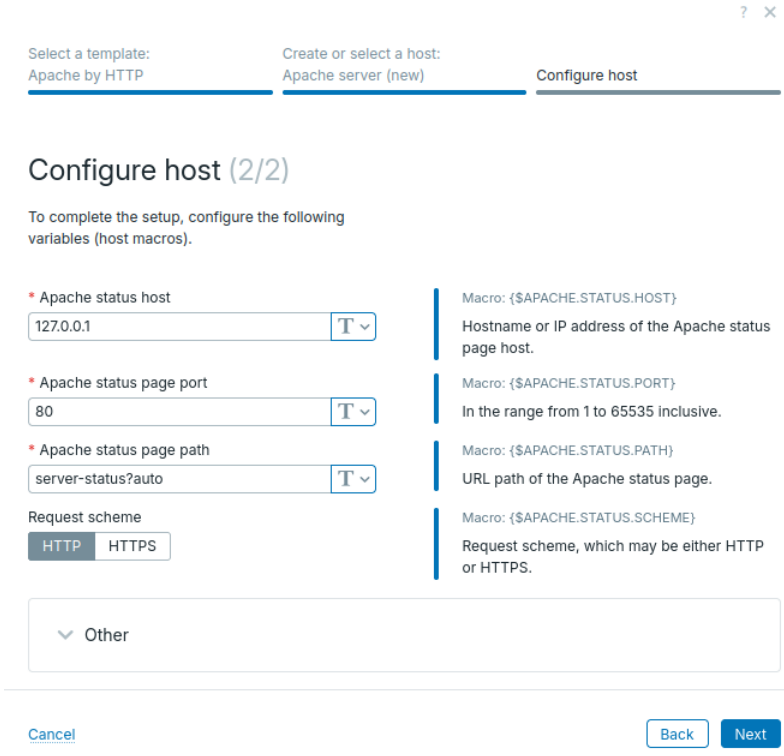
1. Avviare la Host Wizard in *Data collection* > *Hosts*.
2. Selezionare il template *Apache by HTTP*.



3. Creare un host Apache e aggiungervi un gruppo. Questo host virtuale conterrà i dati relativi ad Apache.



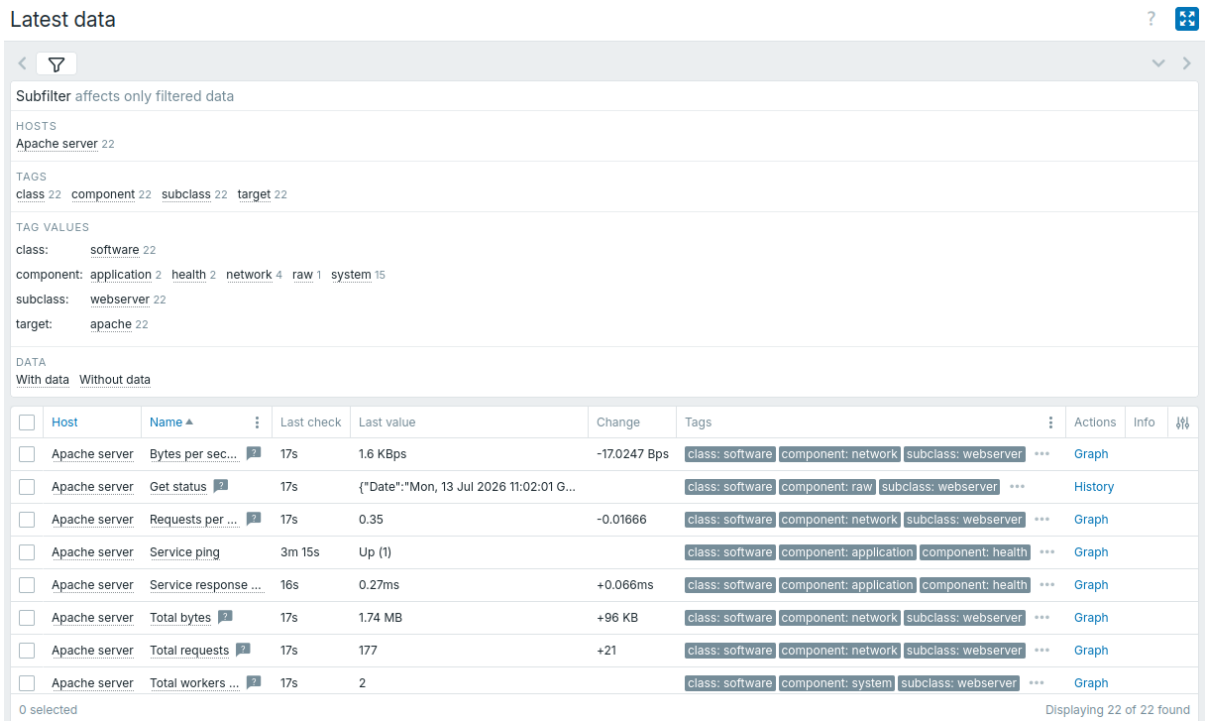
4. Configurare l'host. In particolare, inserire l'indirizzo del web server Apache (127.0.0.1 su una macchina locale, indirizzo IP su una macchina remota).



È possibile lasciare le altre impostazioni ai valori predefiniti.

5. Per completare la Host Wizard, fare clic su *Create* e *Finish*.

Visualizza i dati più recenti



Congratulazioni, il monitoraggio del web server Apache è stato avviato!

Zabbix offre opzioni di visualizzazione e invia avvisi se si verificano problemi. Per configurare gli avvisi via email, vedere Ricezione delle notifiche di problema.

Monitoraggio di un server Linux

Per iniziare a monitorare un server Linux, segui questa procedura:

1. Applica il template di monitoraggio Linux.
2. Installa Zabbix agent sul server.
3. Consulta i dati più recenti.

Per iniziare a monitorare un server Linux non locale, devi conoscere il suo indirizzo IP e l'indirizzo IP del server Zabbix.

Su un server Linux verrà utilizzato un Zabbix agent per monitorarlo. Le istruzioni per installare l'agent saranno fornite durante la procedura.

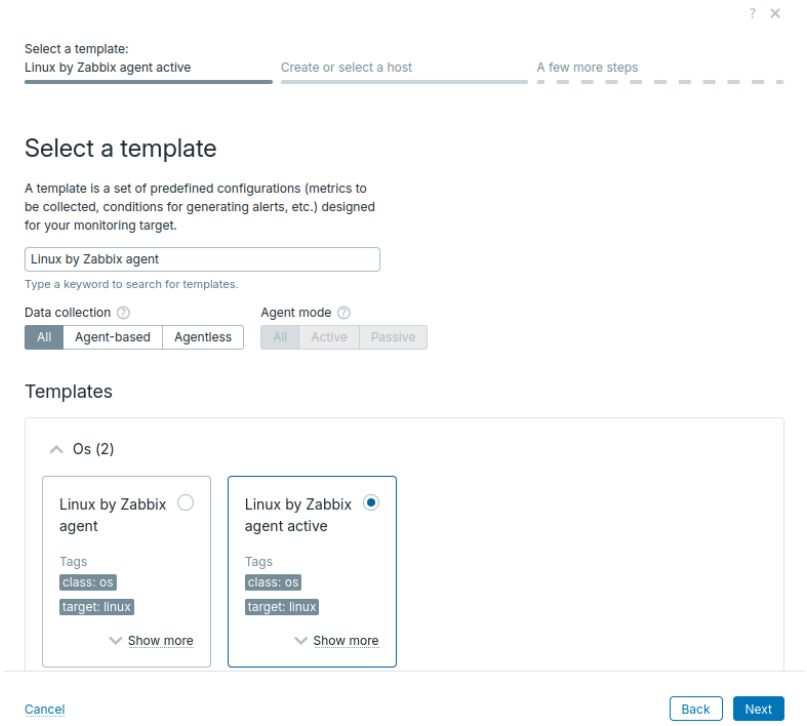
Per usare questa guida devi avere Zabbix server e il frontend installati. Devi aver effettuato l'accesso a Zabbix come utente predefinito *Admin*.

Applicare il template di monitoraggio Linux

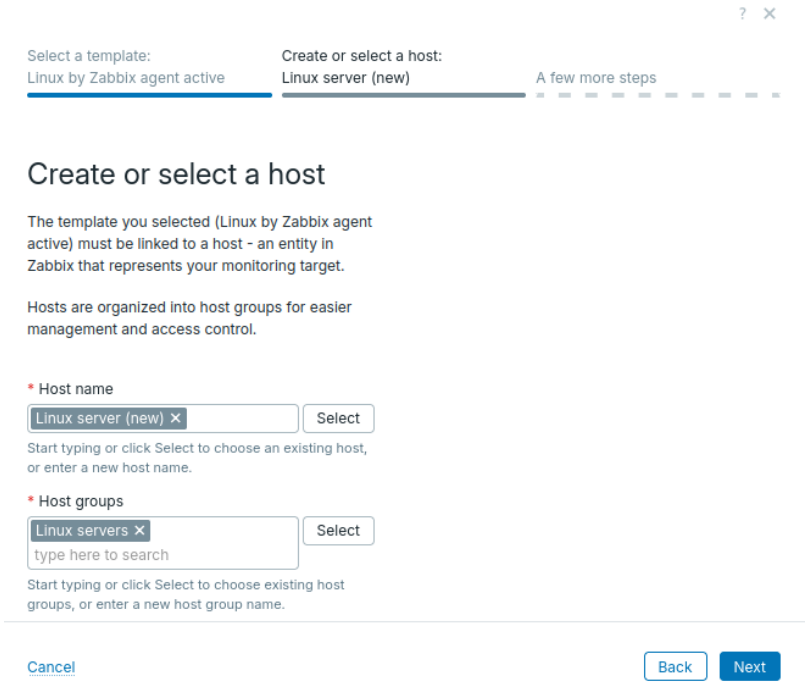
In Zabbix questo template è pronto all'uso. Contiene elementi preconfigurati pronti per la raccolta dati, la visualizzazione e l'analisi.

Per applicare il template:

1. Avvia la Host Wizard in *Data collection* > *Hosts*.
2. Seleziona il template *Linux by Zabbix agent active*.



3. Crea un host per il server Linux e aggiungi un gruppo per esso. Questo host virtuale conterrà tutti i dati relativi al monitoraggio del server Linux.



4. Configura e installa l’agent.

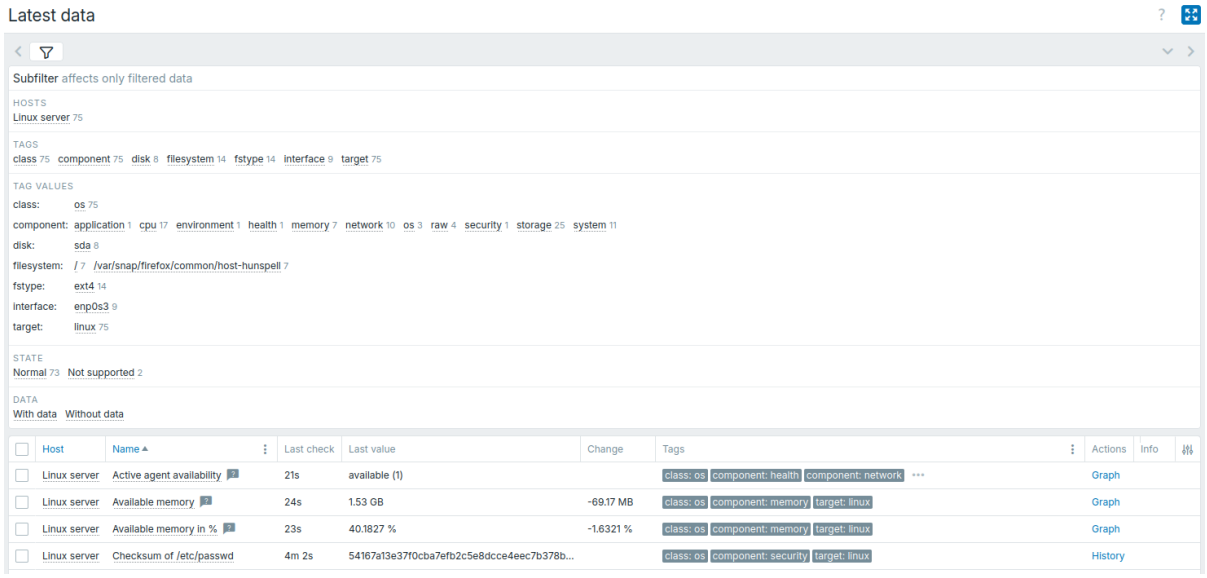


Per installare l'agent, esegui il comando fornito sul server Linux. Installerà l'agent e configurerà la connessione dell'agent a Zabbix server con crittografia.
 Se hai già installato l'agent, puoi saltare questo passaggio (fai clic su *Next*). Tuttavia, ti verrà richiesto di fornire l'identità della chiave precondivisa e la chiave precondivisa per connetterti all'agent.

Puoi lasciare le altre impostazioni ai valori predefiniti.

5. Completa la configurazione dell'host: fai clic su *Create* e *Finish*.

Visualizza i dati più recenti



Congratulazioni, il monitoraggio del server Linux è stato avviato!

Zabbix offre opzioni di visualizzazione e invia avvisi se si verificano problemi. Per configurare gli avvisi via email, consulta Ricezione delle notifiche dei problemi.

Monitoraggio del server MySQL

Per iniziare a monitorare un server MySQL, segui questa procedura:

1. Crea un utente MySQL con privilegi limitati.
2. Applica il template di monitoraggio MySQL.
3. Controlla i dati più recenti.

Per iniziare a monitorare un server MySQL, devi conoscere il suo DSN e l'indirizzo IP del server Zabbix.

Un agent Zabbix verrà utilizzato per monitorare il server MySQL. Le istruzioni per installare l'agent saranno fornite durante la procedura.

Per utilizzare questa guida, devi avere Zabbix server e l'interfaccia web installati. Devi aver effettuato l'accesso a Zabbix come utente predefinito *Admin*.

Creare un utente MySQL con autorizzazioni limitate

Si consiglia di creare un utente MySQL con autorizzazioni limitate esclusivamente a quelle necessarie per il monitoraggio di MySQL. Sostituire <password> con una password complessa e univoca.

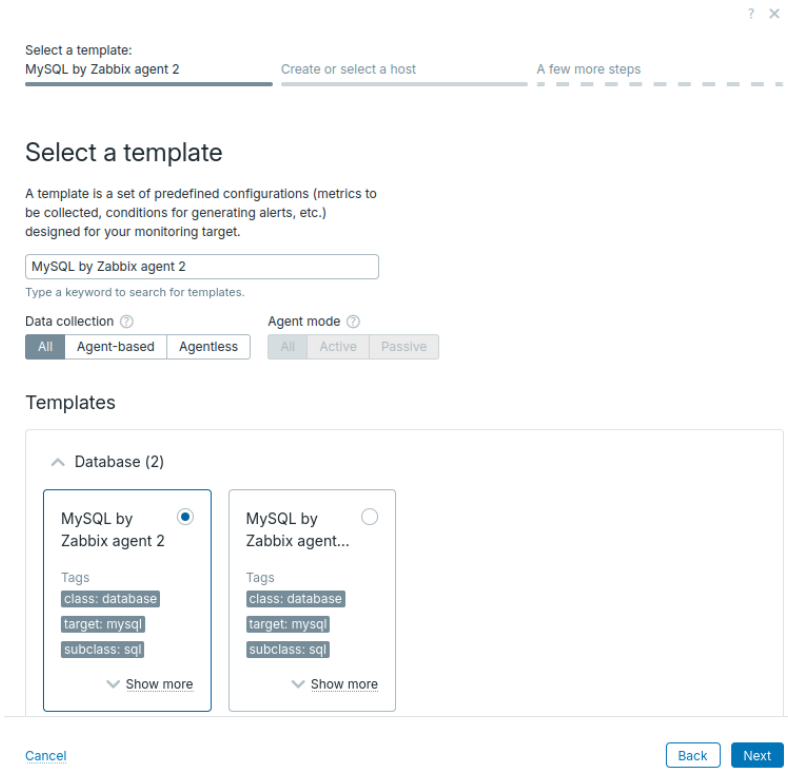
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

Applicare il template di monitoraggio MySQL

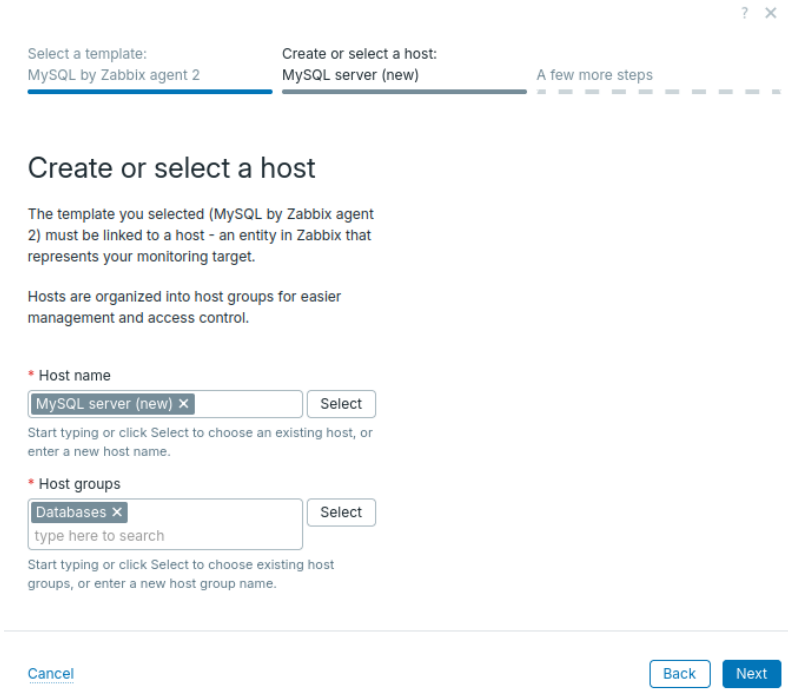
In Zabbix questo template è pronto all'uso. Contiene elementi preconfigurati pronti per la raccolta dei dati, la visualizzazione e l'analisi.

Per applicare il template:

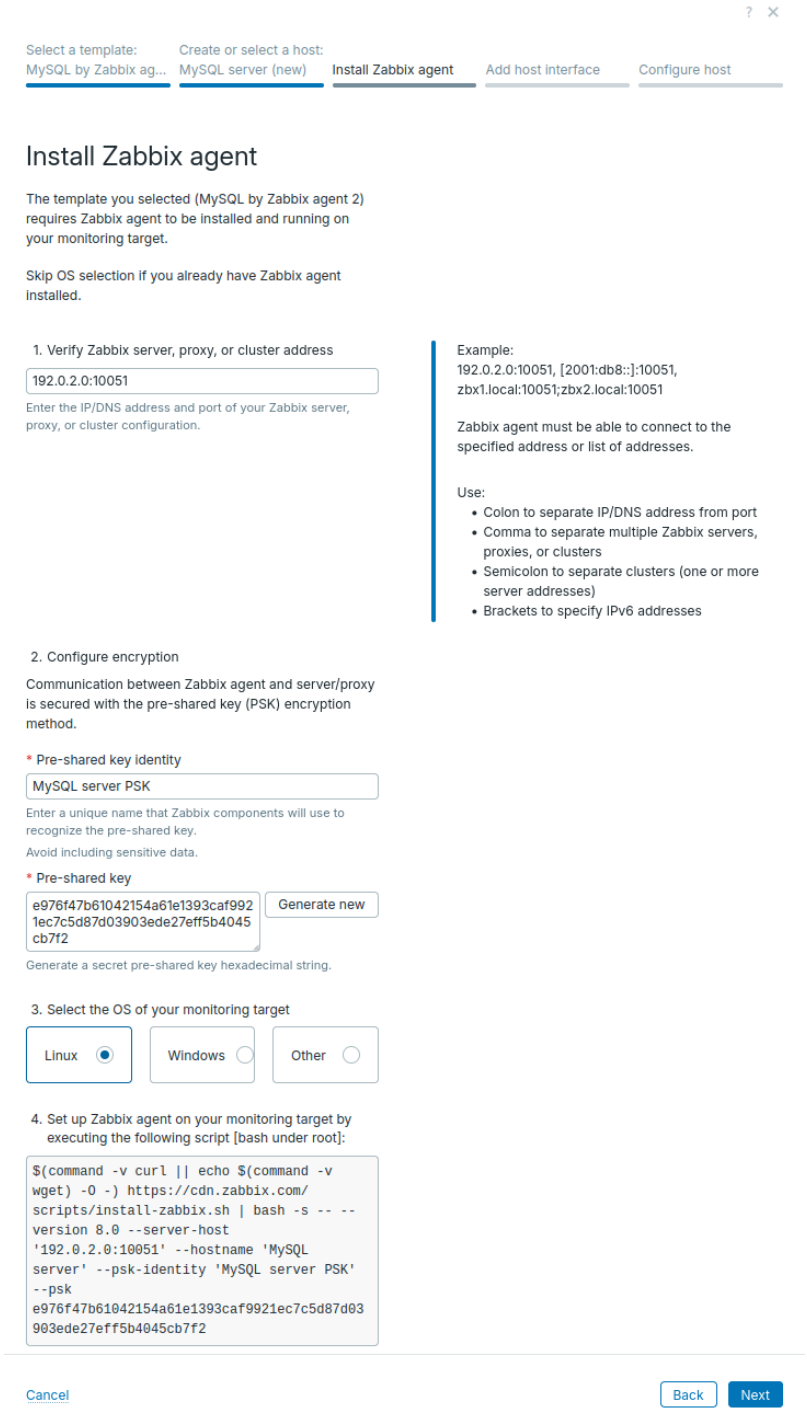
1. Avvia la Host Wizard in *Data collection* > *Hosts*.
2. Seleziona il template *MySQL by Zabbix agent 2*.



3. Crea un host server MySQL e aggiungi un gruppo. Questo host virtuale conterrà tutti i dati relativi al monitoraggio del server MySQL.

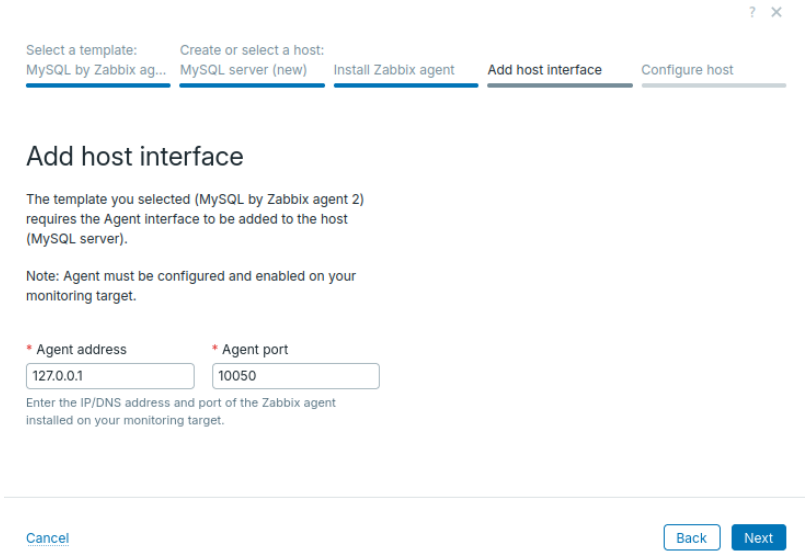


4. Configura e installa agent 2 (Zabbix agent 2 fornisce il monitoraggio MySQL già pronto all'uso).

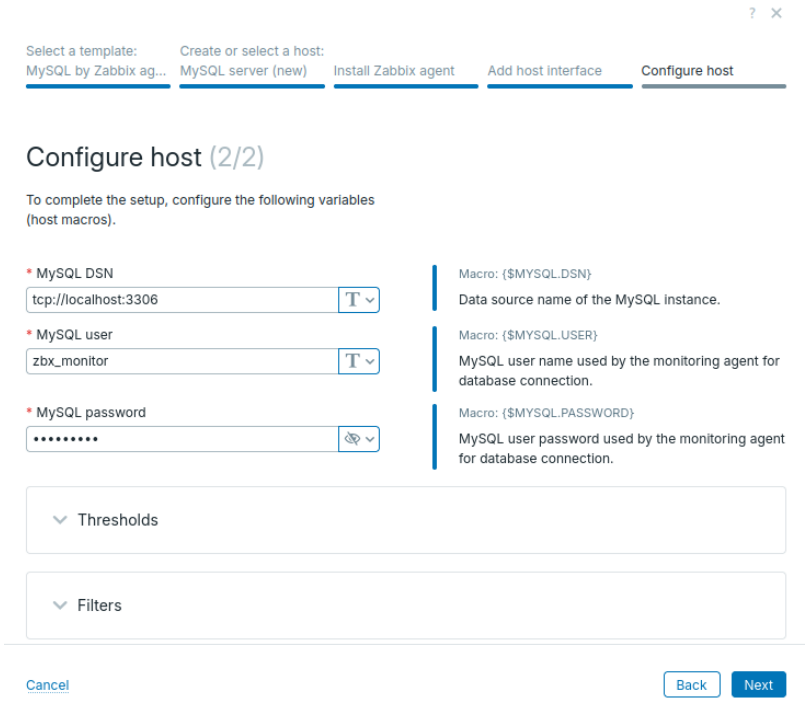


Per installare l'agent, esegui il comando fornito sulla macchina che ospita il server MySQL. Installerà l'agent e configurerà la connessione dell'agent a Zabbix server con crittografia.
 Se hai già installato agent 2, puoi saltare questo passaggio (fai clic su *Next*). Tuttavia, ti verrà richiesto di fornire l'identità della chiave precondivisa e la chiave precondivisa per connetterti all'agent.

5. Inserisci l'indirizzo IP e la porta dell'agent (server Linux).



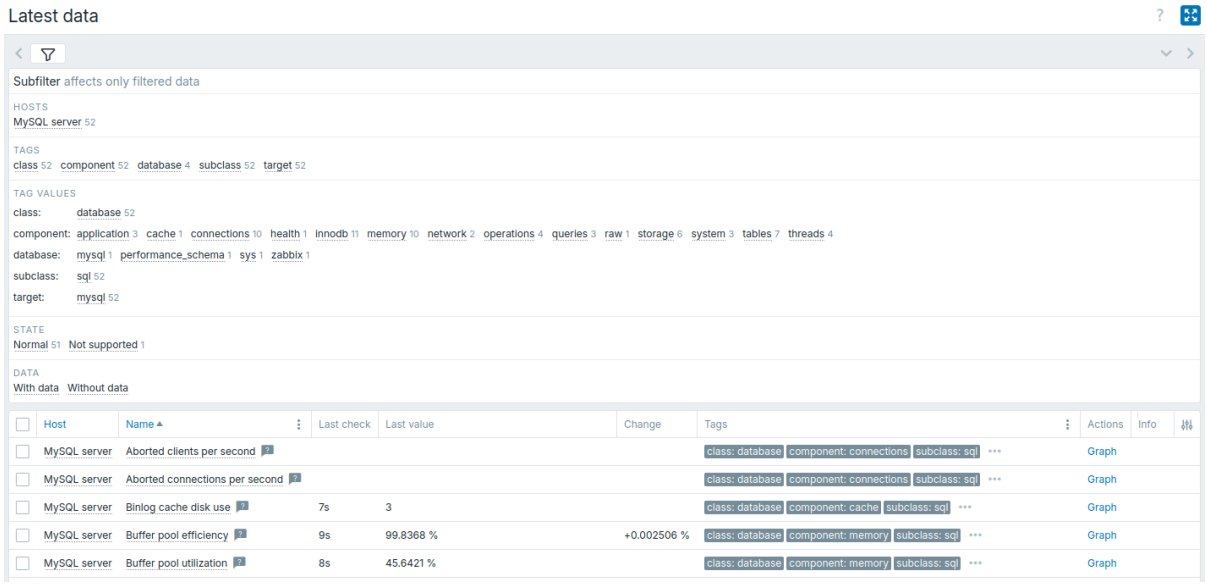
6. Completa la configurazione dell'host. In particolare, ti verrà chiesto di inserire il DSN del server MySQL, l'utente MySQL e la password. Fornisci l'utente `zbx_monitor` con la password che hai creato.



Puoi lasciare le altre impostazioni ai valori predefiniti.

6. Per completare la Host Wizard, fai clic su *Create* e *Finish*.

Visualizza i dati più recenti



Congratulazioni, il monitoraggio del server MySQL è stato avviato!

Zabbix offre opzioni di visualizzazione e invia avvisi se si verificano problemi. Per configurare gli avvisi via email, consulta Ricezione delle notifiche di problema.

Altre guide

Questa sezione della documentazione contiene ricette rapide per configurare Zabbix per alcuni obiettivi di monitoraggio comunemente richiesti.

È pensata per il nuovo utente Zabbix e può essere utilizzata come guida di navigazione attraverso altre sezioni della documentazione che contengono le informazioni necessarie per risolvere il compito.

Sono disponibili le seguenti guide di riferimento rapido:

- Monitorare Linux con Zabbix agent
- Monitorare Windows con Zabbix agent
- Monitorare Apache tramite HTTP
- Monitorare MySQL con Zabbix agent 2
- Monitorare VMware con Zabbix
- Monitorare il traffico di rete con Zabbix
- Monitorare il traffico di rete con Zabbix usando controlli attivi
- Monitorare i siti web con Browser items
- Monitorare i certificati dei siti web con Zabbix agent 2 (passivo)
- Monitorare uno switch di rete o un router con Zabbix
- Monitorare il registro eventi di Windows usando controlli attivi

Monitorare Linux con Zabbix agent

Introduzione

Questa pagina illustra i passaggi necessari per avviare il monitoraggio di base delle macchine Linux con Zabbix. I passaggi descritti in questo tutorial possono essere applicati a qualsiasi sistema operativo basato su Linux.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base della tua macchina Linux. Se stai cercando opzioni di personalizzazione approfondite o hai bisogno di una configurazione più avanzata, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server e Zabbix frontend seguendo le istruzioni per il proprio sistema operativo.

Installare Zabbix agent

Zabbix agent è il processo responsabile della raccolta dei dati.

Verifica la versione del tuo server Zabbix:

```
zabbix_server -V
```

Installa Zabbix agent della stessa versione (consigliato) sulla macchina Linux che desideri monitorare. In base alle tue esigenze di monitoraggio, può trattarsi della stessa macchina su cui è installato il server Zabbix oppure di una macchina completamente diversa.

Scegli il metodo di installazione più adatto:

- Esegui come container Docker - consulta l'elenco delle immagini disponibili nel repository Docker di Zabbix.
- Installa dai pacchetti di Zabbix (disponibili per Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Compila dai sorgenti.

Configurare Zabbix per il monitoraggio

Zabbix agent può raccogliere metriche in modalità attiva o passiva (simultaneamente).

> **Note:**
> Un controllo passivo è una semplice richiesta di dati. Zabbix server o proxy richiede alcuni dati (ad esempio, il carico della CPU) e Zabbix agent invia il risultato al server. I controlli attivi richiedono un'elaborazione più complessa. L'agent deve prima recuperare dal server(s) un elenco di item per l'elaborazione indipendente e poi inviare in blocco i dati indietro. Vedere Controlli agent passivi e attivi per ulteriori informazioni.

I template di monitoraggio forniti da Zabbix in genere offrono due alternative: un template per Zabbix agent e un template per Zabbix agent (active). Con la prima opzione, l'agent raccoglierà le metriche in modalità passiva. Tali template forniranno risultati di monitoraggio identici, ma utilizzando protocolli di comunicazione diversi.

La configurazione successiva di Zabbix dipende dal fatto che si selezioni un template per i controlli Zabbix agent attivi o passivi.

Controlli passivi

Zabbix agent

1. Aprire il file di configurazione dell'agent sulla macchina in cui è installato l'agent.

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

2. Aggiungere l'indirizzo IP o il nome DNS (e la porta facoltativa) del server Zabbix al parametro server. Ad esempio:

```
Server=192.0.2.1:10051
```

Zabbix agent utilizzerà questo indirizzo per accettare connessioni in ingresso solo dai server Zabbix o dai proxy specificati per recuperare i dati.

3. Riavviare Zabbix agent.

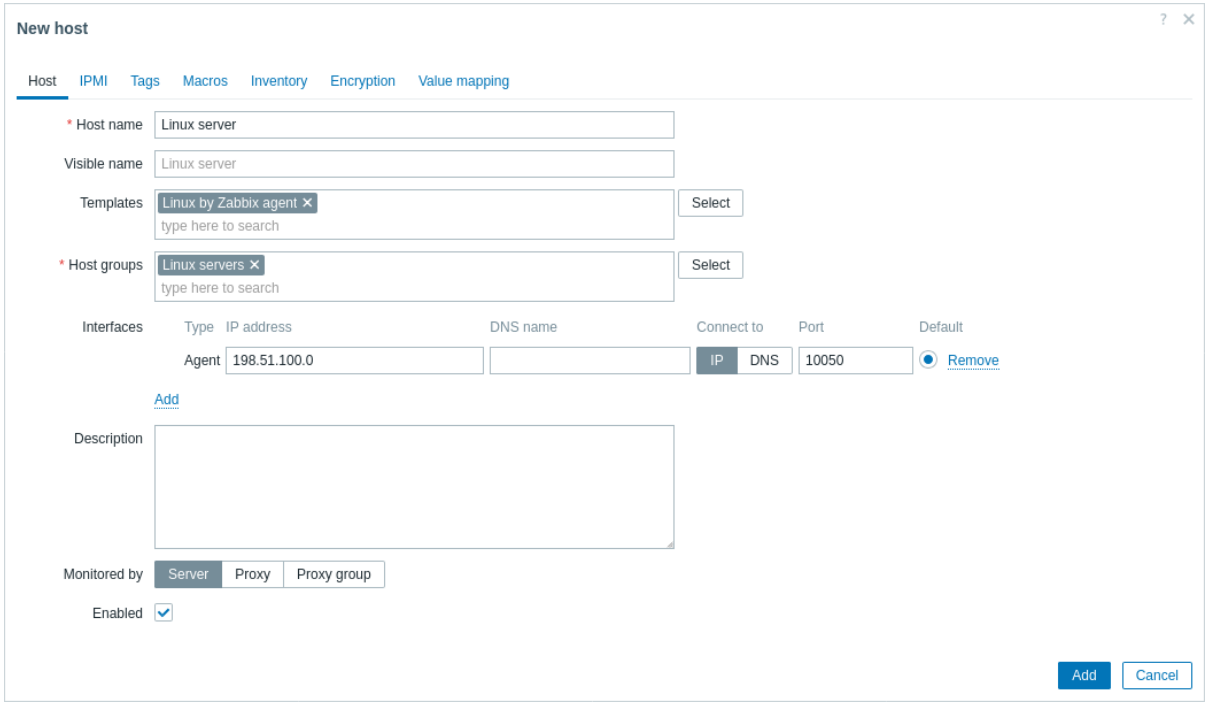
```
systemctl restart zabbix-agent
```

frontend Zabbix

1. Accedi al frontend Zabbix.

2. Crea un host nell'interfaccia web di Zabbix.

- Nel campo *Host name*, inserisci un nome host (ad esempio, "Linux server").
- Nel campo *Templates*, digita o seleziona il template "Linux by Zabbix agent" che verrà collegato all'host.
- Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, "Linux servers").
- Nel parametro *Interfaces*, aggiungi l'interfaccia *Agent* e specifica l'indirizzo IP o il nome DNS della macchina Linux su cui è installato l'agent.



3. Fai clic su *Add* per aggiungere l'host. Questo host rappresenterà la macchina Linux monitorata.

Controlli attivi

Zabbix agent

1. Aprire il file di configurazione dell'agent sulla macchina in cui è installato l'agent:

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

2. Aggiungere l'indirizzo IP o il nome DNS (e la porta opzionale) del server Zabbix al parametro ServerActive. Ad esempio:

```
ServerActive=192.0.2.1:10051
```

Zabbix agent utilizzerà questo indirizzo per connettersi alla porta trapper del server Zabbix (predefinita: 10051) e richiedere i dati di configurazione per i controlli attivi.

3. Definire il parametro Hostname, che deve corrispondere al nome dell'host che verrà definito nel frontend Zabbix. Nel nostro esempio è:

```
Hostname=Linux server
```

Il valore di Hostname deve corrispondere perché, per i controlli attivi, Zabbix agent lo utilizza per recuperare la configurazione corretta dell'host dal server. In particolare, l'agent avvia una connessione al server e si identifica utilizzando il valore di Hostname. Il server fornisce quindi la configurazione di monitoraggio per quell'host. Se questi valori differiscono, l'agent non riceverà la configurazione appropriata, causando metriche mancanti o problemi di monitoraggio.

4. Riavviare Zabbix agent.

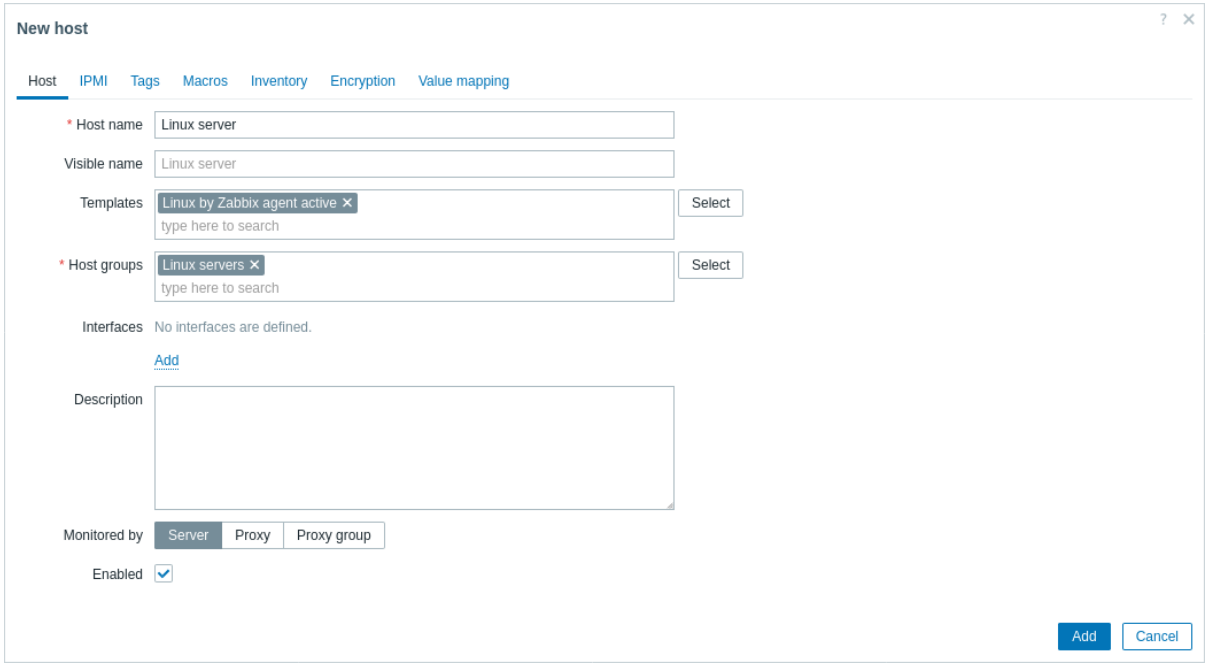
```
systemctl restart zabbix-agent
```

frontend di Zabbix

1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix.

- Nel campo *Host name*, inserisci un nome host (ad esempio, "Linux server") che corrisponda al valore del parametro *Hostname* definito in precedenza nel file di configurazione dell'agent.
- Nel campo *Templates*, digita o seleziona il template "Linux by Zabbix agent active" che verrà collegato all'host.
- Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, "Linux servers").
- Lascia *Interfaces* non definito; un'interfaccia non è necessaria per i controlli attivi perché l'agent avvia la connessione al server invece di attendere una connessione dal server.



3. Fai clic su *Add* per aggiungere l'host. Questo host rappresenterà la macchina Linux monitorata.

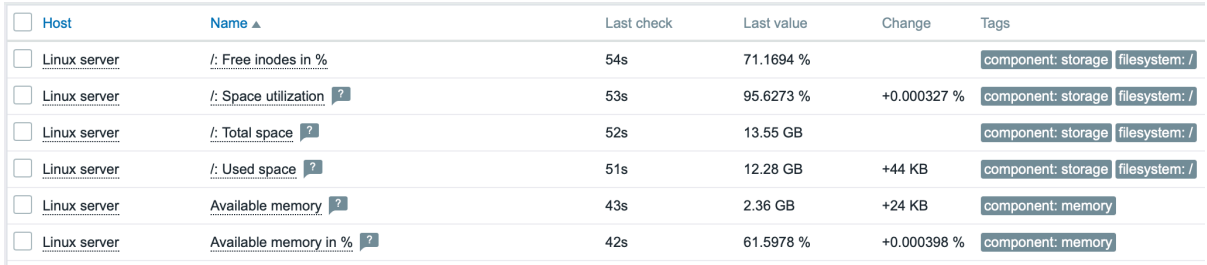
Visualizzare le metriche raccolte

Congratulazioni! A questo punto, Zabbix sta già monitorando la tua macchina Linux.

Per visualizzare le metriche raccolte, apri la sezione di menu *Monitoring* > *Hosts* menu section e fai clic su *Latest data* accanto all'host.



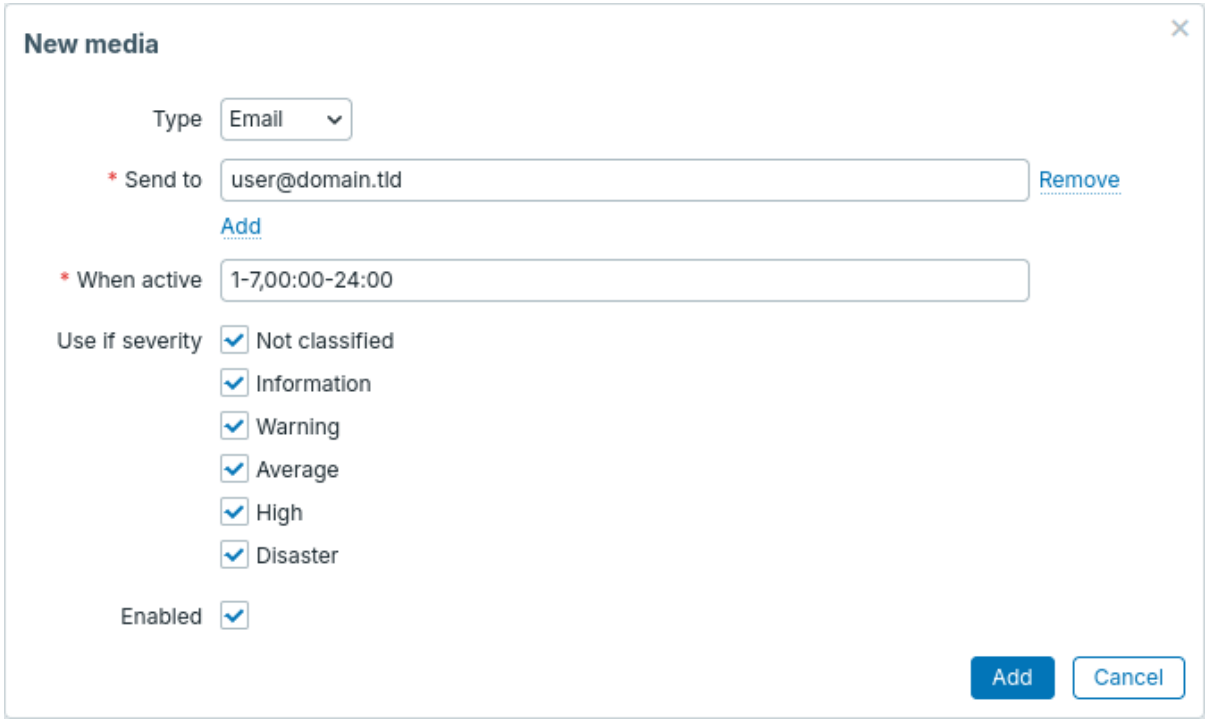
Questa azione aprirà un elenco di tutte le metriche più recenti raccolte dall'host del server Linux.



Configurare gli avvisi di problema

Zabbix può notificarti un problema con la tua infrastruttura usando vari metodi.
Questa guida fornisce i passaggi di configurazione per l'invio di avvisi via email.

1. Vai a *User settings* > *Profile*, passa alla scheda *Media* e
aggiungi la tua email.



2. Segui la guida per Receiving problem notification.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

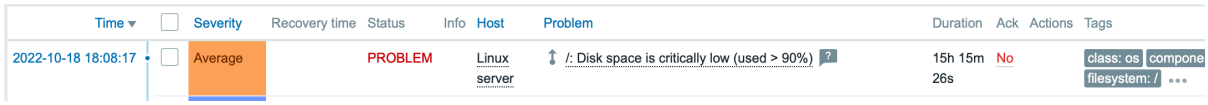
Testa la tua configurazione

Su Linux, puoi simulare un carico elevato della CPU e, di conseguenza, ricevere un alert di problema eseguendo:

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

Potrebbe essere necessario avviare più processi md5sum affinché il carico della CPU superi la soglia.

Quando Zabbix rileva il problema, questo apparirà nella sezione *Monitoring* > *Problems*.



Se gli alert sono configurati, riceverai anche la notifica del problema.

Vedi anche:

- Creazione di un item - come iniziare a monitorare metriche aggiuntive (monitoraggio personalizzato senza template).
- Item dell'agent Zabbix, Item dell'agent Zabbix per Windows - elenco completo delle metriche che puoi monitorare usando l'agent Zabbix su Windows.
- Escalation dei problemi - come creare scenari di avviso multi-step (ad es. inviare prima un messaggio all'amministratore di sistema, poi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Installazione dai pacchetti - come installare i componenti Zabbix usando i pacchetti ufficiali RPM e DEB per varie distribuzioni Linux, garantendo l'accesso alle funzionalità più recenti e alle correzioni dei bug.

Monitorare Windows con Zabbix agent

Introduzione

Questa pagina illustra i passaggi necessari per iniziare il monitoraggio di base delle macchine Windows con Zabbix.

A chi è destinata questa guida

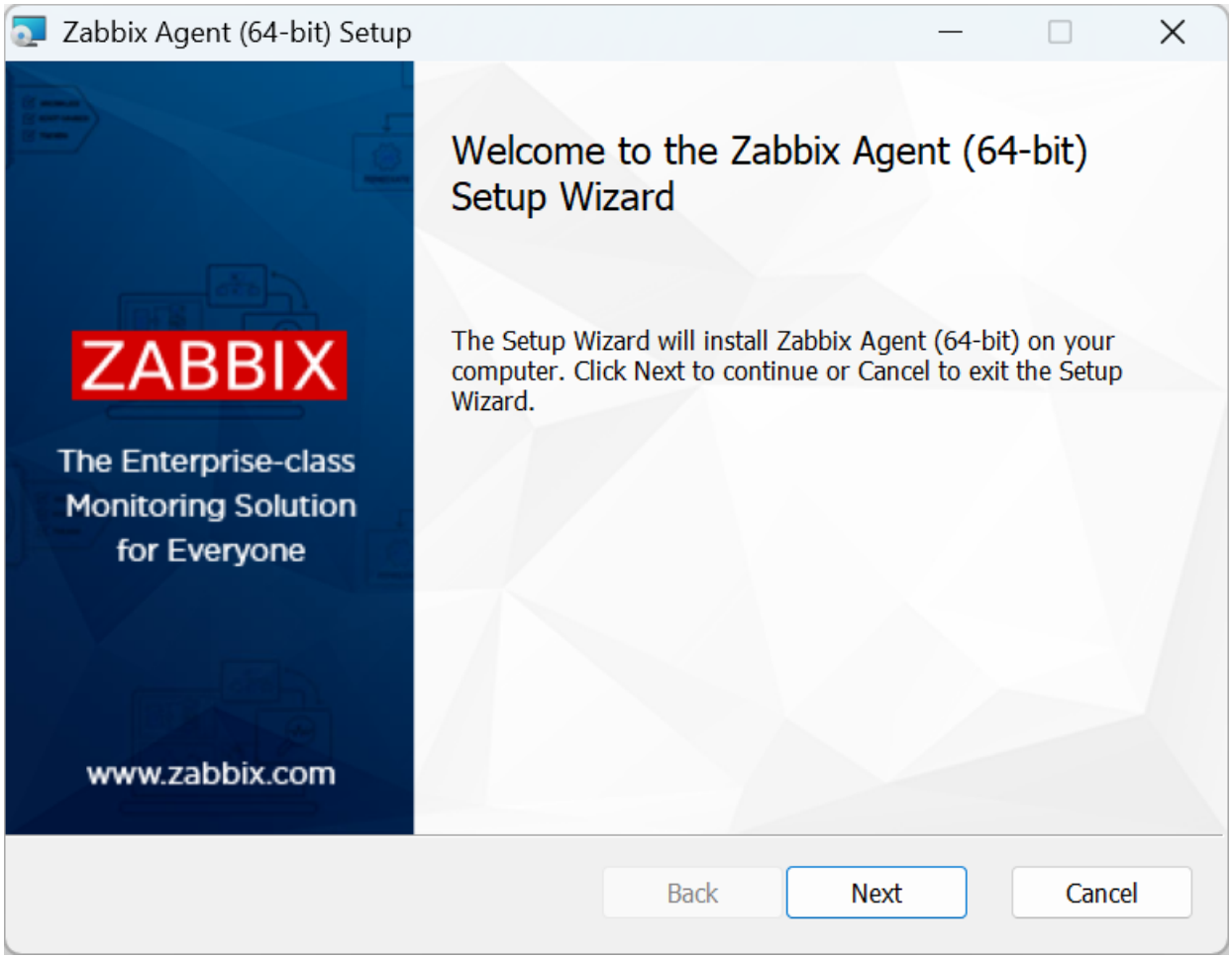
Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base della macchina Windows. Se si cercano opzioni di personalizzazione avanzate o è necessaria una configurazione più complessa, consultare la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida all'installazione, è necessario scaricare e installare il server Zabbix e il frontend Zabbix seguendo le istruzioni per il proprio sistema operativo.

Installare Zabbix agent

Zabbix agent è il processo responsabile della raccolta dei dati. È necessario installarlo sulla macchina Windows che si desidera monitorare. Seguire le istruzioni di installazione di Zabbix agent per Windows.



Configurare Zabbix per il monitoraggio

Zabbix agent può raccogliere metriche in modalità attiva o passiva (simultaneamente).

Note:
Un controllo passivo è una semplice richiesta di dati. Zabbix server o proxy richiede alcuni dati (ad esempio, il carico della CPU) e Zabbix agent invia il risultato al server. I controlli attivi richiedono un'elaborazione più complessa. L'agent deve prima recuperare dal server(s) un elenco di item per l'elaborazione indipendente e poi inviare in blocco i dati al server. Per ulteriori informazioni, vedere Passive and active agent checks.

I template di monitoraggio forniti da Zabbix in genere offrono due alternative: un template per Zabbix agent e un template per Zabbix agent (active). Con la prima opzione, l'agent raccoglierà le metriche in modalità passiva. Questi template forniranno risultati di monitoraggio identici, ma utilizzando protocolli di comunicazione diversi.

La configurazione successiva di Zabbix dipende dal fatto che si selezioni un template per i controlli Zabbix agent attivi o passivi.

Controlli passivi

Frontend di Zabbix

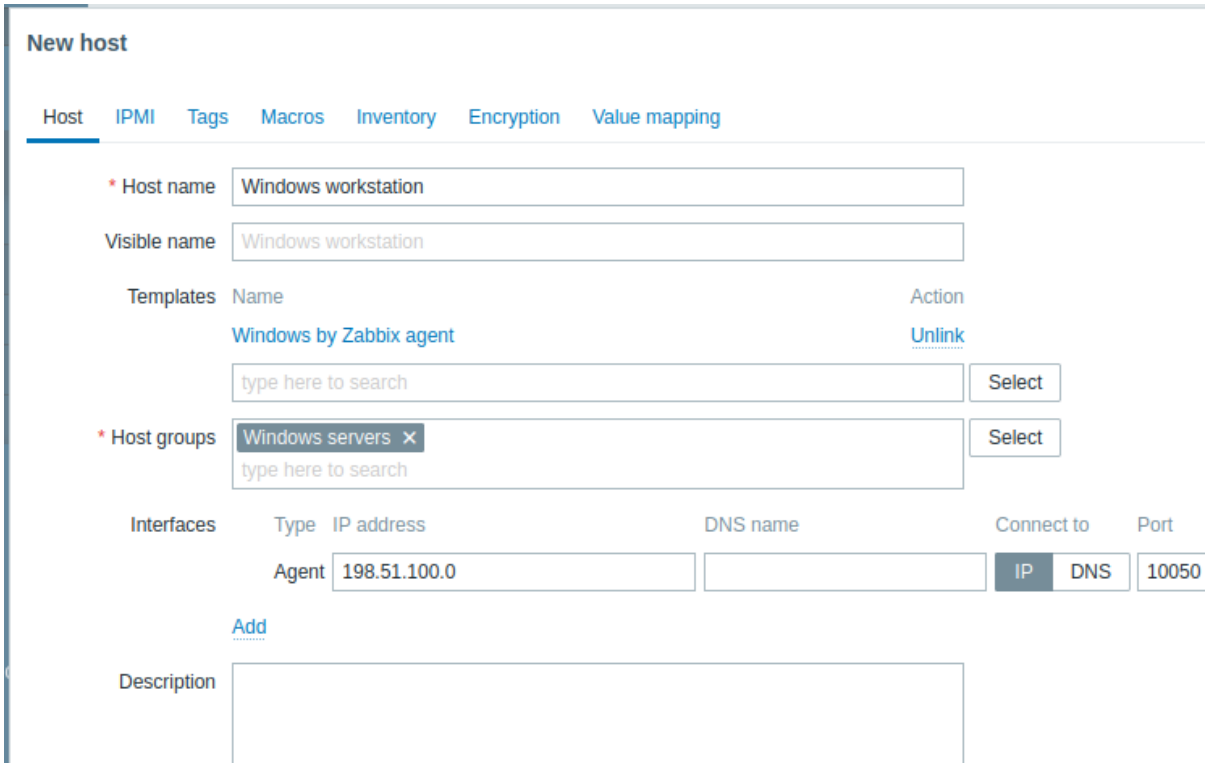
1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix.

Questo host rappresenterà la tua macchina Windows.

3. Nel parametro *Interfaces*, aggiungi l'interfaccia *Agent* e specifica l'indirizzo IP o il nome DNS della macchina Windows in cui è installato l'agent.

4. Nel parametro *Templates*, digita o seleziona *Windows by Zabbix agent*.



Zabbix agent

Per i controlli passivi, Zabbix agent deve conoscere l'indirizzo IP o il nome DNS del server Zabbix. Se sono state fornite le informazioni corrette durante l'installazione di Zabbix agent, il file di configurazione è già stato aggiornato. In caso contrario, è necessario specificarle manualmente. Aprire il file di configurazione di Zabbix agent e aggiungere l'indirizzo IP/DNS del server Zabbix al parametro `Server`.

Esempio:

```
Server=192.0.2.22
```

Controlli attivi

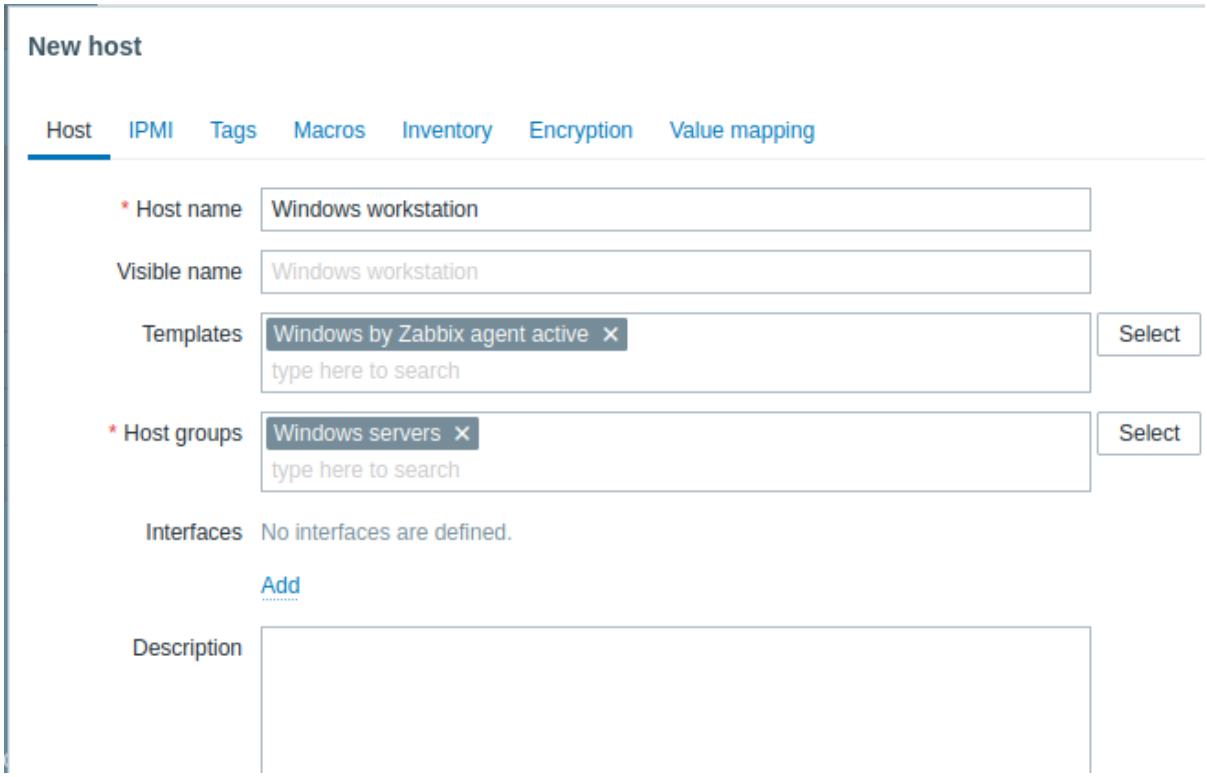
Frontend di Zabbix

1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix.

Questo host rappresenterà la tua macchina Windows.

3. Nel parametro *Templates*, digita o seleziona *Windows by Zabbix agent active*.



Zabbix agent

Nella cartella `C:\Program files\Zabbix Agent` apri il file `zabbix_agentd.conf` e aggiungi:

- Il nome dell'host che hai creato nel frontend di Zabbix al parametro *Hostname*.
- L'indirizzo IP o il nome DNS della macchina del tuo server Zabbix al parametro *ServerActive* (potrebbe essere già compilato se lo hai specificato durante la configurazione di Zabbix agent).

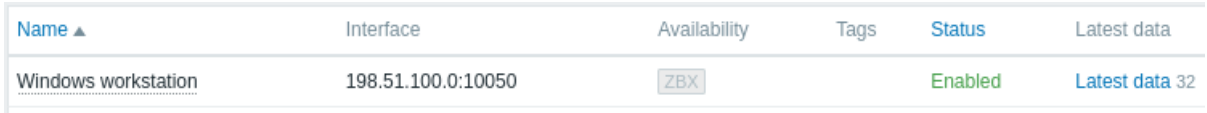
Esempio:

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

Visualizza le metriche raccolte

Congratulazioni! A questo punto, Zabbix sta già monitorando la tua macchina Windows.

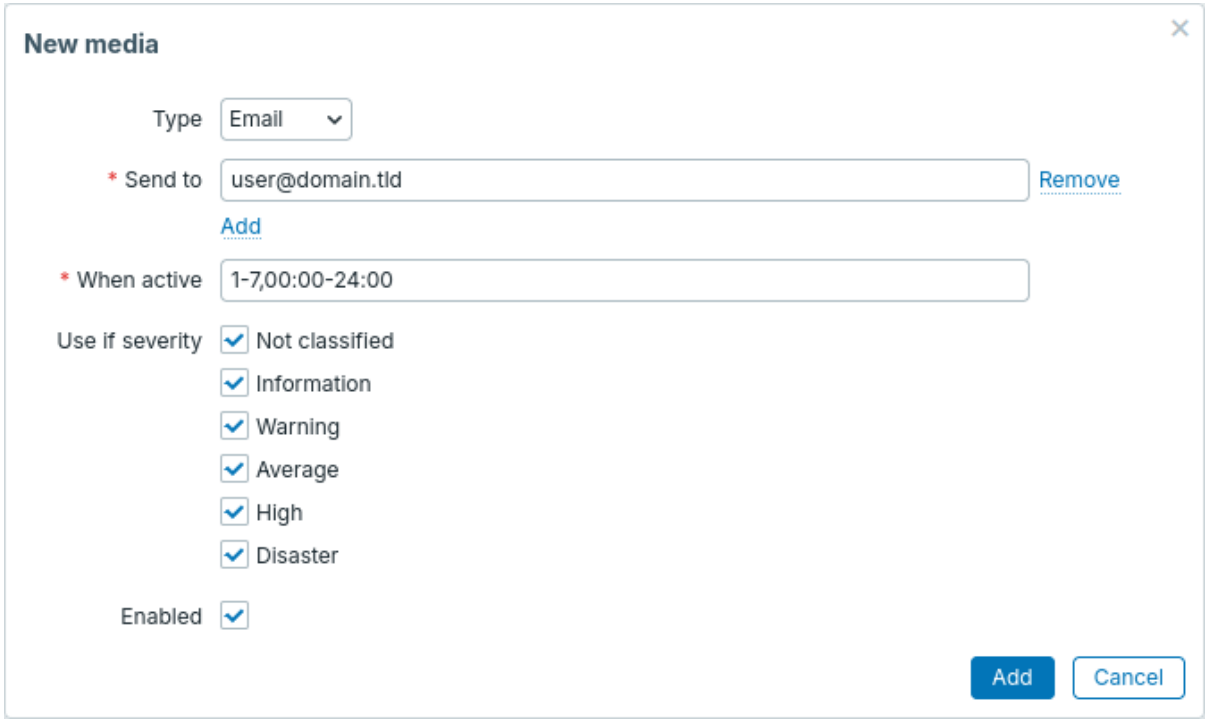
Per visualizzare le metriche raccolte, apri la sezione di menu *Monitoring* > *Hosts* menu section e fai clic su *Latest data* accanto all'host.



Configurare gli avvisi di problema

Zabbix può notificarti un problema con la tua infrastruttura usando vari metodi. Questa guida fornisce i passaggi di configurazione per l'invio di avvisi via email.

1. Vai su *Impostazioni utente* > *Profilo*, passa alla scheda *Media* e aggiungi la tua email.



2. Segui la guida per Ricevere notifiche di problema.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Note:
Su Windows, puoi usare l'utility CpuStres per simulare un carico elevato della CPU e, di conseguenza, ricevere un avviso di problema.

Vedi anche:

- Creare un item - come iniziare a monitorare metriche aggiuntive (monitoraggio personalizzato senza template).
- Item dell'agent Zabbix, Item dell'agent Zabbix per Windows - elenco completo delle metriche che puoi monitorare usando l'agent Zabbix su Windows.
- Escalation dei problemi - come creare scenari di avviso multi-passaggio (ad esempio, prima inviare un messaggio all'amministratore di sistema, poi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).

Monitorare Apache tramite HTTP

Introduzione

Questa pagina mostra un modo rapido e semplice per iniziare a monitorare un server web Apache senza installare alcun software aggiuntivo.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base della tua installazione di Apache. Se stai cercando opzioni di personalizzazione avanzate o hai bisogno di una configurazione più complessa, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida all'installazione, è necessario scaricare e installare Zabbix server e Zabbix frontend seguendo le istruzioni per il proprio sistema operativo.

Preparare Apache

1. Verifica quale versione di Apache stai utilizzando:

Su un sistema basato su RHEL, esegui:

```
httpd -v
```

Su Debian/Ubuntu, esegui:

```
apache2 -v
```

2. Assicurati che il modulo Status sia abilitato nella tua istanza di Apache.

Su un sistema basato su RHEL, esegui:

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

Su Debian/Ubuntu, esegui:

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

Se non vedi status_module nell'elenco, abilita il modulo eseguendo:

Su un sistema basato su RHEL, esegui:

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

Su Debian/Ubuntu, esegui:

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

3. Modifica il file di configurazione di Apache per consentire l'accesso ai report di stato dall'IP del server Zabbix.

Su un sistema basato su RHEL: `/etc/httpd/conf.modules.d/status.conf`:

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

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

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

Aggiungi le seguenti righe al file (**sostituisci 198.51.100.255** con l'indirizzo IP del tuo server Zabbix):

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

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

4. Riavvia Apache

Su un sistema basato su RHEL, esegui:

```
sudo systemctl restart httpd
```

Su Debian/Ubuntu, esegui:

```
sudo systemctl restart apache2
```

5. Per verificare che tutto sia configurato correttamente, esegui (**sostituisci 198.51.100.255** con l'indirizzo IP del tuo server Zabbix):

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

La risposta dovrebbe contenere le statistiche del web server Apache.

Configurare Zabbix per il monitoraggio

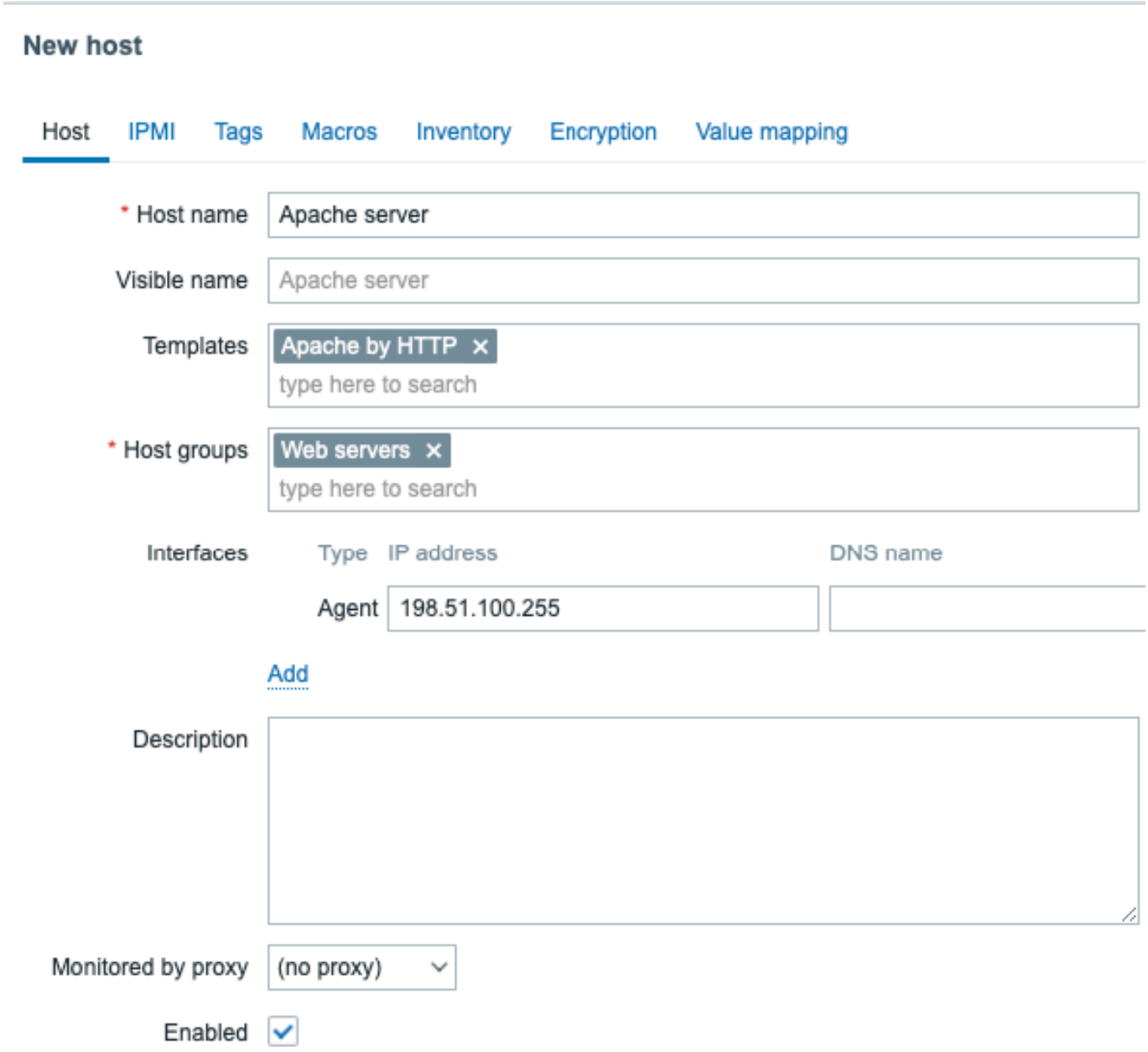
1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix.

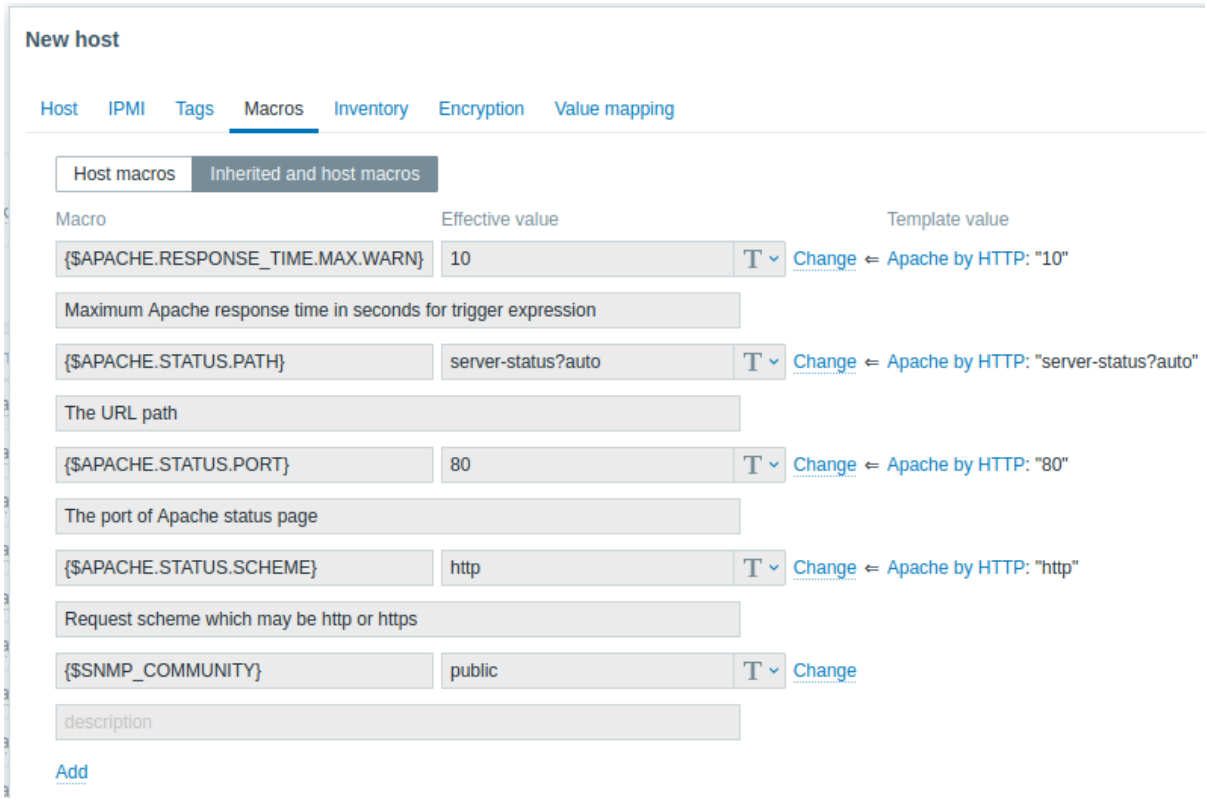
Questo host rappresenterà il tuo server Apache.

3. Nel parametro *Interfaces*, aggiungi l'interfaccia *Agent* e specifica l'indirizzo IP della tua istanza Apache. **Non è necessario installare Zabbix agent sulla macchina**, l'interfaccia verrà usata solo per risolvere la macro {HOST.CONN}. Questa macro viene usata negli item del template per individuare l'istanza Apache.

4. Nel parametro *Templates*, digita o seleziona *Apache by HTTP*.



5. Passa alla scheda **Macros** e seleziona la modalità *Inherited and host macros*. Verifica che i valori delle macro {$APACHE.STATUS.PORT} e {$APACHE.STATUS.SCHEME} siano adatti alle impostazioni della tua installazione. Per impostazione predefinita, la porta è 80 e lo schema è http. Modifica i valori delle macro se utilizzi una porta e/o uno schema diversi.



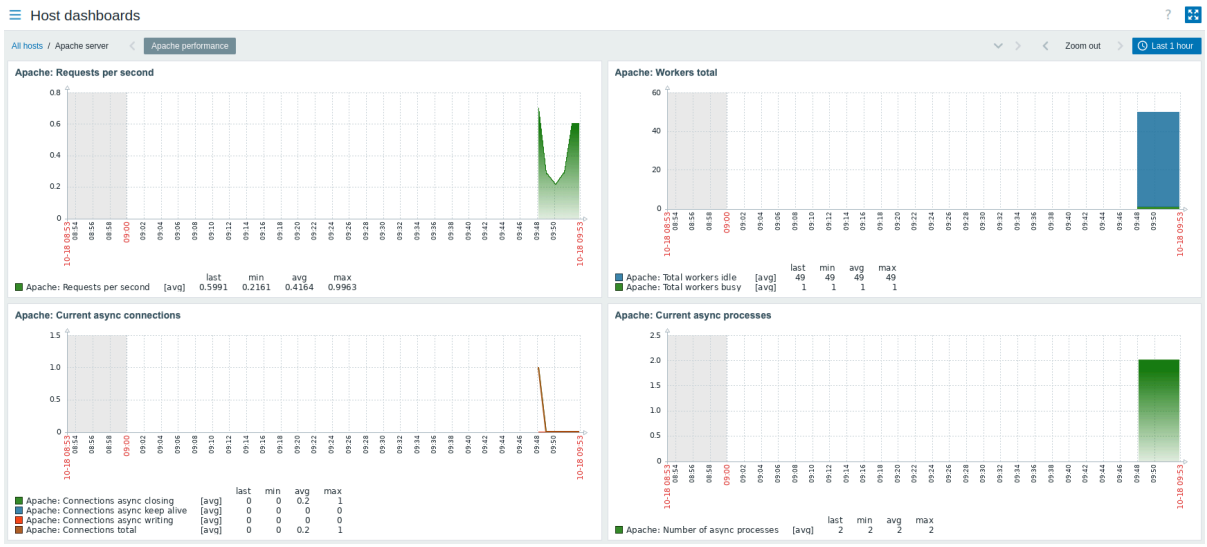
Visualizzare le metriche raccolte

Congratulazioni! A questo punto, Zabbix sta già monitorando il tuo web server Apache.

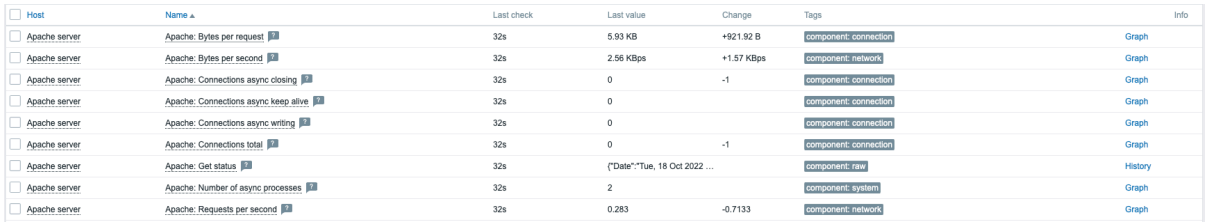
Per visualizzare le metriche raccolte, apri la sezione di menu *Monitoring* > *Hosts* menu section e fai clic su *Dashboards* accanto all'host.



Questa azione ti porterà alla dashboard dell'host con le metriche più importanti raccolte dalla pagina /server-status di Apache.



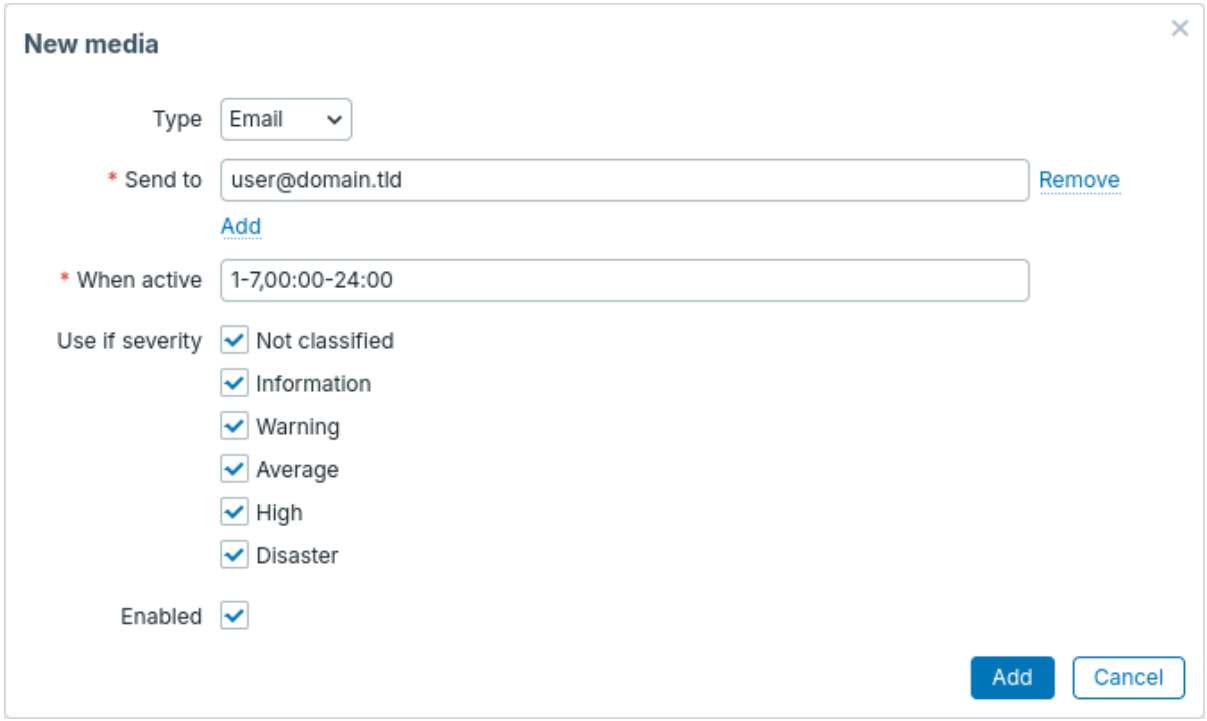
In alternativa, da *Monitoring* > *Hosts*, puoi fare clic su *Latest data* per visualizzare tutte le metriche raccolte più di recente in un elenco.



Configurare gli avvisi sui problemi

Zabbix può notificarti un problema nella tua infrastruttura usando diversi metodi.
Questa guida fornisce i passaggi di configurazione per l'invio di avvisi via email.

1. Vai a *User settings* > *Profile*, passa alla scheda *Media* e aggiungi la tua email.



2. Segui la guida per Receiving problem notification.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Testa la configurazione

Per simulare un problema reale e ricevere un alert di test:

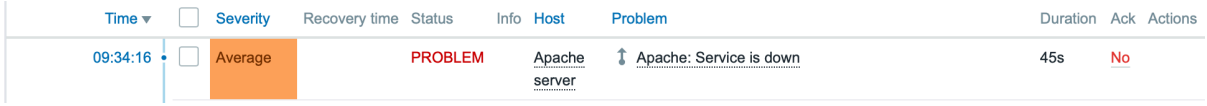
1. Apri la configurazione dell'host *Apache server* in Zabbix.

2. Passa alla scheda Macros e seleziona *Inherited and host macros*.

3. Fai clic su *Change* accanto alla macro {$APACHE.STATUS.PORT} e imposta una porta diversa.

4. Fai clic su *Update* per salvare la configurazione dell'host.

5. Tra pochi minuti, Zabbix rileverà il problema *Apache service is down*, perché ora non riesce a connettersi all'istanza. Comparirà nella sezione *Monitoring* > *Problems*.



Se gli alert sono configurati, riceverai anche la notifica del problema.

6. Ripristina il valore della macro per risolvere il problema e continuare a monitorare Apache.

Vedi anche:

- Web server hardening - impostazioni consigliate per una maggiore sicurezza del web server.
- Creating an item - come iniziare a monitorare metriche aggiuntive.
- HTTP items - come monitorare metriche personalizzate usando HTTP agent.
- Problem escalations - come creare scenari di avviso multi-step (ad esempio, inviare prima un messaggio all'amministratore di sistema, poi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).

Monitorare MySQL con Zabbix agent 2

Introduzione

Questa pagina illustra i passaggi necessari per avviare il monitoraggio di base di un server MySQL.

Per monitorare un server MySQL, esistono diversi approcci: Zabbix agent, Zabbix agent 2 oppure lo standard Open Database Connectivity (ODBC). L'obiettivo principale di questa guida è il monitoraggio di un server MySQL con Zabbix agent 2, che è l'approccio **consigliato** grazie alla sua configurazione semplice e uniforme in diversi scenari. Tuttavia, questa pagina fornisce anche istruzioni per gli altri approcci, quindi puoi scegliere liberamente quello più adatto alle tue esigenze.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base di un server MySQL. Se stai cercando opzioni di personalizzazione approfondita o hai bisogno di una configurazione più avanzata, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server, Zabbix frontend e Zabbix agent 2 seguendo le istruzioni per il proprio sistema operativo.

A seconda del proprio ambiente, alcuni passaggi di questa guida potrebbero differire leggermente. Questa guida si basa su un ambiente che esegue Ubuntu.

Creare un utente MySQL

Per monitorare un server MySQL, Zabbix richiede l'accesso a esso e ai suoi processi. La tua installazione MySQL dispone già di un utente con il livello di accesso richiesto (l'utente "zabbix" creato durante l'installazione di Zabbix), tuttavia, questo utente ha più privilegi del necessario per un semplice monitoraggio (privilegi per DROP dei database, DELETE delle voci dalle tabelle, ecc.). Pertanto, è necessario creare un utente MySQL con il solo scopo di monitorare il server MySQL.

1. Connettiti al client MySQL, crea un utente "zbx_monitor" (sostituisci <*password*> per l'utente "zbx_monitor" con una password a tua scelta), e GRANT i privilegi necessari all'utente:

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

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

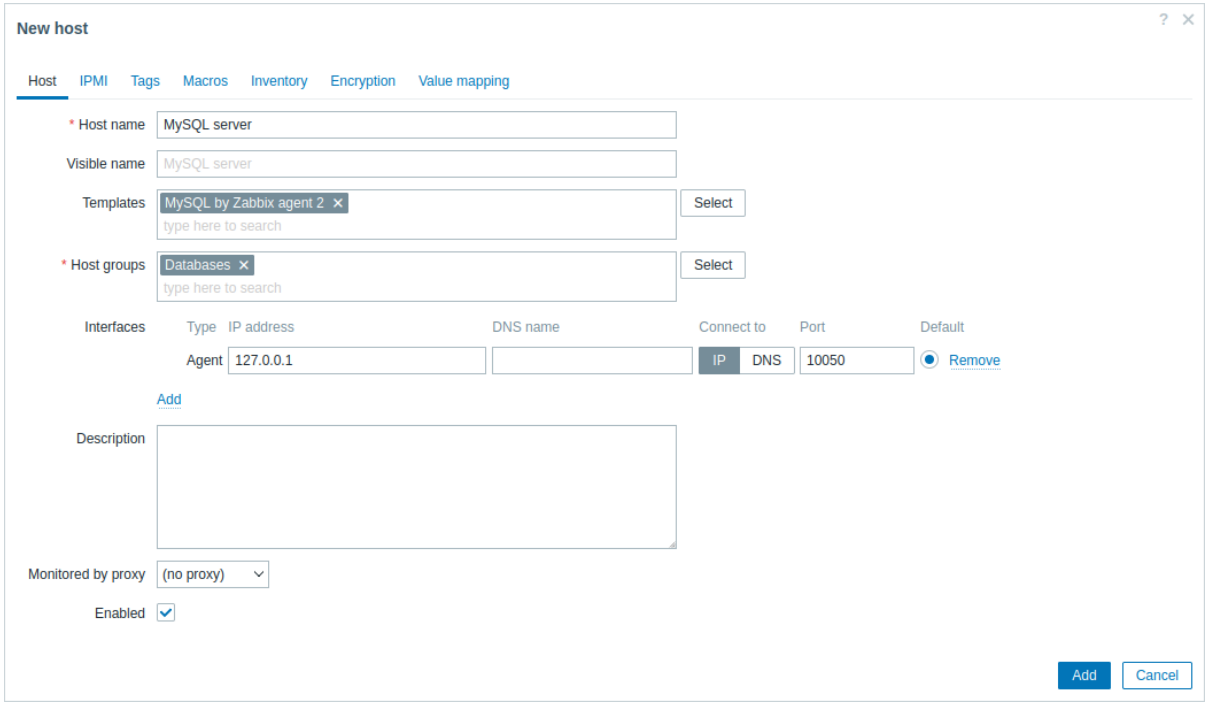
Una volta creato l'utente, puoi passare al passaggio successivo.

Configurare il frontend di Zabbix

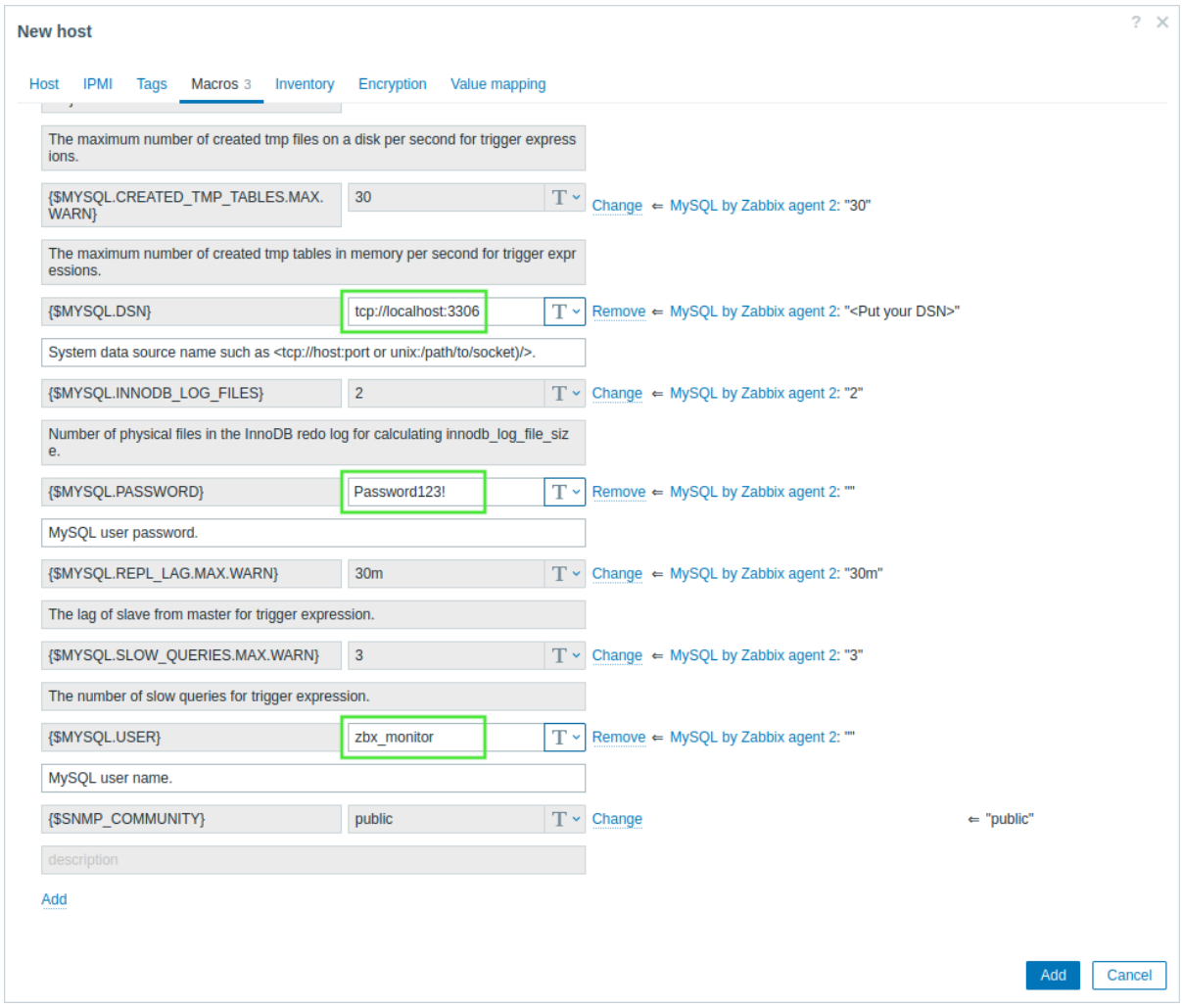
1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix:

   - Nel campo *Host name*, inserisci un nome host (ad esempio, "MySQL server").
   - Nel campo *Templates*, digita o seleziona il template "MySQL by Zabbix agent 2" che verrà collegato all'host.
   - Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, "Databases").
   - Nel campo *Interfaces*, aggiungi un'interfaccia di tipo "Agent" e specifica l'indirizzo IP del tuo server MySQL. Questa guida usa "127.0.0.1" (localhost) per monitorare un server MySQL installato sulla stessa macchina di Zabbix server e Zabbix agent 2.



- Nella scheda *Macros*, passa a *Inherited and host macros*, cerca le seguenti macro e fai clic su *Change* accanto al valore della macro per aggiornarlo:
  - {$MYSQL.DSN} - imposta l'origine dati del server MySQL (la stringa di connessione di una sessione denominata dal file di configurazione del plugin MySQL di Zabbix agent 2). Questa guida usa l'origine dati predefinita "tcp://localhost:3306" per monitorare un server MySQL installato sulla stessa macchina di Zabbix server e Zabbix agent 2.
  - {$MYSQL.PASSWORD} - imposta la password del precedente utente MySQL creato "zbx_monitor".
  - {$MYSQL.USER} - imposta il nome del precedente utente MySQL creato "zbx_monitor".

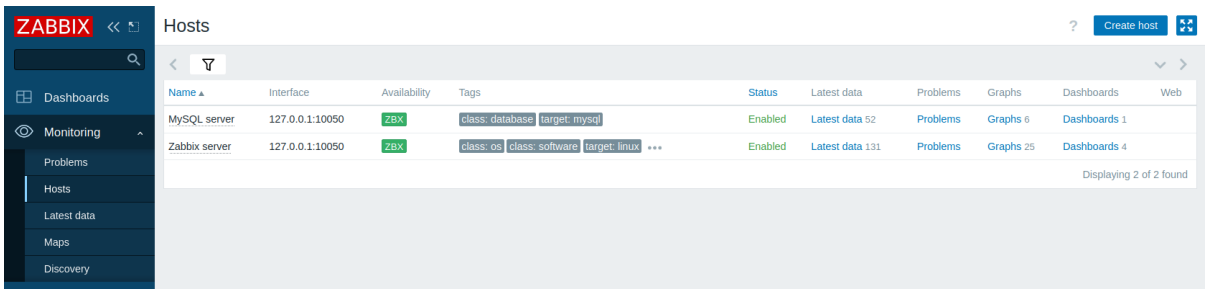


3. Fai clic su *Add* per aggiungere l'host. Questo host rappresenterà il tuo server MySQL.

Visualizzare le metriche raccolte

Congratulazioni! A questo punto, Zabbix sta già monitorando il tuo server MySQL.

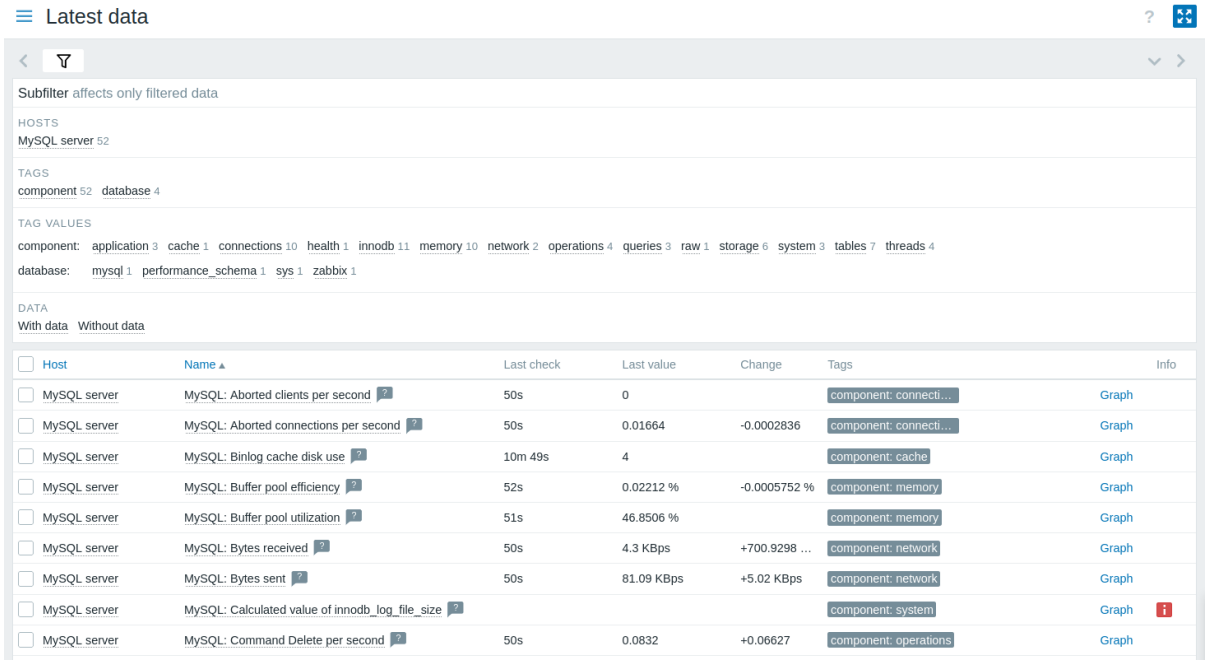
Per visualizzare le metriche raccolte, vai alla sezione di menu *Monitoring > Hosts* e fai clic su *Dashboards* accanto all'host.



Questa azione ti porterà alla dashboard dell'host (configurata a livello di template) con le metriche più importanti raccolte dal server MySQL.



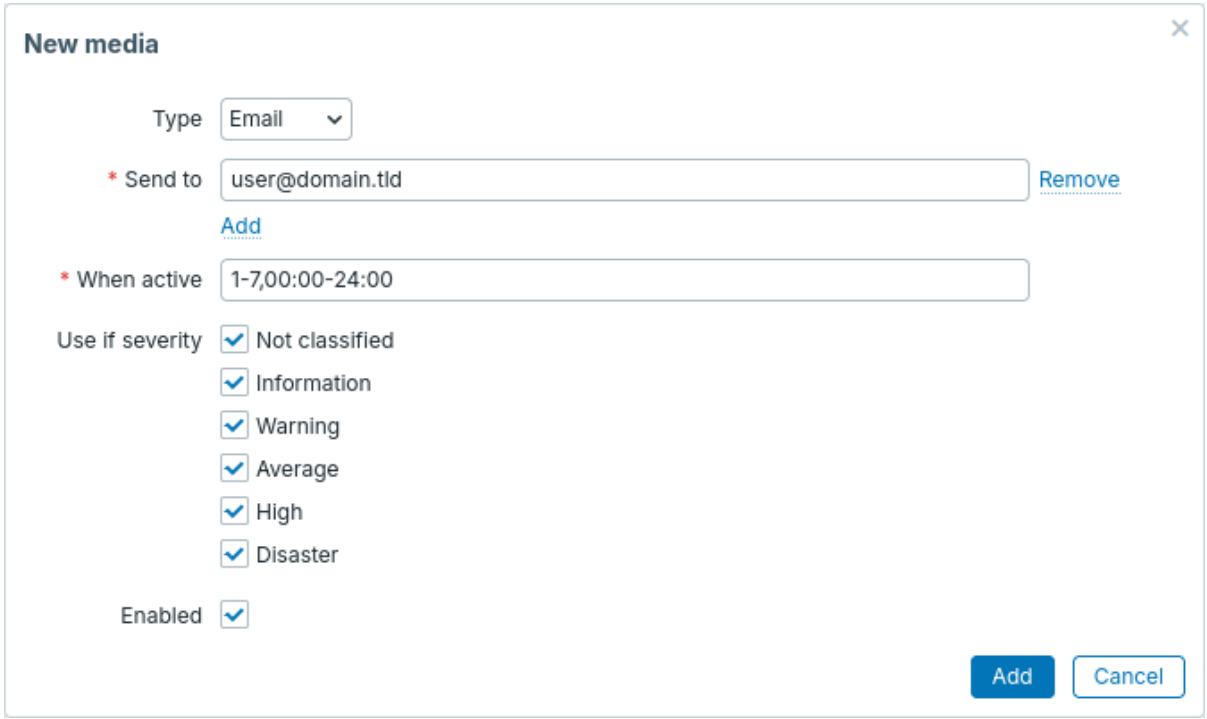
In alternativa, dalla sezione di menu *Monitoring > Hosts*, puoi fare clic su *Latest data* per visualizzare in un elenco tutte le metriche raccolte più recenti. Nota che l'item *MySQL: Calculated value of innodb_log_file_size* è previsto che non abbia dati, poiché il valore verrà calcolato dai dati dell'ultima ora.



Configurare gli avvisi di problema

Zabbix può notificarti un problema con la tua infrastruttura usando diversi metodi. Questa guida fornisce i passaggi di configurazione di base per inviare avvisi via email.

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Media* e aggiungi la tua email.



2. Segui la guida per Ricevere una notifica di problema.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Testare la configurazione

Per testare la configurazione, possiamo simulare un problema reale aggiornando la configurazione dell'host nel frontend di Zabbix.

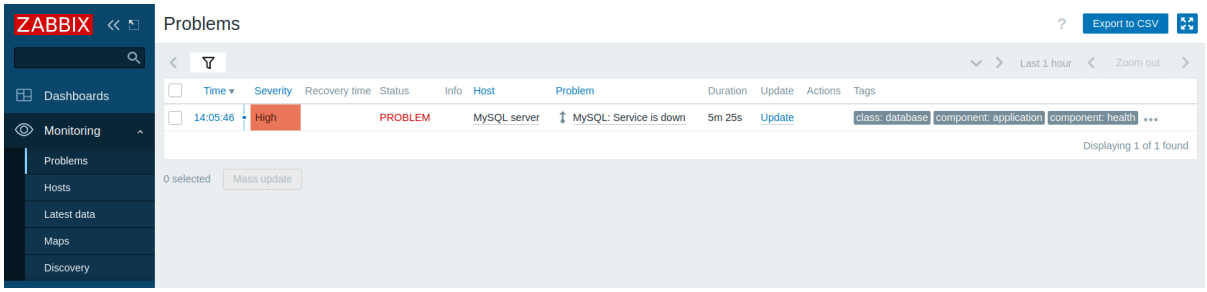
1. Apri la configurazione dell'host del tuo server MySQL in Zabbix.

2. Passa alla scheda *Macros* e seleziona *Inherited and host macros*.

3. Fai clic su *Change* accanto, ad esempio, al valore della macro {$MYSQL.USER} configurato in precedenza e imposta un nome utente MySQL diverso.

4. Fai clic su *Update* per aggiornare la configurazione dell'host.

5. Dopo qualche istante, Zabbix rileverà il problema ”MySQL: Service is down”, perché non sarà in grado di connettersi al server MySQL. Il problema apparirà in *Monitoring > Problems*.



Se gli avvisi sono configurati, riceverai anche la notifica del problema.

6. Riporta il valore della macro al valore precedente per risolvere il problema e continuare a monitorare il server MySQL.

Altri approcci per monitorare MySQL

Invece di monitorare un server MySQL con Zabbix agent 2, puoi anche usare Zabbix agent oppure lo standard Open Database Connectivity (ODBC). Sebbene l’uso di Zabbix agent 2 sia consigliato, potrebbero esserci alcune configurazioni che non supportano Zabbix agent 2 o che richiedono un approccio personalizzato.

La differenza principale tra Zabbix agent e ODBC risiede nel metodo di raccolta dei dati: Zabbix agent viene installato direttamente sul server MySQL e raccoglie i dati usando le funzionalità integrate, mentre ODBC si basa su un driver ODBC per stabilire una connessione al server MySQL e recuperare i dati tramite query SQL.

Sebbene molti passaggi di configurazione siano simili al monitoraggio di un server MySQL con Zabbix agent 2, esistono alcune differenze significative: devi configurare Zabbix agent oppure ODBC per poter monitorare un server MySQL. Le istruzioni seguenti ti guidano attraverso queste **differenze**.

Monitorare MySQL con Zabbix agent

Per monitorare un server MySQL con Zabbix agent, è necessario scaricare e installare Zabbix server, Zabbix frontend e Zabbix agent seguendo le istruzioni per il proprio sistema operativo.

Una volta installati correttamente i componenti Zabbix richiesti, è necessario creare un utente MySQL come descritto nella sezione *Create MySQL user*.

Dopo aver creato l’utente MySQL, è necessario configurare Zabbix agent in modo che possa stabilire una connessione con il server MySQL e monitorarlo. Questo include la configurazione di più user parameters per eseguire controlli personalizzati dell’agent, oltre a fornire a Zabbix agent le credenziali necessarie per connettersi al server MySQL come utente ”zbx_monitor” creato in precedenza.

Configura l’agent Zabbix

1. Accedi alla directory delle configurazioni aggiuntive dell’agent Zabbix.

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

> **Attention:**
> La directory delle configurazioni aggiuntive dell’agent Zabbix deve trovarsi nella stessa directory del file di configurazione dell’agent Zabbix; il percorso predefinito dipende dal sistema operativo e dal metodo di installazione di Zabbix.

Invece di definire tutti i parametri utente necessari per il monitoraggio del server MySQL nel file di configurazione dell’agent Zabbix, questi parametri verranno definiti in un file separato nella directory delle configurazioni aggiuntive.

2. Crea un file `template_db_mysql.conf` nella directory delle configurazioni aggiuntive dell’agent Zabbix.

```
vi template_db_mysql.conf
```

3. Copia il contenuto del file `template_db_mysql.conf` (situato nel repository Zabbix) nel file `template_db_mysql.conf` creato e salva.

4. Riavvia l’agent Zabbix per aggiornare la configurazione.

```
systemctl restart zabbix-agent
```

Dopo aver configurato i parametri utente dell’agent Zabbix, puoi procedere alla configurazione delle credenziali che consentiranno all’agent Zabbix di accedere al server MySQL.

5. Passare alla directory home dell’agent Zabbix (se non esiste nel sistema, è necessario crearla; predefinita: */var/lib/zabbix*).

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

6. Creare un file *.my.cnf* nella directory home dell'agent Zabbix.

```
vi .my.cnf
```

7. Copiare il seguente contenuto nel file *.my.cnf* (sostituire *<password>* con la password dell'utente "zbx_monitor").

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

Configura il frontend di Zabbix e verifica la tua configurazione

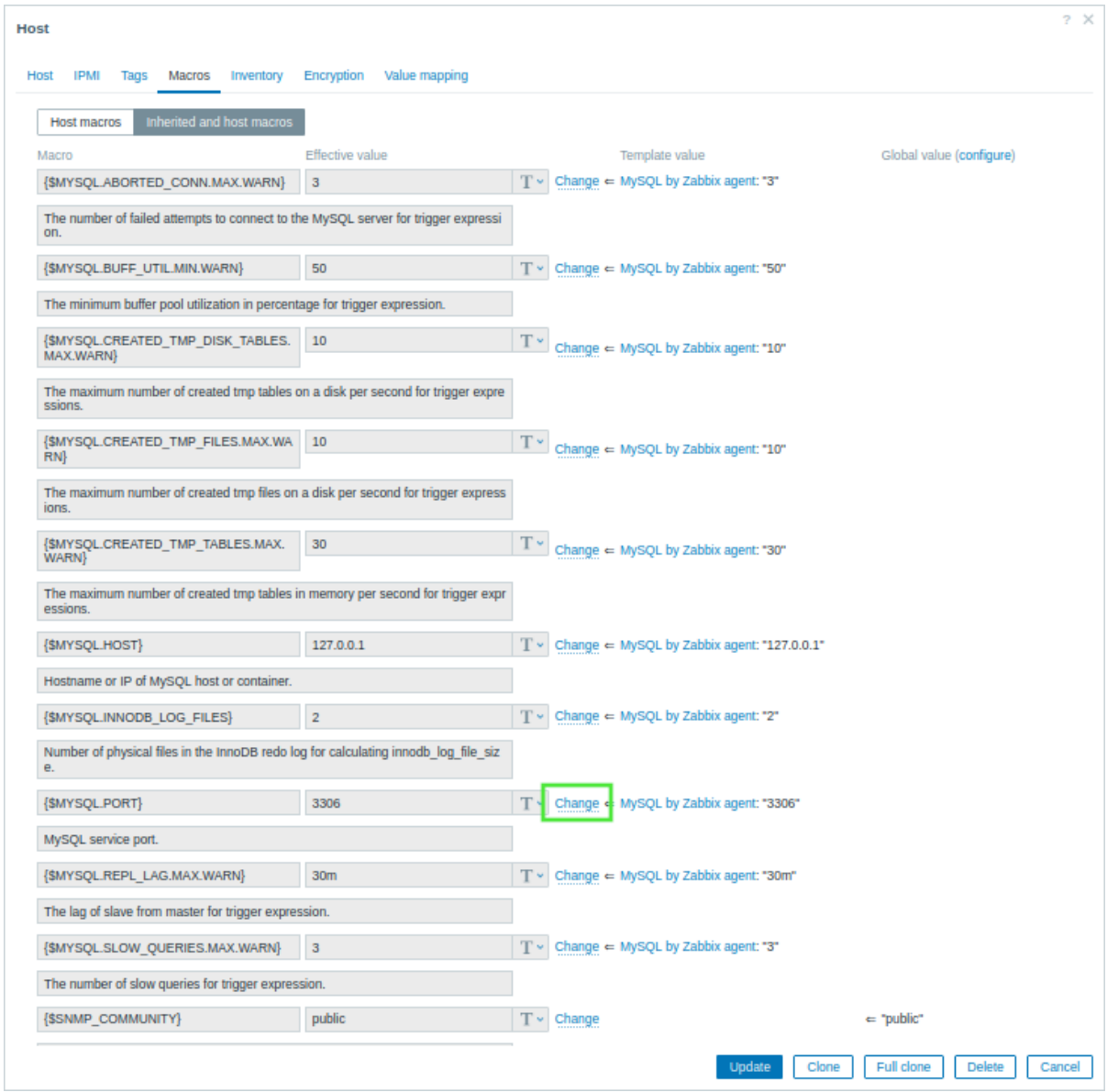
Per configurare il frontend di Zabbix, segui le istruzioni nella sezione *Configura il frontend di Zabbix* con le seguenti modifiche:

- Nel campo *Templates*, digita o seleziona il template "MySQL by Zabbix agent" che verrà collegato all'host.
- La configurazione di *Macros* non è necessaria.

Una volta configurato il frontend di Zabbix, puoi visualizzare le metriche raccolte e impostare gli avvisi sui problemi.

Per verificare la tua configurazione, segui le istruzioni nella sezione *Verifica la tua configurazione* con le seguenti modifiche:

- Nella sezione *Inherited and host macros* della configurazione dell'host del server MySQL, fai clic su *Change* accanto al valore della macro {$MYSQL.PORT} e imposta una porta diversa (ad esempio, "6033").



Monitorare MySQL con ODBC

Per monitorare un server MySQL con ODBC, è necessario scaricare e installare Zabbix server e Zabbix frontend.

Una volta installati correttamente i componenti Zabbix richiesti, è necessario creare un utente MySQL come descritto nella sezione *Create MySQL user*.

Dopo aver creato l'utente MySQL, è necessario configurare ODBC. Questo include l'installazione di una delle implementazioni open source dell'API ODBC più comunemente utilizzate - unixODBC - e di un driver unixODBC, oltre alla modifica del file di configurazione del driver ODBC.

Configurare ODBC

1. Installare unixODBC. Il modo consigliato per installare unixODBC è usare i repository di pacchetti predefiniti del sistema operativo Linux.

```
apt install unixodbc
```

2. Installare il driver database unixODBC di MariaDB. Anche se si dispone di un database MySQL, il driver unixODBC di MariaDB viene usato per motivi di compatibilità.

```
apt install odbc-mariadb
```

3. Verificare la posizione dei file di configurazione ODBC *odbcinst.ini* e *odbc.ini*.

```
odbcinst -j
```

Il risultato dell'esecuzione di questo comando dovrebbe essere simile al seguente.

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

4. Per configurare il driver ODBC per il monitoraggio di un database MySQL, è necessario il nome del driver, che si trova nel file *odbcinst.ini*. Nell'esempio seguente del file *odbcinst.ini*, il nome del driver è "MariaDB Unicode".

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

5. Copiare il seguente contenuto nel file *odbc.ini* (sostituire <*password*> con la password dell'utente "zbx_monitor"). Questa guida usa "127.0.0.1" (localhost) come indirizzo del server MySQL per monitorare un server MySQL installato sulla stessa macchina del driver ODBC. Si noti il nome dell'origine dati (DSN) "test", che sarà necessario quando si configura il frontend di Zabbix.

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

Configurare il frontend di Zabbix e testare la configurazione

Per configurare il frontend di Zabbix, segui le istruzioni nella sezione *Configurare il frontend di Zabbix* con le seguenti modifiche:

- Nel campo *Templates*, digita o seleziona il template "MySQL by ODBC" che verrà collegato all'host.
- La configurazione di *Interfaces* non è necessaria.
- Il valore della macro {$MYSQL.DSN} nella sezione *Inherited and host macros* della configurazione dell'host del server MySQL deve essere impostato sul nome DSN del file *odbc.ini*.

Una volta configurato il frontend di Zabbix, puoi visualizzare le metriche raccolte, configurare gli avvisi di problema e testare la configurazione.

Vedi anche

- Creazione di un item - come iniziare a monitorare metriche aggiuntive.
- Escalation dei problemi - come creare scenari di avviso multi-step (ad es., inviare prima un messaggio all'amministratore di sistema, poi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Monitoraggio ODBC - come configurare ODBC su altre distribuzioni Linux e come iniziare a monitorare metriche aggiuntive relative al database con ODBC.
- Template *MySQL by Zabbix agent* - informazioni aggiuntive sul template *MySQL by Zabbix agent*.
- Template *MySQL by Zabbix agent 2* - informazioni aggiuntive sul template *MySQL by Zabbix agent 2*.
- Template *MySQL by ODBC* - informazioni aggiuntive sul template *MySQL by ODBC*.

Monitorare VMware con Zabbix

Introduzione

Questa pagina illustra i passaggi necessari per avviare il monitoraggio di base di VMware.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base di VMware. Se stai cercando opzioni di personalizzazione avanzate o hai bisogno di una configurazione più complessa, consulta la sezione Virtual machine monitoring o la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server e Zabbix frontend seguendo le istruzioni per il proprio sistema operativo.

A seconda del proprio ambiente, alcuni passaggi di questa guida potrebbero differire leggermente. Questa guida si basa su un ambiente che esegue Ubuntu.

Si presume che VMware sia già configurato. Questa guida non copre la configurazione di VMware.

Configurare il server Zabbix

Per monitorare VMware, è necessario abilitare i processi Zabbix *vmware collector*. Per ulteriori informazioni su come viene eseguito il monitoraggio VMware, vedere Monitoraggio delle macchine virtuali.

1. Aprire il file di configurazione del server Zabbix.

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

2. Individuare e impostare il parametro `StartVMwareCollectors` nel file di configurazione del server Zabbix su 2 o un valore superiore (il valore predefinito è 0).

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

StartVMwareCollectors=2
```

3. Riavviare il server Zabbix.

```
systemctl restart zabbix-server
```

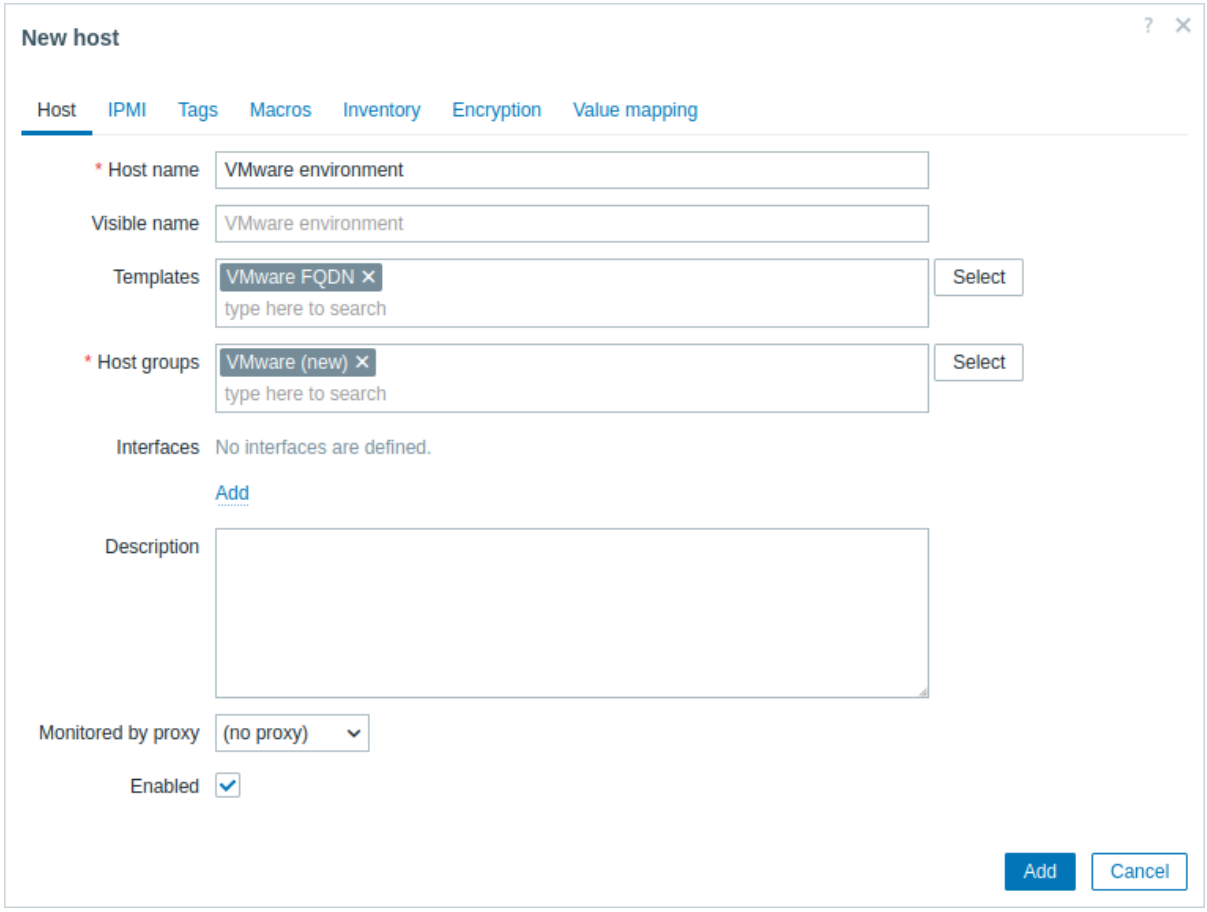
Dopo l'avvio dei processi *vmware collector*, procedere al passaggio successivo.

Configurare il frontend di Zabbix

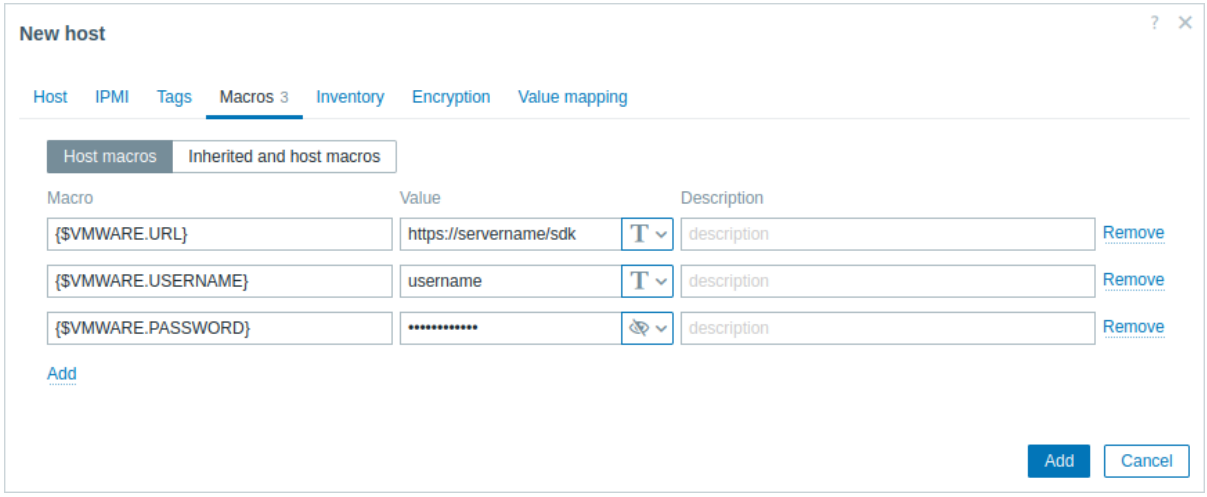
1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix:

- Nel campo *Host name*, inserisci un nome host (ad esempio, "VMware environment").
- Nel campo *Templates*, digita o seleziona il template "VMware FQDN" (o "VMware"). Per ulteriori informazioni su questi template, consulta Virtual machine monitoring.
- Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, un nuovo gruppo host "VMware").



- Nella scheda *Macros*, imposta le seguenti macro dell'host:
  - {$VMWARE.URL} - URL SDK del servizio VMware (vCenter o hypervisor ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - nome utente del servizio VMware
  - {$VMWARE.PASSWORD} - password dell'utente {$VMWARE.USERNAME} del servizio VMware



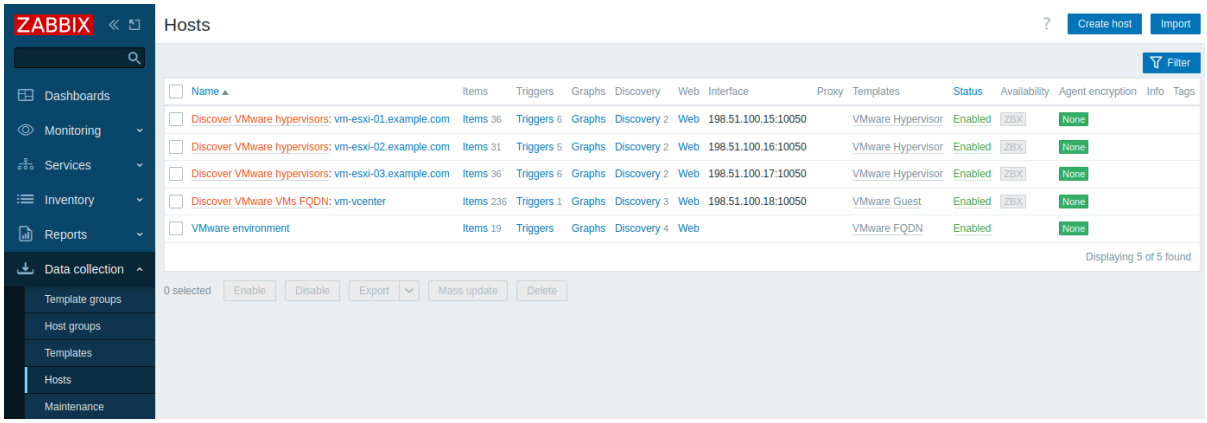
3. Fai clic sul pulsante *Add* per creare l'host. Questo host rappresenterà il tuo ambiente VMware.

Visualizzare le metriche raccolte

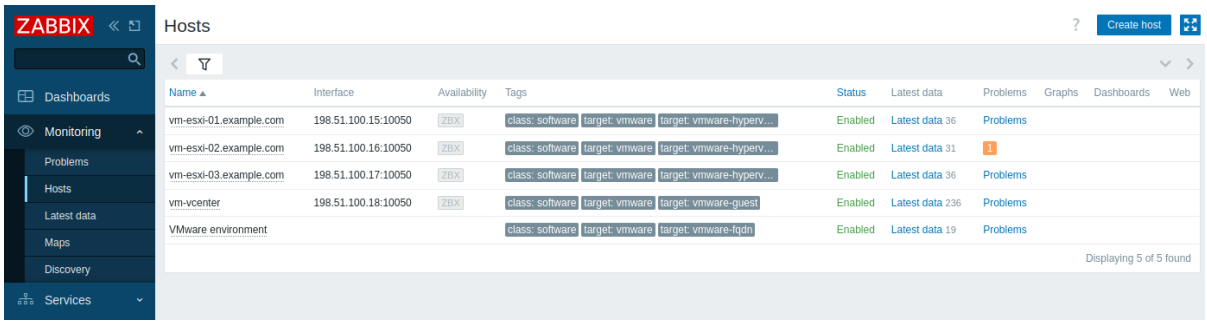
Congratulazioni! A questo punto, Zabbix sta già monitorando il tuo ambiente VMware.

A seconda della configurazione del tuo ambiente VMware, Zabbix può rilevare e quindi creare host per le entità rilevate. Tieni presente che il rilevamento e la creazione degli host possono anche essere eseguiti manualmente, se necessario.

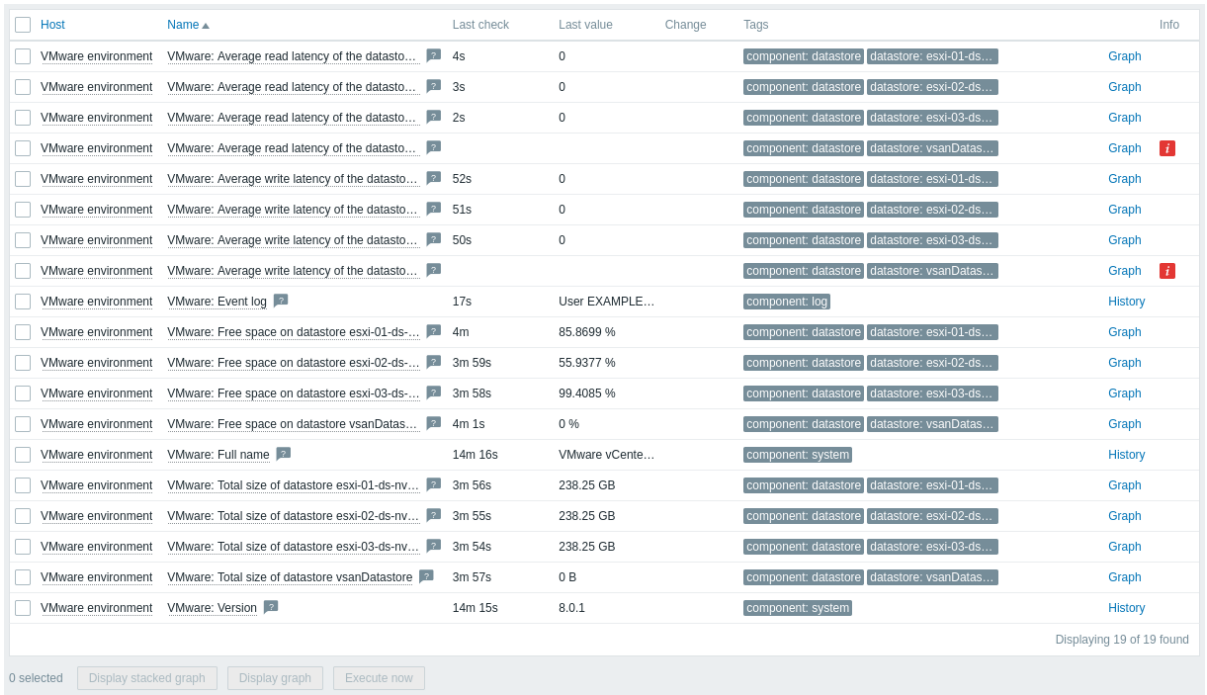
Per visualizzare gli host creati, vai alla sezione di menu *Raccolta dati > Host*.



Per visualizzare le metriche raccolte, vai alla sezione di menu *Monitoraggio > Host* e fai clic su *Dati più recenti* accanto all'host creato "VMware environment" o a uno degli host creati per le entità rilevate.



Questa azione aprirà un elenco di tutte le metriche più recenti raccolte dall'host selezionato.

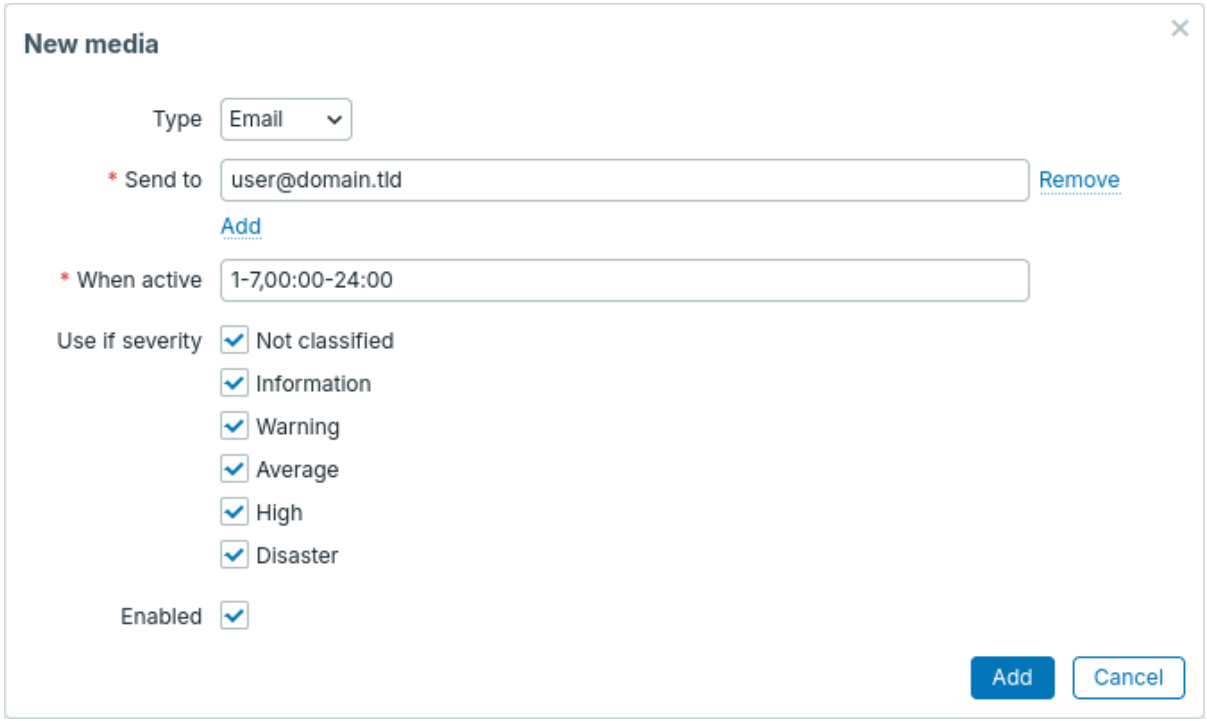


Tieni presente che alcuni item non hanno dati e si trovano nello stato *Non supportato*. Questo accade perché Zabbix non riesce a trovare contatori di prestazioni validi sul datastore specifico, poiché non è abilitato nell'ambiente VMware monitorato.

Configurare gli avvisi di problema

Zabbix può notificarti un problema con la tua infrastruttura usando vari metodi. Questa guida fornisce i passaggi di configurazione di base per l'invio di avvisi via email.

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Media* e aggiungi il tuo indirizzo email.



2. Segui la guida per Ricevere una notifica di problema.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Vedi anche

- Creazione di un item - come iniziare a monitorare metriche aggiuntive.
- Escalation dei problemi - come creare scenari di avviso multi-step (ad esempio, inviare prima un messaggio all'amministratore di sistema, poi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Monitoraggio delle macchine virtuali - informazioni aggiuntive sul monitoraggio VMware (processo di raccolta dati, opzioni di configurazione del server, indicazioni per la risoluzione dei problemi, ecc.).
- Chiavi degli item per il monitoraggio VMware - un elenco completo delle metriche VMware che possono essere monitorate usando Zabbix.
- Template *VMware* - informazioni aggiuntive sul template *VMware*.
- Template *VMware FQDN* - informazioni aggiuntive sul template *VMware FQDN*.

Monitorare il traffico di rete con Zabbix

Introduzione

Questa pagina ti guida attraverso i passaggi necessari per iniziare il monitoraggio di base del traffico di rete con Zabbix.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base del traffico di rete. Se stai cercando opzioni di personalizzazione avanzate o hai bisogno di una configurazione più complessa, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server, Zabbix frontend e Zabbix agent seguendo le istruzioni per il proprio sistema operativo.
Si noti che è necessario installare Zabbix agent sulla macchina che richiede il monitoraggio del traffico.
Può trattarsi dello stesso host in cui è installato Zabbix server oppure di un host diverso.

Questa guida fornirà le istruzioni per configurare il monitoraggio del traffico di rete dell'interfaccia *eth0* su una macchina separata denominata *Remote host*.

Configurare Zabbix per il monitoraggio

L'agent Zabbix può raccogliere metriche in modalità attiva o passiva (simultaneamente). Per ulteriori informazioni, vedere Passive and active agent checks. In questa guida verrà descritto il monitoraggio tramite controlli passivi.

Configurazione dell'agent Zabbix

1. Aprire il file di configurazione dell'agent sulla macchina in cui è installato l'agent:

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

2. Aggiungere l'indirizzo IP o il nome DNS del server Zabbix al parametro *Server*. Ad esempio:

```
Server=192.0.2.22
```

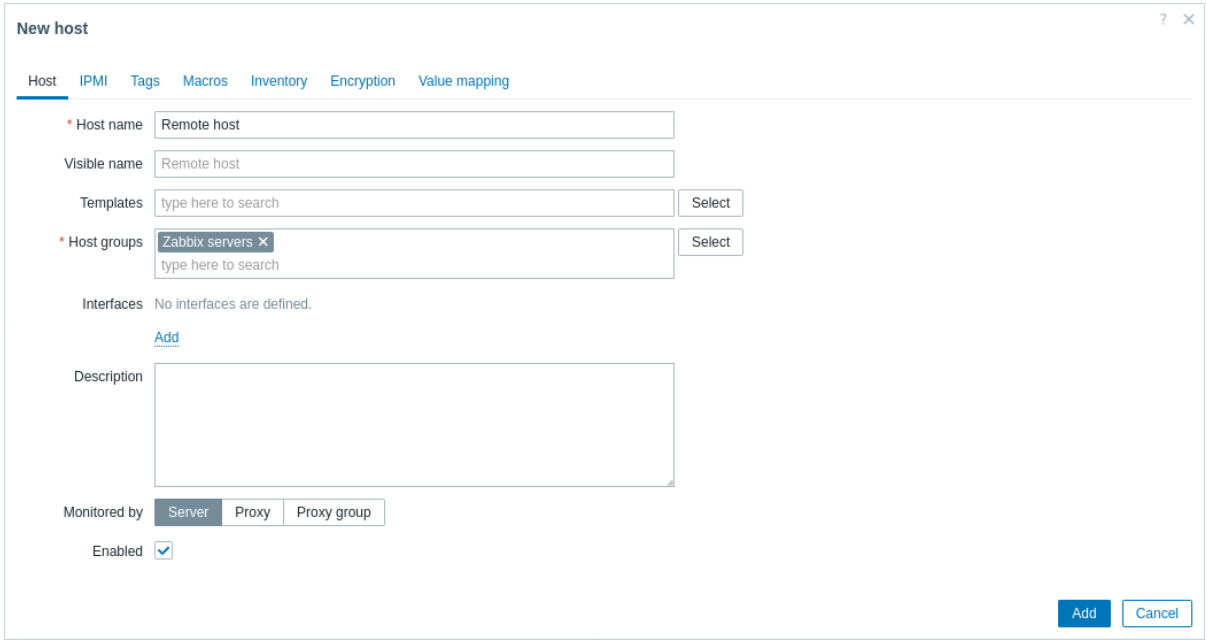
3. Riavviare l'agent Zabbix:

```
systemctl restart zabbix-agent
```

Frontend di Zabbix

1. Accedere al frontend di Zabbix.

2. Creare un host nell'interfaccia web di Zabbix, specificando l'indirizzo IP o il nome DNS della macchina su cui è installato l'agent.

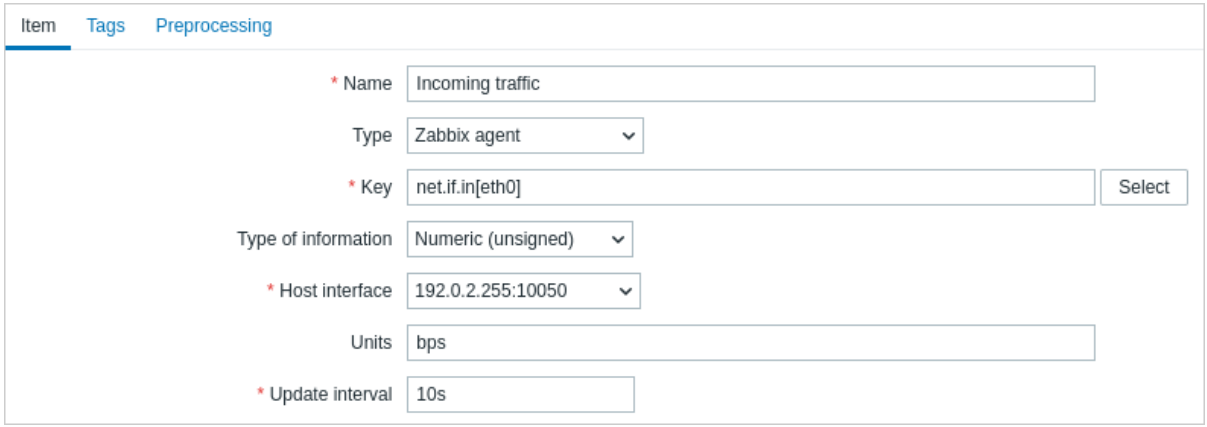


Creare gli item

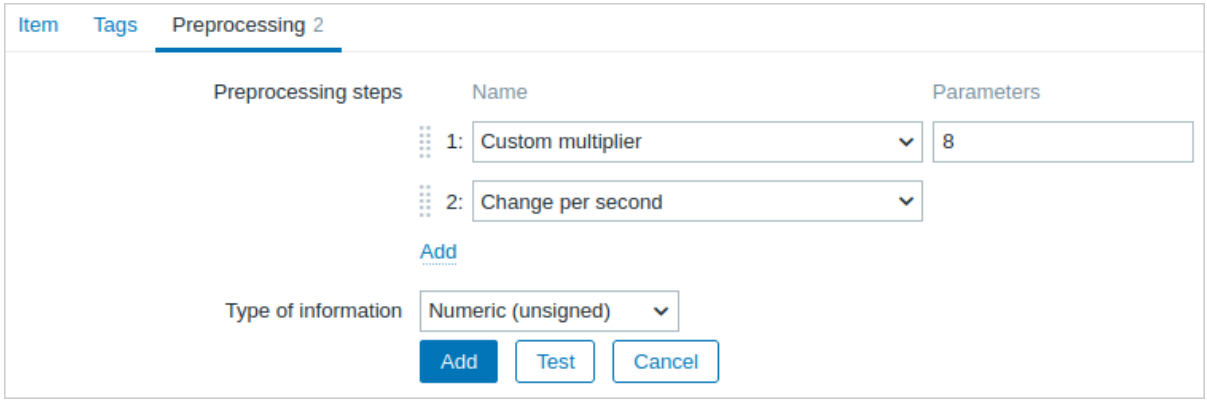
Segui le istruzioni su creazione di un item per aggiungere gli item per il monitoraggio del traffico, ovvero:

- Traffico in ingresso
- Traffico in uscita
- Traffico totale

Una configurazione semplice per l'item di monitoraggio del traffico in ingresso sarebbe la seguente:



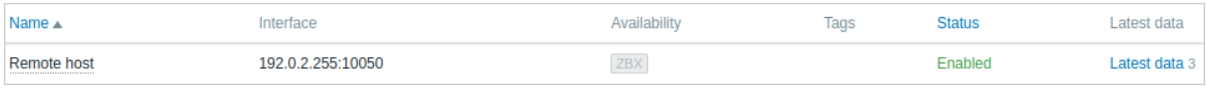
Per rendere i dati raccolti adatti all'uso pratico, puoi impostare alcuni passaggi di preprocessing durante la creazione degli item. Nel caso specifico, questi possono essere la moltiplicazione per 8 (per convertire i byte in bit) e la presentazione come variazione al secondo.



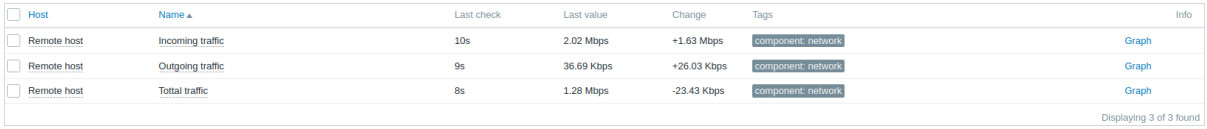
Visualizzazione dei dati raccolti

Congratulazioni! A questo punto, Zabbix sta già monitorando il traffico di rete.

Per visualizzare le metriche raccolte, aprire la sezione di menu *Monitoring > Hosts* e fare clic su *Latest data* nella riga dell'host.

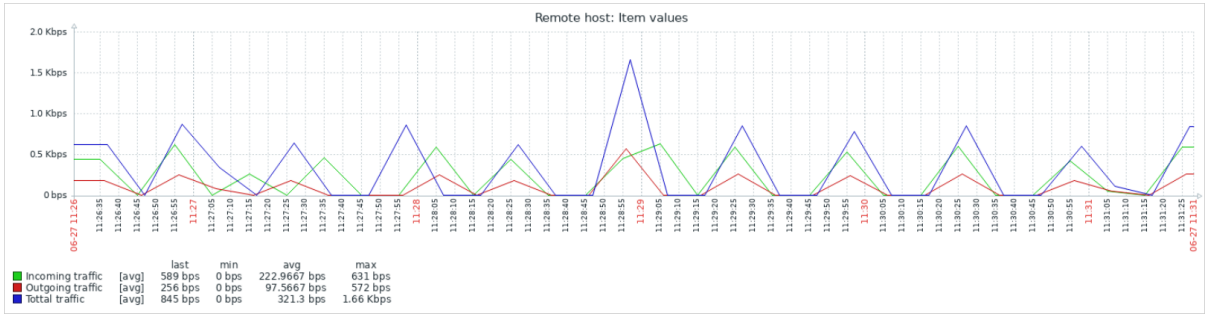


Verranno visualizzati i dati sul traffico.



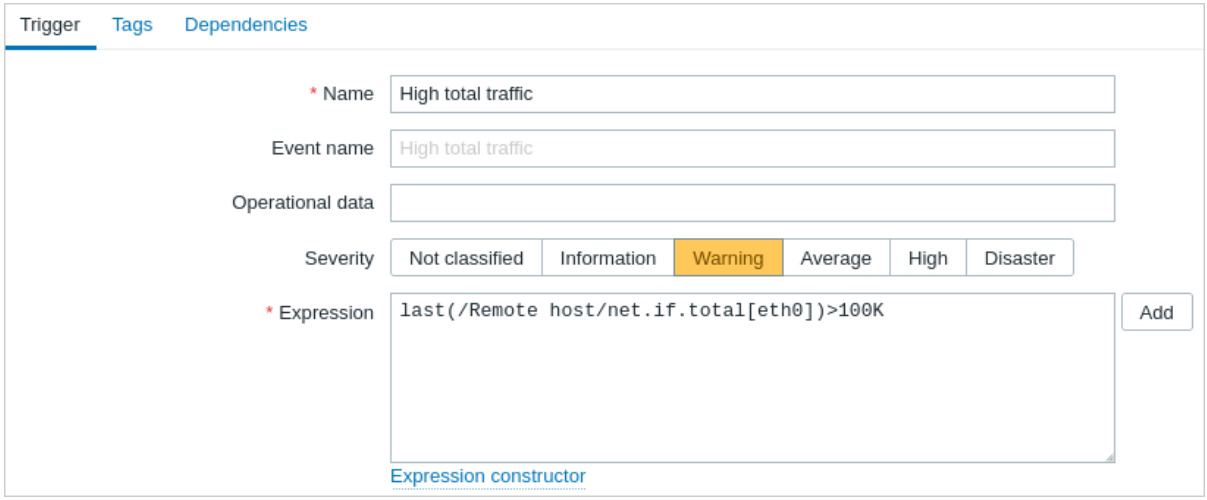
Visualizzare i grafici

I dati raccolti possono essere visualizzati come grafici. Per visualizzarli, nella sezione *Dati recenti* fai clic su *Grafico* nella riga dell'item oppure seleziona gli item richiesti e fai clic su *Mostra grafico* in basso.

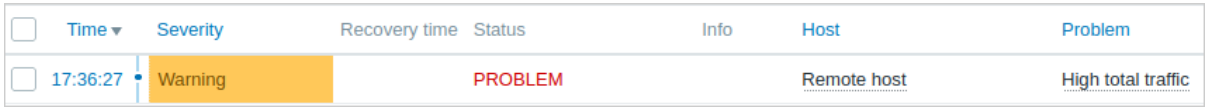


Configurare i trigger

Puoi impostare i trigger per rilevare traffico di rete anomalo. Consulta le istruzioni su come configurare un trigger e aggiungi il trigger che segnala che il traffico totale è troppo elevato, ad esempio:



Ora fai superare al traffico la soglia che hai impostato nell'espressione del trigger e vai in *Monitoring > Problems* per verificare che il problema sia elencato lì.



Time ▾	Severity	Recovery time	Status	Info	Host	Problem
17:36:27	Warning		PROBLEM		Remote host	High total traffic

Configurare gli avvisi di problema

Esistono diversi modi per ricevere notifiche sul problema. L'email è il metodo più diffuso; segui le istruzioni per configurare una notifica di problema via email. Puoi anche scegliere altri tipi di media da utilizzare per la consegna delle notifiche.

Vedi anche:

- Escalation dei problemi - come creare scenari di avviso multi-step (ad esempio, inviare prima un messaggio all'amministratore di sistema, quindi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Conferma del problema - come indicare che il problema è noto, aggiungere commenti sulla sua risoluzione, sopprimere o chiudere il problema.
- Monitora Linux con agent Zabbix - come avviare il monitoraggio di base degli item più importanti collegando un template preconfigurato.

Monitorare il traffico di rete usando controlli attivi

Introduzione

Questa pagina illustra i passaggi necessari per iniziare il monitoraggio di base del traffico di rete con Zabbix utilizzando i controlli attivi.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base del traffico di rete utilizzando i controlli attivi. Se stai cercando opzioni di personalizzazione approfondite o hai bisogno di una configurazione più avanzata, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server, Zabbix frontend e Zabbix agent seguendo le istruzioni per il proprio sistema operativo. Si noti che è necessario installare Zabbix agent sulla macchina che richiede il monitoraggio del traffico. Può trattarsi dello stesso host su cui è installato Zabbix server oppure di un host diverso.

Questa guida fornirà le istruzioni per configurare il monitoraggio del traffico di rete dell'interfaccia *eth0* su una macchina separata denominata *Remote host*.

Configurare Zabbix per il monitoraggio

L'agent Zabbix può raccogliere metriche in modalità attiva o passiva (simultaneamente). Per ulteriori informazioni, vedere Passive and active agent checks. In questa guida verrà descritto il monitoraggio tramite **active checks**.

Configurazione di Zabbix agent

1. Aprire il file di configurazione dell'agent sulla macchina in cui è installato l'agent.

Se si utilizza Zabbix agent:

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

Se si utilizza Zabbix agent 2:

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

2. Aggiungere l'indirizzo IP o il nome DNS (e la porta facoltativa) del proprio Zabbix server al parametro ServerActive. Ad esempio:

```
ServerActive=192.0.2.1:10051
```

Zabbix agent utilizzerà questo indirizzo per connettersi alla porta trapper del Zabbix server (predefinita: 10051) e richiedere i dati di configurazione per i controlli attivi.

3. Definire il parametro Hostname, che deve corrispondere al nome dell'host che verrà definito nel frontend di Zabbix. Nel nostro esempio:

```
Hostname=Remote host
```

Il valore di Hostname deve corrispondere perché, per i controlli attivi, Zabbix agent lo utilizza per recuperare dal server la configurazione corretta dell'host. In particolare, l'agent avvia una connessione al server e si identifica utilizzando il valore di Hostname. Il server fornisce quindi la configurazione di monitoraggio per quell'host. Se questi valori differiscono, l'agent non riceverà la configurazione appropriata, causando metriche mancanti o problemi di monitoraggio.

4. Riavviare Zabbix agent.

Se si utilizza Zabbix agent:

```
systemctl restart zabbix-agent
```

Se si utilizza Zabbix agent 2:

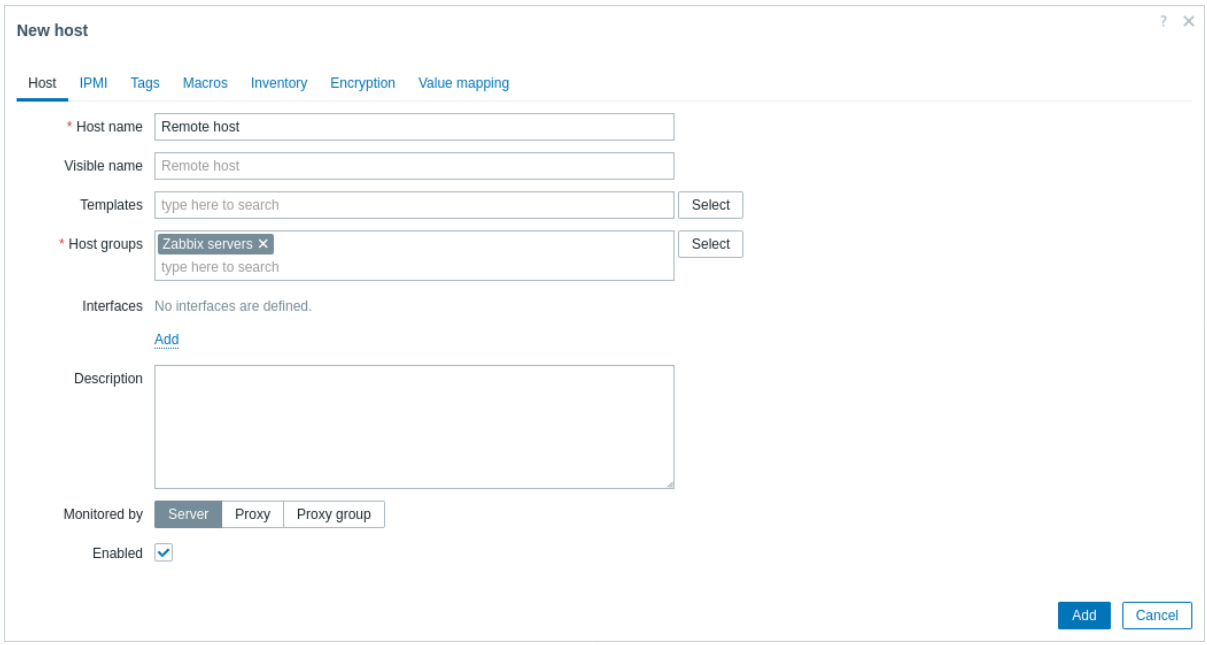
```
systemctl restart zabbix-agent2
```

frontend di Zabbix

1. Accedere al frontend di Zabbix.

2. Creare un host nell'interfaccia web di Zabbix.

- Nel campo *Host name*, inserire un nome host (ad esempio, "Remote host") che corrisponda al valore del parametro Hostname definito in precedenza nel file di configurazione dell'agent.
- Nel campo *Host groups*, digitare o selezionare un gruppo di host (ad esempio, "Zabbix servers").
- Lasciare *Interfaces* non definito; un'interfaccia non è necessaria per i controlli attivi perché l'agent avvia la connessione al server invece di attendere una connessione dal server.



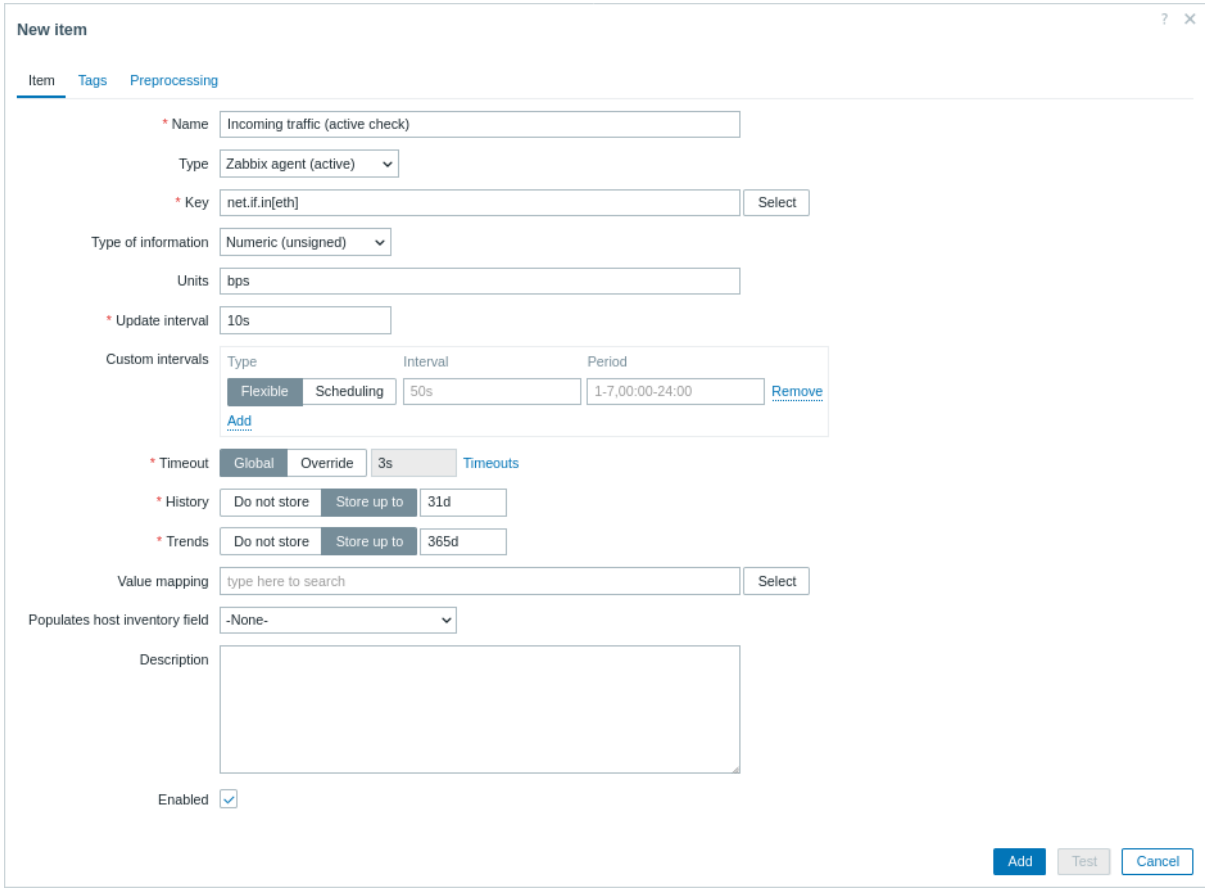
3. Fare clic su *Add* per aggiungere l'host. Questo host rappresenterà la macchina Linux monitorata.

Creare item

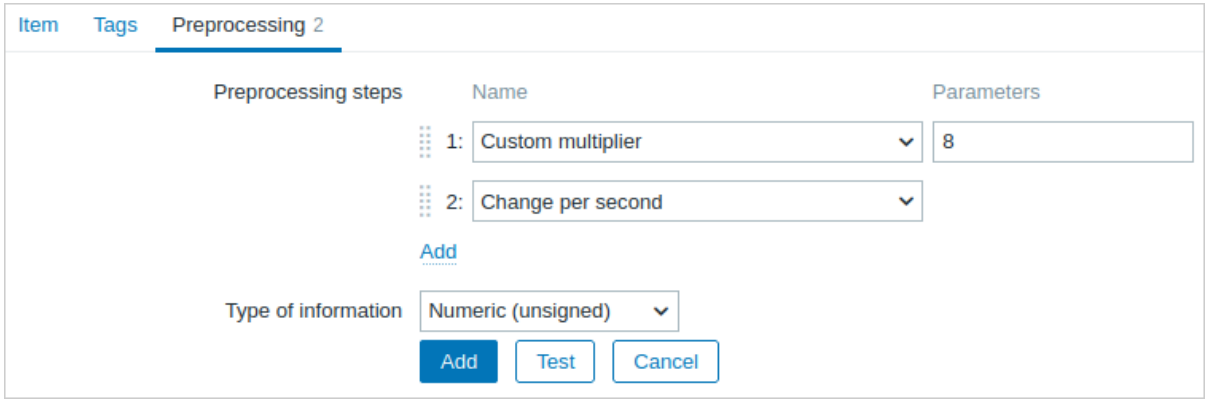
Segui le istruzioni su creazione di un item per aggiungere gli item per il monitoraggio del traffico, vale a dire:

- Traffico in ingresso
- Traffico in uscita
- Traffico totale

Una configurazione semplice per l'item di monitoraggio del traffico in ingresso con un controllo attivo sarebbe la seguente:



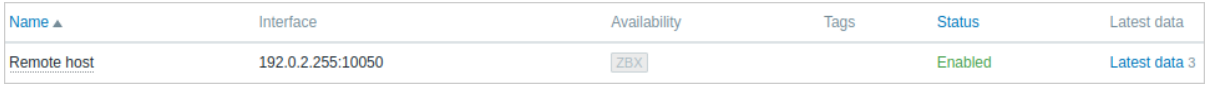
Per rendere i dati raccolti adatti all'uso pratico, puoi impostare alcuni passaggi di preprocessing quando crei gli item. Nel caso in esame, questi possono essere la moltiplicazione per 8 (per convertire i byte in bit) e la presentazione come variazione al secondo.



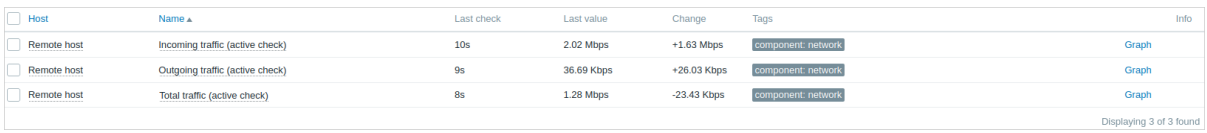
Visualizzare i dati raccolti

Congratulazioni! A questo punto, Zabbix sta già monitorando il traffico di rete.

Per visualizzare le metriche raccolte, apri la sezione di menu *Monitoring > Hosts* e fai clic su *Latest data* nella riga dell'host.



Vedrai i dati del traffico.



Vedi anche:

- Visualizza i grafici - come visualizzare i dati raccolti come grafici.
- Configura i trigger - come impostare i trigger per rilevare traffico di rete anomalo.
- Configura gli avvisi sui problemi - come impostare le notifiche relative a situazioni di problema.

- Escalation dei problemi - come creare scenari di avviso in più fasi (ad esempio, inviare prima un messaggio all'amministratore di sistema, poi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Conferma del problema - come indicare che il problema è noto, aggiungere commenti sulla sua risoluzione, sopprimere o chiudere il problema.
- Monitora Linux con Zabbix agent - come avviare il monitoraggio di base degli item più importanti collegando un template preconfigurato.
- Installazione dai pacchetti - come installare i componenti Zabbix utilizzando i pacchetti ufficiali RPM e DEB per varie distribuzioni Linux, garantendo l'accesso alle funzionalità più recenti e alle correzioni dei bug.

Monitorare siti web con gli item Browser

Introduzione

Questa pagina illustra i passaggi necessari per avviare il monitoraggio di base dei siti web con gli item Browser.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base dei siti web con gli item Browser. Se stai cercando opzioni di personalizzazione avanzate o hai bisogno di una configurazione più complessa, consulta la pagina Browser items o la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server e Zabbix frontend seguendo le istruzioni per il proprio sistema operativo.

A seconda del proprio ambiente, alcuni passaggi di questa guida potrebbero differire leggermente. Questa guida si basa su un ambiente che esegue Ubuntu.

Configurare WebDriver

Gli item browser richiedono un framework di automazione (Selenium Server oppure un semplice WebDriver, ad esempio ChromeDriver) come endpoint di test web che controlla e interagisce con un browser, eseguendo comandi di test come fare clic sui pulsanti o inserire testo. Come esempio, questa guida utilizzerà Selenium Server con Chrome in un container Docker.

Si presume che Docker sia già configurato. Questa guida non copre la configurazione di Docker. Per le istruzioni di installazione, vedere Install Docker Engine on Ubuntu.

1. Avviare Selenium Server con Chrome in un container Docker con le seguenti opzioni:

- **docker run --name browser** - avvia un nuovo container Docker denominato "browser";
- **-p 4444:4444** - mappa la porta 4444 sulla macchina host alla porta 4444 nel container (questa è la porta usata da Selenium Server per accettare i comandi);
- **-p 7900:7900** - mappa la porta 7900 sulla macchina host alla porta 7900 nel container (questa è la porta usata dal server Virtual Network Computing (VNC), consentendo di visualizzare da remoto la GUI del browser; richiede un client VNC);
- **--shm-size="2g"** - assegna 2 GB di memoria condivisa al container (questo è importante per il corretto funzionamento di Chrome, poiché può richiedere una quantità significativa di memoria condivisa per evitare arresti anomali);
- **-d** - esegue il container in modalità detached, cioè in background;
- **selenium/standalone-chrome:latest** - specifica l'immagine Docker da usare; in questo caso, l'ultima versione di Selenium Server con Chrome.

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

2. Verificare che il container Docker `browser` sia in esecuzione e accessibile.

- Recuperare l'indirizzo IP del container (in questo esempio, 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
#####     ...
```

- Testare la connessione al container con Ncat:

```
nc -zv 192.0.2.1 4444

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

- Recuperare il contenuto della pagina web da Selenium Server con curl:

```
curl -L 192.0.2.1:4444

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

Note:
Per la risoluzione dei problemi, fare riferimento alla documentazione di Docker.

Configurare il server Zabbix

Gli item browser vengono eseguiti ed elaborati dai processi *browser poller* di Zabbix, che devono essere abilitati modificando il parametro di configurazione del server StartBrowserPollers. Inoltre, il parametro WebDriverURL deve specificare l'endpoint di test web configurato in precedenza.

Per impostazione predefinita, il parametro StartBrowserPollers è impostato su 1; pertanto, è necessario specificare solo l'endpoint di test web.

1. Aprire il file di configurazione del server Zabbix:

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

2. Individuare e impostare il parametro WebDriverURL nel file di configurazione del server Zabbix:

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

WebDriverURL=192.0.2.1:4444
```

3. Riavviare il server Zabbix.

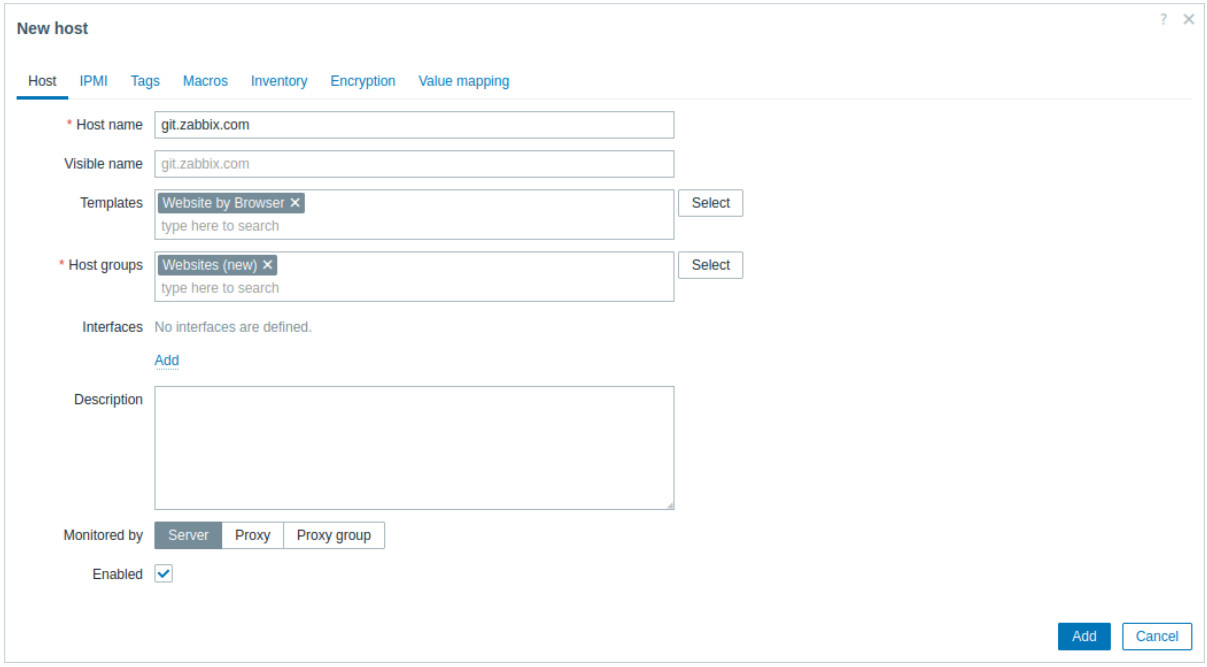
```
systemctl restart zabbix-server
```

Configurare il frontend di Zabbix

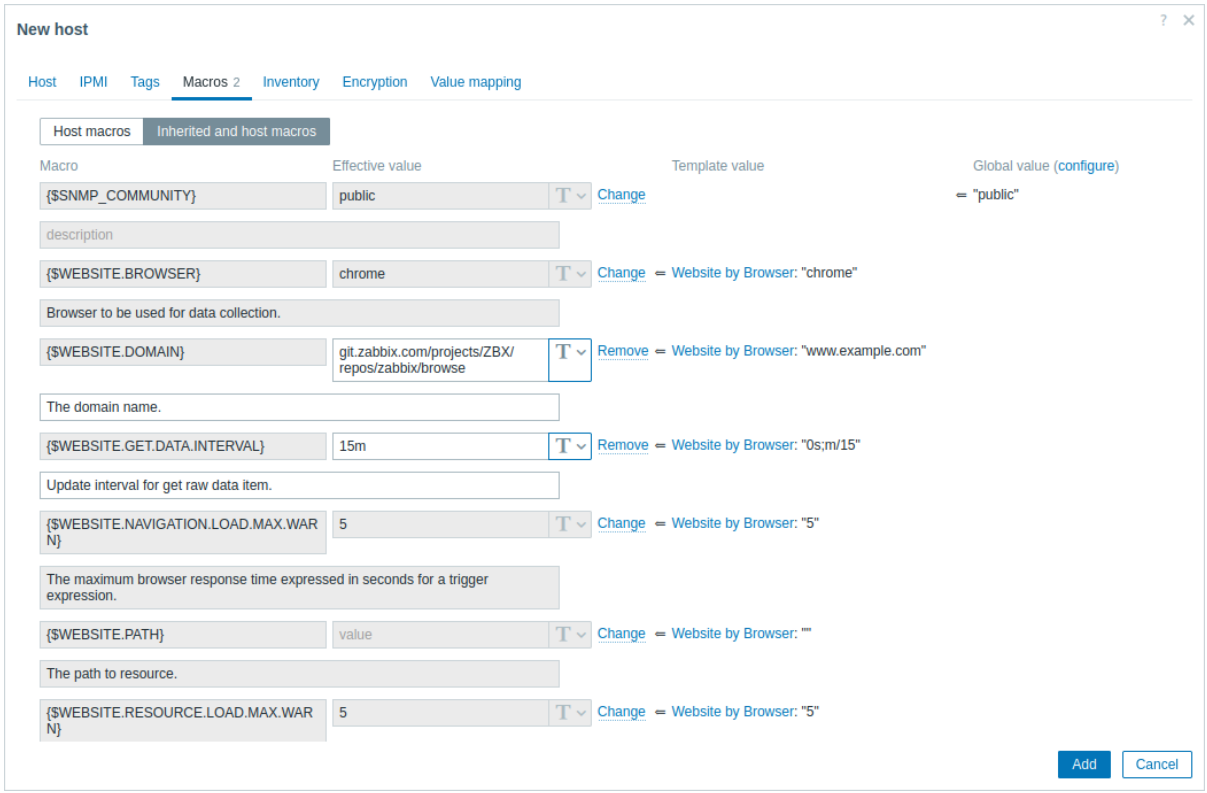
1. Accedi al frontend di Zabbix.

2. Crea un host nell'interfaccia web di Zabbix:

- Nel campo *Host name*, inserisci un nome host (ad esempio, "git.zabbix.com").
- Nel campo *Templates*, digita o seleziona il template "Website by Browser". Per ulteriori informazioni su questo template, consulta Website by Browser.
- Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, un nuovo gruppo host "Websites").



- Nella scheda *Macros*, passa a *Inherited and host macros*, cerca le seguenti macro e fai clic su *Change* accanto al valore della macro per aggiornarlo:
  - {$WEBSITE.DOMAIN} - Nome di dominio (ad esempio, git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Intervallo di aggiornamento dei dati dell'item (ad esempio, 15m)

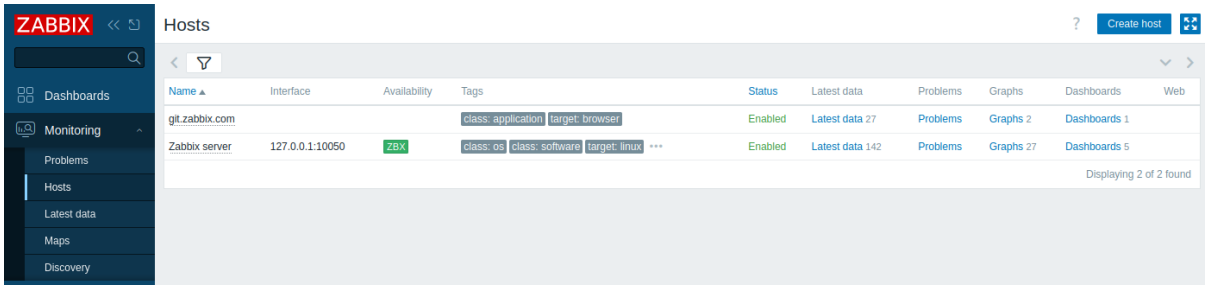


3. Fai clic sul pulsante *Add* per creare l'host. Questo host rappresenterà il sito web che desideri monitorare.

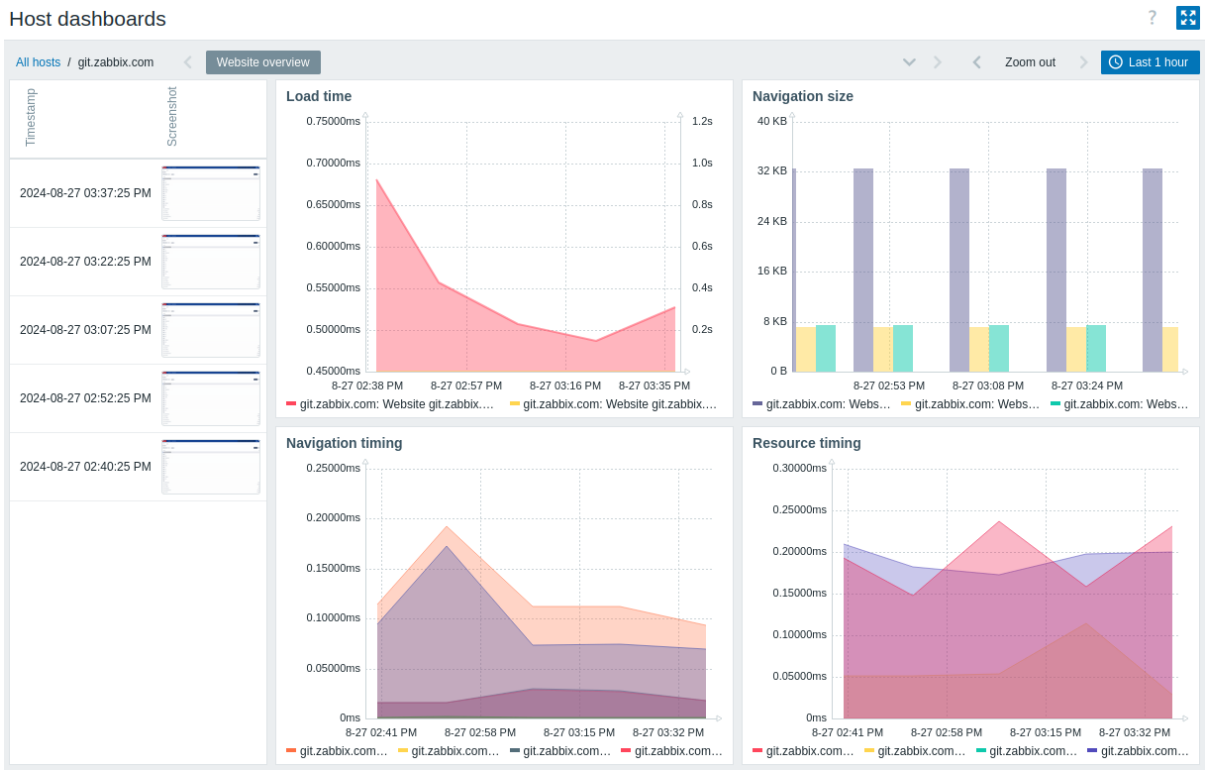
Visualizzare le metriche raccolte

Congratulazioni! A questo punto, Zabbix sta già monitorando il sito web che hai specificato.

Per visualizzare le metriche raccolte, vai alla sezione di menu *Monitoring > Hosts* e fai clic su *Dashboards* accanto all'host.



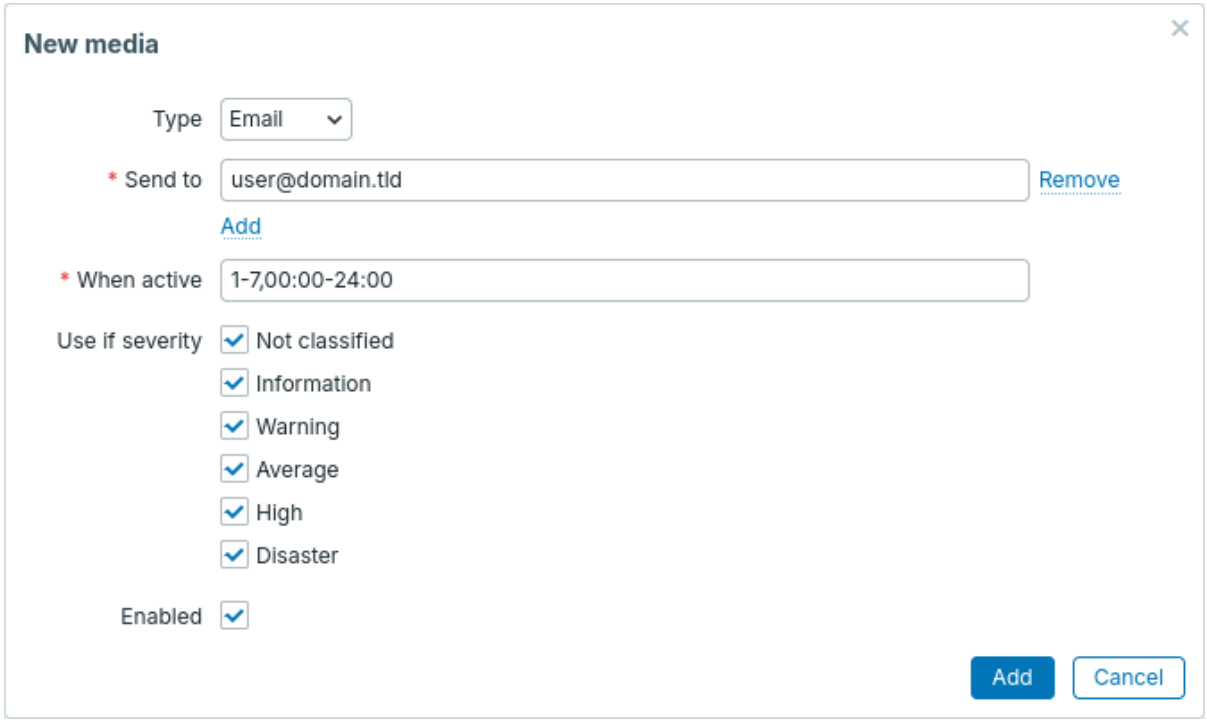
Questa azione ti porterà alla dashboard dell'host (configurata a livello di template) con le metriche più importanti raccolte dal sito web.



Configurare gli avvisi di problema

Zabbix può notificarti un problema con la tua infrastruttura usando diversi metodi. Questa guida fornisce i passaggi di configurazione di base per inviare avvisi via email.

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Media* e aggiungi la tua email.



2. Segui la guida per Ricevere una notifica di problema.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Test della configurazione

Per testare la configurazione, possiamo simulare un problema reale aggiornando la configurazione dell'host nel frontend di Zabbix.

1. Apri la configurazione dell'host del tuo sito web in Zabbix.

2. Passa alla scheda *Macros* e seleziona *Inherited and host macros*.

3. Fai clic su *Change* accanto, ad esempio, al valore della macro {$WEBSITE.DOMAIN} configurato in precedenza e imposta un nome di dominio errato (ad esempio, /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

4. Fai clic su *Update* per aggiornare la configurazione dell'host.

5. Dopo qualche istante, Zabbix rileverà il problema "Failed to get JSON of the requested website", perché non sarà in grado di connettersi al sito web specificato. Il problema apparirà in *Monitoring > Problems*.



Se gli alert sono configurati, riceverai anche la notifica del problema.

6. Riporta il valore della macro al valore precedente per risolvere il problema e continuare a monitorare il sito web.

Vedi anche

- Creazione di un item - come iniziare a monitorare metriche aggiuntive.
- Escalation dei problemi - come creare scenari di avviso multi-step (ad esempio, inviare prima un messaggio all'amministratore di sistema, quindi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Item del browser - come configurare gli item del browser.
- template Website by Browser - informazioni aggiuntive sul template *Website by Browser*.

Monitorare i certificati dei siti web con Zabbix agent 2 (passive)

Introduzione

Questa guida fornisce una panoramica completa su come configurare e monitorare i certificati SSL/TLS utilizzando la chiave `web.certificate.get` in Zabbix agent 2. È progettata per semplificare il monitoraggio dei certificati per uno o più siti web, consentendo agli amministratori di identificare rapidamente potenziali problemi come certificati scaduti o non validi.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e contiene il set minimo di passaggi necessari per abilitare il monitoraggio di base dei certificati dei siti web. Se stai cercando opzioni di personalizzazione avanzate o hai bisogno di una configurazione più complessa, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server, Zabbix frontend e Zabbix agent 2 seguendo le istruzioni per il proprio sistema operativo. Questo tutorial presuppone che sia Zabbix server sia agent siano installati sulla stessa macchina; pertanto, `127.0.0.1` viene utilizzato nella configurazione.

Configurare Zabbix agent 2

1. Aprire il file di configurazione dell'agent di Zabbix:

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

2. Impostare il parametro Server su 127.0.0.1, poiché l'agent e il server sono in esecuzione sulla stessa macchina:

```
Server=127.0.0.1
```

3. Salvare il file e riavviare il servizio Zabbix agent 2:

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

4. Dopo aver configurato Zabbix agent 2, verificarne la disponibilità con:

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

Zabbix agent 2 include il plugin WebCertificate per impostazione predefinita, pertanto non sono necessarie installazioni o configurazioni separate.

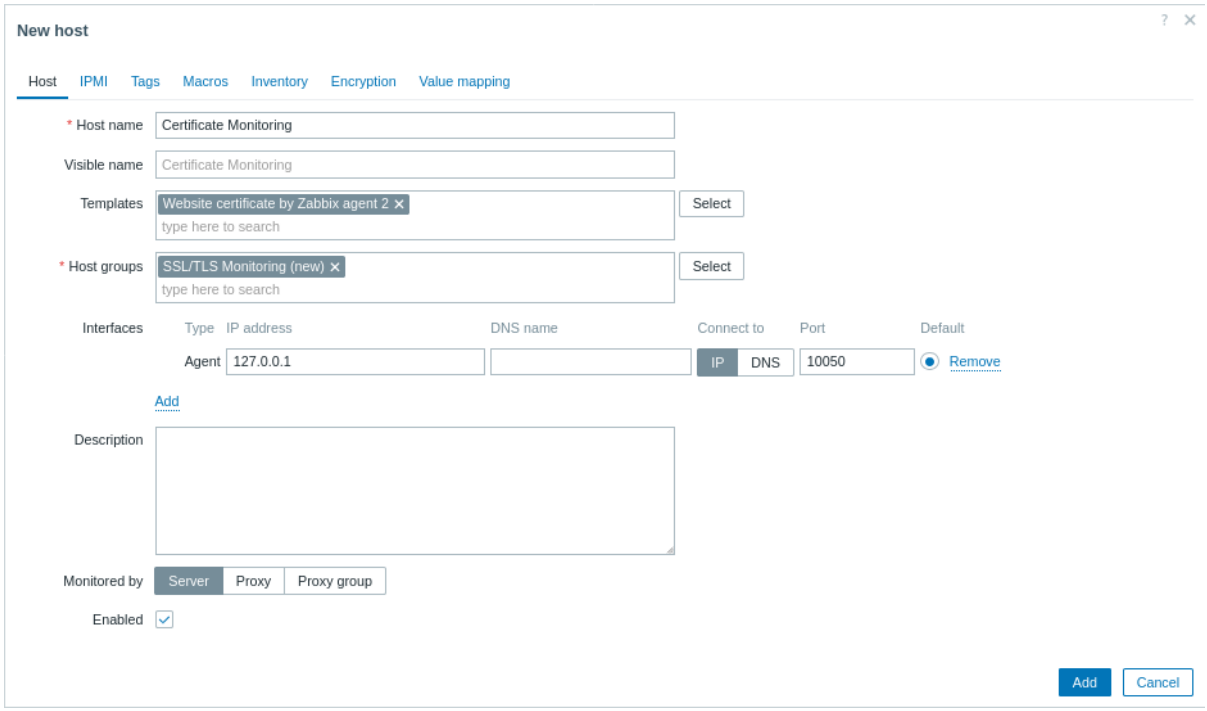
Configurare il frontend di Zabbix

1. Accedere al frontend di Zabbix.

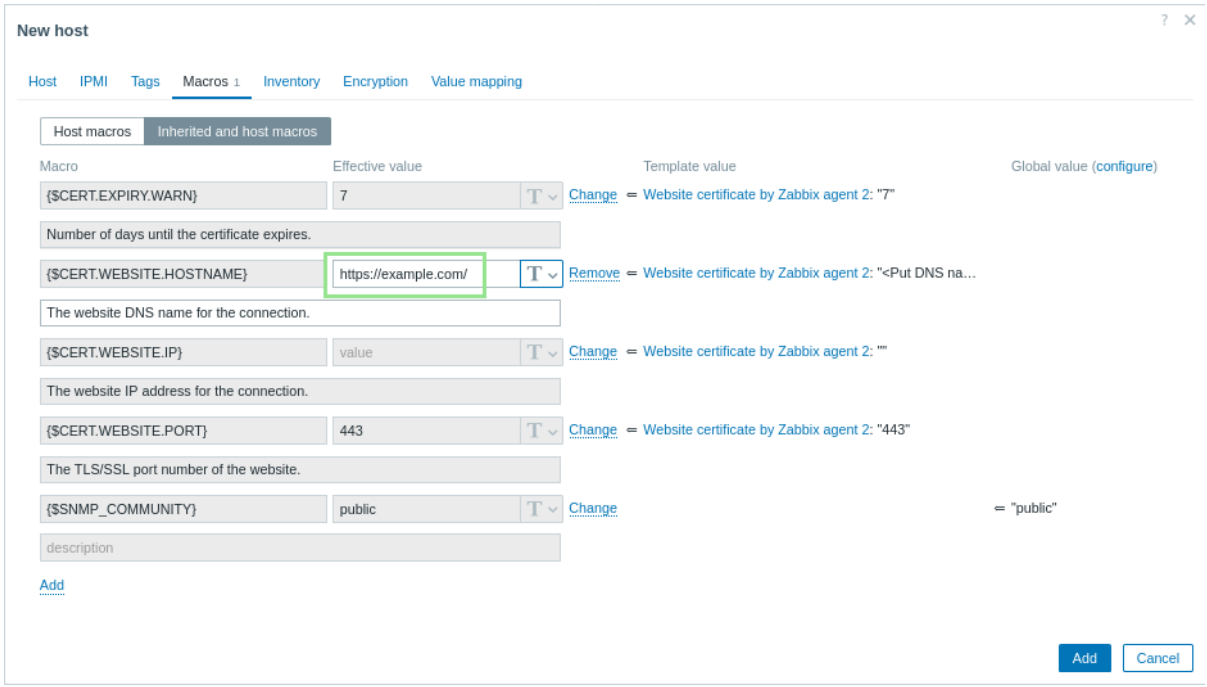
2. Andare su *Monitoring* > *Hosts*.

3. Fare clic su un host esistente in cui si desidera monitorare i certificati del sito web, oppure creare un host se necessario:

- Nel campo *Host name*, inserire un nome host (ad esempio, "Certificate Monitoring").
- Nel campo *Templates*, digitare o selezionare il template "Website certificate by Zabbix agent 2" che verrà collegato all'host.
- Nel campo *Host groups*, digitare o selezionare un gruppo host (ad esempio, "SSL/TLS Monitoring").
- Nel campo *Interfaces*, aggiungere un'interfaccia di tipo "Agent" e specificare un indirizzo IP. Per questo esempio, viene utilizzato "127.0.0.1".



- Nella scheda *Macros*, passare a *Inherited and host macros*, cercare le seguenti macro e fare clic su *Change* accanto al valore della macro per aggiornarlo:
  - `{$CERT.WEBSITE.HOSTNAME}` - inserire come valore il nome DNS del sito web desiderato.



4. Fare clic su *Add* per aggiungere l'host.

Per monitorare più siti web, è possibile specificare un elenco separato da virgole dei rispettivi hostname nella macro `{$CERT.WEBSITE.HOSTNAME}`. Facoltativamente, è anche possibile fornire un elenco separato da virgole delle porte nella macro `{$CERT.WEBSITE.PORT}`, dove ogni porta corrisponde al rispettivo hostname nell'ordine. Ad esempio, per monitorare example.com ed example.org sulla porta 8443:

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

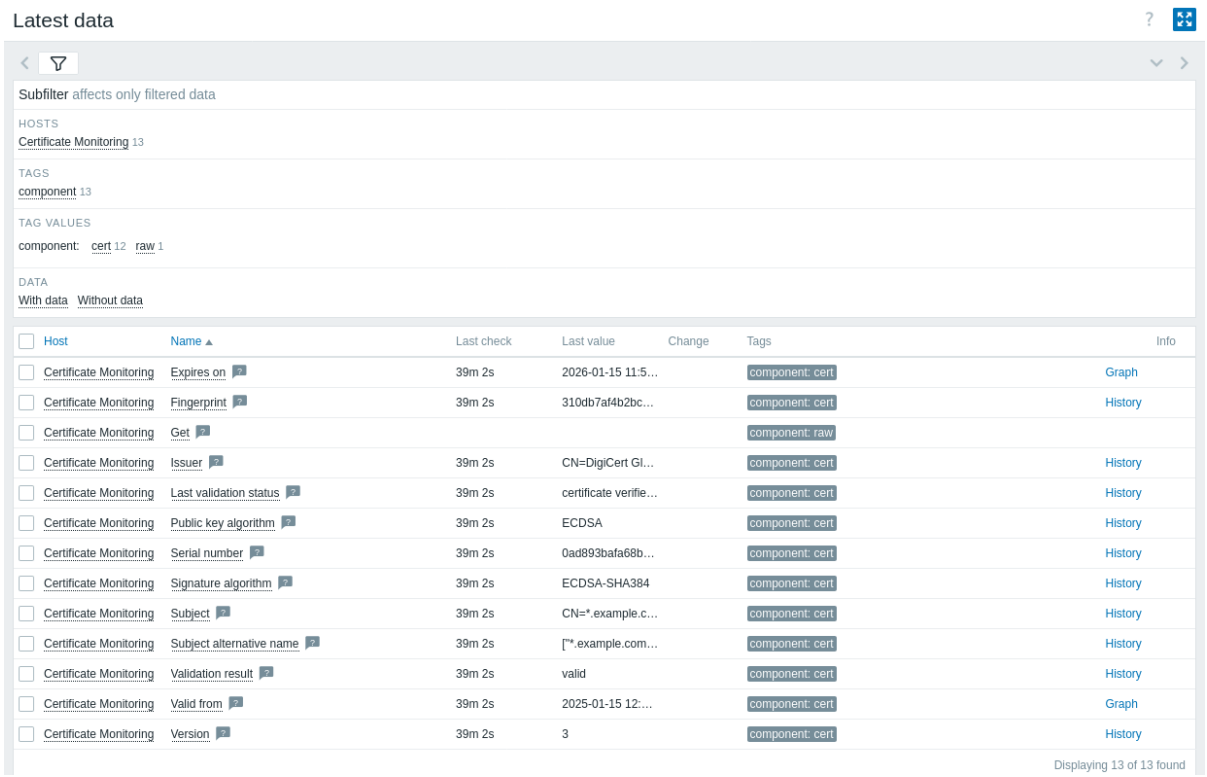
Per ogni sito web specificato nella macro `{$CERT.WEBSITE.HOSTNAME}`, Zabbix creerà un insieme corrispondente di item e trigger. Ciò consente il monitoraggio e gli avvisi separati per il certificato SSL di ciascun sito web.

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

Visualizzare le metriche raccolte

Congratulazioni! A questo punto, Zabbix sta già monitorando il certificato web desiderato.

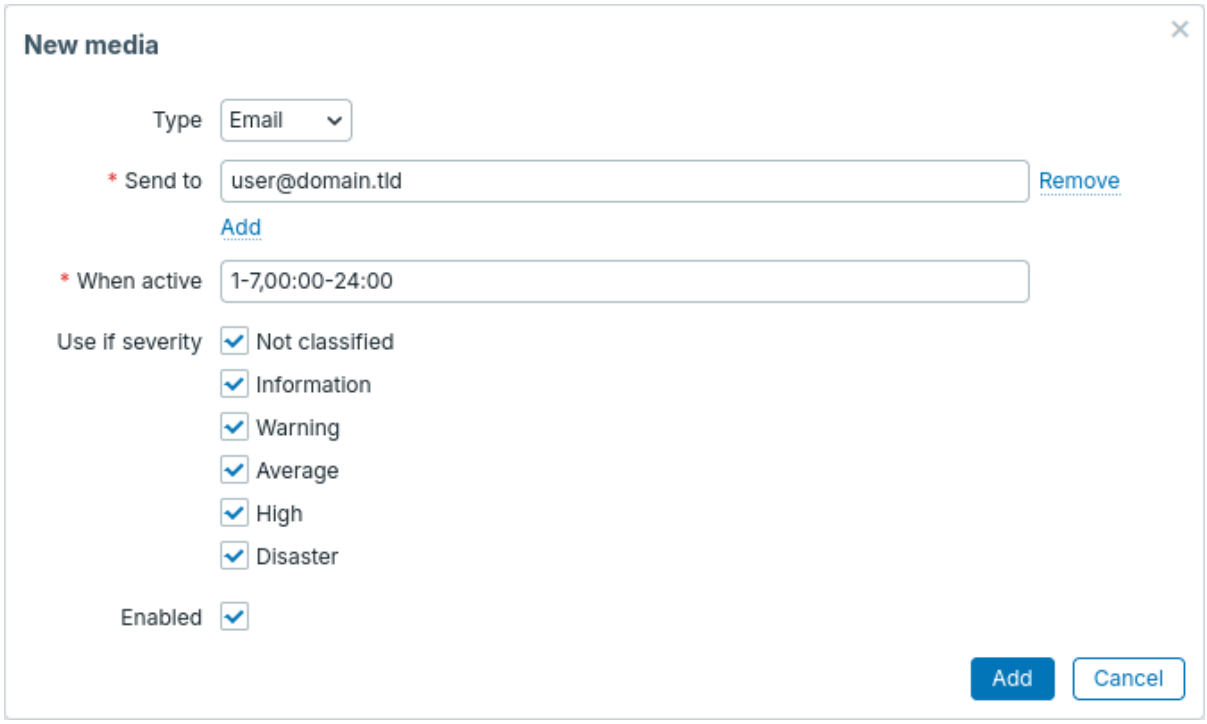
Per visualizzare le metriche raccolte, vai alla sezione di menu *Monitoring > Hosts* e fai clic su *Latest data* accanto all'host per visualizzare in un elenco tutte le metriche raccolte più recenti, come la data di scadenza, l'emittente e il soggetto.



Configurare gli avvisi sui problemi

Zabbix può notificarti problemi nell'infrastruttura. Questa guida fornisce i passaggi di configurazione di base per l'invio di avvisi via email.

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Media* e aggiungi la tua email.



2. Segui la guida per Ricevere una notifica di problema.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Test della configurazione

Per testare la configurazione, possiamo simulare un problema reale aggiornando la configurazione dell'host nel frontend di Zabbix.

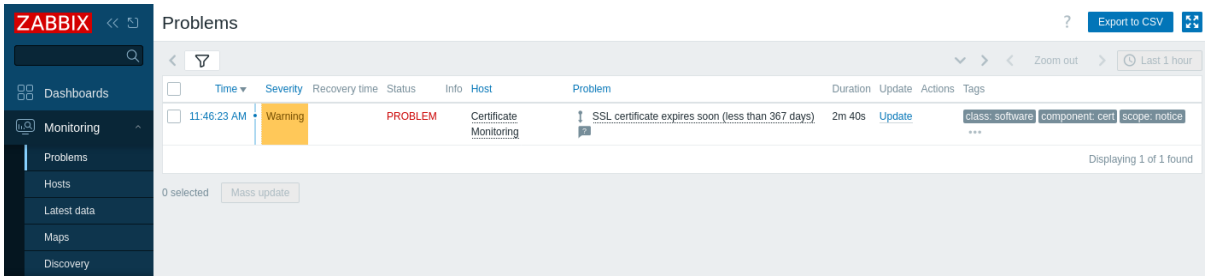
1. Apri la configurazione dell'host "Certificate Monitoring" in Zabbix.

2. Passa alla scheda *Macros* e seleziona *Inherited and host macros*.

3. Fai clic su *Change* accanto al valore della macro `{$CERT.EXPIRY.WARN}` configurato in precedenza e imposta un numero di giorni molto alto (un valore superiore a 365 giorni dovrebbe essere sufficiente) per ricevere un avviso prima della scadenza del certificato.

4. Fai clic su *Update* per aggiornare la configurazione dell'host.

5. Dopo qualche istante, Zabbix rileverà il problema "SSL certificate expires soon", con il numero di giorni rimanenti alla scadenza. Il problema verrà visualizzato in *Monitoring > Problems*.



Se gli avvisi sono configurati, riceverai anche la notifica del problema.

6. Riporta il valore della macro al valore precedente per risolvere il problema e continuare a monitorare i valori del certificato.

Vedi anche

- Zabbix agent 2 - elenca le chiavi degli item.
- Template *Website certificate by Zabbix agent 2* - informazioni aggiuntive sul template *Website certificate by Zabbix agent 2*.
- Template *Website certificate by Zabbix agent 2 active* - informazioni aggiuntive sul template *Website certificate by Zabbix agent 2 active*.
- Monitor websites with Browser items - come avviare il monitoraggio di base dei siti web con Browser items.

Monitorare uno switch di rete o un router con Zabbix

Introduzione

Questa guida ti accompagna attraverso i passaggi necessari per iniziare il monitoraggio di base del tuo switch o router di rete usando Zabbix. Un router Cisco viene usato come esempio, ma la procedura si applica a qualsiasi dispositivo di rete abilitato SNMP.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e per gli amministratori di rete che desiderano abilitare rapidamente il monitoraggio di base dei dispositivi di rete. Se hai bisogno di una personalizzazione approfondita o di opzioni di configurazione avanzate, fai riferimento alla pagina SNMP agent o alla sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida, assicurati di avere:

- Zabbix server e Zabbix frontend installati: installa seguendo le istruzioni per il tuo sistema operativo (vedi Installation from packages e Web interface installation).
- Zabbix agent installato se stai monitorando metriche di rete locali.
- Dispositivo abilitato SNMP: uno switch o router di rete (ad esempio, un router Cisco) con SNMP abilitato.
- File MIB installati: l'installazione dei file MIB consente a Zabbix di tradurre gli OID numerici in nomi e descrizioni leggibili dall'uomo. Senza un supporto MIB adeguato, potresti vedere solo valori numerici, rendendo più difficile configurare gli item e risolvere i problemi.

Per installare i file MIB su Ubuntu:

1. Installa il pacchetto downloader dei MIB:

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

Se devi aggiungere MIB specifici del vendor (ad esempio, da Cisco, Juniper), inseriscili nella directory MIB appropriata:

- Per i sistemi basati su Linux, le posizioni comuni includono `/usr/share/snmp/mibs/` o `/usr/local/share/snmp/mibs/`.
- Per le installazioni di Zabbix, i file MIB possono essere archiviati in `/var/lib/zabbix/mibs/`.

Assicurati che la variabile di ambiente `MIBDIRS` o il file `snmp.conf` includa il percorso corretto.

Per verificare che il sistema riconosca i nuovi MIB, usa:

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

Per istruzioni dettagliate, fai riferimento alla documentazione della tua libreria SNMP:

- Cisco MIBs
- Juniper MIBs

2. Modifica `/etc/snmp/snmp.conf` e commenta la riga che inizia con `mibs :` per consentire al sistema di caricare tutti i MIB disponibili.

3. Verifica eseguendo un `snmpwalk` (ad esempio, `snmpwalk -v 2c -c <your_community_string> <device_IP>`) e controlla che gli OID vengano visualizzati con nomi descrittivi.

A seconda del tuo ambiente, alcuni passaggi di questa guida potrebbero differire leggermente. Questa guida si basa su un ambiente che esegue Ubuntu e su un dispositivo di rete Cisco Catalyst 3750V2-24FS monitorato.

Si presume che il tuo dispositivo di rete sia già installato fisicamente e connesso.

Configurare il dispositivo di rete (esempio di router Cisco)

Per il monitoraggio tramite SNMP, è necessario configurare il dispositivo di rete per consentire le query SNMP. L'esempio seguente è per SNMPv2 e non tiene conto delle impostazioni esistenti. Attenzione: l'applicazione di questi comandi potrebbe sovrascrivere le configurazioni SNMP correnti.

Per un router Cisco, la configurazione in genere prevede i passaggi descritti di seguito.

Esempio SNMPv2

1. Abilita SNMP e imposta la community string.

Accedi alla console del tuo router Cisco ed entra in modalità di configurazione:

```
configure terminal
```

Quindi, abilita SNMP specificando una community string di sola lettura. Ad esempio:

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

Sostituisci `<your_community_string>` con la tua community string sicura. Nota: l'opzione RO (Read-Only) consente a SNMP di recuperare dati dal dispositivo, ma impedisce qualsiasi modifica alla configurazione.

Per motivi di sicurezza, è consigliato limitare l'accesso SNMP solo ai dispositivi necessari. Per ulteriori indicazioni sulla configurazione delle liste di controllo degli accessi (ACL), consulta la documentazione ufficiale Cisco.

2. Salva la configurazione.

Salva le modifiche per assicurarti che le impostazioni SNMP rimangano attive dopo un riavvio:

```
write memory
```

Esempio SNMPv3

SNMPv3 offre una sicurezza avanzata con autenticazione e crittografia. La sua configurazione è più sicura di SNMPv2 e dovrebbe essere verificata in base alla documentazione specifica del dispositivo.

1. Creare un gruppo SNMP.

Configurare un gruppo SNMPv3 con privacy (crittografia) abilitata:

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

2. Creare un utente SNMP.

Aggiungere un utente SNMPv3 con autenticazione e privacy. Sostituire i segnaposto con i valori desiderati:

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

3. Salvare la configurazione:

```
write memory
```

Per ulteriori dettagli o istruzioni specifiche del modello, è possibile fare riferimento a tutorial esterni sulla configurazione SNMP Cisco. Tuttavia, questa guida fornisce i passaggi di base per abilitare il monitoraggio SNMP.

Configurare il frontend di Zabbix

Creare un host nel frontend di Zabbix

1. Accedi al frontend di Zabbix.

2. Aggiungi un nuovo host.

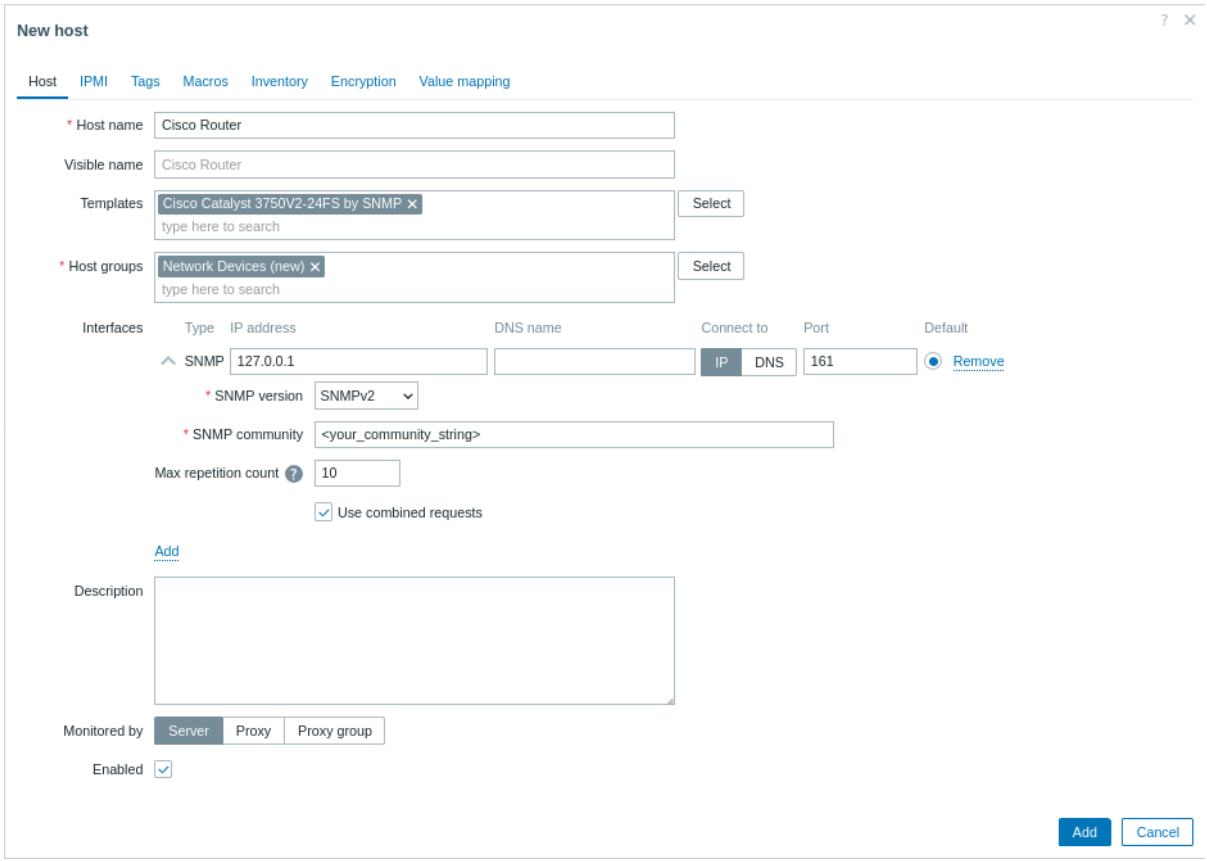
Vai su *Data collection* > *Hosts* e fai clic su *Create host*.

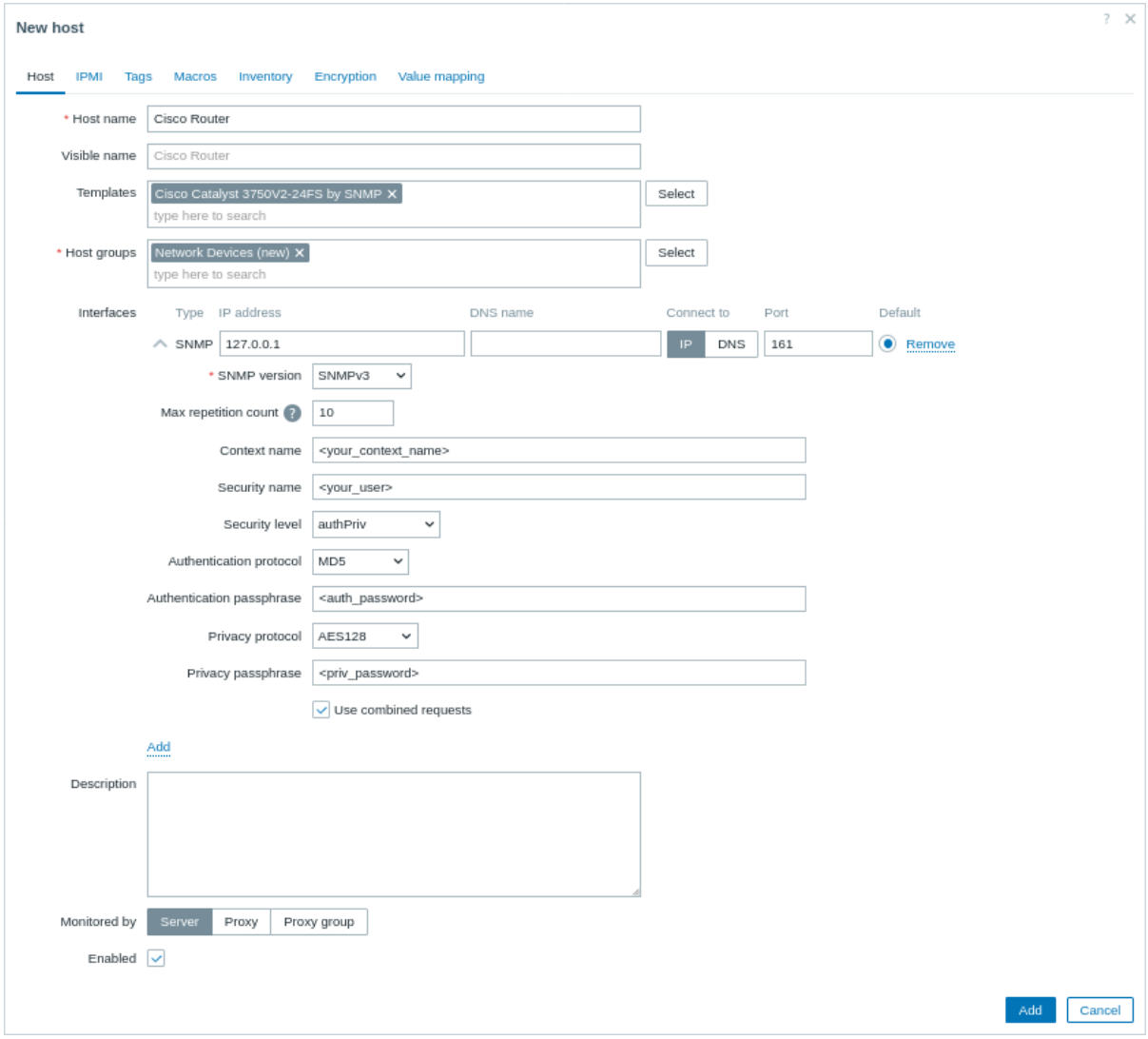
- *Host name*: inserisci un nome per il tuo dispositivo (ad esempio, ”Cisco Router”).
- Host groups: seleziona un gruppo esistente oppure crea un nuovo gruppo, ad esempio ”Network Devices”.
- Interfaces:
  - Fai clic su *Add* sotto Interfaces.
  - Scegli *SNMP* come tipo di interfaccia.
  - Inserisci l’indirizzo IP o il nome DNS del tuo router Cisco.
  - Imposta la porta SNMP predefinita (di solito 161).
  - Usa il menu a discesa per selezionare la versione SNMP appropriata (ad esempio, SNMPv2).
  - Per SNMPv1/v2, inserisci la community string nel campo *SNMP community*. Per SNMPv3, verranno richieste credenziali aggiuntive (*Context name*, *Security name* e *Security level*, ecc.).

3. Collega i template

Nel campo *Templates*, seleziona il template SNMP che corrisponde meglio al tuo dispositivo. Zabbix fornisce una gamma di template SNMP predefiniti per molte famiglie di dispositivi. Ad esempio, se stai monitorando un dispositivo Cisco, scegli il template che corrisponde al sistema operativo o al modello del dispositivo (come Cisco IOS SNMP o Cisco Catalyst 3750<device model> SNMP).

4. Fai clic su *Add* per salvare l’host.



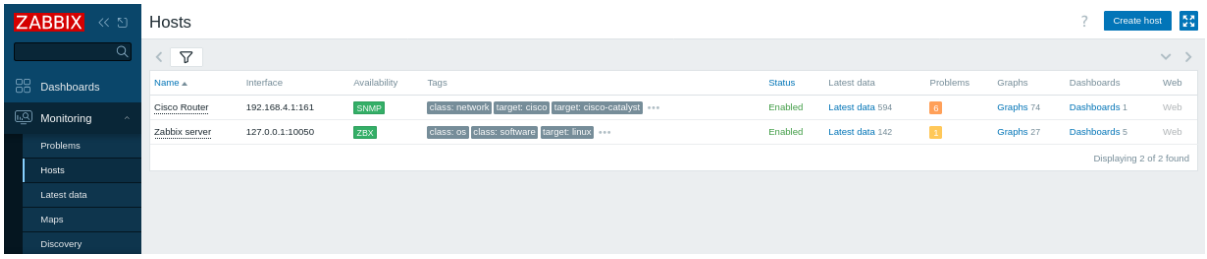


Visualizza le metriche raccolte

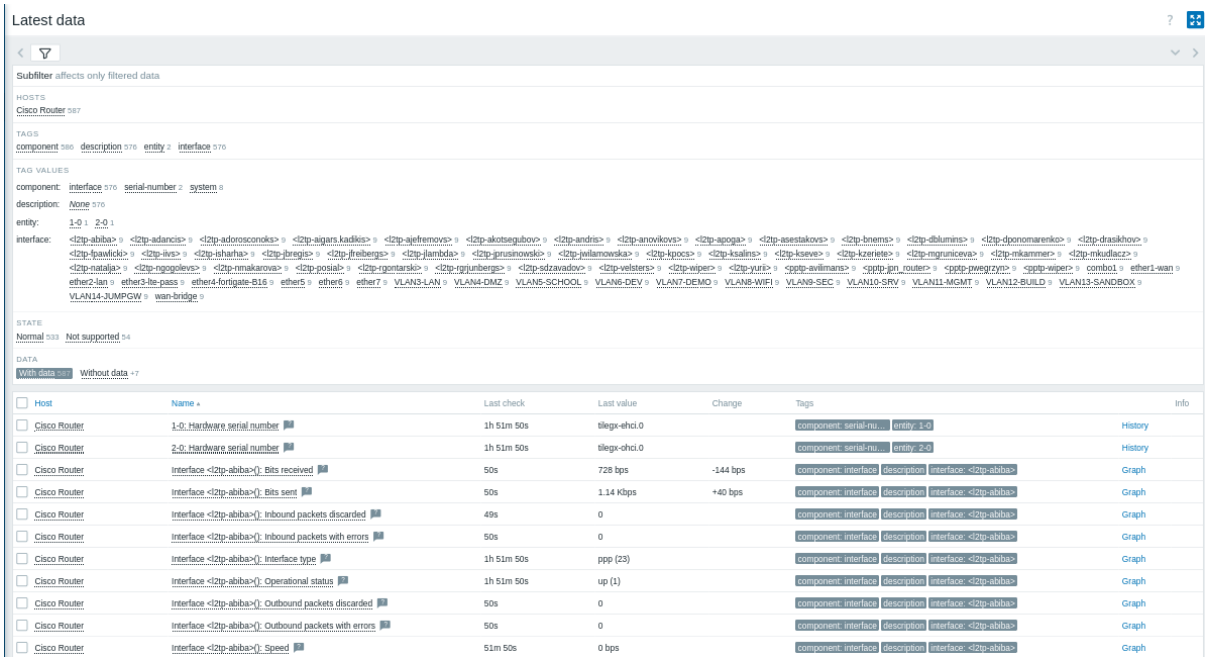
Congratulazioni! Zabbix è ora configurato per monitorare il tuo dispositivo di rete.

Dati più recenti:

- Vai a Monitoring > Latest data nel frontend di Zabbix.



- Seleziona il tuo host ”Cisco Router” (o gli host rilevati) per visualizzare metriche come l’uptime hardware e di rete, la perdita ICMP, il ping e il tempo di risposta, ecc.



- Grafici e schermate:

Per visualizzare i dati sulle prestazioni, fai clic su *Graphs* accanto agli item SNMP per vedere metriche dettagliate.

Come passo successivo, puoi:

- Aggiungere item SNMP personalizzati per monitorare metriche aggiuntive.
- Configurare gli avvisi sui problemi per ricevere notifiche su potenziali problemi.

Creare item SNMP

Una volta configurato l'host, puoi creare item per monitorare metriche specifiche. Nota: questo passaggio è facoltativo se stai usando un template, poiché i template contengono già set predefiniti di item.

1. Identifica l'OID SNMP:

Usa il comando `snmpwalk` per elencare gli OID disponibili sul tuo dispositivo. Ad esempio:

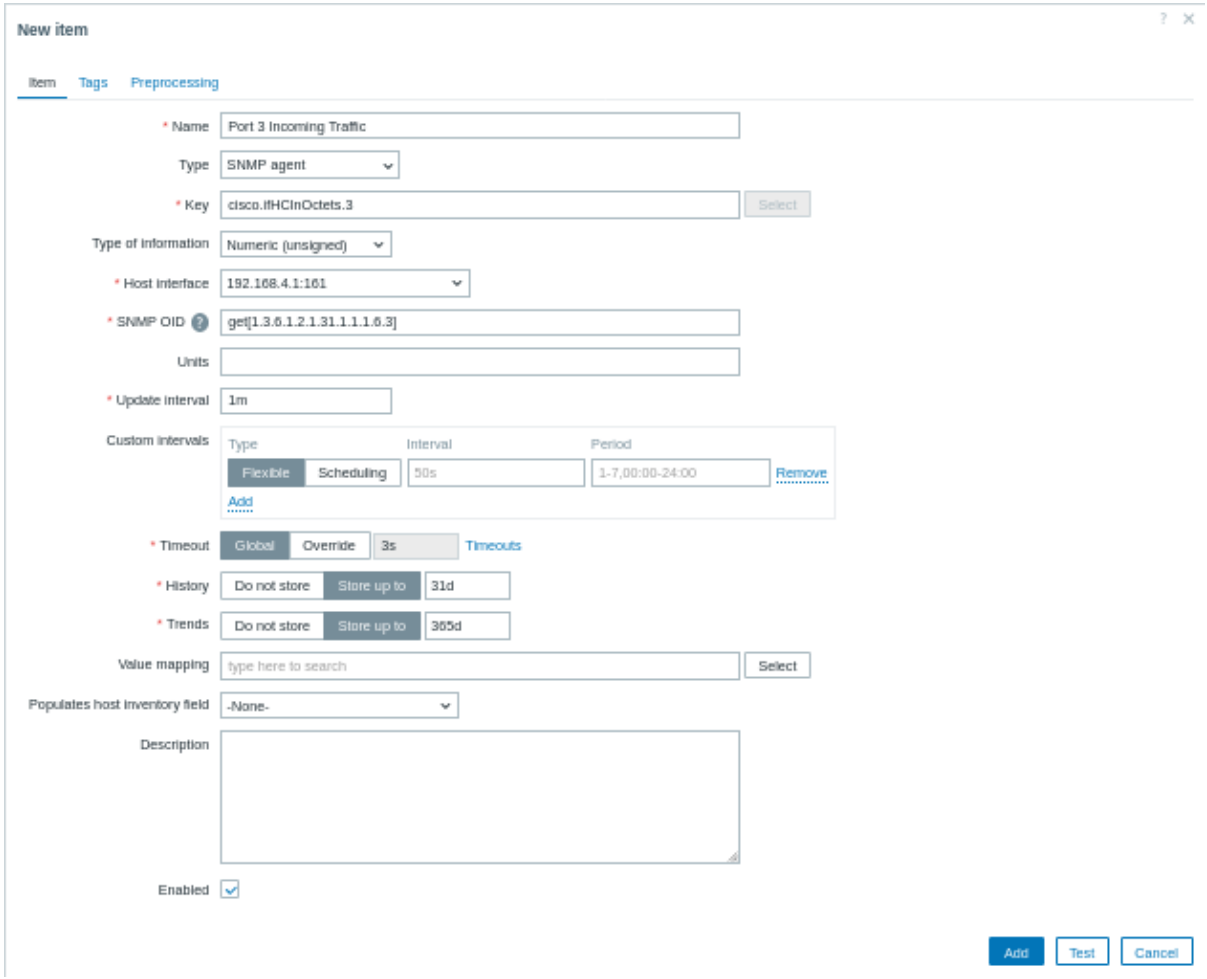
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Trova l'OID per la metrica che desideri monitorare (ad esempio, IF-MIB::ifHCInOctets.3 per il traffico in ingresso sulla porta 3). Per ottenere l'OID numerico, puoi usare:

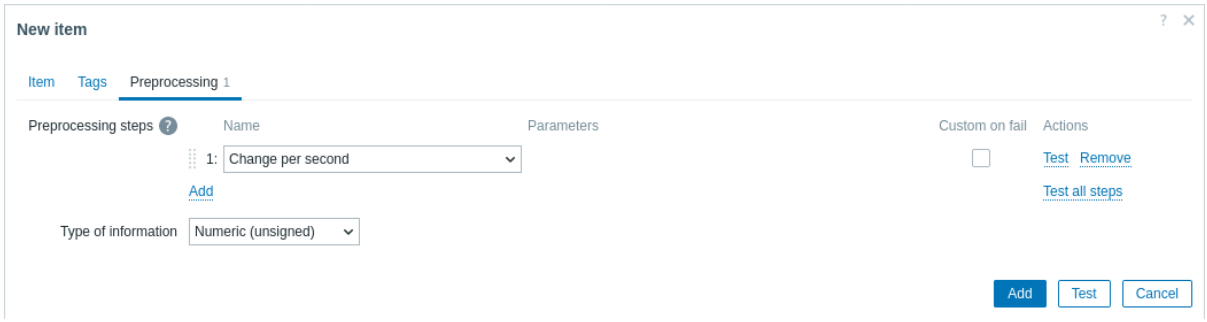
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Crea un item SNMP:

- Vai a *Data collection* > *Hosts* e fai clic sulla scheda *Items* per il tuo host SNMP, quindi fai clic su *Create item*.
- *Name*: inserisci un nome descrittivo (ad esempio, "Traffico in ingresso porta 3").
- *Type*: seleziona *SNMP agent*.
- *Key*: fornisci una chiave significativa (ad esempio, `cisco.ifHCInOctets.3`).
- *Host interface*: assicurati che sia selezionata l'interfaccia SNMP.
- *SNMP OID*: inserisci l'OID usando uno dei formati supportati, ad esempio:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` per un singolo valore;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` per recuperare in modo asincrono un sottoalbero di valori.



- *Preprocessing* (se necessario): se l'item restituisce un contatore cumulativo (come il traffico dell'interfaccia), vai alla scheda *Preprocessing* e aggiungi un passaggio di preprocessing come "Change per second" per calcolare il tasso.



Per recuperare più valori in una singola transazione SNMP, puoi specificare diversi OID usando la sintassi `walk[OID1,OID2,...]`.

Traduzione degli OID tra nomi numerici e nomi MIB

Quando si lavora con SNMP, potrebbe essere necessario convertire tra OID numerici e i relativi nomi MIB. Questa traduzione aiuta a identificare e risolvere i problemi relativi alle metriche in modo più semplice.

- Per tradurre un nome MIB in un OID numerico: usa il comando `snmptranslate` con l'opzione `-On`. Ad esempio, per tradurre il nome MIB `IF-MIB::ifHCInOctets.3` nel suo OID numerico, esegui:

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

Questo comando potrebbe restituire:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Per tradurre un OID numerico nel suo nome MIB: usa il comando `snmptranslate` con l'opzione `-IR` (o `-m ALL`) per invertire la traduzione. Ad esempio, per tradurre l'OID numerico `.1.3.6.1.2.1.31.1.1.1.6.3` di nuovo nel suo nome MIB, esegui:

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

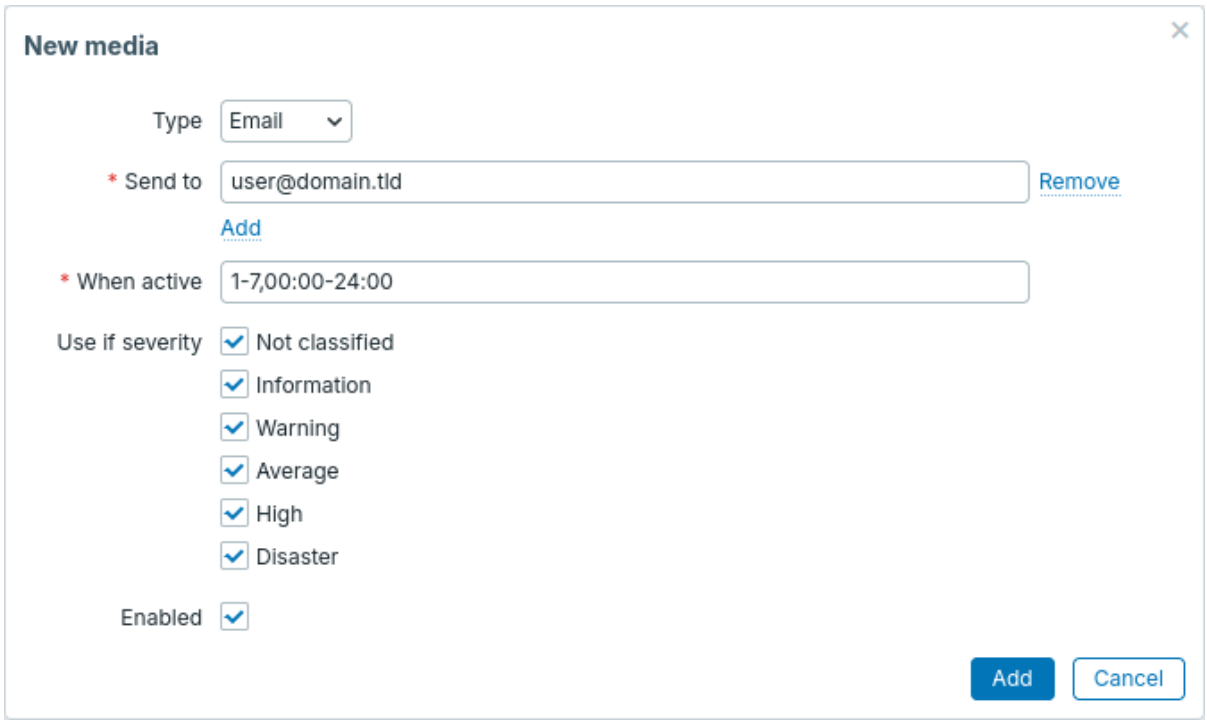
Questo comando potrebbe restituire:

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

Configurare gli avvisi sui problemi

Questa guida fornisce i passaggi di configurazione di base per l'invio di avvisi via email.

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Media* e aggiungi il tuo indirizzo email.



2. Segui la guida per Ricevere una notifica di problema.

La prossima volta, quando Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Testa la tua configurazione

Per assicurarti che Zabbix rilevi correttamente i problemi di prestazioni di rete, simula un problema reale aumentando la soglia del tempo di risposta del ping ICMP.

1. Apri la configurazione dell'host "Cisco Router" in Zabbix.

2. Vai alla scheda *Macros* e seleziona *Inherited and host macros*.

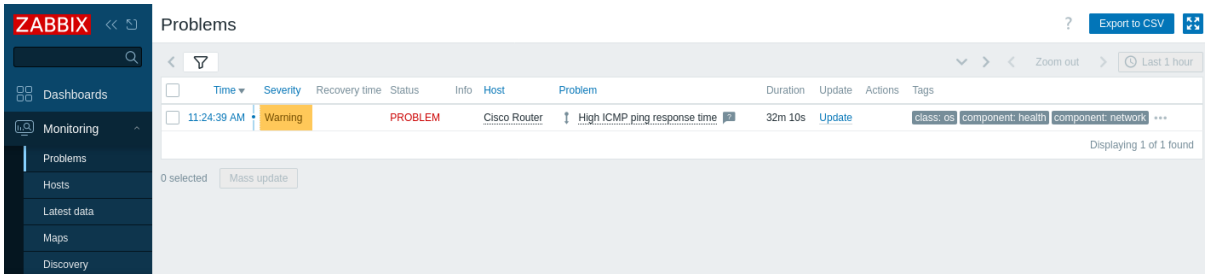
3. Individua la macro `{$ICMP_RESPONSE_TIME_WARN}` (o una macro simile per la soglia del tempo di risposta).

4. Imposta un valore molto basso (ad esempio 0.001) per attivare un avviso quando il tempo di risposta del ping supera questo valore.

5. Fai clic su *Update* per applicare le modifiche.

6. Attendi qualche istante affinché Zabbix rilevi il problema simulato.

7. Vai a *Monitoring > Problems* per verificare che venga visualizzato un avviso (ad esempio, "High ICMP ping response time").



Se gli avvisi sono configurati, dovresti ricevere anche un avviso relativo al problema.

8. Ripristina il valore della macro a quello originale e fai clic su *Update* per salvare le modifiche.

9. Conferma che il problema sia risolto e scompaia dalla sezione *Problems*.

Risoluzione dei problemi del monitoraggio SNMP

Se noti che l'icona SNMP nel frontend di Zabbix appare ROSSA oppure non vengono raccolti dati, prova i seguenti passaggi:

1. Verifica la connettività SNMP.

Per SNMPv2 esegui il seguente comando dal tuo server Zabbix:

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

Questo comando verifica che il dispositivo risponda alle query SNMP.

Per SNMPv3, includi le credenziali SNMPv3 appropriate:

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

Questo verifica che le credenziali SNMPv3 siano corrette e che il dispositivo risponda in modo sicuro.

2. Assicurati che i file MIB siano installati e abilitati come descritto nei prerequisiti. Per verificarlo, il seguente comando non deve restituire errori quando interroghi un dispositivo di rete:

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

Questo dovrebbe restituire OID tradotti senza errori.

3. Conferma che la versione SNMP e le credenziali configurate in Zabbix corrispondano a quelle impostate sul dispositivo. Ad esempio, controlla le impostazioni SNMP nella configurazione dell'host in Zabbix e confrontale con la configurazione del dispositivo. Su un dispositivo Cisco, puoi verificare le impostazioni SNMP eseguendo:

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

Questo assicura che la community string (per SNMPv2) o i dettagli dell'utente SNMPv3 siano corretti.

4. Verifica che SNMP sia abilitato correttamente sul tuo dispositivo di rete. Su un router Cisco, accedi alla console ed esegui:

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

Questo comando mostra la configurazione SNMP attiva e aiuta a confermare che SNMP sia configurato correttamente.

5. Assicurati che nessun firewall o problema di rete stia bloccando il traffico SNMP (tipicamente sulla porta 161) tra il server Zabbix e il dispositivo. Puoi testare la connettività usando:

nc -zv <device_IP> 161

`nc -zv` verifica se la porta 161 è aperta e in ascolto sul dispositivo.

Inoltre, se stai usando UFW su Ubuntu, controlla lo stato del firewall:

```
sudo ufw status
```

Oppure, per iptables:

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

6. Esamina i file di log del server Zabbix per eventuali errori relativi a SNMP, così da individuare meglio il problema:

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

`tail -f` consente di monitorare gli aggiornamenti del log in tempo reale.

Vedi anche:

- Creazione di un item - scopri come aggiungere metriche aggiuntive.
- SNMP agent - informazioni aggiuntive sul monitoraggio SNMP con Zabbix.
- Template standardizzati per dispositivi di rete - informazioni sui template SNMP disponibili.
- Scoperta degli OID SNMP - informazioni aggiuntive sulla scoperta SNMP su uno switch.
- Configurazione di una regola di discovery di rete - informazioni aggiuntive su come configurare una regola di discovery di rete utilizzata da Zabbix per individuare host e servizi.

Monitorare il registro eventi di Windows usando controlli attivi

Introduzione

Questa guida spiega come monitorare i registri eventi di Windows con Zabbix usando controlli attivi. Con le chiavi item specifiche per Windows di Zabbix, puoi raccogliere e analizzare eventi critici (come tentativi di accesso non riusciti, errori di sistema, ecc.) in tempo reale.

A chi è destinata questa guida

Questa guida è pensata per i nuovi utenti di Zabbix e per gli amministratori di rete che desiderano monitorare i registri eventi di Windows. Per le opzioni di configurazione avanzate, consulta la documentazione sulle chiavi item specifiche per Windows.

Prerequisiti

Prima di procedere con questa guida, devi scaricare e installare Zabbix server e Zabbix frontend seguendo le istruzioni per il tuo sistema operativo. Devi inoltre avere Zabbix agent scaricato e installato sulla macchina Windows che desideri monitorare.

Configurare l'agent Zabbix per il monitoraggio del registro eventi di Windows

1. Aprire il file di configurazione dell'agent Zabbix sul proprio host Windows e assicurarsi che il parametro `ServerActive` sia impostato sull'indirizzo IP del server Zabbix e che il parametro `Hostname` corrisponda al nome dell'host che verrà definito nel frontend Zabbix. In questo modo l'agent può richiedere i controlli attivi per il proprio host e dal server Zabbix specificato. Ad esempio:

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

2. Riavviare il servizio dell'agent Zabbix per applicare le modifiche:

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

3. Verificare che l'host Windows sia in esecuzione:

- Assicurarsi che il servizio dell'agent Zabbix sia in esecuzione sull'host Windows.
- Verificare che l'host Windows possa connettersi al server Zabbix sulla porta 10051. Per testare la connettività dall'host Windows, aprire PowerShell ed eseguire il comando seguente:

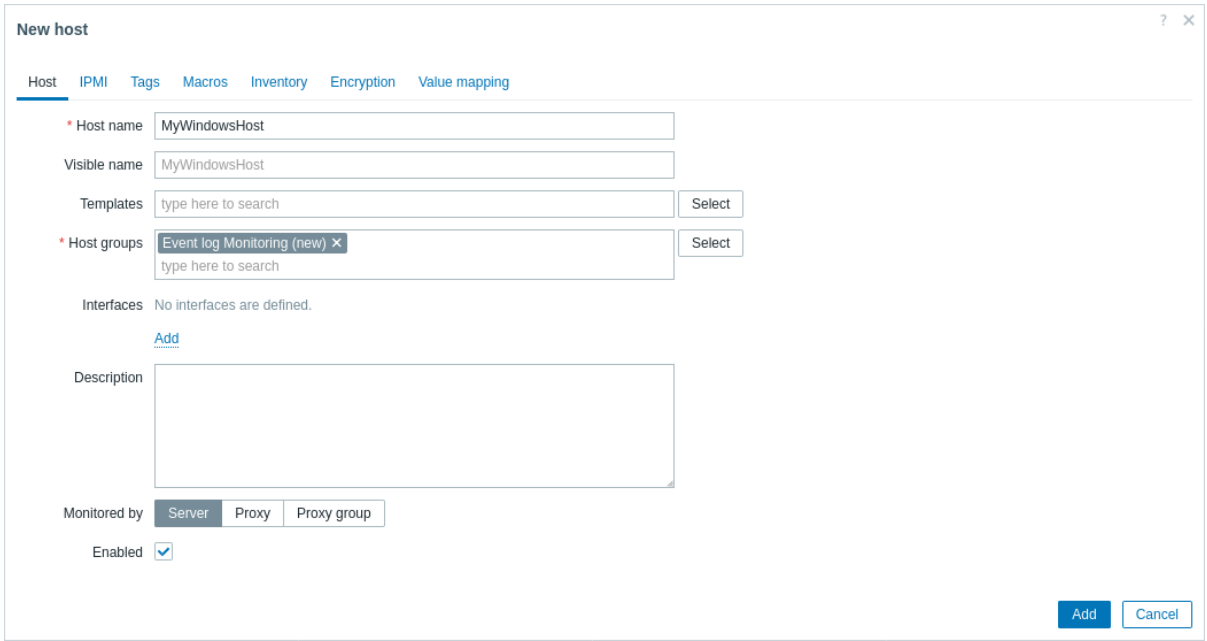
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configurare il frontend di Zabbix

1. Vai a *Data collection* > *Hosts* e crea un host:

- Nel campo *Host name*, inserisci un nome host (ad esempio, "MyWindowsHost").
- Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, "Event log Monitoring").
- Fai clic su *Add* per salvare l'host configurato.

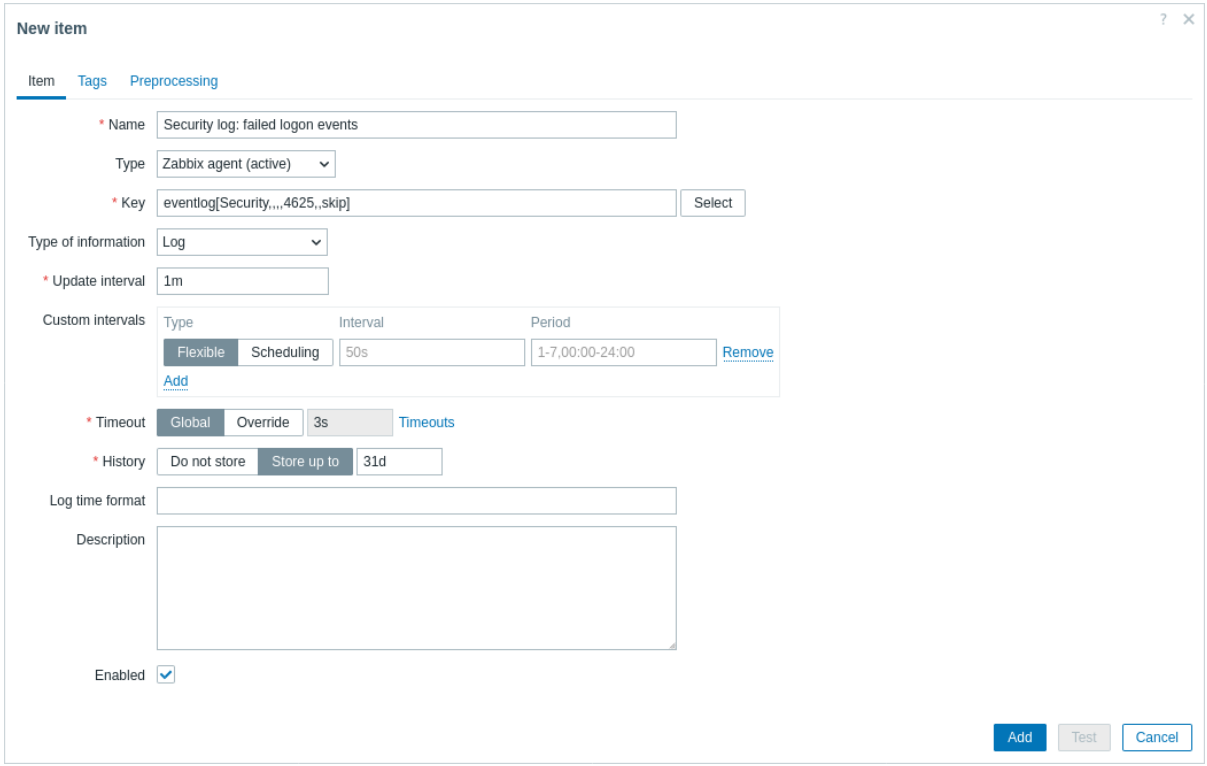
Note:
Nel campo *Templates* puoi aggiungere il template "Windows by Zabbix agent active" per aiutarti nella risoluzione dei problemi osservando se altri item attivi sullo stesso host vengono aggiornati.



2. Crea un nuovo item con i seguenti parametri:

- Nel campo *Name*, inserisci un nome descrittivo per l'item (ad esempio, "Security log: failed logon events").
- Nel menu a discesa *Type*, seleziona "Zabbix agent (active)" (richiesto per il monitoraggio del registro eventi).
- Nel campo *Key*, usa la chiave item eventlog. Ad esempio, per monitorare i tentativi di accesso non riusciti (Event ID: 4625) nel registro Security e ignorare le voci più vecchie dell'ultimo controllo dell'item (usando il parametro `skip`), inserisci la seguente chiave item: `eventlog[Security,,,,4625,,skip]`

- Nel menu a discesa *Type of information*, seleziona ”Log”.



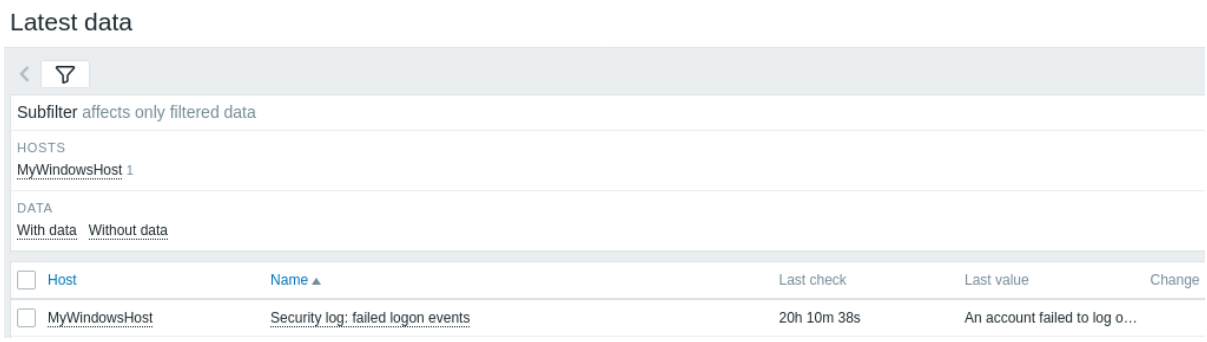
3. Fai clic su *Add* per salvare l’item.

Test e visualizza le metriche raccolte

Congratulazioni! Zabbix è ora configurato per raccogliere i log eventi di Windows. Per verificare che i log eventi vengano raccolti, puoi testare l’item ”Security log: failed logon events” disconnettendoti dal tuo account Windows e tentando di accedere usando credenziali errate.

Quindi, visualizza i log raccolti nel frontend di Zabbix:

1. Vai a *Monitoring* > *Latest data* nel frontend di Zabbix.



2. Filtra per il tuo host ”MyWindowsHost” nel campo *Name*.

3. Fai clic su *History* per visualizzare i valori dei log registrati.

MyWindowsHost: Security log: failed logon events

Timestamp	Local time	Source	Severity	Event ID	Value
2025-04-07 11:13:11 PM	2025-04-07 11:12:19 PM	Microsoft-Windows-Security-Auditing	Failure Audit	4625	An account failed to log on. Subject: Security ID: NT AUTHORITY\SYSTEM Account Name: DESKTOP-NNEM26U$ Account Domain: WORKGROUP Logon ID: 0x3E7 Logon Type: 2 Account For Which Logon Failed: Security ID: NULL SID Account Name: Admin Account Domain: DESKTOP-NNEM26U Failure Information: Failure Reason: Unknown user name or bad password. Status: 0xC000006D Sub Status: 0xC000006A

4. Se i valori dei log sono assenti, procedi alla sezione Risoluzione dei problemi della guida.

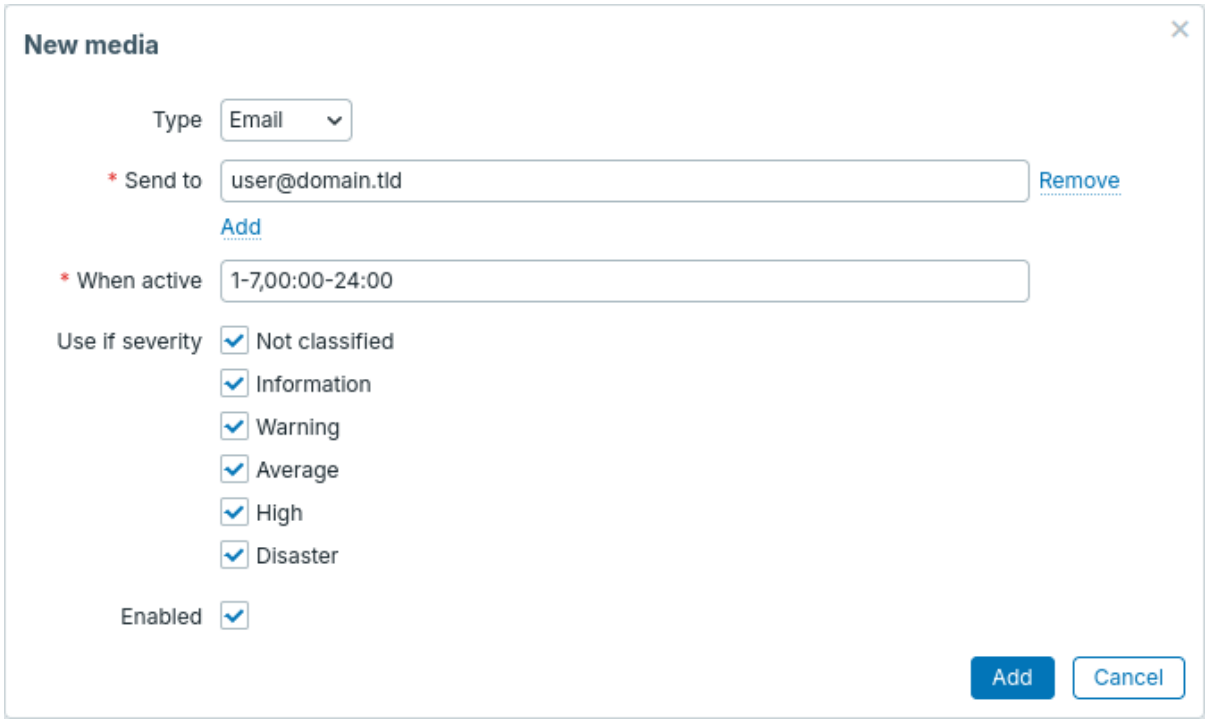
Configurare gli avvisi di problema

Questa guida fornisce i passaggi di configurazione di base per l'invio di avvisi via email.

1. Vai su Raccolta dati > Host per definire un trigger che si attivi quando il tuo item di log eventi registra il pattern che ti interessa. Ad esempio, per intercettare i tentativi di accesso non riusciti nel log Security, usa la funzione find():

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

2. Vai su *Impostazioni utente > Profilo*, passa alla scheda *Media* e aggiungi il tuo indirizzo email.



3. Segui la guida per Ricevere una notifica di problema.

La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Risoluzione dei problemi

Se riscontri problemi nella raccolta o nella visualizzazione dei log eventi di Windows, usa i suggerimenti riportati di seguito per identificare e risolvere i problemi più comuni:

1. Sul server Zabbix (Linux), elenca le regole iptables con il seguente comando:

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

e verifica che sia presente una regola ACCEPT per la porta TCP 10051.

2. Assicurati che la chiave `eventlog[...]` utilizzi il nome esatto del log (con distinzione tra maiuscole e minuscole), l'ID evento, la modalità (ad esempio, skip) e gli altri parametri esattamente come mostrato nelle chiavi item specifiche per Windows.

Vedi anche:

- Creazione di un item - scopri come aggiungere metriche aggiuntive.
- Zabbix agent su Microsoft Windows - istruzioni dettagliate per l'installazione.
- Monitorare Windows con Zabbix agent - una guida completa alla configurazione del monitoraggio di base per le macchine Windows usando Zabbix agent.
- Chiavi item specifiche per Windows - informazioni dettagliate sulle chiavi item specifiche per Windows supportate dagli Zabbix agent, incluse quelle per il monitoraggio del registro eventi.
- Monitoraggio dei file di log - istruzioni per configurare Zabbix per il monitoraggio centralizzato e l'analisi dei file di log, applicabili ai registri eventi di Windows.

Installazione: guide aggiuntive

Se stai cercando il modo più semplice per installare Zabbix, consulta Installazione e primi passi.

Le guide seguenti illustrano modalità **aggiuntive** o specifiche del sistema per installare Zabbix o i suoi componenti:

- Installazione da container
- Installazione dai sorgenti
- Note sull'installazione dai pacchetti
- Installazione di Zabbix agent da Windows MSI
- Compilazione di Zabbix agent 2 su Windows
- Compilazione di Zabbix agent su Windows
- Installazione di agent su macOS da PKG
- Zabbix agent su IBM AIX
- Compilazione di Zabbix agent su macOS
- Zabbix appliance
- Guida alla distribuzione di Zabbix server su AWS
- Guida alla distribuzione di Zabbix proxy su AWS
- Guida alla distribuzione di Zabbix server su Azure
- Guida alla distribuzione di Zabbix proxy su Azure
- Guida alla distribuzione su Google Cloud

Installazione da container

Panoramica

Questa pagina descrive come distribuire Zabbix usando Docker o Docker Compose.

Docker Compose è il modo più rapido per distribuire Zabbix. Legge un file di configurazione e avvia automaticamente tutti i container che compongono una configurazione Zabbix completa, nell'ordine corretto.

Docker (distribuzione manuale) ottiene lo stesso risultato distribuendo ogni componente passo dopo passo.

Prerequisiti

Prima di iniziare, assicurarsi che Docker (1.12.0 o versione successiva) sia installato nel sistema. In caso contrario, seguire la guida all'installazione di Docker. Per la distribuzione con Docker Compose, è inoltre necessario Docker Compose (2.24.0 o versione successiva).

Alcuni componenti Zabbix richiedono che determinate porte siano aperte sul host che esegue Docker (ad esempio, 10051/TCP per il server Zabbix, 162/UDP per i trap SNMP). Consultare Requisiti per l'elenco completo delle porte utilizzate dai componenti Zabbix.

Per il server e l'agent Zabbix, la porta predefinita può essere modificata impostando la variabile di ambiente `ZBX_LISTENPORT` sul container corrispondente.

Immagini Docker disponibili

Zabbix fornisce un'immagine Docker per ogni componente Zabbix, tutte pubblicate su Docker Hub. Ogni immagine viene utilizzata per creare un container che esegue il componente corrispondente.

Componente Zabbix	Immagine Docker
Agent	zabbix/zabbix-agent
Agent 2	zabbix/zabbix-agent2
Server (MySQL)	zabbix/zabbix-server-mysql
Server (PostgreSQL)	zabbix/zabbix-server-pgsql
Interfaccia web (Apache + MySQL)	zabbix/zabbix-web-apache-mysql
Interfaccia web (Apache + PostgreSQL)	zabbix/zabbix-web-apache-pgsql
Interfaccia web (Nginx + MySQL)	zabbix/zabbix-web-nginx-mysql
Interfaccia web (Nginx + PostgreSQL)	zabbix/zabbix-web-nginx-pgsql
Proxy (SQLite3)	zabbix/zabbix-proxy-sqlite3
Proxy (MySQL)	zabbix/zabbix-proxy-mysql
Java gateway	zabbix/zabbix-java-gateway
Servizio web	zabbix/zabbix-web-service
Trap SNMP	zabbix/zabbix-snmptraps
Server MCP	zabbix/zabbix-mcp-server

Note:
Per utilizzare le trap SNMP, il container delle trap SNMP deve condividere un volume con il container del server o del proxy Zabbix (vedere come utilizzare questa immagine su Docker Hub e l'esempio riportato di seguito).

Tutte le immagini dei componenti Zabbix su Docker Hub sono basate sulle versioni principali più recenti dei sistemi operativi supportati. Queste immagini vengono ricreate automaticamente quando vengono aggiornate le immagini del sistema operativo sottostante.

Tag delle immagini

Ogni immagine supporta tag per selezionare il sistema operativo di base e la versione di Zabbix:

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

Valori `<os>` supportati:

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

Valori `<version>` supportati:

- `latest` - Ultima versione stabile di Zabbix su Alpine Linux
- `<os>-latest` - Ultima versione stabile di Zabbix sul sistema operativo selezionato
- `<os>-trunk` - Ultima build di sviluppo (nightly) sul sistema operativo selezionato
- `<os>-X.X-latest` - Ultima minor release di Zabbix di una specifica major version di Zabbix sul sistema operativo selezionato
- `<os>-X.X.*` - Specifica minor release di Zabbix sul sistema operativo selezionato

Esempi:

```
#### Ultimo Zabbix proxy stabile (SQLite3) su Alpine Linux:
docker pull zabbix/zabbix-proxy-sqlite3:latest

#### Ultimo Zabbix proxy stabile (SQLite3) su Ubuntu:
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

#### Ultima build di sviluppo (nightly) di Zabbix server (MySQL) su Ubuntu:
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk
```

```
#### Ultima minor release 8.0 di Zabbix server (MySQL) su Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest

#### Versione 8.0.1 di Zabbix server (MySQL) su Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

Docker Compose è il modo più rapido per distribuire Zabbix. Legge un file di configurazione (file Compose) e gestisce l'intera configurazione: scarica le immagini Docker, crea una rete interna tra i container, configura lo storage, inizializza il database e avvia tutto nell'ordine corretto.

Il repository ufficiale dei Dockerfile di Zabbix fornisce file Docker Compose pronti all'uso e un sistema di configurazione basato su `.env`, che supporta diversi sistemi operativi, backend di database e configurazioni dei componenti di Zabbix.

1. Clonare il repository, accedere alla relativa directory e passare alla versione 8.0:

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

2. (Facoltativo) Personalizzare la distribuzione con le variabili d'ambiente. Questo passaggio può essere ignorato per una configurazione predefinita.

3. Distribuire la configurazione predefinita: server Zabbix, frontend web (in esecuzione su NGINX) e un backend di database (MySQL o PostgreSQL), ciascuno in esecuzione nel proprio container su Alpine Linux.

```
#### Con MySQL come database:
docker compose -f ./compose.yaml up -d

#### Con PostgreSQL come database:
docker compose -f ./compose_pgsql.yaml up -d
```

Dopo che i container sono stati avviati (in genere entro 1–3 minuti), il server Zabbix si avvia e il frontend web diventa disponibile per il login.

> **Note:**
> Utilizzare `docker compose ps` per verificare lo stato dei container. Tutti i container (ad eccezione del container `zabbix-docker-server-db-init-1`) devono avere lo stato `Up`. Se uno di essi mostra lo stato `Exited`, controllare i relativi log per individuare eventuali errori con `docker logs -f <container-name>`.

Variabili d'ambiente

Il comportamento di Docker Compose e la configurazione dei componenti Zabbix possono essere personalizzati utilizzando variabili d'ambiente.

Le **variabili a livello di Compose** (definite nel file `.env`) controllano quali immagini Docker, porte e intervalli di indirizzi IP di rete vengono utilizzati. È possibile utilizzare queste variabili inline prima del comando `docker compose` oppure modificare il file `.env`.

Ad esempio, il comando seguente distribuisce una configurazione Zabbix completa composta da più container utilizzando un'immagine basata su Ubuntu (anziché l'immagine predefinita basata su Alpine Linux) ed espone l'interfaccia web tramite NGINX sulle porte HTTP (8282) e HTTPS (8443) personalizzate:

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

Le **variabili a livello di componente** (definite nei file `env_vars/.env_<component>`) controllano la configurazione dei componenti Zabbix. Modificare il file `.env_<component>` corrispondente prima di eseguire il comando `docker compose`.

Ad esempio, è possibile modificare il numero di poller passivi e le impostazioni della cache del server Zabbix, nonché configurare il fuso orario dell'interfaccia web, modificando le variabili seguenti:

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

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

> **Note:**
> Le variabili d'ambiente a livello di componente corrispondono ai parametri di configurazione dei componenti Zabbix, utilizzando uno stile di denominazione diverso (ad esempio, ZBX_STARTPOLLERS corrisponde a StartPollers). Alcune variabili sono specifiche di Docker e alcuni parametri di configurazione non possono essere modificati (ad esempio, PIDFile e LogType). Quando si utilizzano variabili d'ambiente, consultare la sezione *Environment variables* nella descrizione dell'immagine Docker di ciascun componente su Docker Hub.

Volumi

Docker Compose memorizza i dati persistenti nella directory zbx_env/ creata accanto al file Compose. Questa directory conserva i dati dei componenti tra i riavvii e gli aggiornamenti dei container.

Il contenuto di zbx_env/ è predefinito per l'immagine di ciascun componente. Ad esempio:

- zbx_env/etc/ - utilizzata dal container dell'interfaccia web di Zabbix
- zbx_env/usr/ e zbx_env/var/ - utilizzate dal container del server Zabbix

Per ulteriori informazioni su ciascun volume, vedere la sezione *Allowed volumes* nella descrizione dell'immagine Docker di ciascun componente su Docker Hub.

Makefile

Il repository dei Dockerfile di Zabbix fornisce anche un Makefile come scorciatoia per le attività comuni di Docker Compose. Invece di usare il comando completo `docker compose`, è possibile usare comandi `make` più brevi (eseguire `make help` per visualizzare le opzioni disponibili):

```
#### Distribuzione predefinita (server Zabbix, frontend, MySQL, tutto su Alpine Linux)
make up

#### Distribuzione personalizzata (server Zabbix, frontend con porte NGINX personalizzate, PostgreSQL, tut
make up \
  OS=ubuntu \
  DB=pgsql \
  ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
  ZABBIX_WEB_NGINX_HTTPS_PORT=8443
```

> **Attention:**
> Quando si arrestano o rimuovono i container, specificare sempre lo stesso tipo di database utilizzato per la distribuzione (ad esempio, `make down DB=pgsql`).

Per impostazione predefinita, `make up` avvia solo un set minimo di servizi (server Zabbix, frontend e database). In questo modo la configurazione rimane leggera ed evita di avviare componenti che potrebbero non essere necessari. Per includere componenti aggiuntivi, è possibile usare i seguenti profili Compose:

```
#### Distribuzione predefinita + Zabbix agent, Java gateway, servizio web e trap SNMP:
make up COMPOSE_PROFILES=full

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

Docker (distribuzione manuale)

Usa la distribuzione manuale quando vuoi distribuire Zabbix passo dopo passo, eseguire singoli componenti, integrarli con un ambiente esistente o usare un runtime di container alternativo come Podman.

Ad esempio, per distribuire un proxy Zabbix, esegui il seguente comando:

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

```
  -d \
  zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

Questo comando:

- Scarica la Docker image `zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest`.
- Crea e avvia un container `zabbix-proxy-sqlite3` basato sulla Docker image (con i flag `--init` e `-d`).
- Configura il parametro `Server` del proxy Zabbix tramite la variabile di ambiente `ZBX_SERVER_HOST`. Questo parametro specifica l'indirizzo IP del server Zabbix da cui il proxy recupererà i dati di configurazione e a cui invierà i dati raccolti. Tieni presente che le altre variabili richieste per il funzionamento del proxy attivo usano valori predefiniti e possono essere omesse.

Note:
Le variabili di ambiente corrispondono ai parametri di configurazione dei componenti Zabbix usando uno stile di denominazione diverso (ad esempio, `ZBX_SERVER_HOST` corrisponde a `Server`). Alcune variabili sono specifiche di Docker e alcuni parametri di configurazione non possono essere modificati (ad esempio, `PIDFile` e `LogType`). Quando usi le variabili di ambiente, fai riferimento alla sezione *Environment variables* nella descrizione della Docker image di ciascun componente su Docker Hub.

- Collega una posizione di archiviazione gestita da Docker (ad esempio, `/var/lib/docker/volumes/zabbix-proxy-data/`) alla directory `/var/lib/zabbix` del container, in modo che i dati del proxy Zabbix vengano archiviati in modo permanente, anche se il container viene rimosso.

Note:
Per ulteriori informazioni su ciascun volume, consulta la sezione *Allowed volumes* nella descrizione della Docker image di ciascun componente su Docker Hub.

Dopo aver distribuito il container del proxy Zabbix, puoi procedere ad aggiungere il proxy nella tua interfaccia web Zabbix e a configurare gli host da monitorare con questo proxy.

Gli esempi seguenti coprono tre scenari aggiuntivi di distribuzione:

- Zabbix server (MySQL) con Java gateway
- Zabbix server (PostgreSQL) con SNMP traps
- Zabbix server (MySQL) con Java gateway su RHEL 8–10

Per altri esempi, fai riferimento alla descrizione della Docker image di ciascun componente su Docker Hub.

Implementare Zabbix server (MySQL) cu Java gateway

Acest exemplu arată cum se implementează un Zabbix server cu un backend MySQL, un Java gateway pentru monitorizarea JMX și o interfață web bazată pe NGINX.

1. Creați o rețea Docker dedicată, astfel încât toate containerele componentelor Zabbix să se poată conecta între ele folosind numele containerelor:

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

2. Porniți un container gol pentru serverul MySQL (înlocuiți `zabbix_pwd` și `root_pwd` cu parole puternice și utilizați aceste valori în pașii următori):

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

3. Urmăriți jurnalele MySQL și așteptați până când MySQL este pregătit să accepte conexiuni (utilizați `Ctrl+C` pentru a ieși din jurnale):

```
docker logs -f mysql-server

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

4. Porniți un container Zabbix Java gateway:

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

5. Activați `log_bin_trust_function_creators` în containerul serverului MySQL. Această setare relaxează temporar restricțiile de securitate pentru funcțiile stocate; setarea va fi dezactivată la pașii următori. Aceasta permite utilizatorului bazei de date Zabbix să creeze funcții stocate fără privilegiul `SUPER`, necesar în MySQL atunci când este activată înregistrarea binară (implicit începând cu MySQL 8.0). Executați următoarea comandă:

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

6. Porniți un container Zabbix server și conectați-l la containerele Java Gateway și serverului MySQL:

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

7. Urmăriți jurnalele Zabbix server și așteptați până când Zabbix server finalizează inițializarea schemei bazei de date (utilizați `Ctrl+C` pentru a ieși din jurnale):

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

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

8. Dezactivați `log_bin_trust_function_creators` în containerul serverului MySQL. Aceasta reactivează restricția de securitate care împiedică utilizatorii fără `SUPER` să creeze funcții stocate. Executați următoarea comandă:

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

9. Porniți un container pentru frontend-ul web Zabbix și conectați-l la containerele Zabbix server și serverului MySQL:

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

După ce containerele sunt pornite și funcționează, Zabbix server pornește, iar frontend-ul web devine disponibil pentru autentificare.

> **Note:**
> Utilizați `docker ps` pentru a verifica starea containerelor. Toate containerele trebuie să aibă starea `Up`. Dacă vreun container afișează `Exited`, verificați jurnalele acestuia pentru erori folosind `docker logs -f <container-name>`.

Implementare Zabbix server (PostgreSQL) cu capcane SNMP

Acest exemplu arată cum se implementează un Zabbix server cu un backend PostgreSQL, capcane SNMP și o interfață web bazată pe NGINX.

1. Creați o rețea Docker dedicată, astfel încât toate containerele componentelor Zabbix să se poată conecta între ele folosind numele containerelor:

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

2. Porniți un container gol pentru serverul PostgreSQL (înlocuiți `zabbix_pwd` cu o parolă puternică și utilizați această valoare în pașii următori):

```
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. Porniți un container Zabbix pentru capcane SNMP:

```
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. Porniți un container Zabbix server și conectați-l la containerele serverului PostgreSQL și capcanelor SNMP:

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

5. Urmăriți jurnalele Zabbix server și așteptați până când Zabbix server finalizează inițializarea schemei bazei de date (utilizați `Ctrl+C` pentru a ieși din jurnale):

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

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

6. Porniți un container pentru frontend-ul web Zabbix și conectați-l la containerele Zabbix server și serverului PostgreSQL:

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

După ce containerele sunt pornite și funcționează, Zabbix server pornește, iar frontend-ul web devine disponibil pentru autentificare.

Note:
Utilizați `docker ps` pentru a verifica starea containerelor. Toate containerele trebuie să aibă starea `Up`. Dacă vreun container afișează `Exited`, verificați jurnalele acestuia pentru erori folosind `docker logs -f <container-name>`.

Implementare Zabbix server (MySQL) cu Java gateway pe RHEL 8-10

Acest exemplu arată cum se implementează un Zabbix server cu un backend MySQL, un Java gateway pentru monitorizarea JMX și o interfață web bazată pe NGINX, toate rulând pe Red Hat Enterprise Linux 8, 9 sau 10.

Pe Red Hat Enterprise Linux, runtime-ul de containere recomandat este Podman în locul Docker. Podman funcționează similar cu Docker, dar nu necesită un serviciu de fundal care să ruleze ca root, ceea ce îl face mai potrivit pentru mediile Red Hat.

1. Creați un pod nou cu numele `zabbix` și porturile expuse pentru frontend-ul web Zabbix și trapper-ul Zabbix server:

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

2. (Opțional) Porniți un container Zabbix agent în podul `zabbix`:

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

3. Creați directorul `./mysql/` pe hostul RHEL:

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

4. Porniți un container gol pentru serverul MySQL (înlocuiți `zabbix_pwd` și `root_pwd` cu parole puternice și utilizați aceste valori în pașii următori):

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

5. Urmăriți jurnalele MySQL și așteptați până când MySQL este pregătit să accepte conexiuni (utilizați `Ctrl+C` pentru a ieși din jurnale):

```
podman logs -f mysql-server

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

6. Porniți un container Zabbix Java gateway:

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

7. Activați `log_bin_trust_function_creators` în containerul serverului MySQL. Această setare relaxează temporar restricțiile de securitate pentru funcțiile stocate; setarea va fi dezactivată la pașii următori. Aceasta permite utilizatorului bazei de date Zabbix să creeze funcții stocate fără privilegiul `SUPER`, necesar în MySQL atunci când este activată înregistrarea binară (implicit începând cu MySQL 8.0). Executați următoarea comandă:

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

8. Porniți un container Zabbix server și conectați-l la Java Gateway și la containerele serverului MySQL:

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

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

9. Urmăriți jurnalele Zabbix server și așteptați până când Zabbix server finalizează inițializarea schemei bazei de date (utilizați `Ctrl+C` pentru a ieși din jurnale):

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

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

10. Dezactivați `log_bin_trust_function_creators` în containerul serverului MySQL. Aceasta reactivează restricția de securitate care împiedică utilizatorii fără `SUPER` să creeze funcții stocate. Executați următoarea comandă:

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

11. Porniți un container pentru frontend-ul web Zabbix și conectați-l la containerele Zabbix server și serverului MySQL:

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

> **Note:**
> Podul `zabbix` expune portul `80/TCP` (HTTP) către mașina host, de la portul `8080/TCP` al containerului `zabbix-web-mysql`.

După ce containerele sunt pornite și funcționează, Zabbix server pornește, iar frontend-ul web devine disponibil pentru autentificare.

> **Note:**
> Utilizați `docker ps` pentru a verifica starea containerelor. Toate containerele trebuie să aibă starea `Up`. Dacă vreun container afișează `Exited`, verificați jurnalele acestuia pentru erori folosind `docker logs -f <container-name>`.

Installazione dai sorgenti

Puoi ottenere la versione più recente di Zabbix compilandola dai sorgenti. Vedi anche Ottenere il codice sorgente di Zabbix.

Qui è disponibile una guida passo passo per installare Zabbix dai sorgenti.

Installazione dei daemon Zabbix

1 Scaricare l'archivio dei sorgenti

Andare alla pagina di download di Zabbix e scaricare l'archivio dei sorgenti. Una volta scaricato, estrarre i sorgenti eseguendo:

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

> **Note:**
> Inserire nel comando la versione corretta di Zabbix. Deve corrispondere al nome dell'archivio scaricato.

2 Creare un account utente

Tutti i processi daemon di Zabbix vengono eseguiti con utenti di sistema senza privilegi.
Se un daemon di Zabbix viene avviato da un account utente senza privilegi, continuerà a essere eseguito come quell'utente.

Nella configurazione predefinita, se un daemon viene avviato come `root`, passerà all'account utente `zabbix`, che deve essere presente. Per creare un utente e un gruppo di sistema `zabbix`, eseguire i comandi elencati di seguito.

Sistema basato su RedHat:

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

Sistema basato su Debian:

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

Non è necessario creare un account utente separato per il frontend di Zabbix.

Raccomandazione di sicurezza

Se lo server e l'agent di Zabbix vengono eseguiti sulla stessa macchina, si consiglia di eseguirli con **account utente separati**. Eseguire entrambi con lo stesso utente consente all'agent di accedere al file di configurazione del server, il che potrebbe esporre informazioni sensibili, come la password del database, a qualsiasi utente di livello Admin in Zabbix.

> **Attention:**
> Eseguire Zabbix come `root`, `bin` o qualsiasi altro account con privilegi speciali rappresenta un rischio per la sicurezza.

Directory home (opzionale)

I processi di Zabbix non richiedono una directory home, quindi in genere non è consigliato crearne una. Tuttavia, se è necessaria una funzionalità che richiede una directory home (ad esempio, per memorizzare le credenziali MySQL in $HOME/.my.cnf), è possibile crearla utilizzando i comandi elencati di seguito.

Sui sistemi basati su RedHat, eseguire:

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

Sui sistemi basati su Debian, eseguire:

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

3 Creare il database Zabbix

Per i demoni Zabbix server e proxy, così come per il frontend Zabbix, è richiesto un database. Non è necessario per eseguire Zabbix agent.

Sono forniti script SQL per creare lo schema del database e inserire il set di dati. Il database di Zabbix proxy richiede solo lo schema, mentre il database di Zabbix server richiede anche il set di dati oltre allo schema.

Dopo aver creato un database Zabbix, procedere con i passaggi successivi della compilazione di Zabbix.

4 Configurazione dei sorgenti

Per compilare Zabbix server, Zabbix proxy o Zabbix agent è richiesto C99 con estensioni GNU. Questa versione può essere specificata esplicitamente impostando CFLAGS="-std=gnu99":

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

> **Note:**
> Se si esegue l'installazione dal repository Git di Zabbix, è necessario eseguire prima:
> `./bootstrap.sh`

Quando si configurano i sorgenti per un Zabbix server o proxy, è necessario specificare il tipo di database da utilizzare. È possibile compilare un solo tipo di database alla volta con un processo server o proxy.

Per visualizzare tutte le opzioni di configurazione supportate, eseguire il comando seguente nella directory dei sorgenti di Zabbix estratta:

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

Per configurare i sorgenti per un Zabbix server e agent, è possibile eseguire un comando simile al seguente:

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

Per configurare i sorgenti per un Zabbix server (con PostgreSQL ecc.), è possibile eseguire:

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

Per configurare i sorgenti per un Zabbix proxy (con SQLite ecc.), è possibile eseguire:

```
./configure --prefix=/usr --enable-proxy --with-net-snmp --with-sqlite3 --with-ssh2
```

Per configurare i sorgenti per un Zabbix agent, è possibile eseguire:

```
./configure --enable-agent
```

oppure, per Zabbix agent 2:

```
./configure --enable-agent2
```

Note:
Per compilare Zabbix agent 2 è necessario installare una versione di Go supportata.

Note sulle opzioni di compilazione:

- `--enable-agent` - compila Zabbix agent, nonché le utilità da riga di comando Zabbix get e Zabbix sender.
- `--with-libcurl` - richiesto per il monitoraggio delle macchine virtuali, l'autenticazione SMTP e gli item di Zabbix agent `web.page.*`, nonché per gli item di query di telemetria. Vedere anche: Requisiti (libcurl).
- `--with-libxml2` - richiesto per il monitoraggio delle macchine virtuali.
- `--with-libpcre2[=DIR]` - Zabbix viene sempre compilato con la libreria PCRE2; questa opzione consente solo di specificare un percorso personalizzato per l'installazione di PCRE2.
- `--with-mysql=/path/to/mysql_config` - specifica il percorso alla configurazione di una particolare libreria client MySQL. Utile quando sono installate più versioni di MySQL o MariaDB.
- `--enable-static` - collega staticamente le librerie (non supportato su Solaris). Utilizzare questa opzione se si prevede di distribuire i binari compilati su sistemi privi delle librerie richieste. Non è consigliata durante la compilazione di Zabbix server. Per compilare staticamente il server, è necessaria una versione statica di ogni libreria esterna. Lo script configure non verifica automaticamente questo requisito.
- `--with-stacksize=<value>` - imposta la dimensione dello stack per thread in kilobyte (ad esempio, `--with-stacksize=512`). È possibile aumentare questo valore se Zabbix si arresta in modo anomalo o si blocca a causa di overflow dello stack (ad esempio, durante il preprocessing su sistemi con limiti predefiniti ridotti per lo stack dei thread).
- `--with-apm` (utilizzare solo con `--enable-proxy`) - richiesto per il monitoraggio delle prestazioni delle applicazioni (APM) sul proxy Zabbix. Vedere anche: Requisiti (g++, gRPC, protobuf, Abseil).

Attention:
Se `./configure` non riesce a causa di librerie mancanti o di altri problemi, controllare il file `config.log` per informazioni dettagliate sull'errore.
Ad esempio, se manca `libssl`, il messaggio di errore immediato potrebbe essere fuorviante:

```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```

In questo caso, `config.log` mostra la causa effettiva:

```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Vedere anche:

- Compilazione di Zabbix con supporto per la crittografia
- Problemi di compilazione noti.

5 Creare e installare tutto

Note:
Se si installa da Zabbix Git repository, è necessario eseguire prima:

```
$ make dbschema
```

```
make install
```

Questo passaggio deve essere eseguito da un utente con permessi sufficienti (in genere 'root', oppure usando `sudo`).

L'esecuzione di `make install` installerà per impostazione predefinita i binari del daemon (zabbix_server, zabbix_agentd, zabbix_proxy) in /usr/local/sbin e i binari client (zabbix_get, zabbix_sender) in /usr/local/bin.

Note:
Per specificare una posizione diversa da `/usr/local`, usare una chiave `--prefix` nel passaggio precedente di configurazione dei sorgenti, ad esempio `--prefix=/home/zabbix`. In questo caso i binari del daemon verranno installati in `<prefix>/sbin`, mentre le utility in `<prefix>/bin`. Le pagine man verranno installate in `<prefix>/share`.

6 Esaminare e modificare i file di configurazione

- modificare il file di configurazione dell'agent di Zabbix (`zabbix_agentd.conf`).

È necessario configurare questo file per ogni host in cui è installato `zabbix_agentd`.

Nel file è necessario specificare l'**indirizzo IP** del server Zabbix. Le connessioni provenienti da altri host verranno rifiutate.

- modificare il file di configurazione del server di Zabbix (`zabbix_server.conf`).

È necessario specificare il nome del database, l'utente e la password (se utilizzata).

Il resto dei parametri sarà adeguato con i valori predefiniti se si dispone di una piccola installazione (fino a dieci host monitorati). Tuttavia, è consigliabile modificare i parametri predefiniti se si desidera massimizzare le prestazioni del server (o del proxy) Zabbix.

- se è stato installato un proxy Zabbix, modificare il file di configurazione del proxy (`zabbix_proxy.conf`).

È necessario specificare l'indirizzo IP del server e il nome host del proxy (che deve essere noto al server), nonché il nome del database, l'utente e la password (se utilizzata).

Note:
Con SQLite è necessario specificare il percorso completo del file del database; l'utente e la password del DB non sono richiesti.

7 Avviare i daemon

Eseguire zabbix_server sul lato server.

`zabbix_server`

Note:
Assicurarsi che il sistema consenta l'allocazione di 36 MB (o poco più) di memoria condivisa, altrimenti il server potrebbe non avviarsi e nel file di log del server verrà visualizzato "Cannot allocate shared memory for <type of cache>.". Questo può accadere su FreeBSD, Solaris 8.

Eseguire zabbix_agentd su tutte le macchine monitorate.

`zabbix_agentd`

Note:
Assicurarsi che il sistema consenta l'allocazione di 2 MB di memoria condivisa, altrimenti l'agent potrebbe non avviarsi e nel file di log dell'agent verrà visualizzato "Cannot allocate shared memory for collector.". Questo può accadere su Solaris 8.

Se è stato installato Zabbix proxy, eseguire zabbix_proxy.

`zabbix_proxy`

Installazione dell'interfaccia web di Zabbix

Copia dei file PHP

Il frontend di Zabbix è scritto in PHP, quindi per eseguirlo è necessario un webserver con supporto PHP. L'installazione si esegue semplicemente copiando i file PHP dalla directory ui nella directory dei documenti HTML del webserver.

Le posizioni comuni delle directory dei documenti HTML per i web server Apache includono:

- /usr/local/apache2/htdocs (directory predefinita quando si installa Apache dai sorgenti)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

Si consiglia di usare una sottodirectory invece della root HTML. Per creare una sottodirectory e copiare i file del frontend di Zabbix al suo interno, eseguire i seguenti comandi, sostituendo <htdocs> con la directory effettiva:

```
mkdir <htdocs>/zabbix
cd ui
cp -a . <htdocs>/zabbix
```

Se si prevede di usare una lingua diversa dall'inglese, vedere Installazione di lingue aggiuntive del frontend per le istruzioni.

Installazione del frontend

Per informazioni sulla procedura guidata di installazione del frontend di Zabbix, consultare la pagina Installazione dell'interfaccia web.

Installazione dei plugin caricabili di Zabbix agent 2

L'installazione dei plugin caricabili di Zabbix agent 2 è necessaria solo se si desidera monitorare target non coperti dai plugin integrati (ad esempio server o cluster MongoDB, PostgreSQL e i suoi fork, ecc.). Consultare l'elenco completo dei plugin caricabili e dei plugin integrati.

Attention:
Prima di installare un plugin, controllarne il file README. Può contenere requisiti specifici e istruzioni di installazione.

Per installare dai sorgenti, prima scaricare ed estrarre l'archivio sorgente del plugin caricabile.

Per compilare il plugin, accedere alla directory del plugin estratta ed eseguire `make`:

```
make
```

Note:
Per compilare i plugin caricabili di Zabbix agent 2 deve essere installata una versione supportata di Go.

L'eseguibile del plugin può essere collocato in qualsiasi posizione, purché sia caricabile da Zabbix agent 2. Specificare il percorso del binario del plugin nel file di configurazione del plugin, ad esempio in `postgresql.conf` per il plugin PostgreSQL:

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Il percorso del file di configurazione del plugin deve essere specificato nel parametro `Include` del file di configurazione di Zabbix agent 2:

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Per ulteriori dettagli sulla configurazione dei plugin, procedere alla sezione setup.

I plugin caricabili forniti da Zabbix utilizzano semplici makefile con i seguenti target di compilazione:

- `make` - compila il plugin
- `make clean` - elimina tutti i file creati durante la compilazione del plugin
- `make check` - esegue i self-test (richiede un target di monitoraggio reale, ad esempio un database PostgreSQL)
- `make style` - verifica lo stile del codice Go con `golangci-lint`
- `make format` - formatta il codice Go con `go fmt`
- `make dist` - crea un archivio sorgente includendo tutte le dipendenze

Installazione di Java gateway

È necessario installare Java gateway solo se si desidera monitorare applicazioni JMX. Java gateway è leggero e non richiede un database.

Per installare dai sorgenti, prima scaricare ed estrarre l'archivio dei sorgenti.

Per compilare Java gateway, eseguire lo script `./configure` con l'opzione `--enable-java`. Si consiglia di specificare l'opzione `--prefix` per richiedere un percorso di installazione diverso da quello predefinito /usr/local, perché l'installazione di Java gateway creerà un'intera struttura di directory, non solo un singolo eseguibile.

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

Per compilare e impacchettare Java gateway in un file JAR, eseguire `make`. Si noti che per questo passaggio saranno necessari gli eseguibili `javac` e `jar` nel percorso.

```
make
```

Ora si dispone di un file zabbix-java-gateway-$VERSION.jar in `src/zabbix_java/bin`. Se si preferisce eseguire Java gateway da `src/zabbix_java` nella directory di distribuzione, è possibile procedere alle istruzioni per configurare ed eseguire Java gateway. In caso contrario, assicurarsi di avere privilegi sufficienti ed eseguire `make install`.

```
make install
```

Procedere a setup per ulteriori dettagli sulla configurazione e sull'esecuzione di Java gateway.

Installazione di Zabbix web service

L'installazione di Zabbix web service è necessaria solo se si desidera utilizzare i report pianificati.

Per installare dai sorgenti, prima scaricare ed estrarre l'archivio dei sorgenti.

Per compilare Zabbix web service, eseguire lo script `./configure` con l'opzione `--enable-webservice`.

> **Note:**
> Per compilare Zabbix web service deve essere installata una versione supportata di Go.

Eseguire zabbix_web_service sulla macchina in cui è installato il web service:

```
zabbix_web_service
```

Procedere alla configurazione per ulteriori dettagli sulla configurazione della generazione dei report pianificati.

Installazione del server MCP di Zabbix

Per installare dai sorgenti, prima scaricare ed estrarre l'archivio dei sorgenti.

Per compilare il server MCP di Zabbix, eseguire il comando seguente nella directory dei sorgenti del server MCP:

```
make build
```

> **Note:**
> Per compilare il server MCP di Zabbix, è necessario installare una versione di Go supportata.

Per riportare una versione specifica nell'output di `zabbix_mcp_server -V` e nell'`User-Agent` delle richieste API di Zabbix, impostare la variabile `VERSION`, ad esempio `make build VERSION=8.0.0`.

Per installare il binario in `/usr/local/sbin` e il file di configurazione di esempio in `/etc/zabbix`, eseguire:

```
sudo make install
```

Modificare il file di configurazione `/etc/zabbix/zabbix_mcp_server.conf` e impostare almeno i seguenti parametri:

- `FrontendURL` - l'URL dell'API di Zabbix, che termina con `api_jsonrpc.php`;
- `AllowedIP` - gli indirizzi dei client MCP autorizzati a connettersi.

Per crittografare le connessioni dai client MCP, impostare `TLSAccept=cert` e specificare `TLSCertFile` e `TLSKeyFile`, oppure eseguire il server MCP dietro un reverse proxy affidabile che gestisca TLS.

Se l'utente `zabbix` non esiste, ad esempio se sulla macchina è installato solo il server MCP, crearlo come descritto in Creare un account utente.

Convalidare il file di configurazione e avviare zabbix_mcp_server:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Consultare Server MCP per ulteriori dettagli sulla configurazione del server e sulla connessione dei client MCP.

Ottenere il codice sorgente di Zabbix

Esistono diversi modi per ottenere il codice sorgente di Zabbix:

- È possibile scaricare le versioni stabili rilasciate dal sito web ufficiale di Zabbix
- È possibile scaricare le build notturne dalla pagina sviluppatori del sito web ufficiale di Zabbix
- È possibile ottenere l'ultima versione di sviluppo dal sistema di repository del codice sorgente Git:
  - La posizione principale del repository completo è https://git.zabbix.com/scm/zbx/zabbix.git
  - Il ramo master e le release supportate sono inoltre replicate su Github all'indirizzo https://github.com/zabbix/zabbix

Per clonare il repository deve essere installato un client Git. Il pacchetto ufficiale del client Git da riga di comando è comunemente chiamato **git** nelle distribuzioni. Per installarlo, ad esempio, su Debian/Ubuntu, eseguire:

```
sudo apt-get update
sudo apt-get install git
```

Per ottenere tutto il codice sorgente di Zabbix, spostarsi nella directory in cui si desidera collocare il codice ed eseguire:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Problemi di compilazione

Questi sono i problemi noti relativi alla compilazione di Zabbix dai sorgenti. Per tutti gli altri casi, vedere la pagina Problemi noti.

Libreria in una posizione non standard

Zabbix consente di specificare una libreria situata in una posizione non standard. Nell'esempio seguente, Zabbix eseguirà `curl-config` dalla posizione non standard specificata e utilizzerà il suo output per determinare la libcurl corretta da usare.

```
$ ./configure --enable-server --with-mysql --with-libcurl=/usr/local/bin/curl-config
```

Questo funzionerà se è l'unica libcurl installata nel sistema, ma potrebbe non funzionare se è presente un'altra libcurl installata in una posizione standard (ad esempio dal gestore pacchetti). Questo è il caso in cui per Zabbix è necessaria una versione più recente della libreria e quella più vecchia serve per altre applicazioni.

Pertanto, specificare un componente in una posizione non standard non funzionerà sempre quando lo stesso componente esiste anche in una posizione standard.

Ad esempio, se si utilizza una libcurl più recente installata in `/usr/local` mentre il pacchetto libcurl è ancora installato, Zabbix potrebbe selezionare quella sbagliata e la compilazione non riuscirà:

```
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
```

Qui, la funzione `curl_easy_header()` non è disponibile nella vecchia `/usr/lib/x86_64-linux-gnu/libcurl.so`, ma è disponibile nella più recente `/usr/local/lib/libcurl.so`.

Il problema riguarda l'ordine dei flag del linker e una soluzione consiste nello specificare il percorso completo della libreria in una variabile LDFLAGS:

```
$ LDFLAGS="-Wl,--no-as-needed /usr/local/lib/libcurl.so" ./configure --enable-server --with-mysql --with-l
```

Si noti l'opzione `-Wl,--no-as-needed`, che potrebbe essere necessaria su alcuni sistemi (vedere anche: opzioni di linking pre-definite sui sistemi basati su Debian).

Dimensione dello stack troppo piccola su alcuni sistemi

Se Zabbix si arresta in modo anomalo o si blocca a causa di overflow dello stack, è possibile aumentare la dimensione dello stack per thread usando l'opzione `--with-stacksize` durante la configurazione dei sorgenti. Questo problema può verificarsi su sistemi con limiti predefiniti bassi per lo stack dei thread, soprattutto durante la pre-elaborazione, quando vengono creati più thread.

L'esempio seguente imposta la dimensione dello stack a 512 KB per thread:

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

È possibile verificare i limiti dello stack dei thread del sistema in fase di esecuzione usando il comando `ulimit -s` sui sistemi basati su Linux.

Note sull'installazione dai pacchetti

Panoramica

I pacchetti ufficiali **pre-release** di Zabbix 8.0 sono disponibili sul sito web di Zabbix, dove è possibile selezionare il sistema operativo e il componente Zabbix per generare le istruzioni e i comandi di installazione.

Oltre alle note sull'installazione dei pacchetti riportate di seguito, consultare anche la pagina Installazione e configurazione.

I pacchetti sono disponibili per le seguenti distribuzioni Linux:

- Red Hat Enterprise Linux e 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:
Alcune distribuzioni del sistema operativo, in particolare quelle basate su Debian, forniscono pacchetti Zabbix propri. Questi pacchetti **non** sono supportati da Zabbix e potrebbero essere obsoleti o privi delle funzionalità e delle correzioni di bug più recenti. Si consiglia di utilizzare esclusivamente i pacchetti ufficiali dello Zabbix Official Repository. Se Zabbix è stato installato in precedenza dal repository del sistema operativo, consultare i passaggi per aggiornare i pacchetti Zabbix dai repository del sistema operativo.

I pacchetti supportano i database MySQL/PostgreSQL e i web server Apache/Nginx. Si noti che il server Zabbix e il proxy non possono condividere lo stesso database; se entrambi sono installati sullo stesso host, utilizzare nomi di database diversi.

Se necessario, pacchetti separati per Zabbix agent/agent 2, Zabbix get e Zabbix sender sono disponibili presso lo Zabbix Official Repository.

Nei sistemi basati su Debian o Ubuntu, l'installazione di un pacchetto Zabbix aggiuntivo con `apt` potrebbe selezionare anche gli aggiornamenti per i pacchetti Zabbix già installati dallo stesso repository. Per installare un pacchetto senza aggiornare un pacchetto server Zabbix o proxy già installato, bloccare temporaneamente tale pacchetto prima dell'installazione e sbloccarlo al termine:

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

Zabbix fornisce inoltre binari precompilati di Zabbix agent per sistemi operativi non Linux; consultare:

- Installazione di Zabbix agent per Windows da MSI
- Installazione di Zabbix agent per macOS da PKG
- Zabbix agent su IBM AIX
- Binari legacy (per sistemi meno recenti o meno comuni, come HP-UX, NetBSD, Tru64 e versioni precedenti di SLES)

Note sull'installazione dei pacchetti

Le seguenti note si applicano a tutti i sistemi:

- Se si utilizza PostgreSQL, `DBHost=localhost` (o un indirizzo IP) nella configurazione di Zabbix server/proxy fa sì che PostgreSQL utilizzi un socket di rete invece di un socket UNIX locale; consultare Configurazione di SELinux per le istruzioni di configurazione correlate.
- Se si utilizza TimescaleDB, consultare la configurazione aggiuntiva di TimescaleDB.
- Se si installa Zabbix Java gateway (per il monitoraggio di applicazioni JMX), consultare le istruzioni di configurazione aggiuntive per i sistemi basati su RHEL e i sistemi basati su Debian.
- Se si installa Zabbix MCP server, installare il pacchetto `zabbix-mcp-server` e aggiornare il file di configurazione (almeno `FrontendURL` e `AllowedIP`) prima di avviare il servizio.
- Per eseguire Zabbix agent come root, consultare Esecuzione dell'agent come root.
- Il monitoraggio delle prestazioni delle applicazioni (APM) sul proxy Zabbix è supportato dai seguenti pacchetti:
  - RHEL 10 e i suoi derivati: installare il pacchetto APM del proxy Zabbix (ad esempio, `zabbix-proxy-mysql-apm` invece di `zabbix-proxy-mysql`); è inoltre necessario installare il repository EPEL e abilitare i repository dei pacchetti di dipendenze CRB.
  - Debian 12 o versioni successive, Raspberry Pi OS (Raspbian) 12 o versioni successive, Ubuntu 24.04 o versioni successive: APM è incluso nei pacchetti del proxy Zabbix.
  - Sugli altri sistemi, è possibile installare il proxy Zabbix dai sorgenti con l'opzione `--with-apm`.

Le seguenti note si applicano a RHEL e ai suoi derivati:

- Se è stato abilitato il repository EPEL (ad esempio, per il pacchetto APM del proxy Zabbix), che fornisce anche pacchetti Zabbix, è necessario escluderlo dalla risoluzione dei pacchetti prima di installare i pacchetti Zabbix ufficiali; consultare Installazione accidentale dei pacchetti Zabbix di EPEL.
- Per installare i pacchetti Zabbix negli ambienti Red Hat UBI, consultare Pacchetti Zabbix per RHEL negli ambienti Red Hat UBI.
- Per utilizzare gli item ping ICMP, sono disponibili anche i pacchetti per `fping` nel repository ufficiale Zabbix.

Configurazione SELinux

Zabbix utilizza la comunicazione inter-processo basata su socket. Sui sistemi in cui Security-Enhanced Linux (SELinux) è abilitato, potrebbe essere necessario aggiungere regole SELinux per consentire a Zabbix di creare/usare socket di dominio UNIX nella directory SocketDir. I file socket sono utilizzati da Zabbix server (alerter, preprocessing, IPMI) e Zabbix proxy (IPMI), e sono presenti mentre il processo è in esecuzione.

Con SELinux abilitato in modalità enforcing, eseguire i seguenti comandi per abilitare la comunicazione tra frontend e server di Zabbix:

Per RHEL 7 (e versioni successive), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (e versioni successive):

```
setsebool -P httpd_can_connect_zabbix on
```

Se il database viene accesso tramite rete (incluso `localhost` per PostgreSQL), consentire anche al frontend di Zabbix di connettersi al database:

```
setsebool -P httpd_can_network_connect_db on
```

Per RHEL precedente alla 7:

```
setsebool -P httpd_can_network_connect on
setsebool -P zabbix_can_network on
```

Dopo aver applicato le impostazioni SELinux, riavviare Apache:

```
systemctl restart httpd
```

In alternativa, è possibile installare un pacchetto `zabbix-selinux-policy` predefinito dal Zabbix Official Repository. Questo pacchetto è fornito per tutte le versioni di OS supportate per semplificare il deployment di Zabbix e impedire agli utenti di disattivare SELinux a causa della complessità della configurazione.
 Per la versione 10 di AlmaLinux, CentOS Stream, RHEL, Rocky Linux e Oracle Linux, la policy SELinux è fornita dal pacchetto `zabbix-selinux-policy-main`; il pacchetto `zabbix-selinux-policy` ha una dipendenza da `zabbix-selinux-policy-main` oppure `selinux-policy-targeted-extra`. Il pacchetto `zabbix-selinux-pol` è in conflitto con `selinux-policy-targeted-extra`; pertanto, entrambi i pacchetti non possono essere installati contemporaneamente.

Attention:
Per la massima sicurezza, si consiglia di impostare configurazioni SELinux personalizzate.

Il pacchetto `zabbix-selinux-policy` (`zabbix-selinux-policy-main` per la versione 10 di AlmaLinux, CentOS Stream, RHEL, Rocky Linux e Oracle Linux) contiene una policy SELinux di base che consente a Zabbix di creare e usare socket e abilita la connessione HTTPd a PostgreSQL (utilizzato da frontend).

Il file sorgente `zabbix_policy.te` contiene le seguenti regole:

```
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;
```

Pacchetti Debuginfo

I pacchetti `Debuginfo` contengono i simboli di debug per i binari di Zabbix. Non sono necessari per una normale installazione o per il funzionamento, ma sono utili per la risoluzione avanzata dei problemi.

Si noti che i pacchetti `Debuginfo` sono disponibili per le seguenti versioni: Red Hat Enterprise Linux 7 e precedenti (incluse le derivate) e SUSE Linux Enterprise Server 15 e precedenti (incluse le derivate).

Per abilitare il repository `zabbix-debuginfo`:

- Su RHEL 7, modificare `/etc/yum.repos.d/zabbix.repo` e impostare `enabled=1` per la sezione `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
```

- Su SUSE, modificare `/etc/zypp/repos.d/zabbix.repo` e impostare `enabled=1` per la sezione `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/$basearch/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/$basearch/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Una volta abilitato, installare i pacchetti:

- Su RHEL, installare un singolo pacchetto con informazioni di debug per tutti i componenti Zabbix:

```
dnf install zabbix-debuginfo
```

- Su SUSE, installare i pacchetti debuginfo specifici per componente:

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

Abilitazione dei repository delle release instabili

Le istruzioni riportate di seguito servono ad abilitare i repository di rilascio Zabbix instabili (disabilitati per impostazione predefinita), utilizzati per le release candidate delle versioni minori di Zabbix.

Per prima cosa, installa o aggiorna il pacchetto zabbix-release all'ultima versione. Per abilitare i pacchetti rc nel tuo sistema, esegui quanto segue:

Red Hat Enterprise Linux

Aprire il file `/etc/yum.repos.d/zabbix.repo` e impostare `enabled=1` per il 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

Aprire `/etc/apt/sources.list.d/zabbix.list` e decommentare "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

Aprire il file `/etc/zypp/repos.d/zabbix.repo` e impostare `enable=1` per il repository `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
```

Installazione dell'agent Windows da MSI

Panoramica

Zabbix agent può essere installato su Windows utilizzando pacchetti di installazione MSI a 32 bit o 64 bit, disponibili per il download.

I requisiti minimi del sistema operativo per l'installazione tramite MSI sono:

- **Per Zabbix agent:** Windows XP (64 bit) o Windows Server 2003
- **Per Zabbix agent 2:** Windows 10 (32 bit) o Windows Server 2016

I pacchetti a 32 bit non possono essere installati su sistemi a 64 bit.

I pacchetti includono:

- supporto TLS (la configurazione TLS è facoltativa)
- utilità Zabbix get e Zabbix sender (possono essere installate insieme a Zabbix agent/agent 2 o separatamente)

Attention:
I pacchetti di Zabbix agent 2 non includono plugin caricabili (MongoDB, PostgreSQL, MSSQL), che devono essere scaricati e installati separatamente.

L'installazione può essere eseguita utilizzando la Procedura guidata di installazione o la riga di comando.

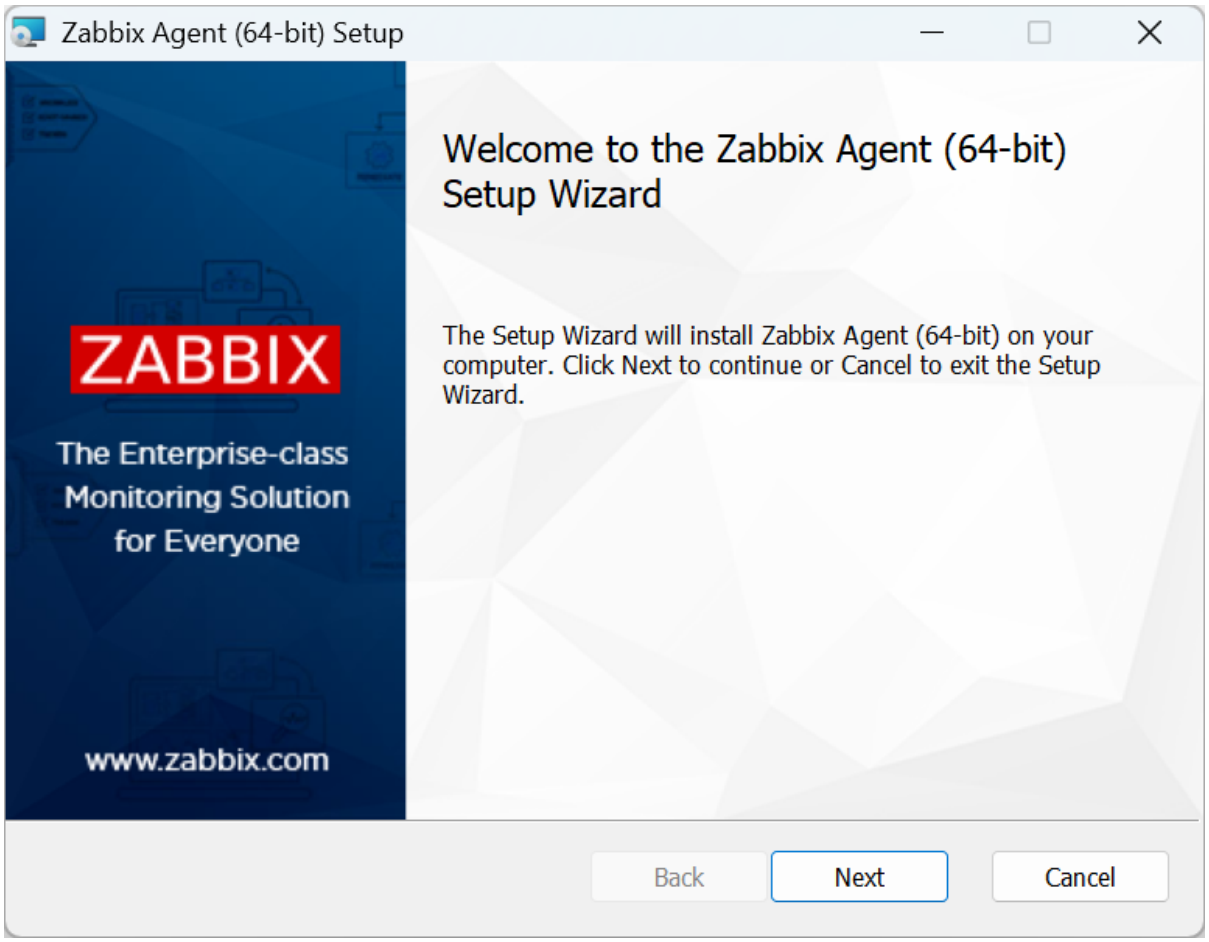
Sebbene l'installazione tramite pacchetti MSI sia pienamente supportata, si consiglia di installare almeno Microsoft .NET Framework 2 per una corretta gestione degli errori.

Attention:
Si consiglia di utilizzare il percorso di installazione predefinito fornito dal programma di installazione. L'utilizzo di un percorso personalizzato senza le autorizzazioni necessarie può compromettere la sicurezza dell'installazione.

Installazione tramite procedura guidata di configurazione

I seguenti passaggi di installazione si applicano sia a Zabbix agent sia a Zabbix agent 2.

1. Fare doppio clic sul file MSI scaricato per avviare l'installazione:

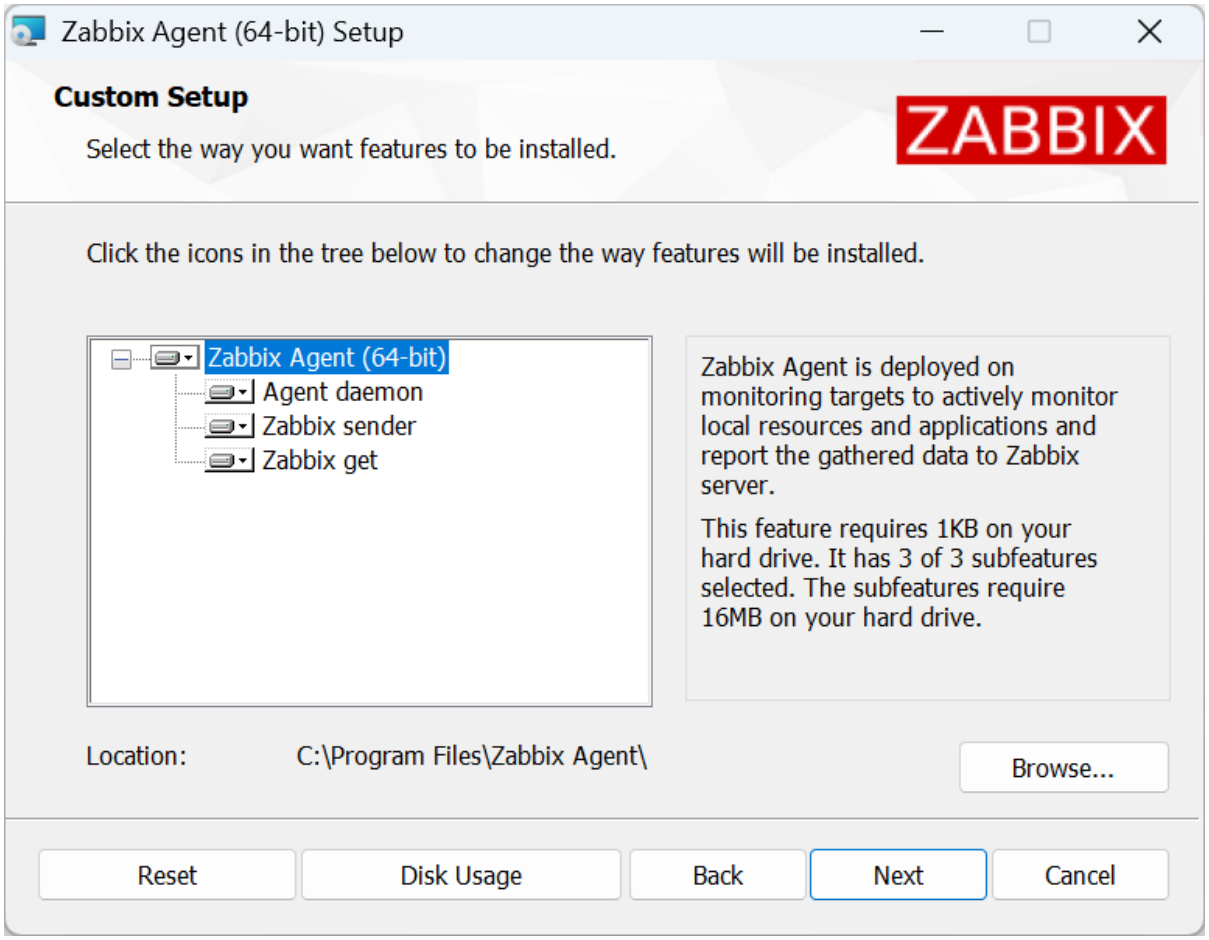


2. Accettare il Contratto di licenza con l'utente finale:



3. Selezionare i componenti Zabbix (demone agent, Zabbix sender, Zabbix get) da installare:

Attention:
Si consiglia di utilizzare il percorso di installazione predefinito fornito dal programma di installazione. L'utilizzo di un percorso personalizzato senza le autorizzazioni necessarie può compromettere la sicurezza dell'installazione.

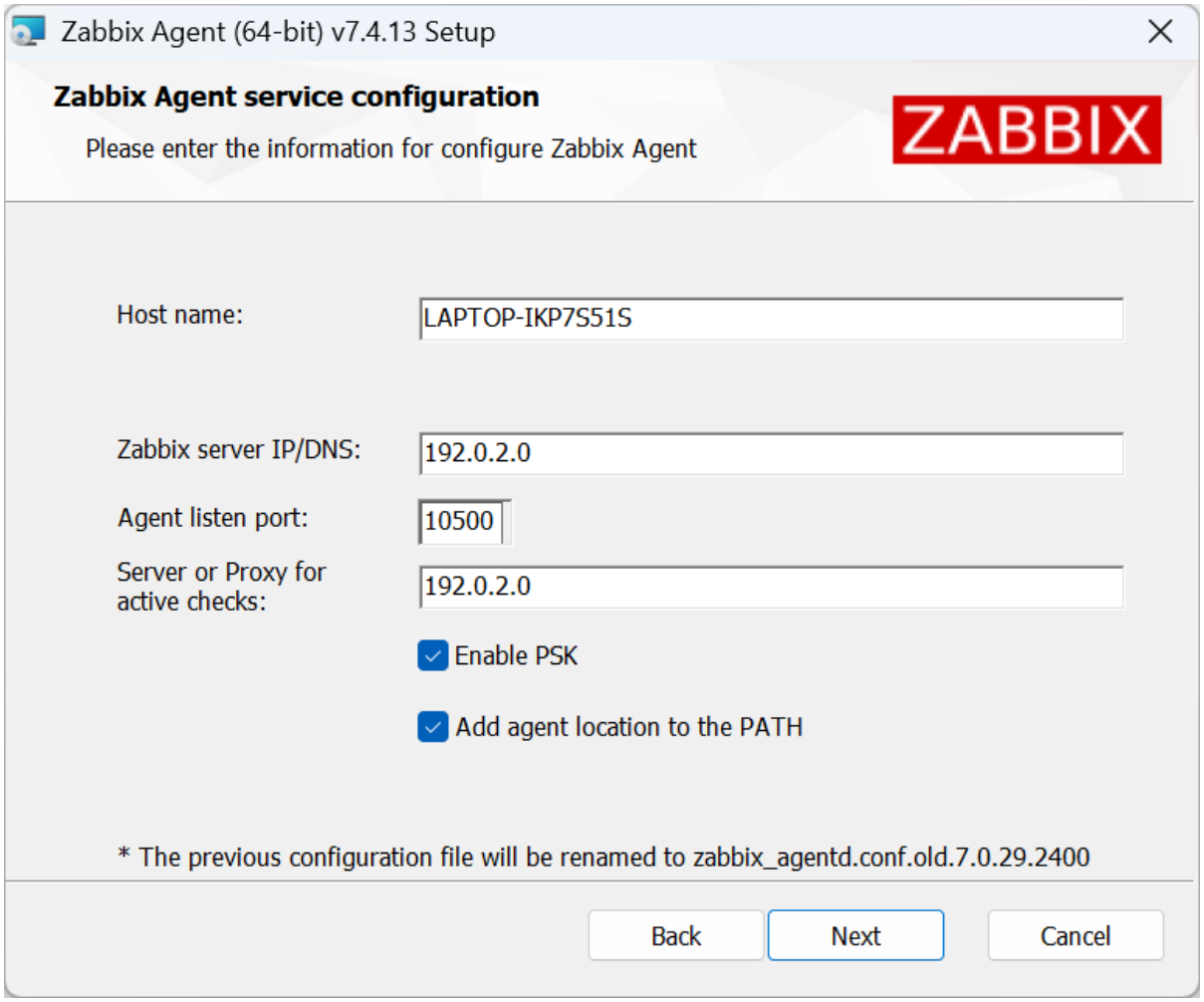


Note:
Se viene utilizzato un percorso personalizzato al di fuori della directory `Program Files` e la cartella di destinazione esiste già, il programma di installazione non ne modificherà le autorizzazioni. Durante una nuova installazione, verrà richiesto di impostare manualmente le autorizzazioni appropriate oppure di continuare senza modificarle.

4. Configurare i seguenti parametri. I relativi valori verranno impostati nel file di configurazione dell'agent Zabbix:

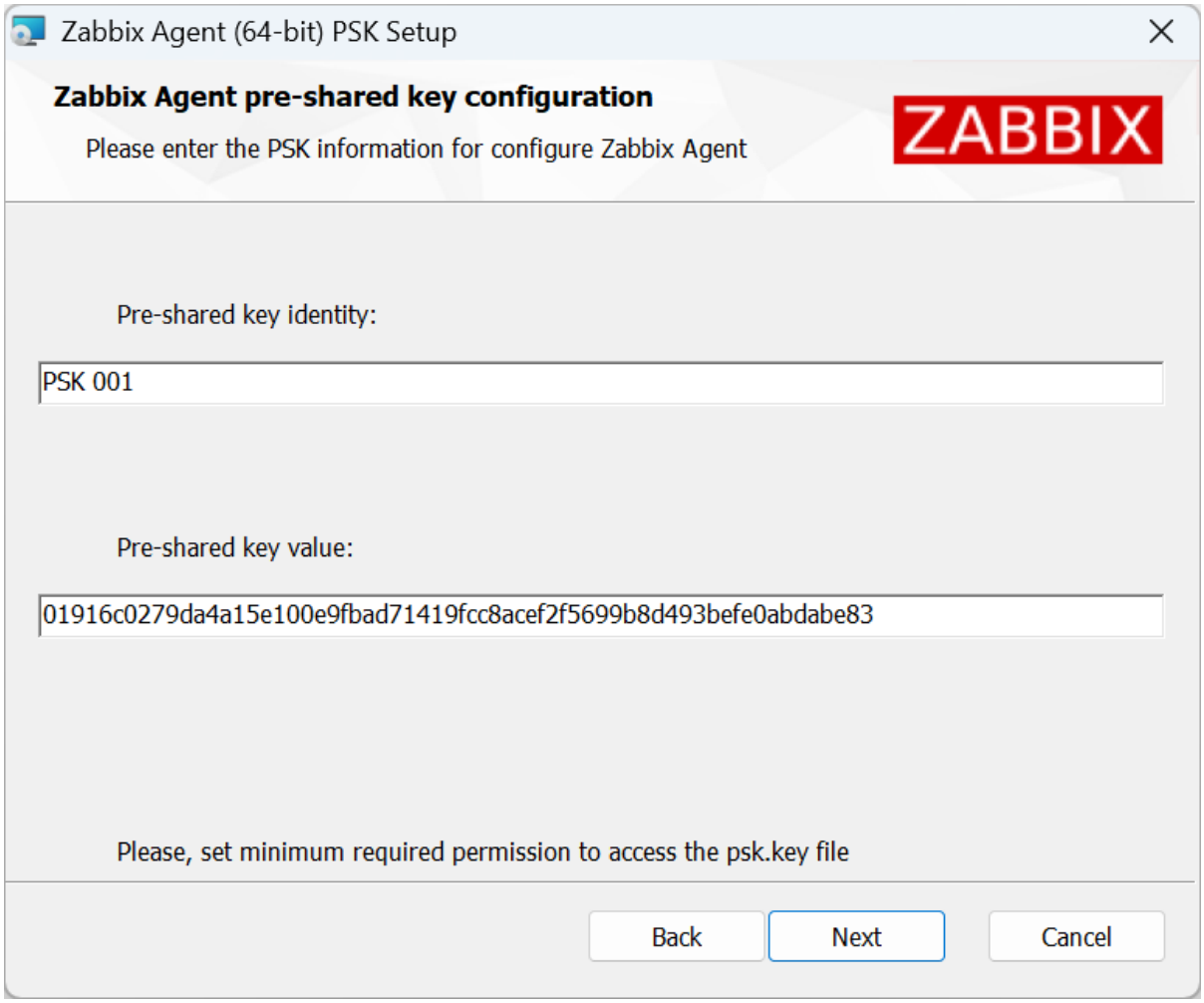
Parametro	Descrizione
Nome host	Il nome host della macchina su cui viene installato Zabbix agent. Imposta il parametro Hostname.
IP/DNS del server Zabbix	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS dei server Zabbix o dei proxy Zabbix. Questo parametro è **obbligatorio**. Imposta il parametro Server.
Porta di ascolto dell'agent	L'agent resterà in ascolto su questa porta per le connessioni dal server. Imposta il parametro ListenPort.
Server o proxy per i controlli attivi	L'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi. L'indirizzo del server/proxy è un indirizzo IP o un nome DNS e una porta facoltativa separata da due punti. Imposta il parametro ServerActive.
Abilita PSK	Selezionare la casella di controllo per abilitare il supporto TLS utilizzando chiavi precondivise. Imposta i parametri TLSConnect e TLSAccept su `psk`.
Aggiungi il percorso dell'agent al PATH	Selezionare la casella di controllo per aggiungere il percorso di Zabbix agent alla variabile PATH di sistema.

Note:
Se viene rilevato un Zabbix agent esistente, verranno visualizzati i valori dei parametri presenti nel relativo file di configurazione. Inoltre, il file di configurazione esistente verrà rinominato durante l'installazione e verrà creato un nuovo file di configurazione.



5. Configurare i parametri PSK se nel passaggio precedente è stata selezionata la casella di controllo *Abilita PSK*. Questi parametri verranno impostati anch'essi nel file di configurazione di Zabbix agent:

Parametro	Descrizione
Identità della chiave precondivisa	La stringa dell'identità della chiave precondivisa. Imposta il parametro TLSPSKIdentity.
Valore della chiave precondivisa	Il valore della stringa della chiave precondivisa. Crea il file `psk.key` contenente la chiave e imposta il parametro TLSPSKFile sul percorso della chiave (predefinito: `C:\Program Files\Zabbix Agent\psk.key`). Si **consiglia** di limitare l'accesso al file della chiave precondivisa modificando le impostazioni di sicurezza del file, in modo che solo Zabbix agent (o l'utente che esegue l'agent) possa leggerlo.

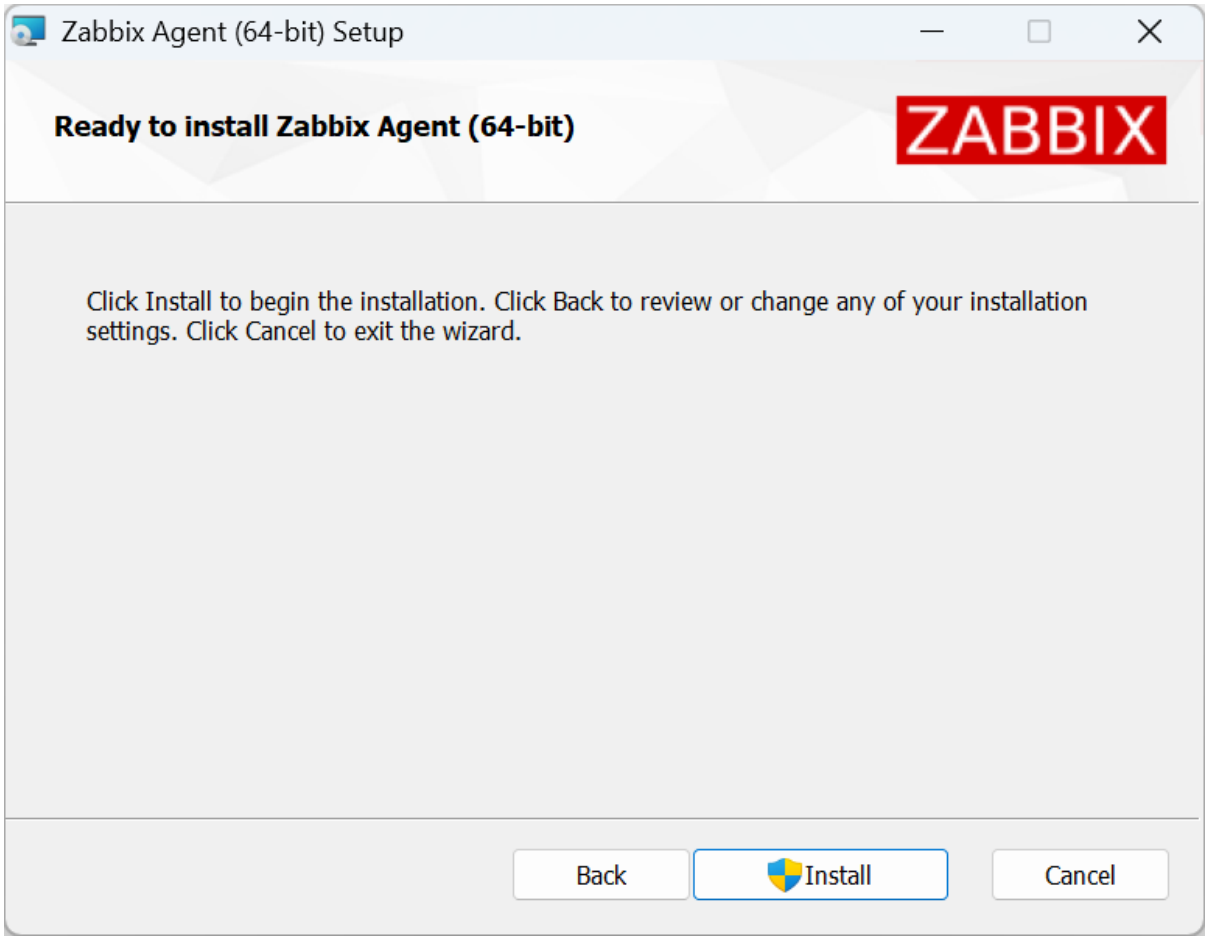


6. Fare clic su *Installa* per iniziare l'installazione.

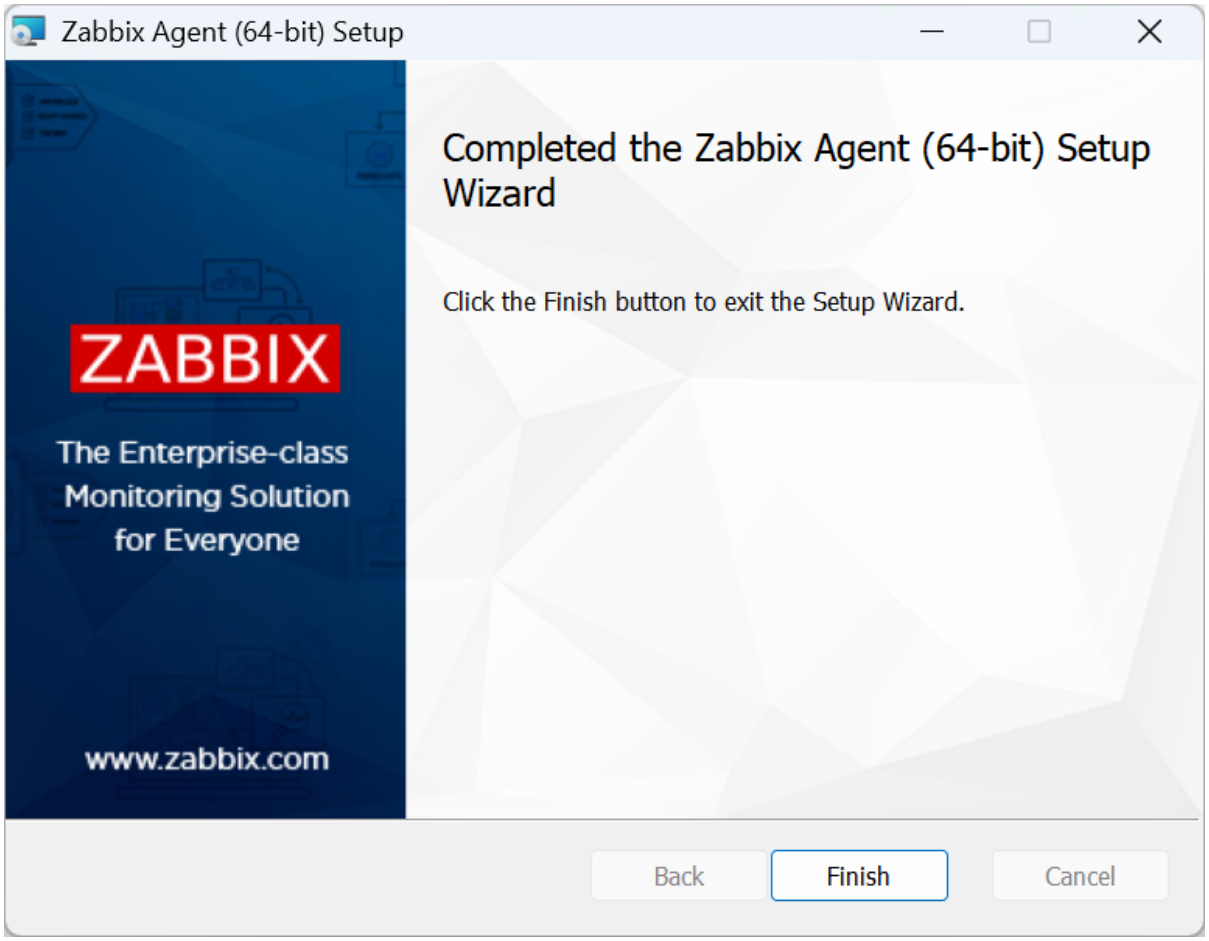
Tutti i componenti Zabbix selezionati e il file di configurazione di Zabbix agent verranno installati nel percorso specificato (predefinito: `C:\Program Files\Zabbix Agent`). Lo stesso vale per Zabbix agent 2, con la differenza che i file di configurazione aggiuntivi per i suoi plugin integrati verranno installati nella sottocartella `zabbix_agent2.d\plugins.d`.

Inoltre, `zabbix_agentd.exe` (o `zabbix_agent2.exe`) verrà configurato come servizio Windows con avvio automatico ritardato (oppure con avvio automatico nelle versioni di Windows precedenti a Windows Vista/Server 2008).

Se durante l'installazione è in esecuzione una versione diversa di Zabbix agent, verrà richiesto di scegliere se chiudere l'applicazione e provare a riavviarla oppure lasciarla aperta; in quest'ultimo caso sarà necessario riavviare il sistema.



7. Fare clic sul pulsante *Fine* per uscire dalla procedura guidata di configurazione.



Installazione da riga di comando

Zabbix agent può essere installato da riga di comando eseguendo il programma di installazione MSI con msiexec. Ad esempio:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

Questo metodo consente installazioni non presidiate e configurazioni personalizzate tramite parametri.

Parametri supportati

I pacchetti di installazione MSI di Zabbix agent supportano i seguenti parametri sia per Zabbix agent sia per Zabbix agent 2.

Note:
I parametri di Zabbix agent/agent2 vengono impostati nel file di configurazione durante l'installazione. Fare clic sul nome di un parametro per visualizzarne la descrizione dettagliata e gli esempi di configurazione nella pagina Zabbix agent (Windows). Per Zabbix agent 2, fare riferimento alla pagina Zabbix agent 2 (Windows).

Parameter	Description
ADDDEFAULT	Un elenco di componenti separati da virgole da installare con la configurazione predefinita. Per ulteriori informazioni, vedere ADDDEFAULT property. Valori possibili: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Esempio: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Un elenco di componenti separati da virgole da installare localmente. Per ulteriori informazioni, vedere ADDLOCAL property. Valori possibili: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Esempio: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Un elenco di parametri AllowKey o DenyKey separati da punto e virgola per limitare i controlli di Zabbix agent. Se necessario, usare una barra rovesciata per eseguire l'escape del delimitatore (`\;`). Imposta i parametri AllowKey e DenyKey nel file di configurazione di agent. Esempio: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	Il percorso completo di un file di configurazione template per Zabbix agent. Durante l'installazione, questo file diventerà il file di configurazione di agent. Il file deve contenere almeno i parametri Server e LogFile. Esempio: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Usare `DONOTSTART=1` per impedire al programma di installazione MSI di avviare il servizio di Zabbix agent.
ENABLEPATH	Usare `ENABLEPATH=1` per aggiungere il percorso di Zabbix agent alla variabile di sistema PATH.
ENABLEPERSISTENTBUFFER	Solo Zabbix agent 2. Abilita l'uso dell'archiviazione persistente locale per gli item attivi.
HOSTINTERFACE	Parametro facoltativo che definisce l'interfaccia dell'host.
HOSTMETADATA	Parametro facoltativo che definisce i metadati dell'host.
HOSTMETADATAITEM	Parametro facoltativo che definisce un item usato per ottenere i metadati dell'host.
HOSTNAME	Parametro facoltativo che definisce il nome dell'host.
HOSTNAMEITEM	Parametro facoltativo che definisce un item usato per ottenere il nome dell'host.
INCLUDE	Un elenco di singoli file separati da punto e virgola o di tutti i file in una directory da includere nel file di configurazione di Zabbix agent.
IGNOREALREADYEXISTSFOLDER	Per le nuove installazioni, usare `IGNOREALREADYEXISTSFOLDER=1` per consentire l'installazione in una cartella di destinazione esistente al di fuori della directory `Program Files`. Se questo parametro non viene specificato, l'installazione non riesce quando tale cartella di destinazione esiste già.
INSTALLFOLDER	Il percorso completo di una cartella in cui verranno installati i componenti di Zabbix e il file di configurazione di Zabbix agent. Per Zabbix agent 2, i file di configurazione aggiuntivi per i plugin integrati verranno installati nella sottocartella `zabbix_agent2.d\plugins.d`. Esempio: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Un elenco di indirizzi IP separati da virgole su cui l'agent deve ascoltare.
LISTENPORT	L'agent ascolterà su questa porta le connessioni dal server.
LOGFILE	Il nome del file di log di Zabbix agent.
LOGTYPE	Il tipo di output del log.
NONMSICONFNAME	Il percorso completo di un file di configurazione personalizzato per Zabbix agent. Durante l'installazione, tutti i parametri di configurazione validi di agent presenti in questo file (limitati a quelli elencati in questa tabella) verranno scritti nel file di configurazione di agent appena creato. Il file deve contenere almeno il parametro Server. Esempio: `NONMSICONFNAME="C:\full\path\to\example.conf"`
PERSISTENTBUFFERFILE	Solo Zabbix agent 2. Il file in cui Zabbix agent 2 deve conservare il database SQLite.

Parameter	Description
PERSISTENTBUFFERPERIOD	Solo Zabbix agent 2. L'intervallo di tempo per cui i dati devono essere memorizzati quando non c'è connessione al server o al proxy.
SERVER	Un elenco di indirizzi IP separati da virgole, opzionalmente in notazione CIDR, oppure nomi DNS di Zabbix server o Zabbix proxy. Questo parametro è **obbligatorio**, tranne quando STARTAGENTS è impostato su 0.
SERVERACTIVE	L'indirizzo di Zabbix server/proxy o la configurazione del cluster da cui ottenere i controlli attivi.
SKIP	Usare `SKIP=fw` per impedire al programma di installazione MSI di aggiungere una regola di eccezione di Windows Firewall per Zabbix agent.
SOURCEIP	L'indirizzo IP di origine per le connessioni in uscita verso Zabbix server o Zabbix proxy, oppure per stabilire connessioni durante l'esecuzione di alcuni item (web.page.get, net.tcp.port, ecc.).
STARTAGENTS	Il numero di istanze pre-fork di zabbix_agentd che elaborano i controlli passivi. Se impostato su 0, i controlli passivi sono disabilitati e l'agent non ascolterà su alcuna porta TCP.
STARTUPTYPE	Tipo di avvio del servizio di Zabbix agent. Valori possibili: **automatic** - avvia il servizio automaticamente all'avvio di Windows; **delayed** - *(predefinito)* ritarda l'avvio del servizio dopo che i servizi avviati automaticamente hanno completato l'avvio (disponibile su Windows Vista/Server 2008 e versioni successive); **manual** - avvia il servizio manualmente (da parte di un utente o di un'applicazione); **disabled** - disabilita il servizio in modo che non possa essere avviato da un utente o da un'applicazione. Esempio: `STARTUPTYPE=disabled`
STATUSPORT	Solo Zabbix agent 2. Se impostato, l'agent ascolterà su questa porta le richieste HTTP di stato (http://localhost:<port>/status).
TIMEOUT	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix proxy o server.
TLSACCEPT	Le connessioni in ingresso da accettare (usato per i controlli passivi). Se impostato su `psk`, anche TLSCONNECT verrà impostato su `psk` (salvo diversa specifica).
TLSCAFILE	Il percorso completo di un file contenente i certificati CA di livello superiore per la verifica del certificato del peer.
TLSCERTFILE	Il percorso completo di un file contenente il certificato di agent o la catena di certificati.
TLSCONNECT	Come l'agent deve connettersi a Zabbix server o proxy (usato per i controlli attivi). Se impostato su `psk`, anche TLSACCEPT verrà impostato su `psk` (salvo diversa specifica).
TLSCRLFILE	Il percorso completo di un file contenente i certificati revocati.
TLSKEYFILE	Il percorso completo di un file contenente la chiave privata di Zabbix agent.
TLSPSKFILE	Il percorso completo di un file contenente la chiave precondivisa di Zabbix agent. Se sia TLSPSKFILE sia TLSPSKVALUE sono impostati, il valore di TLSPSKVALUE verrà scritto nel file specificato in TLSPSKFILE. Si **raccomanda** di limitare l'accesso al file della chiave precondivisa regolando le impostazioni di sicurezza del file in modo che solo Zabbix agent (o l'utente che esegue agent) possa leggerlo.
TLSPSKIDENTITY	La stringa di identità della chiave precondivisa.
TLSPSKVALUE	Il valore della stringa della chiave precondivisa. Se sia TLSPSKFILE sia TLSPSKVALUE sono impostati, il valore di TLSPSKVALUE verrà scritto nel file specificato in TLSPSKFILE. Esempio: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	L'emittente del certificato del server (proxy) consentito.
TLSSERVERCERTSUBJECT	Il soggetto del certificato del server (proxy) consentito.
UNSAFEUSERPARAMETERS	Consente di passare tutti i caratteri negli argomenti ai parametri definiti dall'utente.

Esempi

L'esempio seguente installa Zabbix agent con una configurazione personalizzata. Abilita inoltre il supporto TLS usando chiavi pre-condivise.

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
```

```
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

L'esempio successivo installa una versione più recente di Zabbix agent e usa un **file di configurazione template** (`CONF="C:\agent-template.conf"`). Durante l'installazione, questo file diventerà il file di configurazione dell'agent. Per ereditare i parametri dal vecchio file di configurazione, usa segnaposto di parametro (ad esempio, `[AllowDenyKey]`).

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

#### esempio di agent-template.conf:
LogFile=[LogFile]
[AllowDenyKey]
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
[Include]
[TLSConnect]
[TLSAccept]
[TLSPSKIdentity]
[TLSPSKFile]
```

In alternativa, puoi usare un **file di configurazione personalizzato** (`NONMSICONFNAME="C:\agent-custom.conf"`). Durante l'installazione, tutti i parametri di configurazione validi dell'agent presenti in questo file (limitati a quelli elencati nella tabella sopra) verranno scritti nel nuovo file di configurazione dell'agent creato. Per mantenere la configurazione esistente dell'agent, definisci i parametri da preservare.

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

#### esempio di agent-custom.conf:
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
```

Plugin caricabili di Zabbix agent 2

I plugin caricabili di Zabbix agent 2 possono essere installati su Windows utilizzando pacchetti di installazione MSI a 64 bit, disponibili per il download.

I requisiti minimi del sistema operativo per l'installazione tramite MSI sono Windows 10 (64 bit) o Windows Server 2016.

> **Attention:**
> Prima di installare un plugin, controllare il relativo file README. Potrebbe contenere requisiti specifici e istruzioni di installazione.

Analogamente a Zabbix agent/agent2, i plugin caricabili possono essere installati utilizzando la procedura guidata di installazione o la riga di comando.

Installazione tramite Setup Wizard

1. Fare doppio clic sul file MSI scaricato per avviare l'installazione.
2. Accettare il Contratto di licenza con l'utente finale.
3. Selezionare i plugin caricabili di Zabbix agent 2 da installare.

> **Attention:**
> Si consiglia di utilizzare il percorso di installazione predefinito fornito dal programma di installazione. L'utilizzo di un percorso personalizzato senza le autorizzazioni necessarie può compromettere la sicurezza dell'installazione.

4. Fare clic su *Install* per avviare l'installazione.

Tutti i plugin caricabili di Zabbix agent 2 selezionati verranno installati nel percorso specificato (predefinito: `C:\Program Files\Zabbix Agent 2`), con i relativi file di configurazione installati nella sottocartella `zabbix_agent2.d`.

5. Fare clic sul pulsante *Finish* per uscire dal Setup Wizard.

Installazione da riga di comando

I plugin caricabili di Zabbix agent 2 possono essere installati da riga di comando eseguendo il programma di installazione MSI con `msiexec`. Ad esempio:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent2_plugins-8.0.0-windows-amd64.msi" /qn+
```

I pacchetti di installazione MSI dei plugin caricabili di Zabbix agent 2 supportano i seguenti parametri.

Parameter	Description
ADDDEFAULT	Un elenco di componenti separati da virgole da installare con la configurazione predefinita. Per ulteriori informazioni, vedere ADDDEFAULT property. Valori possibili: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Esempio: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Un elenco di componenti separati da virgole da installare localmente. Per ulteriori informazioni, vedere ADDLOCAL property. Valori possibili: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Esempio: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Il percorso completo della cartella in cui verranno installati i componenti di Zabbix, con i relativi file di configurazione installati nella sottocartella `zabbix_agent2.d`. Esempio: `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

Compilazione di Zabbix agent 2 su Windows

Panoramica

Questa pagina illustra come compilare Zabbix agent 2 dai sorgenti su Windows 10 (64 bit o 32 bit).

Sia la versione a 32 bit sia quella a 64 bit possono essere compilate su una piattaforma a 64 bit, mentre su una piattaforma a 32 bit è possibile compilare solo la versione a 32 bit.

Per compilare Zabbix agent 2 sono necessari:

- Strumenti di compilazione MinGW
- Linguaggio di programmazione Go
- OpenSSL (per le funzionalità di crittografia in Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; per le funzionalità di corrispondenza dei pattern delle espressioni regolari in Zabbix)

È possibile compilare Zabbix agent 2 utilizzando uno dei seguenti metodi:

- Utilizzo di vcpkg - un approccio automatizzato che semplifica la gestione delle dipendenze tramite un gestore di pacchetti C++.
- Compilazione manuale - un approccio manuale che richiede l'installazione di tutte le dipendenze prima di compilare l'agent.

Attention:
Prima di avviare il processo di compilazione, tenere presente quanto segue:

- Per eseguire i comandi, utilizzare il Prompt dei comandi, avviato da un utente con autorizzazioni sufficienti per scrivere nelle cartelle protette. Tuttavia, durante l'installazione di OpenSSL e PCRE2, utilizzare il terminale MSYS2.
- Si consiglia di creare una directory di lavoro in `C:\Zabbix` per tutti i file sorgente e le cartelle di compilazione. Tuttavia, i componenti compilati devono essere installati in `C:\Zabbix\x64` (o `C:\Zabbix\x86` per le compilazioni a 32 bit).

Compilazione di Zabbix agent 2 con vcpkg

Questa sezione contiene le istruzioni per compilare Zabbix agent con vcpkg, un gestore di pacchetti che semplifica la gestione delle dipendenze e l'integrazione con i progetti C++.

1. Scaricare e installare Build Tools for Visual Studio 2022. Durante l'installazione, assicurarsi di selezionare il carico di lavoro *Sviluppo desktop con C++*, che include il gestore di pacchetti vcpkg.

2. Scaricare e installare Go (disponibile come programma di installazione MSI). Durante l'installazione, specificare `C:\Zabbix\Go` come cartella di installazione.

3. Scaricare la distribuzione MinGW che utilizza la libreria runtime Microsoft Visual C; ad esempio:

- Per le compilazioni a 64 bit: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Per le compilazioni a 32 bit: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
La versione di MinGW deve soddisfare i requisiti minimi.

Quindi, estrarla in `C:\Zabbix\mingw64` (o `C:\Zabbix\mingw32` per le compilazioni a 32 bit).

4. Inizializzare vcpkg e installare le dipendenze necessarie per compilare Zabbix agent 2 (l'operazione potrebbe richiedere del tempo):

```
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

#### Per le compilazioni a 64 bit:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

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

5. Scaricare l'archivio dei sorgenti di Zabbix ed estrarlo in `C:\Zabbix\zabbix-8.0.0`.

6. Accedere alla directory di compilazione di Zabbix (`C:\Zabbix\zabbix-8.0.0\build\mingw`) e creare il seguente script `build.bat`:

- Per le compilazioni a 64 bit:

```
:: Aggiungere MinGW e Go alla variabile di sistema `PATH` per la sessione corrente:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Impostare il percorso di installazione di vcpkg:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Impostare i flag del linker per la libreria Crypt32:
SET CGO_LDFLAGS="-lCrypt32"

:: Eseguire il processo di compilazione:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- Per le compilazioni a 32 bit:

```
:: Aggiungere MinGW e Go alla variabile di sistema `PATH` per la sessione corrente:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Impostare il percorso di installazione di vcpkg:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Impostare i flag del linker per la libreria Crypt32:
SET CGO_LDFLAGS="-lCrypt32"

:: Eseguire il processo di compilazione:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Compilare Zabbix agent 2 eseguendo lo script:

```
build.bat
```

Al termine della compilazione, il binario di Zabbix agent 2 si troverà in `C:\Zabbix\zabbix-8.0.0\bin\win64` (per le compilazioni a 64 bit) o in `C:\Zabbix\zabbix-8.0.0\bin\win32` (per le compilazioni a 32 bit). I file di configurazione di Zabbix agent 2 si trovano in `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Per eseguire l'agent, copiare `zabbix_agent2.exe` e i relativi file di configurazione in una cartella dedicata (ad esempio `C:\Zabbix\agent2`), quindi eseguire l'agent:

```
mkdir C:\Zabbix\agent2

#### Per le compilazioni a 64 bit:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### Per le compilazioni a 32 bit:
copy C:\Zabbix\zabbix-8.0.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

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

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

Se necessario, continuare con la compilazione dei plugin caricabili di Zabbix agent 2.

Compilazione dei plugin caricabili di Zabbix agent 2

1. Scaricare il codice sorgente del plugin Zabbix che corrisponde alla versione di Zabbix agent 2 in uso (ad esempio, `zabbix-agent2-plugin-postgresql-8.0.0.tar.gz`) ed estrarlo in `C:\Zabbix`.

> **Attention:**
> Prima di installare un plugin, verificare il relativo file README. Potrebbe contenere requisiti specifici e istruzioni di installazione.

2. Passare alla directory del plugin estratto e compilare il plugin:

```
cd C:\Zabbix\zabbix-agent2-plugin-ember-plus-8.0.0

#### Per build a 64 bit:
mingw32-make ARCH=AMD64

#### Per build a 32 bit:
mingw32-make ARCH=x86
```

Dopo la compilazione, il binario del plugin di Zabbix agent 2 e il relativo file di configurazione si troveranno nella stessa directory del plugin.

L'eseguibile del plugin può essere collocato in qualsiasi posizione, purché sia caricabile da Zabbix agent 2. Specificare il percorso del binario del plugin nel file di configurazione del plugin, ad esempio in postgresql.conf per il plugin PostgreSQL:

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Il percorso del file di configurazione del plugin deve essere specificato nel parametro Include del file di configurazione di Zabbix agent 2:

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Per ulteriori dettagli sulla configurazione dei plugin, consultare setup.

Compilazione manuale di Zabbix agent 2

> **Attention:**
> Questo metodo di compilazione di Zabbix agent 2 è adatto agli utenti che richiedono il pieno controllo dell'ambiente di compilazione o che si trovano in un ambiente con restrizioni in cui l'uso di vcpkg non è possibile.

Questa sezione contiene le istruzioni per compilare manualmente Zabbix agent 2, inclusa l'installazione degli strumenti di compilazione e delle dipendenze richiesti, seguita dalla compilazione dell'agent.

Configurazione degli strumenti di compilazione

1. Scarica e installa MSYS2 (disponibile come programma di installazione MSI). Durante l'installazione, assicurati di specificare `C:\Zabbix\msys64` come cartella di installazione.

2. Scarica e installa Go (disponibile come programma di installazione MSI; vedi le versioni di Go attualmente supportate). Durante l'installazione, assicurati di specificare `C:\Zabbix\Go` come cartella di installazione.

3. Scarica la distribuzione MinGW che utilizza la libreria runtime Microsoft Visual C; ad esempio:

- Per build a 64 bit: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Per build a 32 bit: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
La versione di MinGW deve soddisfare i requisiti minimi.

Quindi, estraila in `C:\Zabbix\mingw64` (oppure in `C:\Zabbix\mingw32` per le build a 32 bit).

Installazione di OpenSSL

Note:
Per compilare Zabbix agent senza il supporto TLS, procedere alla sezione Installazione di PCRE2.

1. Aprire il terminale MSYS2 MSYS con privilegi di amministratore ed eseguire i seguenti comandi:

```
pacman -S perl-Locale-Maketext-Simple
pacman -S nasm
pacman -S make
pacman -S cmake
```

2. Scaricare l'archivio dei sorgenti di OpenSSL ed estrarlo in `C:\Zabbix\openssl-3.5.0`.

3. Accedere alla directory di OpenSSL estratta e creare il seguente script `build.sh`:

- Per le build a 64 bit:

```
####!/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
```

- Per le build a 32 bit:

```
####!/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:
Assicurarsi di revocare agli utenti non amministratori l'accesso in scrittura alla directory `C:\Zabbix\x64\OpenSSL-Win64-350-static`. In caso contrario, l'agent caricherà le impostazioni SSL da un percorso modificabile da utenti senza privilegi, con conseguente potenziale vulnerabilità di sicurezza.

- L'opzione `no-shared` rende le librerie statiche OpenSSL libcrypto.lib e libssl.lib autonome, in modo che i binari Zabbix includano OpenSSL senza richiedere DLL esterne. Ciò significa che i binari Zabbix possono essere copiati su altri computer Windows senza le librerie OpenSSL; tuttavia, quando viene rilasciata una nuova versione correttiva di OpenSSL, sarà necessario ricompilare Zabbix agent.

- Senza l'opzione `no-shared`, Zabbix si affida alle DLL di OpenSSL durante l'esecuzione. Ciò significa che gli aggiornamenti di OpenSSL potrebbero non richiedere la ricompilazione di Zabbix agent; tuttavia, quando lo si copia su altri computer, è necessario copiare anche le DLL di OpenSSL.

Per ulteriori informazioni sulle altre opzioni di configurazione di OpenSSL, consultare la documentazione di OpenSSL.

4. Configurare e installare OpenSSL eseguendo lo script (l'operazione potrebbe richiedere del tempo):

```
cd /c/Zabbix/openssl-3.5.0
./build.sh
```

Installazione di PCRE2

1. Scaricare l'archivio sorgente di PCRE2 ed estrarlo in `C:\Zabbix\pcre2-10.45`.

2. Aprire il terminale MSYS2 MSYS con privilegi di amministratore. Quindi, creare una directory `build` nella directory PCRE2 estratta e accedervi:

```
mkdir /c/Zabbix/pcre2-10.45/build
cd /c/Zabbix/pcre2-10.45/build
```

3. Configurare PCRE2:

```
#### Per build a 64 bit:
export PATH="/c/Zabbix/mingw64/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-O2 -g" -DCMAKE_INSTALL_PREFIX="/c/Zabbix/x64/PCRE2" ..

#### Per build a 32 bit:
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:
Se si verificano errori, si consiglia di eliminare la cache di CMake prima di tentare di ripetere il processo di build con CMake. La cache (`CMakeCachecache.txt`) si trova nella directory build della directory PCRE2 estratta.

4. Installare PCRE2:

```
make install
```

Compilazione di Zabbix agent 2

1. Scaricare l'archivio dei sorgenti di Zabbix ed estrarlo in `C:\Zabbix\zabbix-8.0.0`.

Se è necessario generare un archivio dei sorgenti dal repository dei sorgenti non elaborati (ad esempio, per applicare patch personalizzate o compilare dal codice sorgente più recente), eseguire i seguenti comandi su una macchina **Linux** con Go installato (necessario per configurare Zabbix agent 2):

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Verrà creato un archivio dei sorgenti, che potrà quindi essere copiato su una macchina Windows.

2. Aprire il Prompt dei comandi con privilegi di amministratore. Quindi, accedere alla directory di compilazione di Zabbix e compilare Zabbix agent; assicurarsi di specificare correttamente le directory in cui sono installati OpenSSL e PCRE2:

- Per le compilazioni a 64 bit:

```
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

#### Con il supporto TLS:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### Senza il supporto TLS:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- Per le compilazioni a 32 bit:

```
cd C:\Zabbix\zabbix-8.0.0\build\mingw
set PATH=C:\Zabbix\mingw32\bin;%PATH%

#### Con il supporto TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win32-350-static"

#### Senza il supporto TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Al termine della compilazione, il binario di Zabbix agent 2 si troverà in `C:\Zabbix\zabbix-8.0.0\bin\win64` (oppure in `C:\Zabbix\zabbix-8.0.0\bin\win32` per le compilazioni a 32 bit). I file di configurazione di Zabbix agent 2 si trovano in `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Per eseguire l'agent, copiare il binario `zabbix_agent2.exe` e i relativi file di configurazione in una cartella dedicata (ad esempio, `C:\Zabbix\agent2`), quindi eseguire l'agent:

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

Se necessario, continuare con la compilazione dei plugin caricabili di Zabbix agent 2.

Compilazione di Zabbix agent su Windows

Panoramica

Questa pagina illustra come compilare l'agent Zabbix dai sorgenti su Windows 10 (64 bit).

Queste istruzioni si applicano alle versioni di Windows che supportano Visual Studio 2022.

Per compilare l'agent Zabbix sono necessari:

- Compilatore C (incluso in Build Tools for Visual Studio 2022)
- OpenSSL (per le funzionalità di crittografia in Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; per le funzionalità di corrispondenza dei pattern delle espressioni regolari in Zabbix)

È possibile compilare l'agent Zabbix utilizzando uno dei seguenti metodi:

- Utilizzo di vcpkg - un approccio automatizzato che semplifica la gestione delle dipendenze utilizzando un gestore di pacchetti C++.
- Compilazione manuale - un approccio manuale che richiede l'installazione di tutte le dipendenze prima di compilare l'agent.

A seconda delle esigenze di monitoraggio, potrebbero essere necessarie librerie aggiuntive. Per ulteriori informazioni, vedere Requisiti.

> **Attention:**
> Prima di avviare il processo di compilazione, tenere presente quanto segue:

> - Per eseguire i comandi, utilizzare "x64 Native Tools Command Prompt" (incluso in Build Tools for Visual Studio 2022), avviato da un utente con autorizzazioni sufficienti per scrivere nelle cartelle protette.
> - Si consiglia di creare una directory di lavoro in `C:\Zabbix` per tutti i file sorgente e le cartelle di compilazione. Tuttavia, i componenti compilati devono essere installati in `C:\Program Files\Zabbix\x64`.

Compilazione dell'agent Zabbix con vcpkg

Questa sezione contiene le istruzioni per compilare l'agent Zabbix con vcpkg, un gestore di pacchetti che semplifica la gestione delle dipendenze e l'integrazione con i progetti C++.

1. Scaricare e installare Build Tools for Visual Studio 2022. Durante l'installazione, assicurarsi di selezionare il carico di lavoro *Desktop development with C++*, che include gli strumenti necessari per compilare l'agent con vcpkg:

- Compilatore C (Microsoft Visual C++)
- Strumento da riga di comando NMake
- Gestore di pacchetti vcpkg
- "x64 Native Tools Command Prompt"

2. Aprire `cmd.exe` con privilegi di amministratore, inizializzare vcpkg e installare le dipendenze necessarie per compilare l'agent Zabbix (l'operazione potrebbe richiedere del tempo):

```
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. Scaricare l'archivio dei sorgenti di Zabbix ed estrarlo in `C:\Zabbix\zabbix-8.0.0`.

4. Accedere alla directory di compilazione di Zabbix (`C:\Zabbix\zabbix-8.0.0\build\win32\project`) e creare il seguente script `build.bat`; assicurarsi di specificare correttamente le directory in cui sono installati OpenSSL e PCRE2:

```
:: Imposta il percorso di installazione di vcpkg:
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: Esegue il processo di compilazione:
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. Aprire "x64 Native Tools Command Prompt" e compilare l'agent Zabbix eseguendo lo script:

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

Se è stato aperto un prompt dei comandi diverso, inizializzare prima l'ambiente di compilazione di Visual Studio:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64
cd C:\Zabbix\zabbix-8.0.0\build\win32\project
build.bat
```

Racchiudere tra virgolette i percorsi che contengono spazi.

Al termine della compilazione, i file binari dei componenti Zabbix si troveranno in `C:\Zabbix\zabbix-8.0.0\bin\win64`. Il file di configurazione dell'agent Zabbix si trova in `C:\Zabbix\zabbix-8.0.0\conf`.

Per eseguire l'agent, copiare `zabbix_agentd.exe` e il relativo file di configurazione in una cartella dedicata (ad esempio `C:\Zabbix\agent`), quindi eseguire l'agent:

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

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

Compilazione manuale di Zabbix agent

> **Attention:**
> Questo metodo di compilazione di Zabbix agent è adatto agli utenti che richiedono il pieno controllo dell'ambiente di compilazione o che si trovano in un ambiente con restrizioni in cui l'uso di vcpkg non è possibile.

Questa sezione contiene le istruzioni per compilare manualmente Zabbix agent, inclusa l'installazione degli strumenti di compilazione e delle dipendenze richiesti (Perl, OpenSSL, PCRE2), quindi la compilazione dell'agent.

Installazione degli strumenti di compilazione

1. Scaricare e installare Build Tools for Visual Studio 2022. Durante l'installazione, assicurarsi di selezionare il carico di lavoro *Sviluppo desktop con C++*, che include gli strumenti necessari per compilare manualmente l'agent:

- Compilatore C (Microsoft Visual C++)
- Strumento da riga di comando NMake
- "x64 Native Tools Command Prompt"

Installazione di OpenSSL

> **Note:**
> Per compilare l'agent Zabbix senza il supporto TLS, procedere alla sezione Installazione di PCRE2.

1. Scaricare e installare Strawberry Perl (disponibile come programma di installazione MSI). Durante l'installazione, specificare `C:\Zabbix\Strawberry` come cartella di installazione.

2. Installare il modulo Perl `Text::Template`:

```
cpanm Text::Template
```

3. Verificare che il Netwide Assembler (NASM; necessario per compilare OpenSSL) sia stato compilato durante l'installazione di Strawberry Perl:

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

Se NASM non è stato compilato, installarlo manualmente. Per ulteriori informazioni, consultare la documentazione di NASM.

4. Scaricare l'archivio dei sorgenti di OpenSSL ed estrarlo in `C:\Zabbix\openssl-3.5.0`.

5. Accedere alla directory estratta e configurare OpenSSL, ad esempio:

```
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:**
> Se si sceglie una directory personalizzata per OpenSSL durante la compilazione dell'agent Zabbix su Windows (ad esempio, `C:\zabbix` o `C:\openssl-64bit`), assicurarsi di revocare l'accesso in scrittura a questa directory agli utenti non amministratori. In caso contrario, l'agent caricherà le impostazioni SSL da un percorso che può essere modificato da utenti senza privilegi, causando una potenziale vulnerabilità di sicurezza.

- L'opzione `no-shared` rende le librerie statiche libcrypto.lib e libssl.lib di OpenSSL autonome, in modo che i binari Zabbix includano OpenSSL senza richiedere DLL esterne. Ciò significa che i binari Zabbix possono essere copiati su altri computer Windows senza le librerie OpenSSL; tuttavia, quando viene rilasciata una nuova versione correttiva di OpenSSL, sarà necessario ricompilare l'agent Zabbix.
- Senza l'opzione `no-shared`, Zabbix si affida alle DLL di OpenSSL durante l'esecuzione. Ciò significa che gli aggiornamenti di OpenSSL potrebbero non richiedere la ricompilazione dell'agent Zabbix; tuttavia, quando lo si copia su altri computer, è necessario copiare anche le DLL di OpenSSL.

Per ulteriori informazioni sulle altre opzioni di configurazione di OpenSSL, consultare la documentazione di OpenSSL.

6. Compilare OpenSSL ed eseguire i test (tenere presente che l'operazione potrebbe richiedere del tempo):

> **Attention:**
> Eseguire i test senza privilegi amministrativi; in caso contrario, potrebbero verificarsi risultati imprevisti o rischi per la sicurezza. Se alcuni test hanno esito negativo, consultare la documentazione di OpenSSL per la risoluzione dei problemi.

```
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. Installare OpenSSL:

```
nmake install
```

Per installare solo i componenti software (librerie e file di intestazione, ma non la documentazione), è possibile usare `nmake install_sw`.

Installazione di PCRE2

1. Scaricare e installare CMake (disponibile come programma di installazione MSI). Durante l'installazione, assicurarsi di specificare `C:\Zabbix\CMake` come cartella di installazione e di selezionare l'opzione *Add CMake to the PATH environment variable*.

2. Scaricare l'archivio sorgente di PCRE2 ed estrarlo in `C:\Zabbix\pcre2-10.45`.

3. Creare una directory `build` nella directory PCRE2 estratta e accedervi:

```
mkdir C:\Zabbix\pcre2-10.45\build
cd C:\Zabbix\pcre2-10.45\build
```

4. Configurare PCRE2:

```
cmake -G "NMake Makefiles" -DPCRE_SUPPORT_UNICODE_PROPERTIES=ON -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL
```

Note:
Se si verificano errori, si consiglia di eliminare la cache di CMake prima di tentare di ripetere il processo di build con CMake. La cache (`CMakeCachecache.txt`) si trova nella directory build della directory PCRE2 estratta.

5. Compilare PCRE2 utilizzando NMake:

```
nmake
```

6. Installare PCRE2:

```
cmake --install .
```

Compilazione dell'agent Zabbix

1. Scaricare l'archivio sorgente di Zabbix ed estrarlo in `C:\Zabbix\zabbix-8.0.0`.

Se è necessario generare un archivio sorgente dal repository del codice sorgente (ad esempio, per applicare patch personalizzate o compilare il codice sorgente più recente), eseguire i seguenti comandi su una macchina **Linux**:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Verrà creato un archivio sorgente, che potrà quindi essere copiato su una macchina Windows.

2. Accedere alla directory di compilazione di Zabbix, aprire il prompt dei comandi "x64 Native Tools Command Prompt" e compilare l'agent Zabbix (o altri componenti); assicurarsi di specificare correttamente le directory in cui sono installati OpenSSL e PCRE2:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64

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

#### Con il supporto TLS:
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra

#### Senza il supporto TLS:
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra
```

Racchiudere tra virgolette i percorsi che contengono spazi.

Al termine della compilazione, i file binari dei componenti Zabbix si troveranno in `C:\Zabbix\zabbix-8.0.0\bin\win64`. Il file di configurazione dell'agent Zabbix si trova in `C:\Zabbix\zabbix-8.0.0\conf`.

Per eseguire l'agent, copiare `zabbix_agentd.exe` e il relativo file di configurazione in una cartella dedicata (ad esempio, `C:\Zabbix\agent`), quindi eseguire l'agent:

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

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

Installazione dell'agent macOS da PKG

Panoramica

Zabbix agent può essere installato su macOS utilizzando pacchetti di installazione PKG, disponibili per il download.

I pacchetti di Zabbix agent sono disponibili con o senza crittografia.

Installazione di agent

L'agent può essere installato utilizzando l'interfaccia utente grafica oppure dalla riga di comando, ad esempio:

```
sudo installer -pkg zabbix_agent-8.0.0-macos-arm64-openssl.pkg -target /
```

Assicurati di utilizzare nel comando la versione corretta del pacchetto Zabbix. Deve corrispondere al nome del pacchetto scaricato.

Esecuzione dell'agent

L'agent si avvierà automaticamente dopo l'installazione o il riavvio.

Se necessario, è possibile modificare il file di configurazione in `/usr/local/etc/zabbix/zabbix_agentd.conf`.

Per avviare manualmente l'agent, è possibile eseguire:

```
sudo launchctl start com.zabbix.zabbix_agentd
```

Per arrestare manualmente l'agent:

```
sudo launchctl stop com.zabbix.zabbix_agentd
```

Durante l'aggiornamento, il file di configurazione esistente non viene sovrascritto. Viene invece creato un nuovo file `zabbix_agentd.conf.NEW` da utilizzare per rivedere e aggiornare il file di configurazione esistente, se necessario. Ricordarsi di riavviare l'agent dopo aver apportato modifiche manuali al file di configurazione.

Risoluzione dei problemi e rimozione di agent

Questa sezione elenca alcuni comandi utili che possono essere usati per la risoluzione dei problemi e la rimozione dell'installazione di Zabbix agent.

Verificare se Zabbix agent è in esecuzione:

```
ps aux | grep zabbix_agentd
```

Verificare se Zabbix agent è stato installato dai pacchetti:

```
pkgutil --pkgs | grep zabbix
com.zabbix.pkg.ZabbixAgent
```

Visualizzare i file installati dal pacchetto di installazione (si noti che la / iniziale non viene visualizzata in questa vista):

```
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
```

Arrestare Zabbix agent se è stato avviato con `launchctl`:

```
sudo launchctl unload /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
```

Rimuovere i file (incluse configurazione e log) installati con il pacchetto di installazione:

```
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
```

Dimenticare che Zabbix agent è stato installato:

```
sudo pkgutil --forget com.zabbix.pkg.ZabbixAgent
```

Compilazione di Zabbix agent su macOS

Panoramica

Questa sezione mostra come creare i binari dell'agente macOS Zabbix da sorgenti con o senza TLS.

Prerequisiti

Avrai bisogno degli strumenti di sviluppo da riga di comando (Xcode non è richiesto), di Automake, pkg-config e PCRE (v8.x) oppure PCRE2 (v10.x). Se vuoi compilare i binari dell'agent con TLS, avrai bisogno anche di OpenSSL o GnuTLS.

Per installare Automake e pkg-config, avrai bisogno del gestore di pacchetti Homebrew da https://brew.sh/. Per installarlo, apri il terminale ed esegui il seguente comando:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Quindi installa Automake e pkg-config:

```
brew install automake
brew install pkg-config
```

La preparazione delle librerie PCRE, OpenSSL e GnuTLS dipende dal modo in cui verranno collegate all'agent.

Se intendi eseguire i binari dell'agent su una macchina macOS che dispone già di queste librerie, puoi usare le librerie precompilate fornite da Homebrew. In genere si tratta di macchine macOS che usano Homebrew per compilare i binari dell'agent Zabbix o per altri scopi.

Se i binari dell'agent verranno usati su macchine macOS che non dispongono della versione condivisa delle librerie, dovresti compilare librerie statiche dai sorgenti e collegare l'agent Zabbix con esse.

Compilazione dei binari dell'agent con librerie condivise

Installare PCRE2 (sostituire *pcre2* con *pcre* nei comandi seguenti, se necessario):

```
brew install pcre2
```

Quando si compila con TLS, installare OpenSSL e/o GnuTLS:

```
brew install openssl
brew install gnutls
```

Scaricare i sorgenti di Zabbix:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Compilare l'agent senza TLS:

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6
make
make install
```

Compilare l'agent con OpenSSL:

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-openssl=/usr/local/opt/
make
make install
```

Compilare l'agent con 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
```

Compilazione dei binari dell'agent con librerie statiche senza TLS

Supponiamo che le librerie statiche di PCRE vengano installate in `$HOME/static-libs`. Useremo PCRE2 10.39.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
```

Scaricare e compilare PCRE con il supporto per le proprietà Unicode:

```
mkdir static-libs-source
cd static-libs-source
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
```

Scaricare il codice sorgente di Zabbix e compilare l'agent:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

Compilazione dei binari dell'agent con librerie statiche e OpenSSL

Durante la compilazione di OpenSSL, si consiglia di eseguire `make test` dopo una compilazione riuscita. Anche se la compilazione è andata a buon fine, a volte i test falliscono. In tal caso, i problemi devono essere analizzati e risolti prima di continuare.

Supponiamo che le librerie statiche di PCRE e OpenSSL vengano installate in `$HOME/static-libs`. Useremo PCRE2 10.39 e OpenSSL 1.1.1a.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
OPENSSL_PREFIX="$HOME/static-libs/openssl-1.1.1a"
```

Compiliamo le librerie statiche in `static-libs-source`:

```
mkdir static-libs-source
cd static-libs-source
```

Scaricare e compilare PCRE con il supporto per le proprietà Unicode:

```
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
cd ..
```

Scaricare e compilare OpenSSL:

```
curl --remote-name https://www.openssl.org/source/openssl-1.1.1a.tar.gz
tar xf openssl-1.1.1a.tar.gz
cd openssl-1.1.1a
./Configure --prefix="$OPENSSL_PREFIX" --openssldir="$OPENSSL_PREFIX" --api=1.1.0 no-shared no-capieng no-
make
make test
make install_sw
cd ..
```

Scaricare il codice sorgente di Zabbix e compilare l'agent:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

Compilazione dei binari agent con librerie statiche con GnuTLS

GnuTLS dipende dal backend crittografico Nettle e dalla libreria aritmetica GMP. Invece di usare la libreria GMP completa, questa guida utilizzerà mini-gmp, che è inclusa in Nettle.

Quando si compilano GnuTLS e Nettle, si consiglia di eseguire `make check` dopo una compilazione riuscita. Anche se la compilazione è andata a buon fine, i test a volte falliscono. In tal caso, i problemi devono essere analizzati e risolti prima di continuare.

Supponiamo che le librerie statiche PCRE, Nettle e GnuTLS vengano installate in `$HOME/static-libs`. Useremo PCRE2 10.39, Nettle 3.4.1 e 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"
```

Compiliamo le librerie statiche in `static-libs-source`:

```
mkdir static-libs-source
cd static-libs-source
```

Scaricare e compilare Nettle:

```
curl --remote-name https://ftp.gnu.org/gnu/nettle/nettle-3.4.1.tar.gz
tar xf nettle-3.4.1.tar.gz
cd nettle-3.4.1
./configure --prefix="$NETTLE_PREFIX" --enable-static --disable-shared --disable-documentation --disable-a
make
make check
make install
cd ..
```

Scaricare e compilare GnuTLS:

```
curl --remote-name https://www.gnupg.org/ftp/gcrypt/gnutls/v3.6/gnutls-3.6.5.tar.xz
tar xf gnutls-3.6.5.tar.xz
cd gnutls-3.6.5
PKG_CONFIG_PATH="$NETTLE_PREFIX/lib/pkgconfig" ./configure --prefix="$GNUTLS_PREFIX" --enable-static --dis
make
make check
make install
cd ..
```

Scaricare il codice sorgente di Zabbix e compilare agent:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
CFLAGS="-Wno-unused-command-line-argument -framework Foundation -framework Security" \
> LIBS="-lgnutls -lhogweed -lnettle" \
> LDFLAGS="-L$GNUTLS_PREFIX/lib -L$NETTLE_PREFIX/lib" \
> ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFI
make
make install
```

Guida alla distribuzione di Zabbix server su AWS

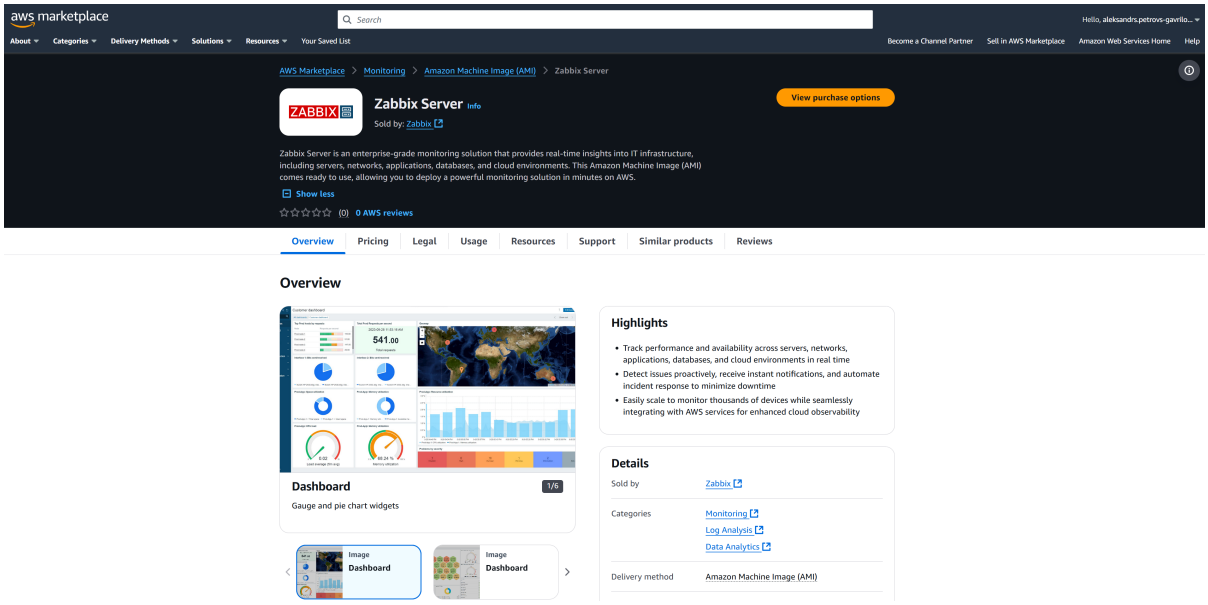
Panoramica

Questa pagina descrive come distribuire Zabbix server utilizzando AWS.

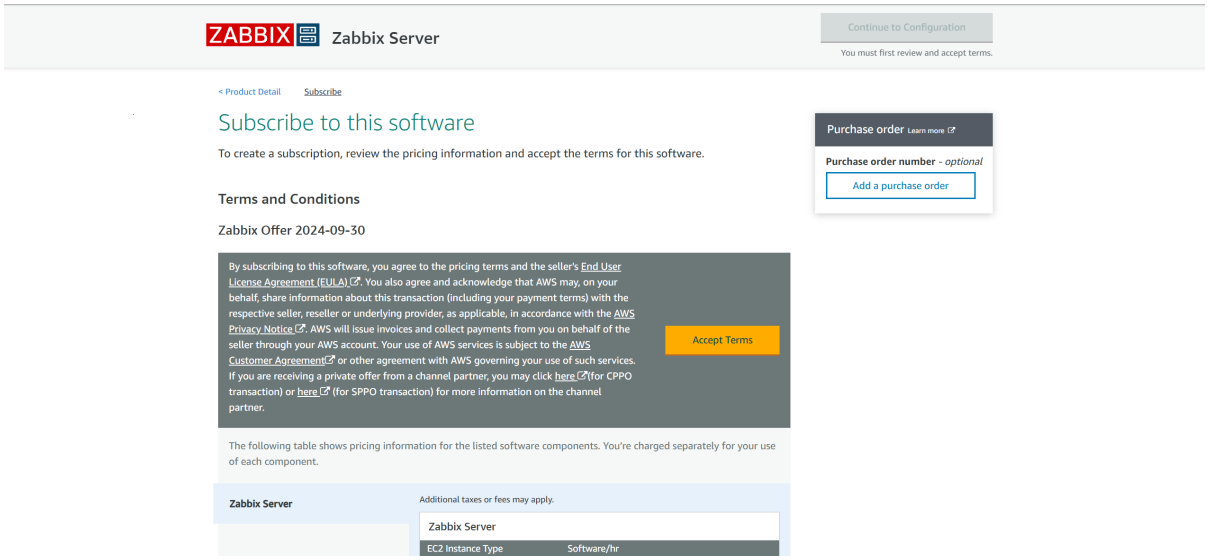
> **Attention:**
> Sono disponibili diverse versioni di Zabbix per la distribuzione sulla piattaforma. Assicurarsi di leggere la documentazione relativa alla stessa versione di Zabbix distribuita. La documentazione per le diverse versioni di Zabbix può essere selezionata dal menu a discesa delle versioni.

Distribuzione dell'appliance

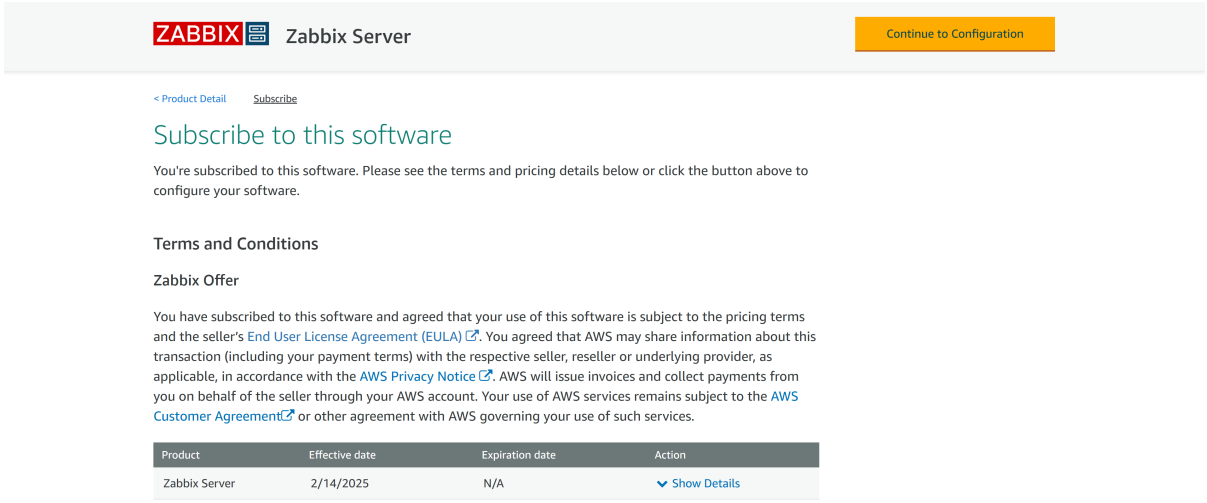
Per distribuire l'appliance Zabbix, accedere innanzitutto alla pagina Zabbix AWS Marketplace. Fare clic su *View purchase options* per distribuire l'immagine e procedere alla pagina di sottoscrizione.



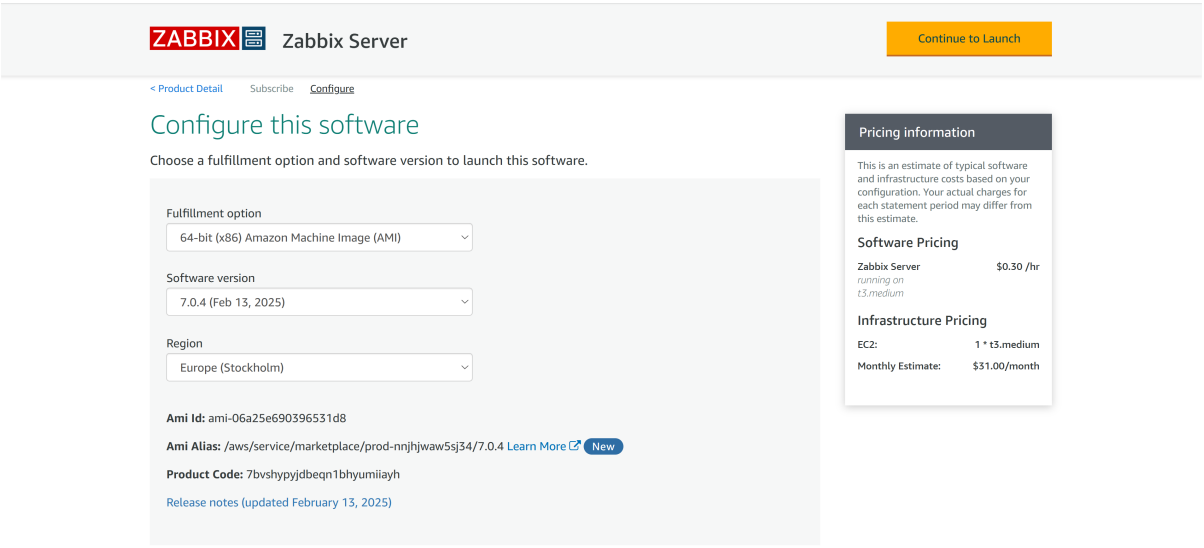
Fare clic sul pulsante *Accept terms* per accettare i *Terms and Conditions*.



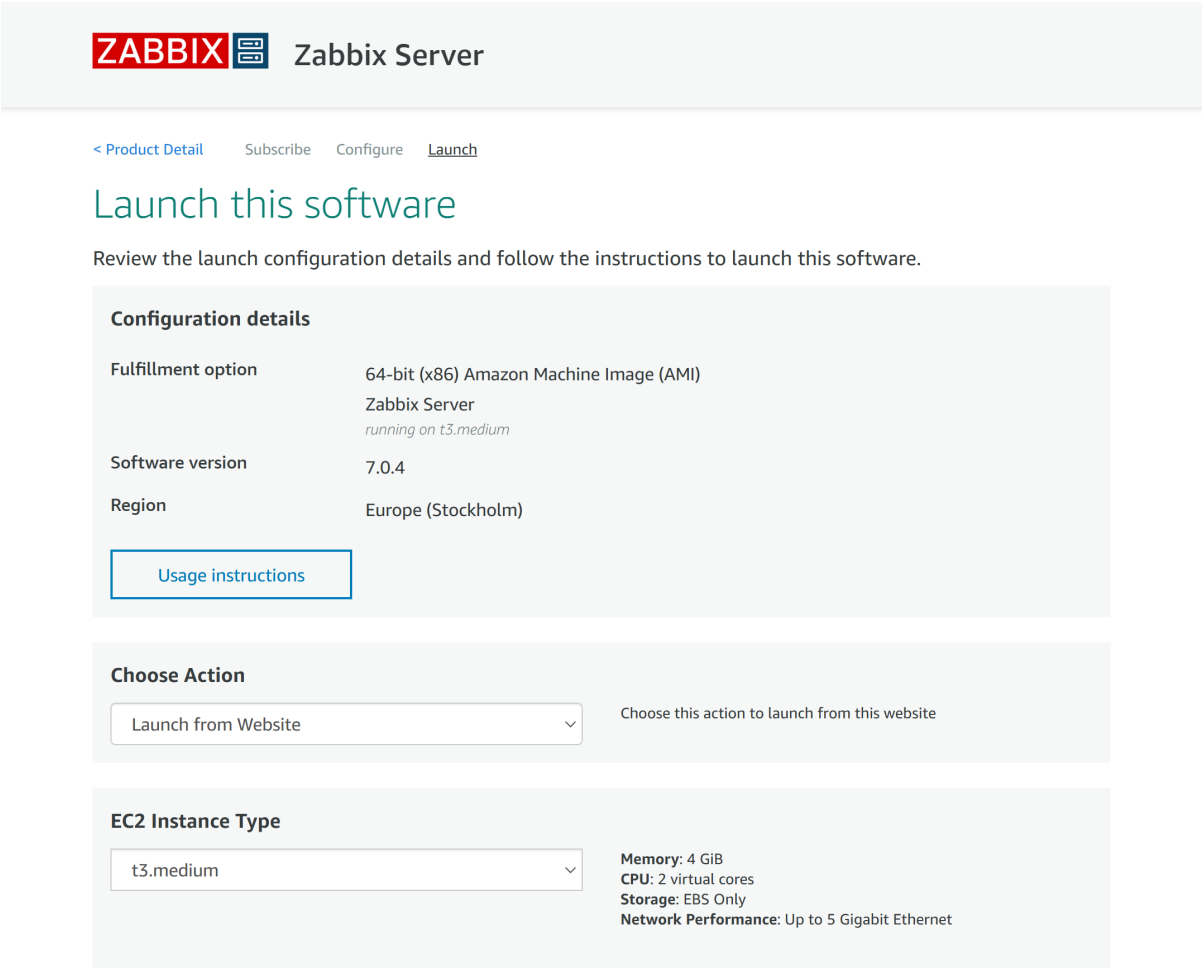
Fare clic sul pulsante *Continue to Configuration* per passare alla sezione Configuration dell'appliance Zabbix.



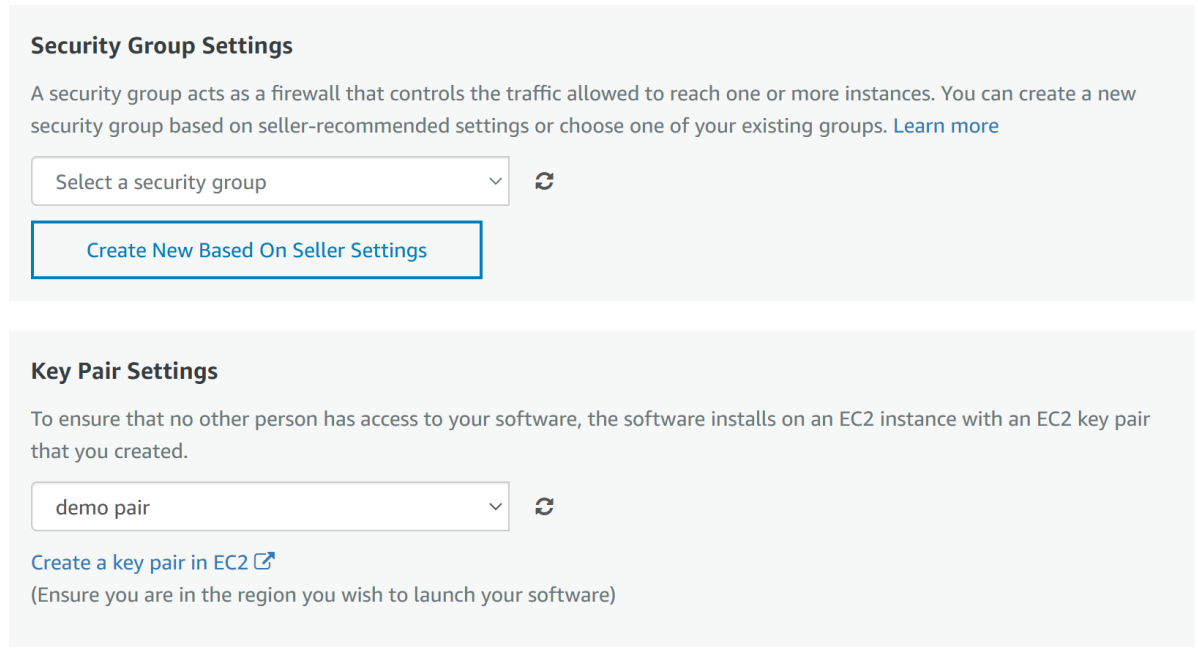
Selezionare l'opzione Fulfilment, la versione di Zabbix e la Region preferita.



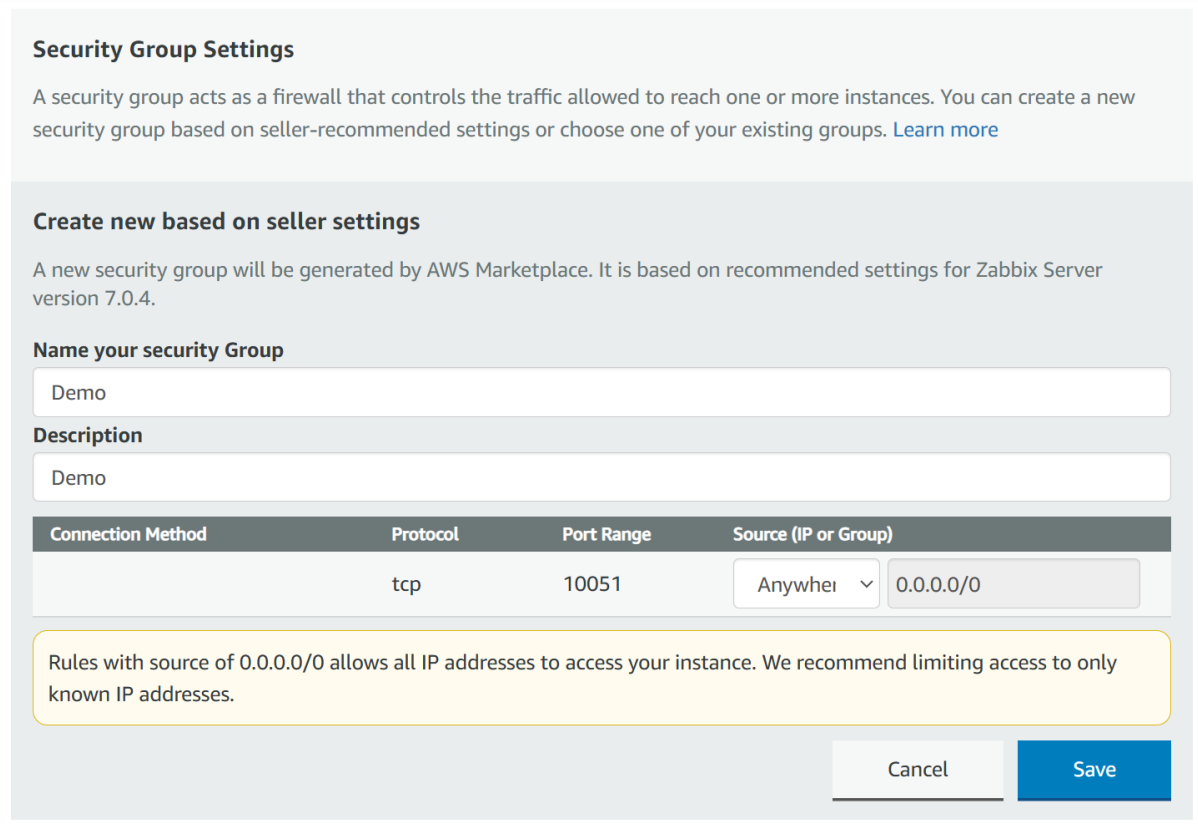
Fare clic sul pulsante *Continue to Launch*. Nella pagina Launch, selezionare l'opzione Launch preferita, EC2 Instance Type, VPC e le impostazioni Subnet.



Nella parte inferiore della pagina Launch, specificare le impostazioni Security Group e selezionare o creare una coppia di chiavi EC2: assicurarsi di salvare la chiave privata in un luogo sicuro!



Se si seleziona *Create a new Security Group based on seller settings*, sarà consentita la porta TCP 10051 (Zabbix server). Nota: se si desidera accedere all'istanza tramite SSH e HTTP/HTTPS (frontend), sarà inoltre necessario consentire le connessioni SSH e HTTP/HTTPS nelle impostazioni Security Group.



Congratulazioni, l'appliance Zabbix è stata distribuita correttamente.

ZABBIX Zabbix Server

Congratulations! An instance of this software is successfully deployed on EC2!

AMI ID: ami-06a25e690396531d8 (View Launch Configuration Details)
You can view this instance on EC2 Console. You can also view all instances on Your Software. Software and AWS hourly usage fees apply when the instance is running and will appear on your monthly bill.

You can launch this configuration again below or go to the configuration page to start a new one.

Configuration details

Fulfillment option	64-bit (x86) Amazon Machine Image (AMI) Zabbix Server *running on t3.medium*
Software version	7.0.4
Region	Europe (Stockholm)

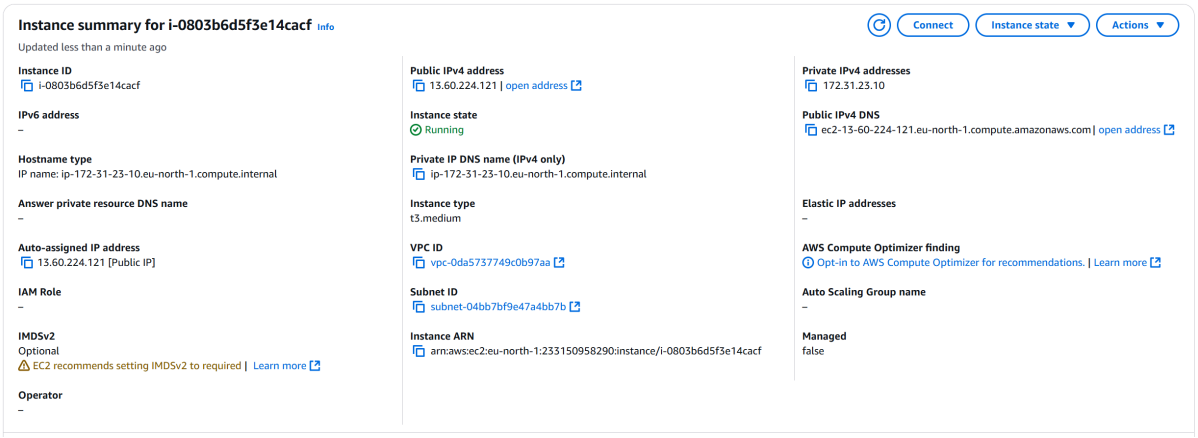
Usage instructions

Choose Action

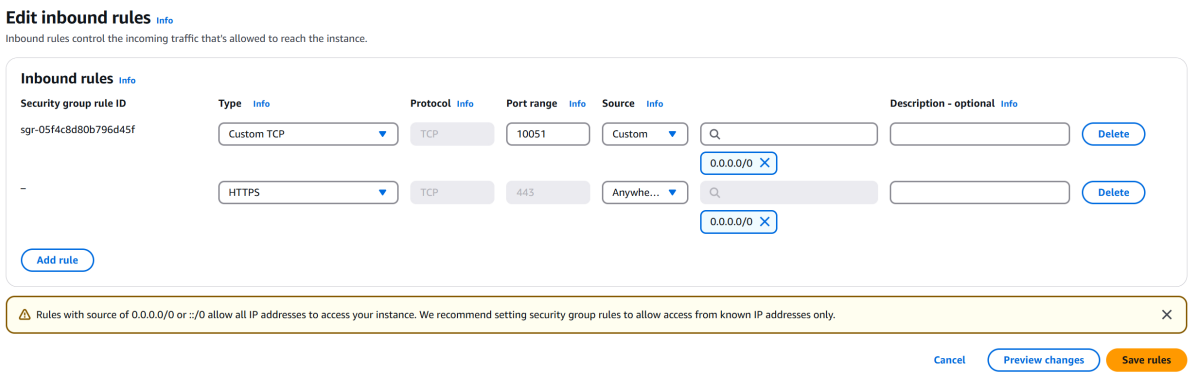
Select a launch action

Accesso all'appliance

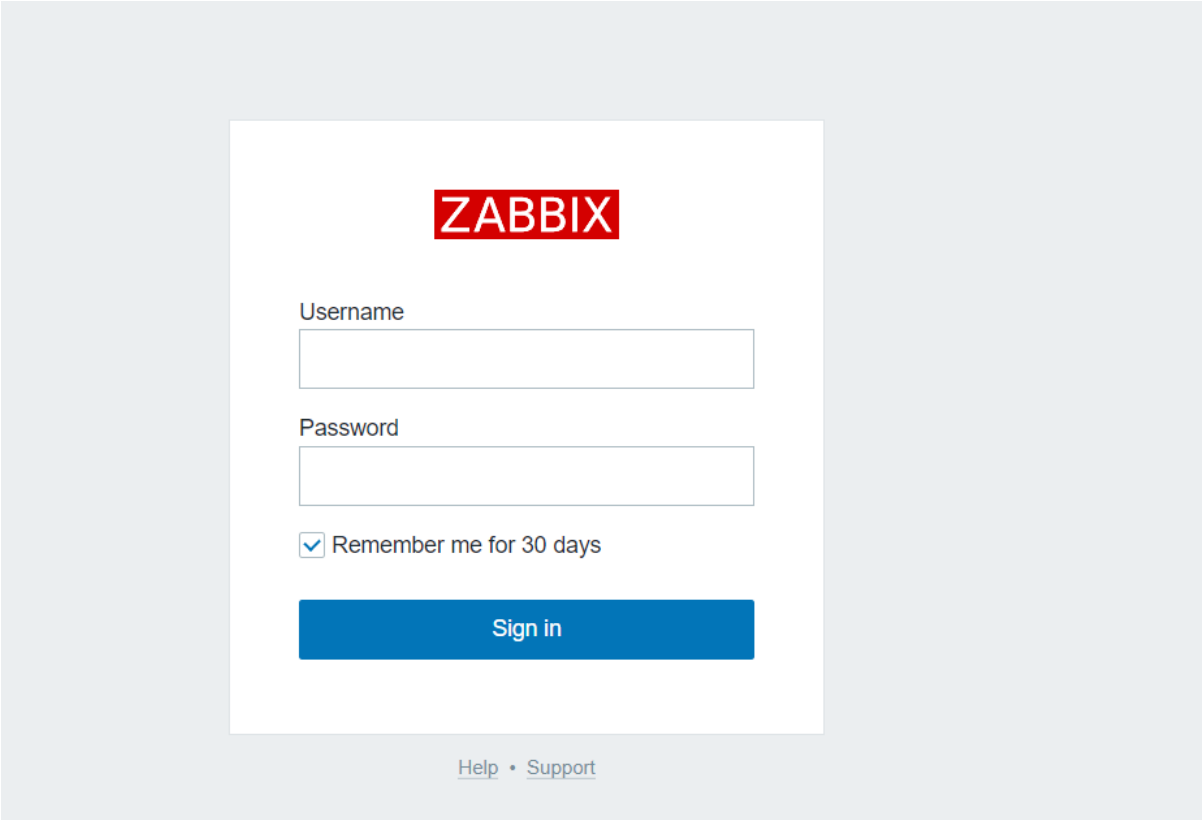
Aprire la sezione *Instances* e selezionare l'istanza dell'appliance Zabbix appena distribuita.



Per accedere al frontend Zabbix utilizzando l'indirizzo IPv4 pubblico o il DNS IPv4 pubblico dell'istanza, consentire il traffico HTTPS in ingresso sulla porta 443 nel Security Group.



Ora è possibile accedere al frontend Zabbix aprendo l'indirizzo IPv4 pubblico o il DNS IPv4 pubblico dell'appliance Zabbix.



Per ottenere la password, è necessario connettersi alla macchina utilizzando la console seriale o SSH. Il nome utente SSH predefinito è rocky. Non dimenticare di utilizzare la coppia di chiavi selezionata o generata in precedenza. La generazione delle credenziali potrebbe richiedere del tempo. La generazione delle credenziali potrebbe richiedere alcuni minuti. Attendere alcuni minuti prima di tentare la connessione.

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

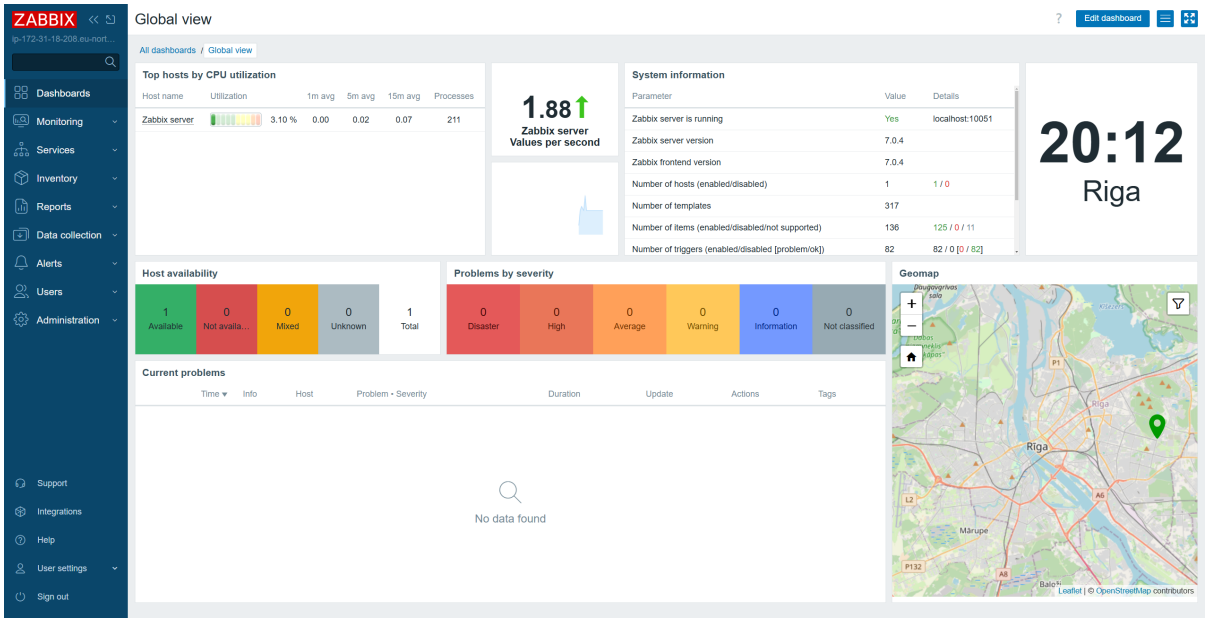
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

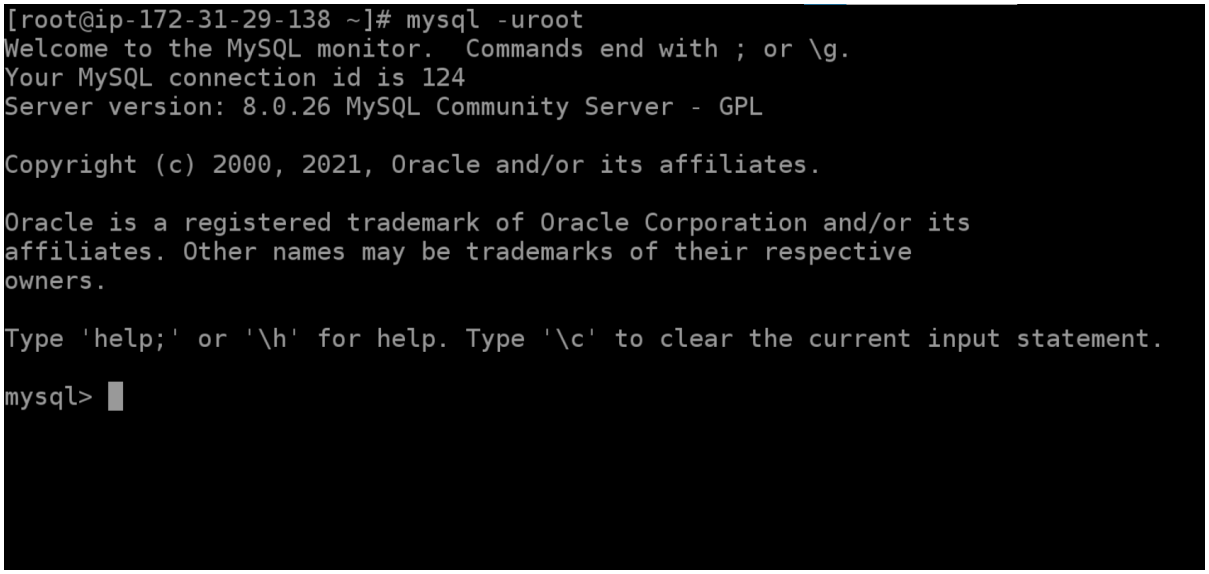
********************************************************************************
```

Ora è possibile utilizzare queste informazioni per accedere al frontend Zabbix.



Accesso al database

È possibile accedere al database di Zabbix utilizzando l'utente `root` senza immettere una password.



Le credenziali sono memorizzate nel file di configurazione `/root/.my.cnf`.

Il fuso orario predefinito del frontend di Zabbix può essere modificato in *Amministrazione* > *Generale* > *GUI*

Guida alla distribuzione di AWS per il proxy Zabbix

Panoramica

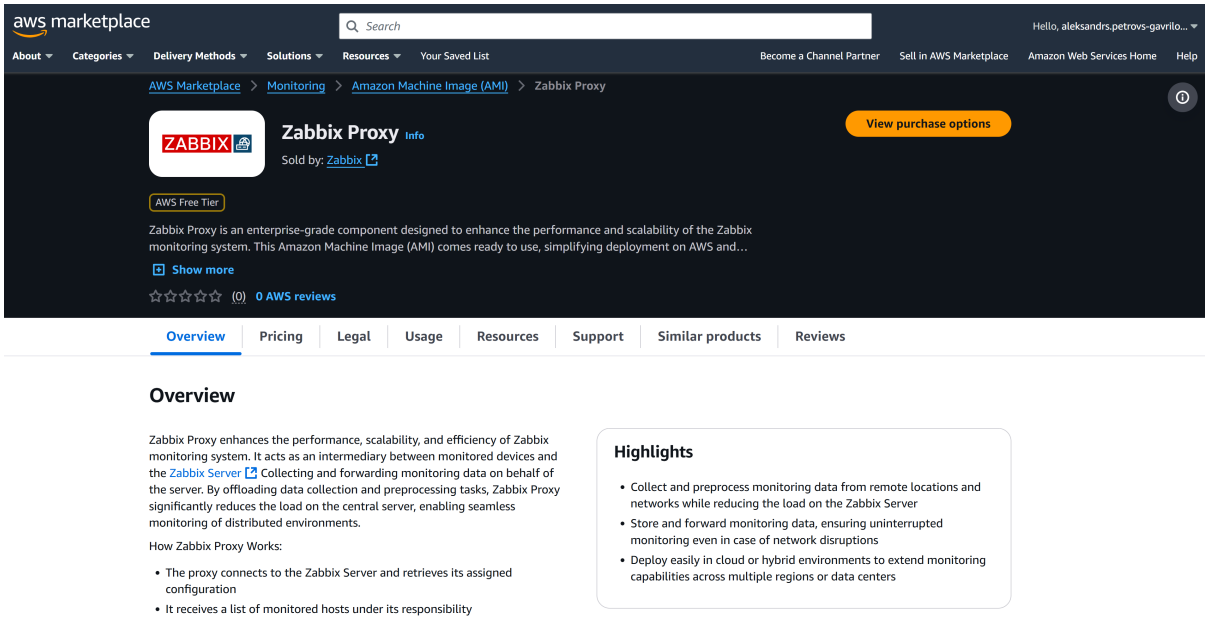
Questa pagina descrive come distribuire Zabbix proxy utilizzando AWS.

> **Attention:**
> Sono disponibili diverse versioni di Zabbix per la distribuzione sulla piattaforma. Assicurarsi di leggere la documentazione relativa alla stessa versione di Zabbix distribuita. La documentazione per le diverse versioni di Zabbix può essere selezionata dal menu a discesa delle versioni.

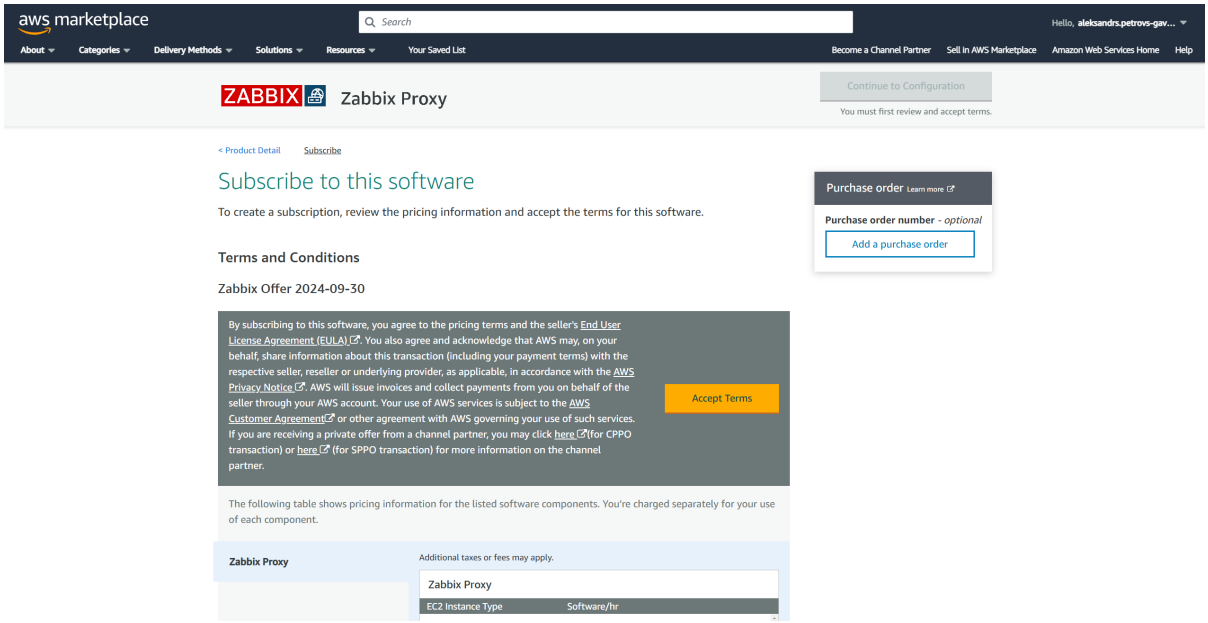
Distribuzione del proxy Zabbix

> **Attention:**
> Per utilizzare il componente proxy Zabbix è necessaria un'istanza funzionante del server Zabbix.

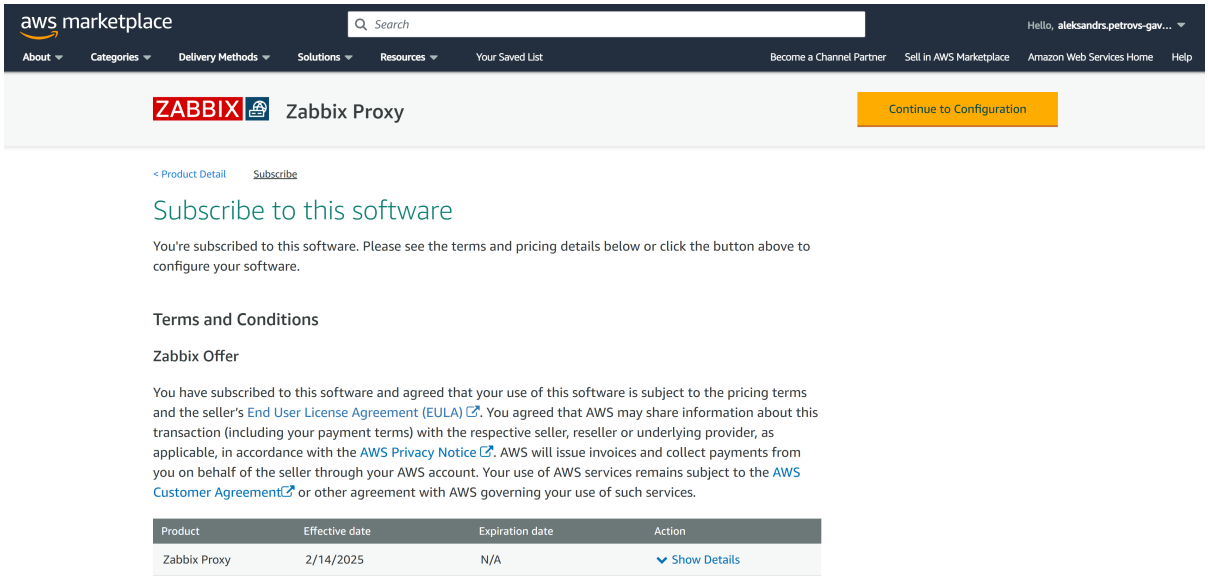
Per distribuire il proxy Zabbix, accedere innanzitutto alla pagina AWS Marketplace di Zabbix. Fare clic su *View purchase options* per distribuire l'immagine e procedere alla pagina di sottoscrizione.



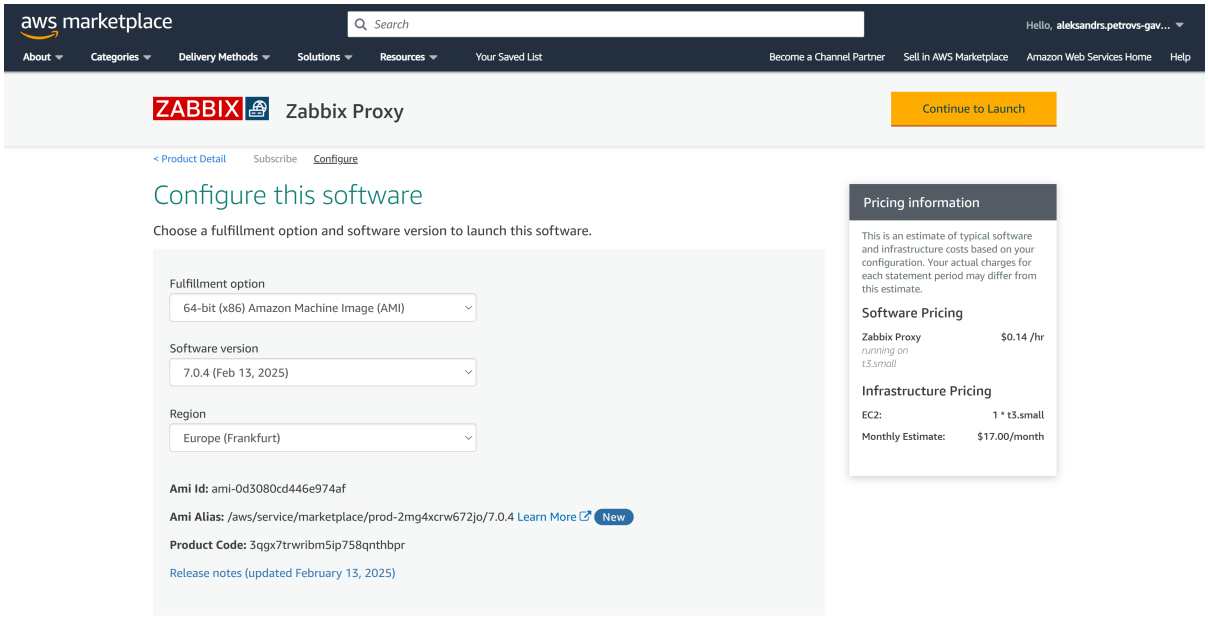
Fare clic sul pulsante *Accept terms* per accettare i *Terms and Conditions*.



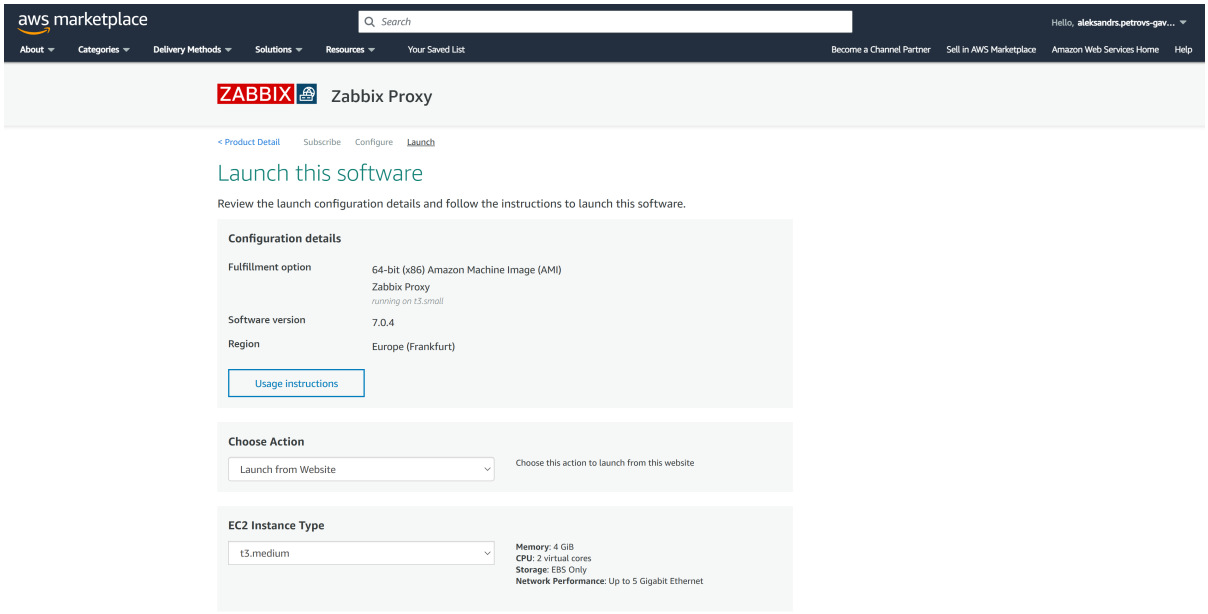
Fare clic sul pulsante *Continue to Configuration* per passare alla sezione Configuration del proxy Zabbix.



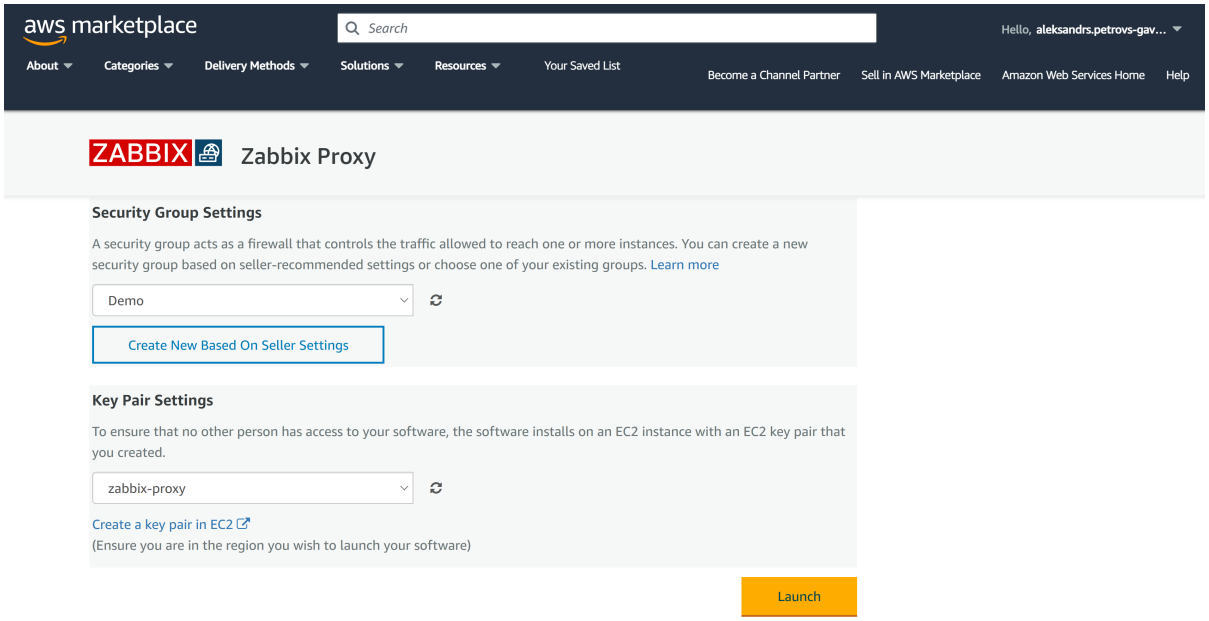
Selezionare l'opzione Fulfilment, la versione di Zabbix e la Region preferita.



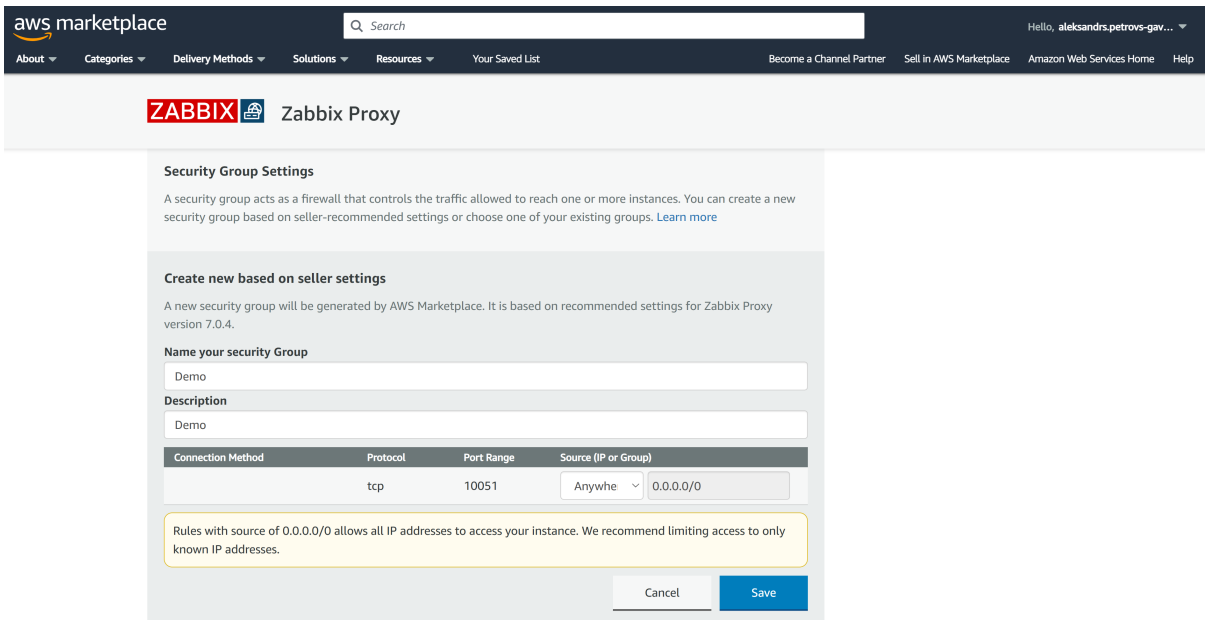
Fare clic sul pulsante *Continue to Launch*. Nella pagina Launch, selezionare l'opzione Launch preferita, EC2 Instance Type, VPC e le impostazioni Subnet.



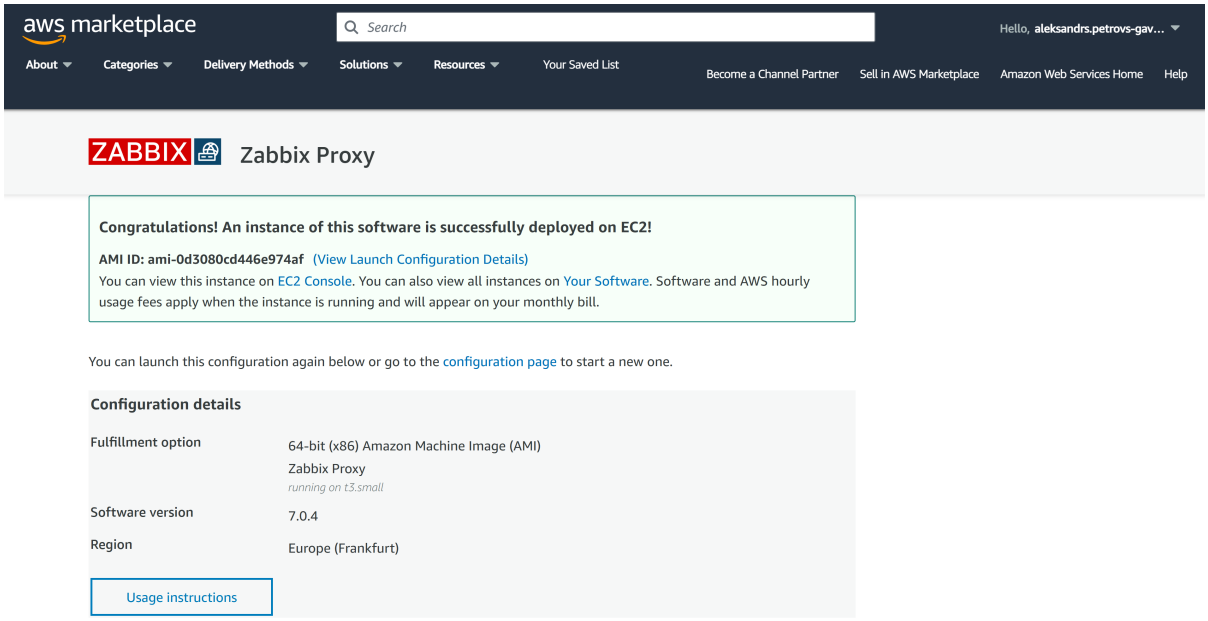
Nella parte inferiore della pagina Launch, specificare le impostazioni Security Group e selezionare o creare una coppia di chiavi EC2. Assicurarsi di salvare la chiave privata in un luogo sicuro.



Se si seleziona *Create a new Security Group based on the Seller settings*, sarà consentito il traffico sulla porta TCP 10051 (server Zabbix). Nota: se si desidera accedere all'istanza tramite SSH, è inoltre necessario consentire le connessioni SSH nelle impostazioni Security Group.

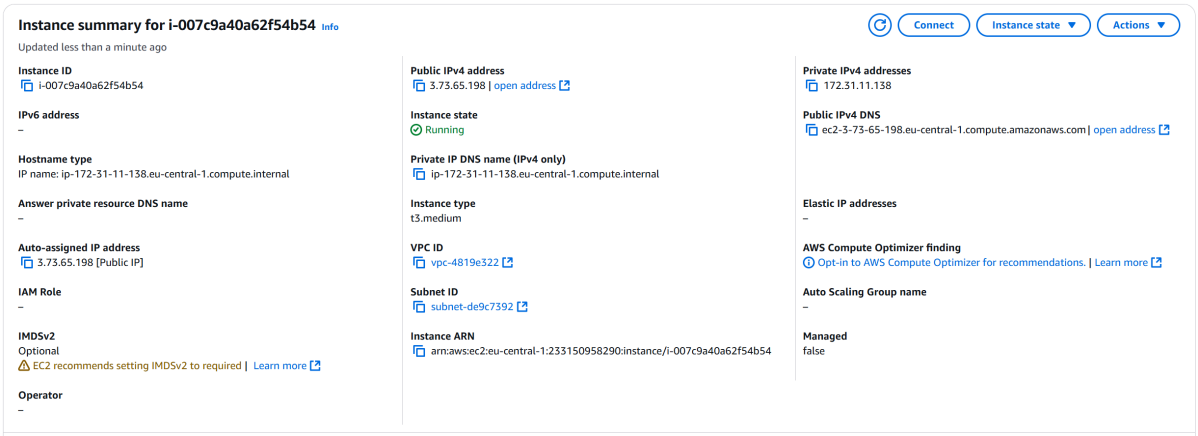


Infine, fare clic sul pulsante *Launch*. Congratulazioni, il proxy Zabbix è ora in fase di distribuzione.



Configurazione e accesso al proxy Zabbix

Aprire la sezione Instances nella console EC2 e selezionare l'istanza proxy Zabbix appena distribuita.



Connettersi all'indirizzo IPv4 pubblico o al DNS IPv4 pubblico del proxy Zabbix tramite SSH. Il nome utente SSH predefinito è `rocky`. Non dimenticare di utilizzare la coppia di chiavi generata in precedenza.

La generazione delle credenziali può richiedere alcuni minuti. Attendere alcuni minuti prima di tentare la connessione.



Il file di configurazione del proxy Zabbix si trova in `/etc/zabbix/zabbix_proxy.conf`.

Il file di configurazione può essere aperto con un editor di testo a scelta, ad esempio vi:

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

La configurazione iniziale del proxy consiste nel modificare i parametri `ProxyMode`, `Server` e `Hostname` nel file di configurazione del proxy Zabbix.

La modalità proxy può essere modificata modificando il parametro `ProxyMode`. Impostare `ProxyMode=0` per la modalità attiva o `ProxyMode=1` per la modalità passiva.

Il parametro `Server` deve contenere l'indirizzo IP del server Zabbix. Ad esempio:

```
Server=192.168.50.141
```

Il parametro `Hostname` deve contenere il nome host del proxy Zabbix. Il nome host definito qui deve corrispondere al nome del proxy specificato durante la configurazione del proxy nel frontend Zabbix. Ad esempio:

```
Hostname=EC2-Frankfurt
```

Una volta completata la configurazione, riavviare il proxy Zabbix con il comando seguente:

```
sudo systemctl restart zabbix-proxy
```

Nota: utilizzare sudo per modificare i file di configurazione ed eseguire comandi dalla riga di comando.

```
### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name of Zabbix server to get configuration data from and send data to.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix
server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated
 equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

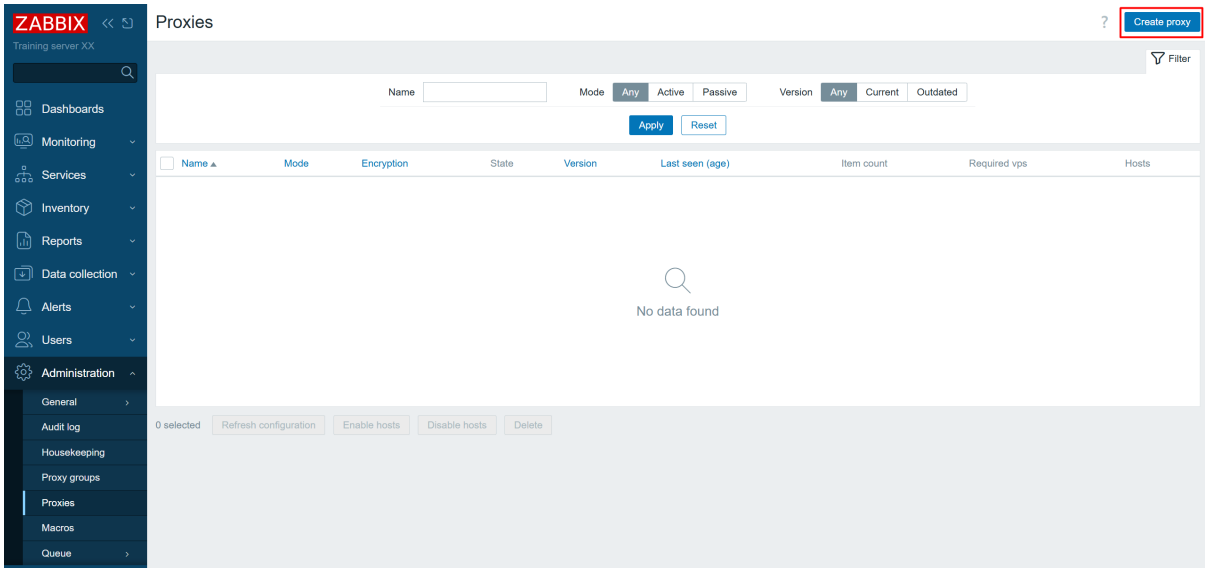
Hostname=EC2-Frankfurt
```

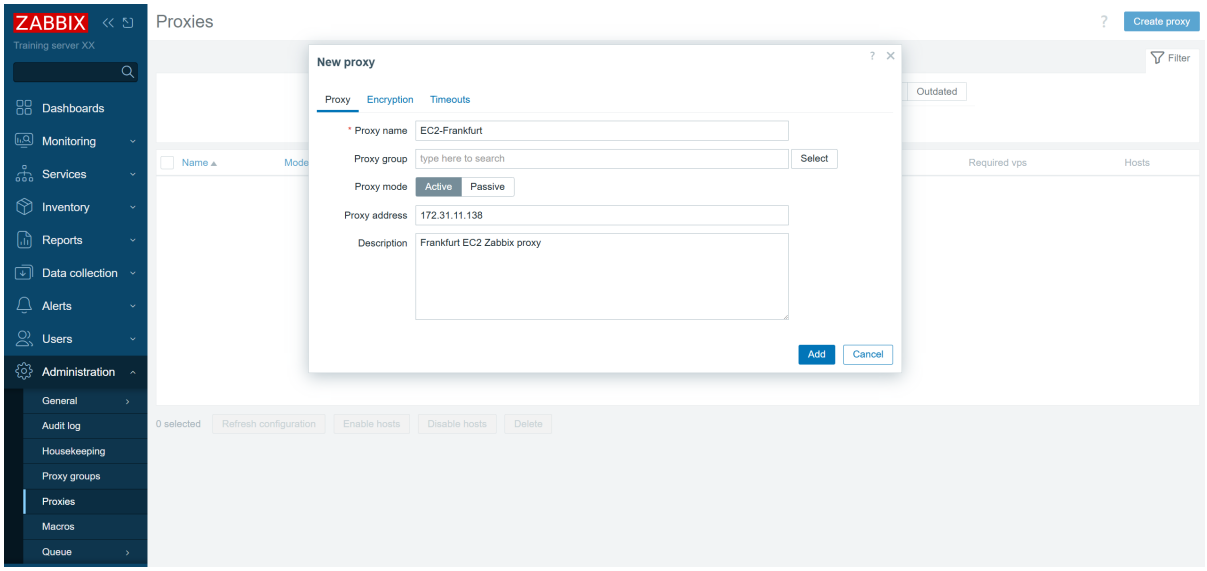
Aprire il frontend Zabbix e accedere a *Administration* > *Proxies*. Fare clic sul pulsante *Create proxy*.

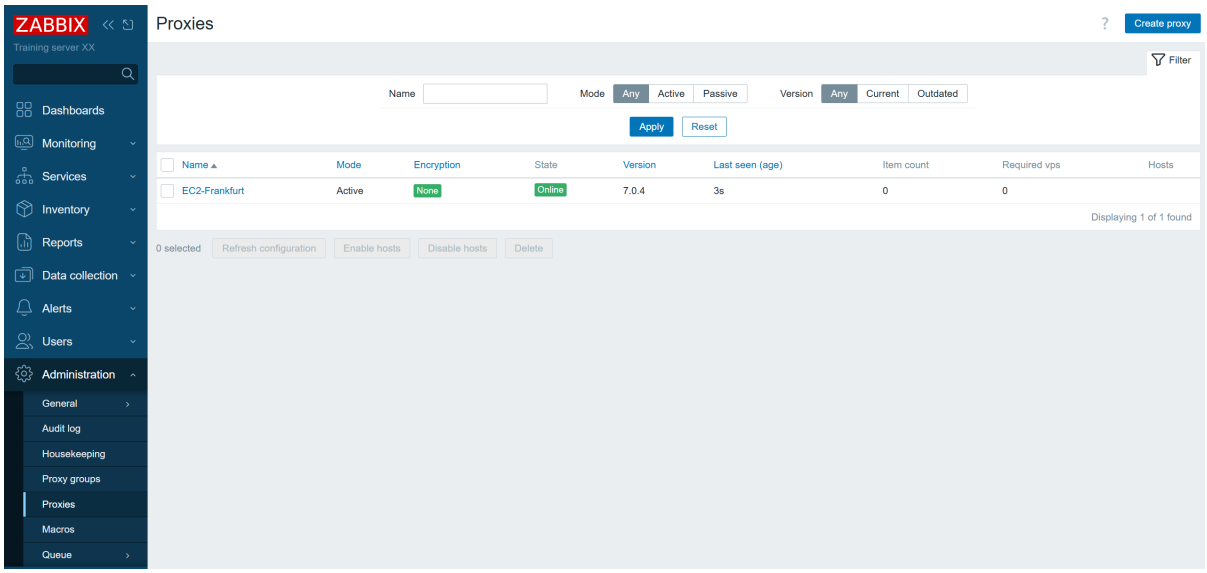
Inserire lo stesso nome del proxy indicato come valore del parametro `Hostname` nel file di configurazione del proxy.

Selezionare la modalità proxy e specificare l'indirizzo IP del proxy (facoltativo per i proxy attivi).

Fare clic su *Create proxy*.







Posizione del database del proxy

Il database SQLite del proxy è archiviato in `/tmp/zabbix_proxy.sqlite3` ed è specificato dal parametro `DBName` nel file di configurazione del proxy Zabbix.

Guida alla distribuzione di Zabbix server su Azure

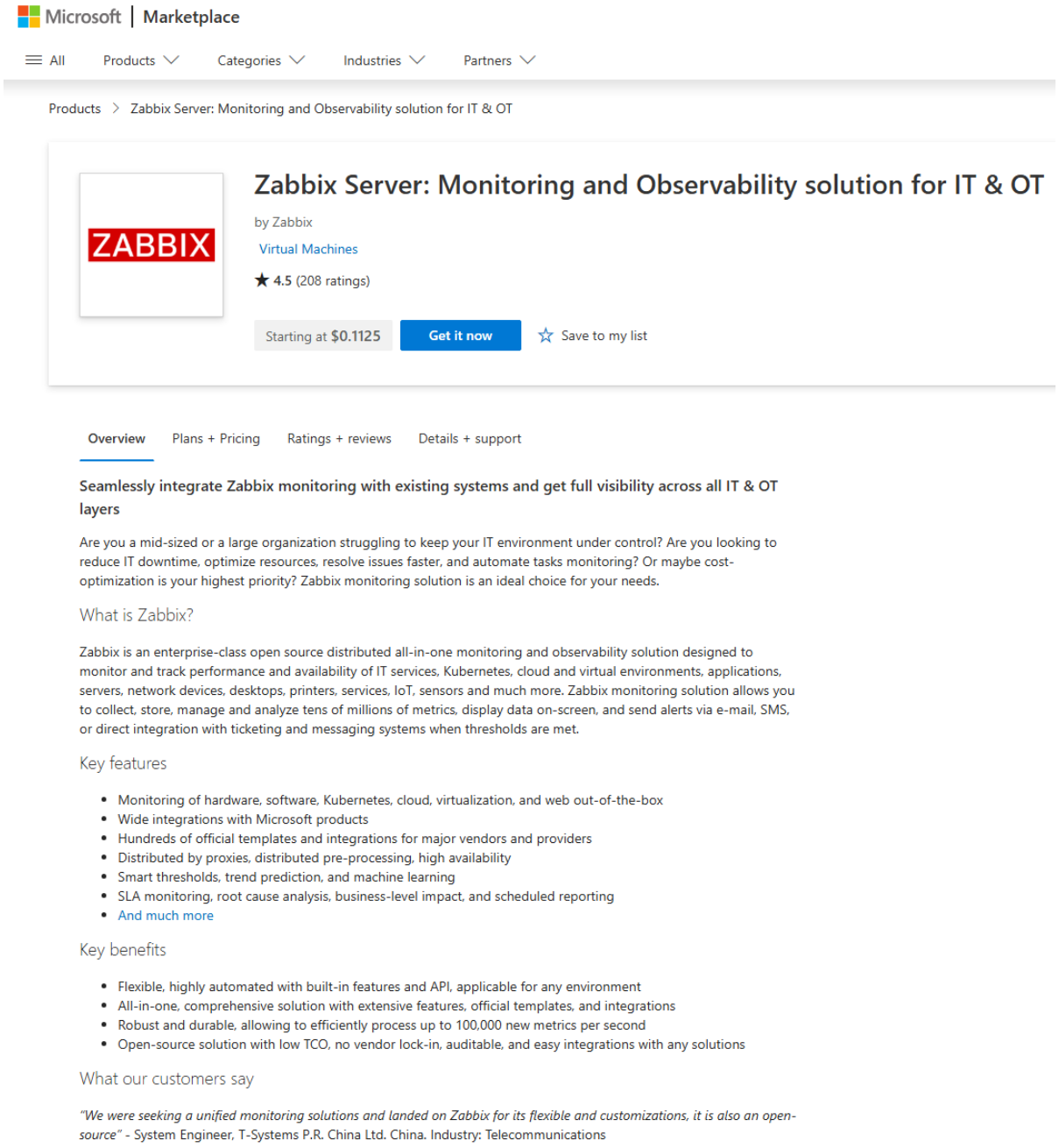
Panoramica

Questa pagina descrive come distribuire Zabbix server utilizzando Azure.

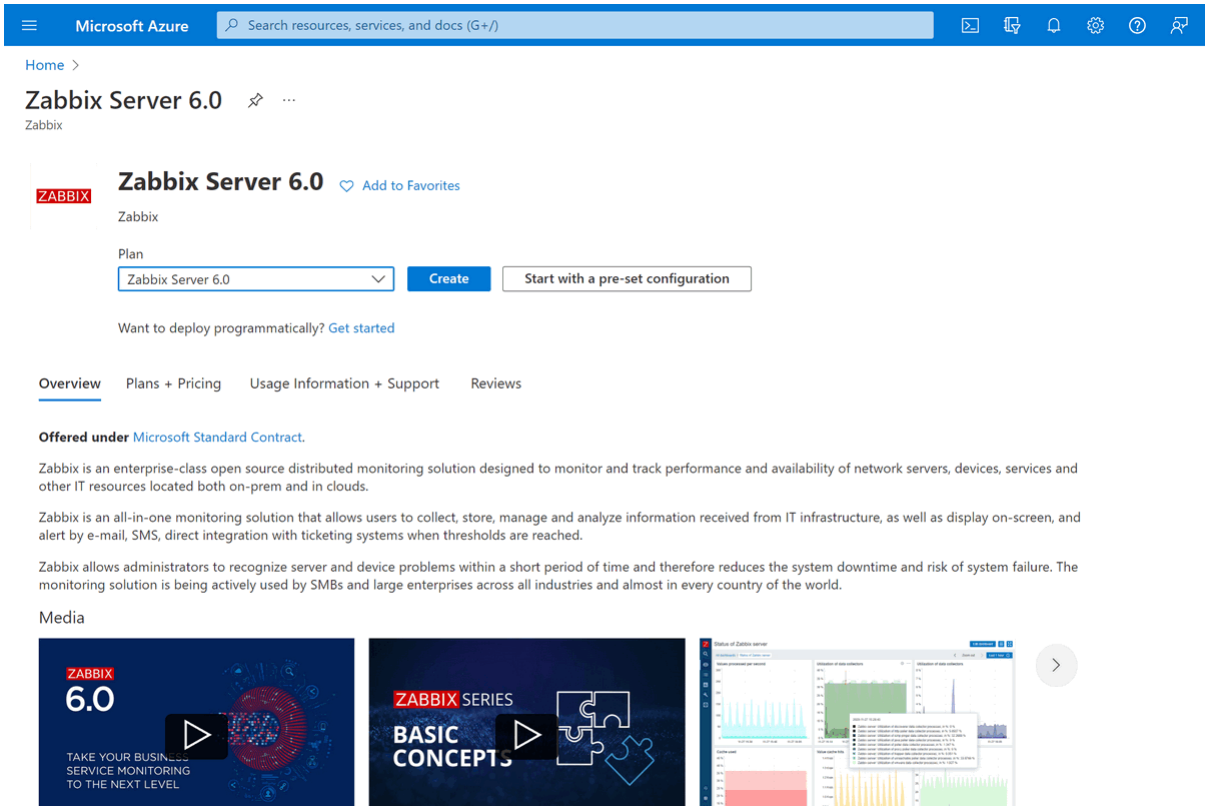
Attention:
Sono disponibili diverse versioni di Zabbix per la distribuzione sulla piattaforma. Assicurarsi di leggere la documentazione relativa alla stessa versione di Zabbix distribuita. La documentazione per le diverse versioni di Zabbix può essere selezionata dal menu a discesa delle versioni.

Distribuzione dell'appliance

Per distribuire l'appliance Zabbix, accedere innanzitutto alla pagina di Zabbix Azure Marketplace. Esaminare la panoramica, quindi fare clic su *Get it now*.



Esaminare la panoramica della distribuzione, i piani tariffari e gli altri dettagli. Quindi, fare clic su *Create* o *Start with a pre-set configuration* per distribuire l'appliance Zabbix.

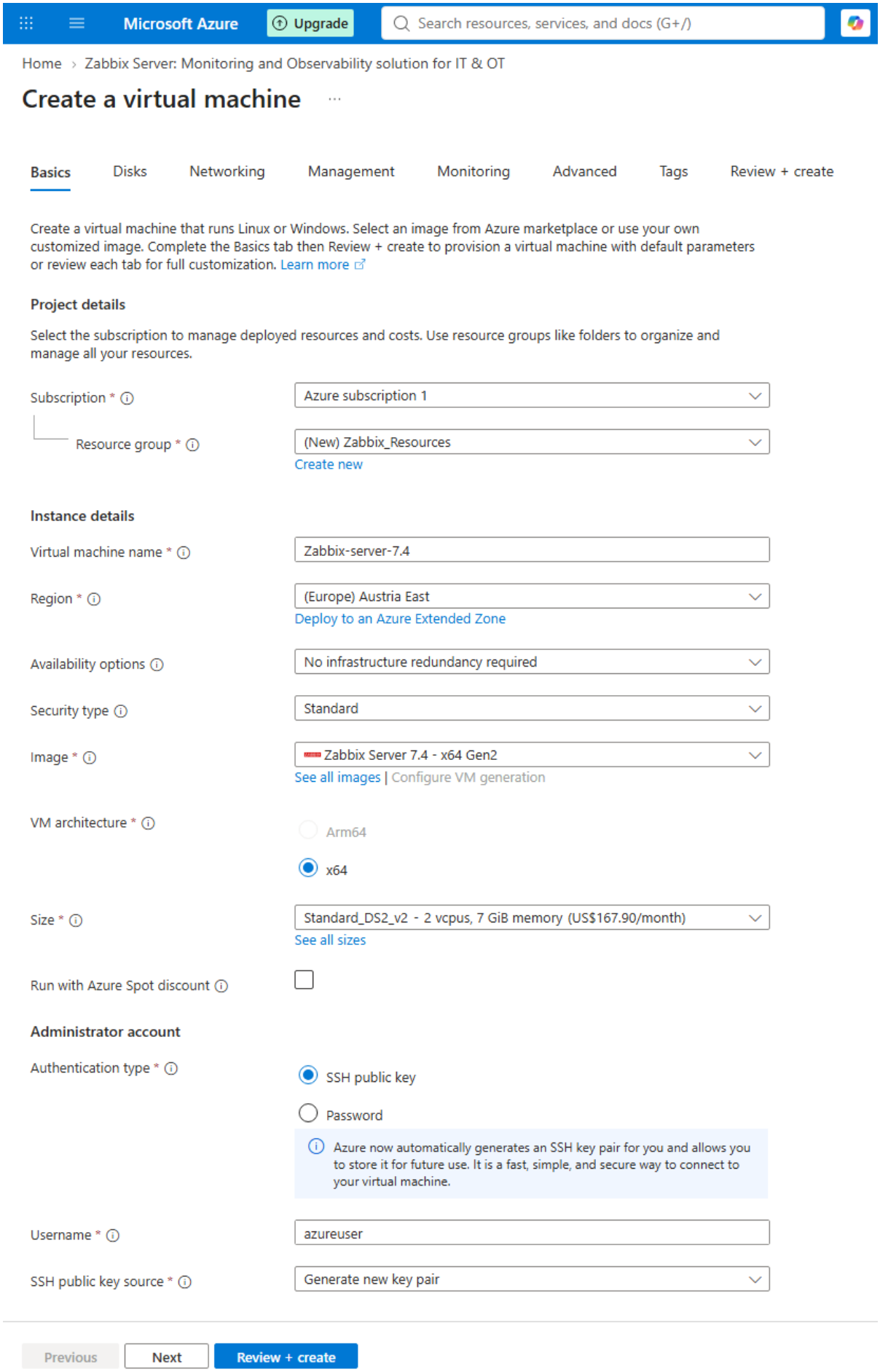


Configurare la VM selezionando le dimensioni della VM, creando un account amministratore e definendo le impostazioni del disco, della rete, di gestione e altre impostazioni. Queste impostazioni possono variare in base ai requisiti dell'infrastruttura.

È possibile utilizzare la seguente configurazione come esempio:

Scheda *Basics*:

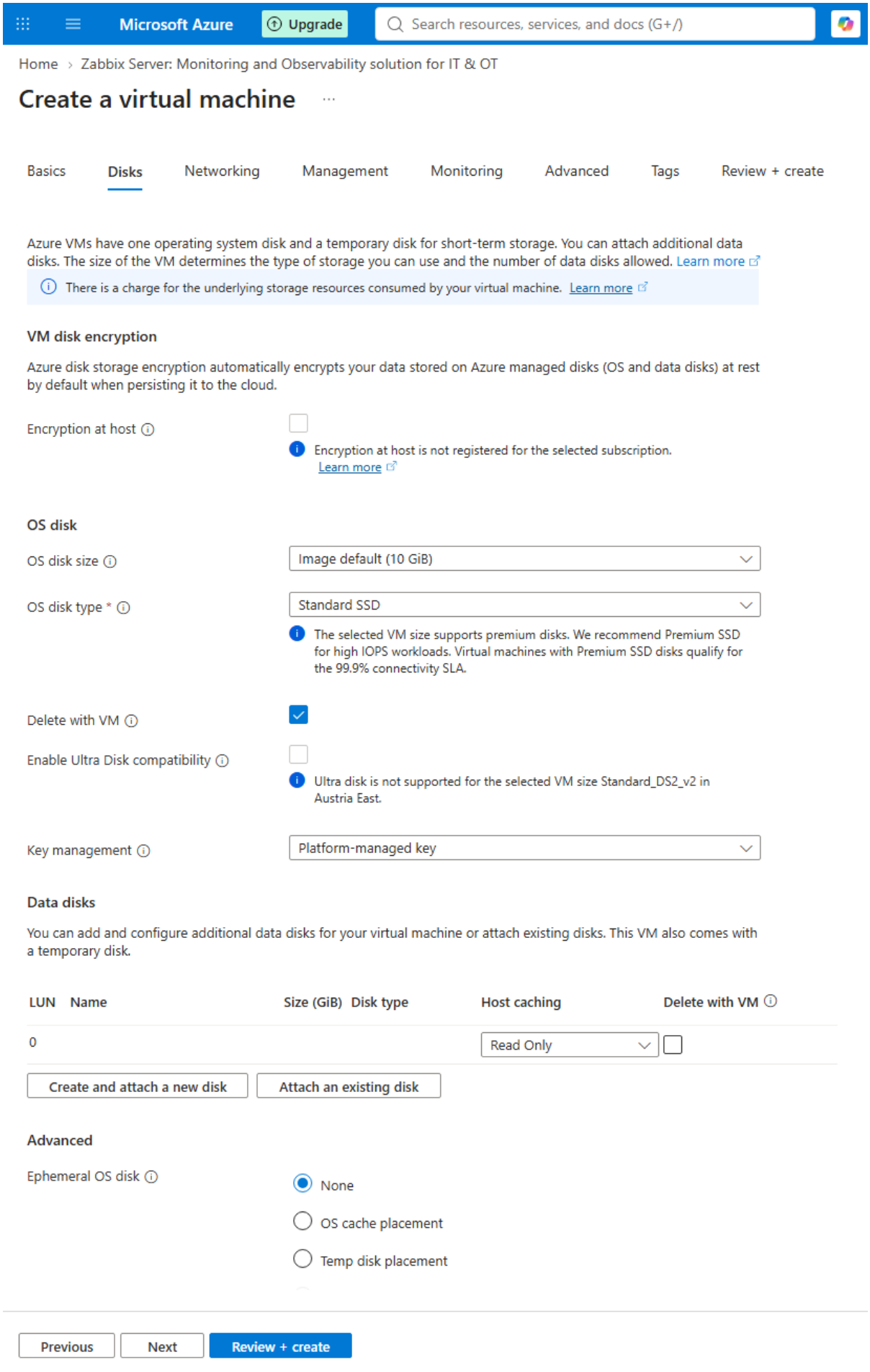
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-server--7.4
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Server 7.4 - x64 Gen2
- *Size*: Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Scheda *Disks*:

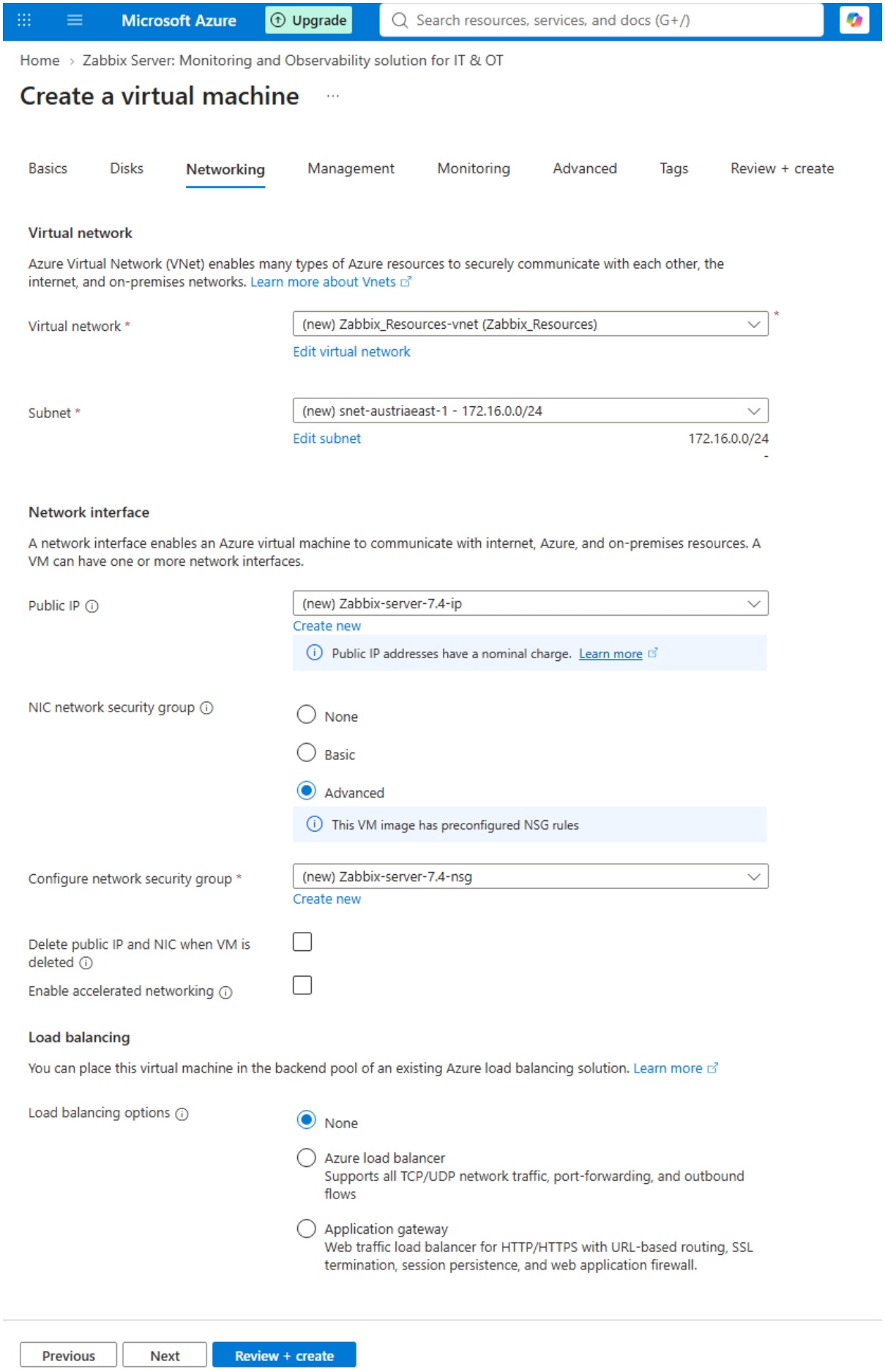
- *Encryption at host*: unchecked
- *OS disk type*: Standard SSD

- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

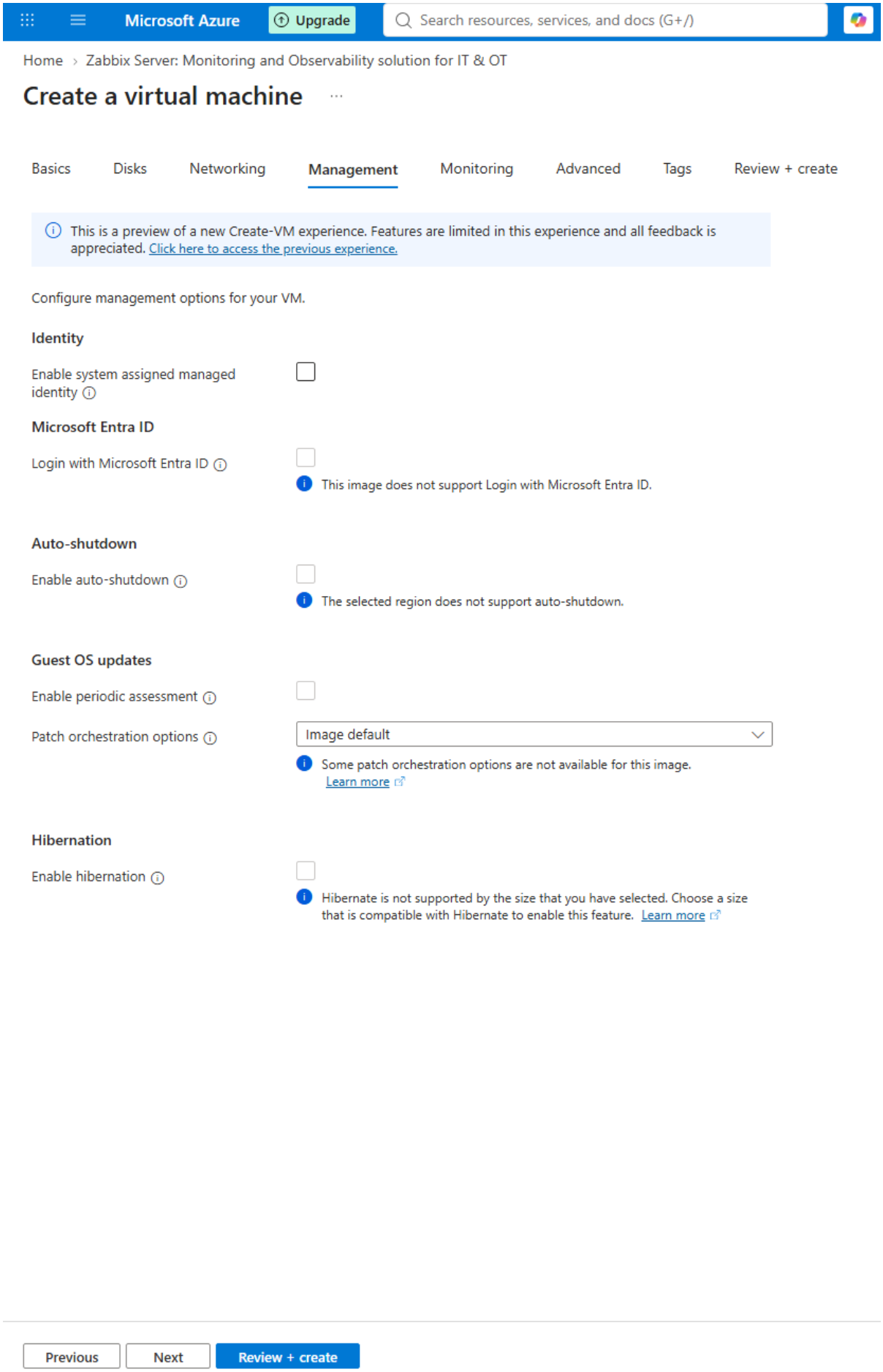


Scheda *Networking*:

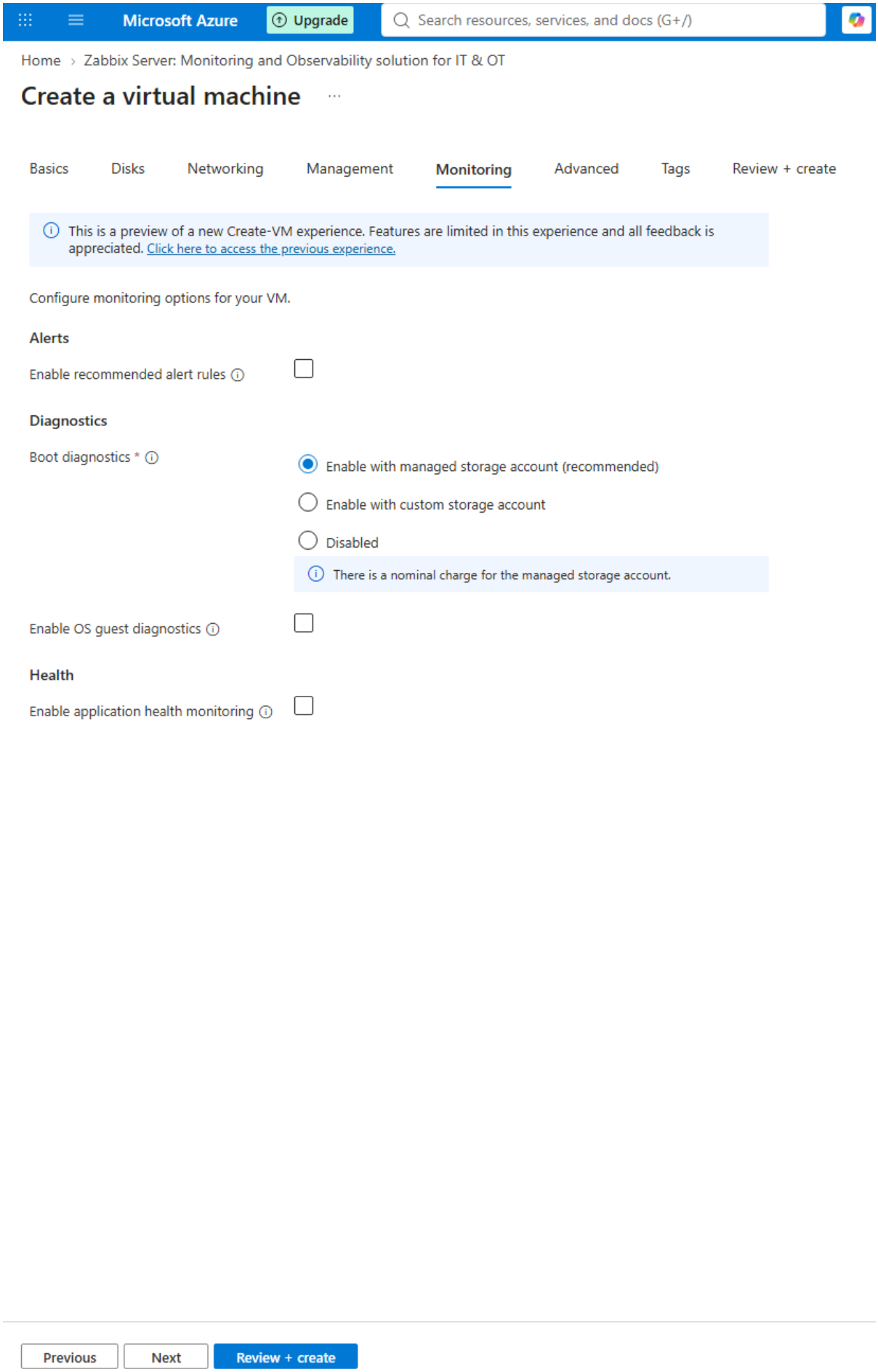
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-server-7.4-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-server-7.4-nsg



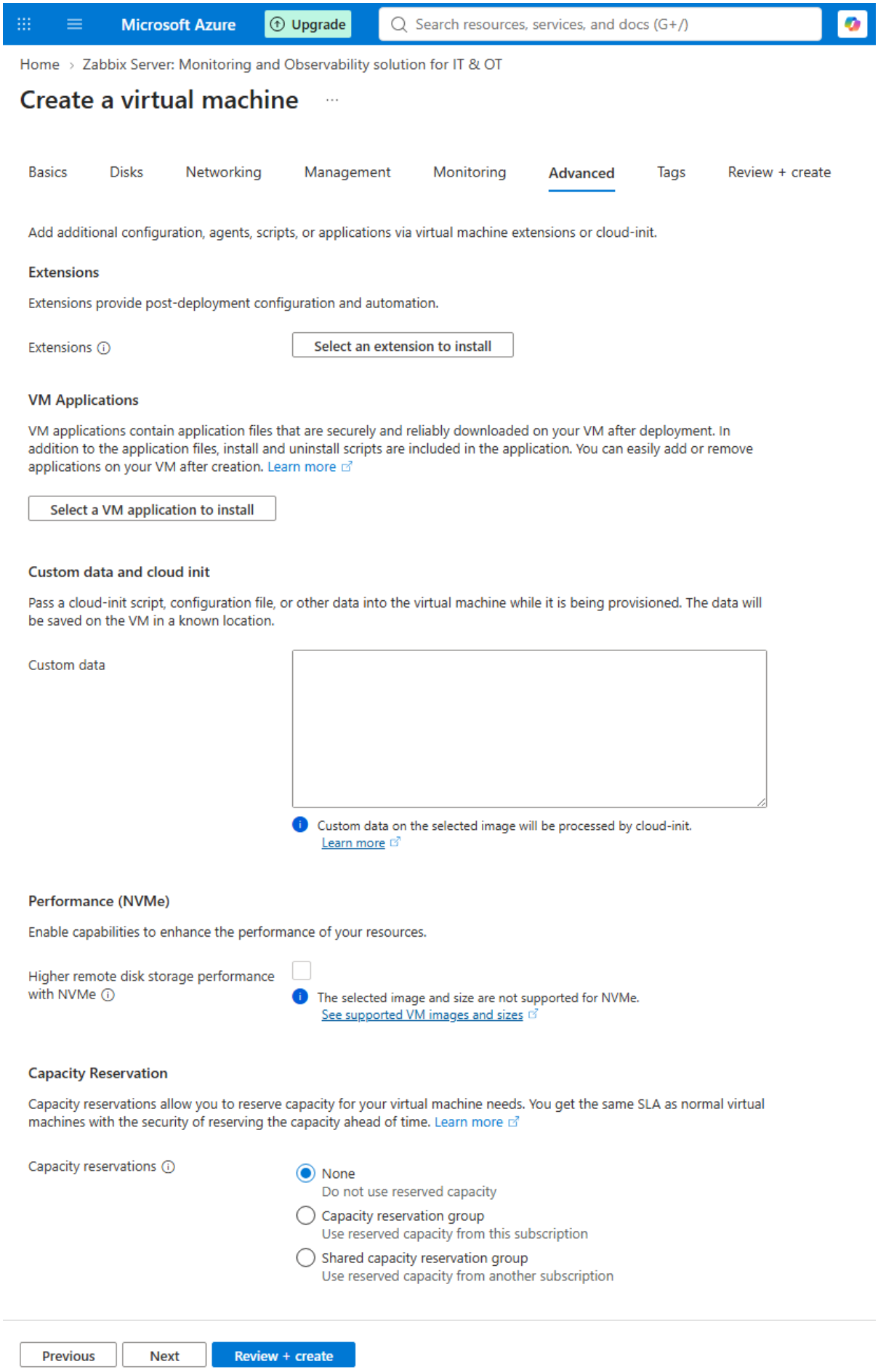
Scheda *Management*: i parametri possono essere lasciati ai valori predefiniti.



Scheda *Monitoring*: i parametri possono essere lasciati ai valori predefiniti.



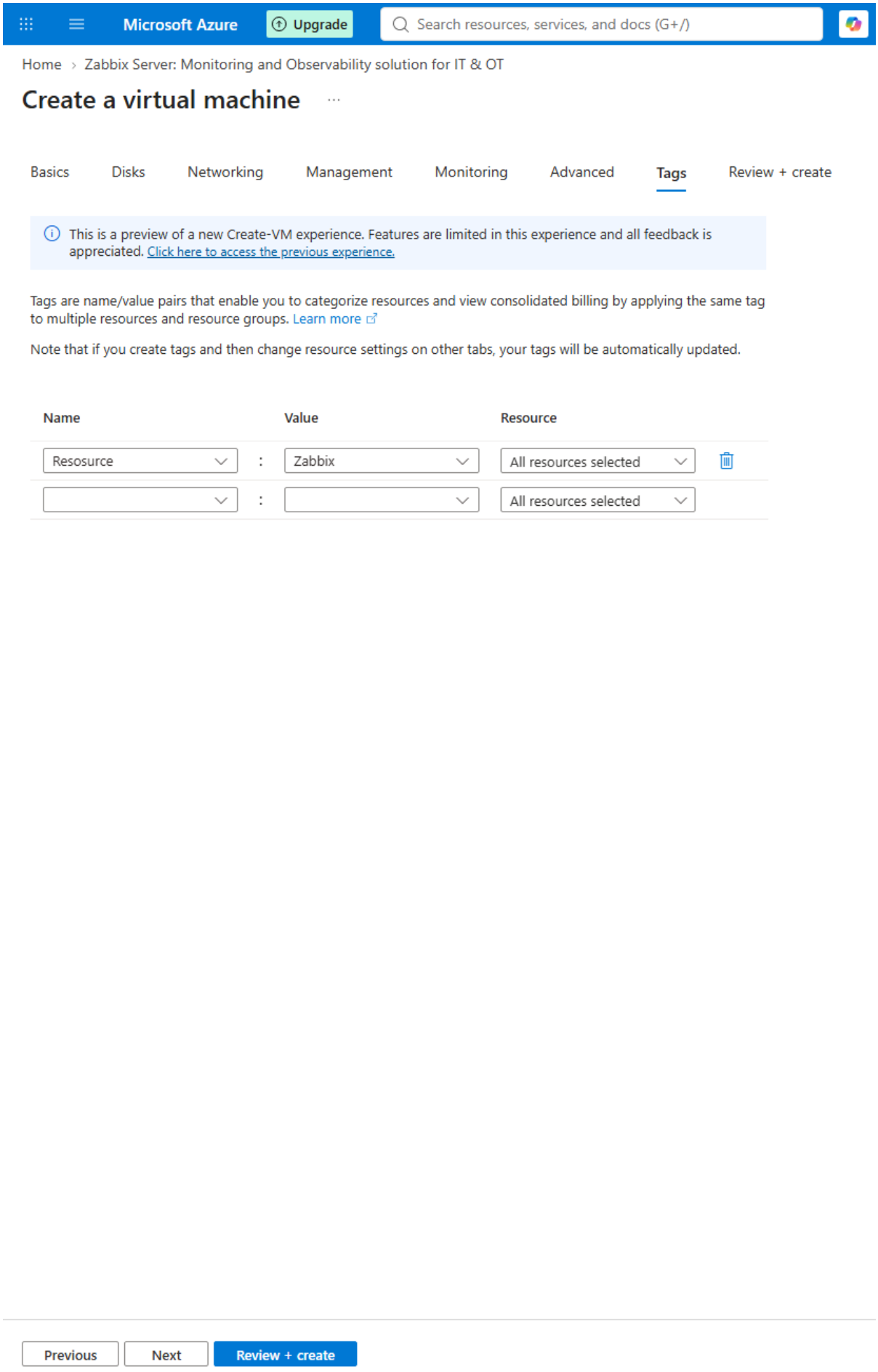
Scheda *Advanced*: i parametri possono essere lasciati ai valori predefiniti.



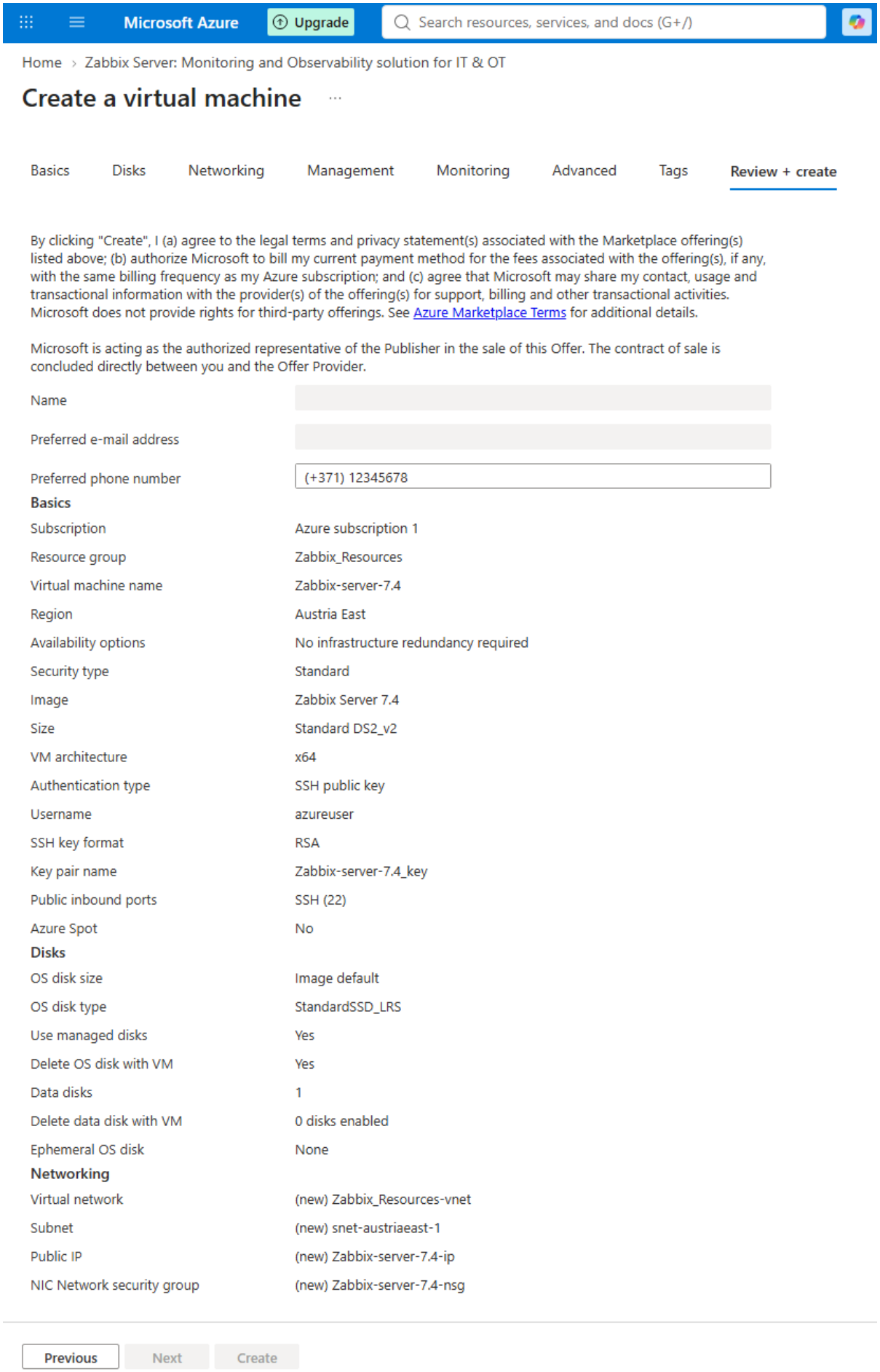
Scheda *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Confermare i dettagli della VM e creare la VM.



Si verrà reindirizzati alla pagina dello stato della distribuzione della VM. Attendere il completamento della distribuzione.

Al termine della distribuzione, aprire la risorsa e annotare il relativo indirizzo IP pubblico. A questo punto sarà necessario connettersi al server Zabbix tramite SSH (o un altro metodo disponibile) per ottenere le credenziali di accesso al frontend Zabbix.

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

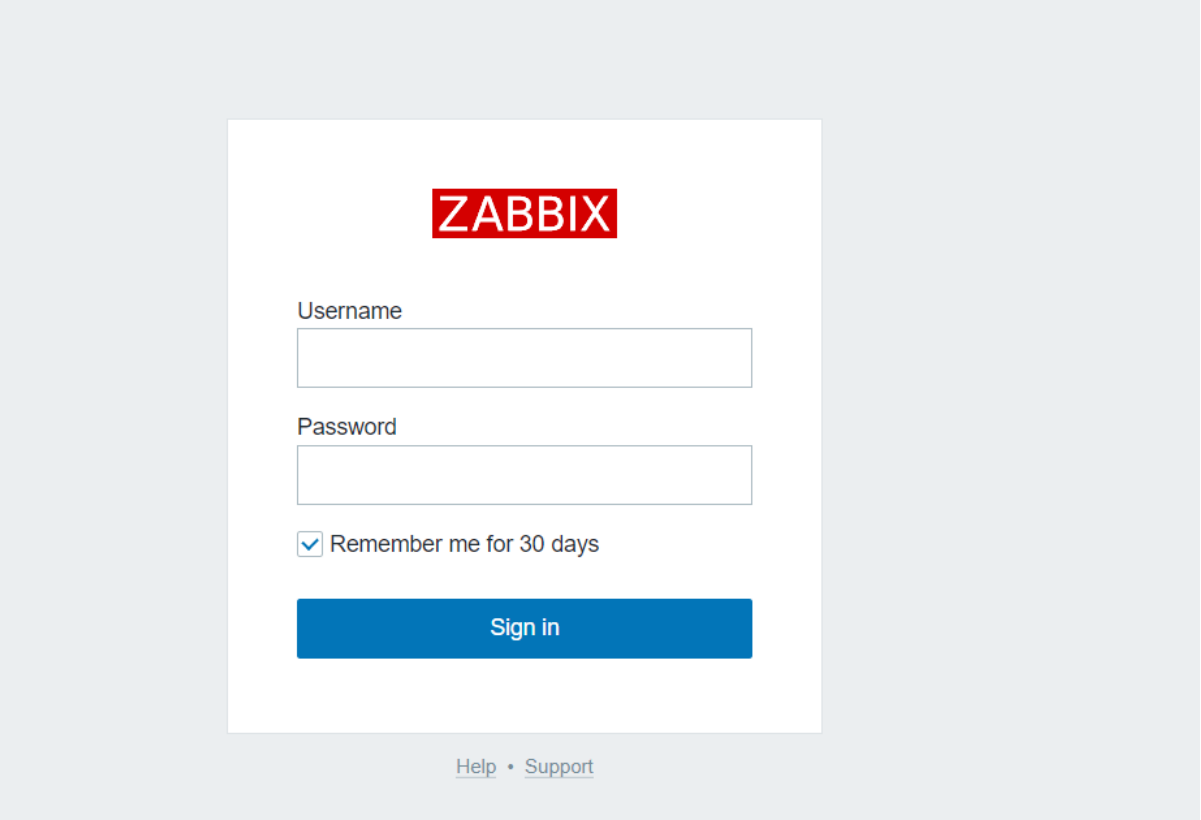

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Accedere al frontend Zabbix aprendo l'indirizzo IP pubblico dell'appliance Zabbix in un browser web. Per effettuare l'accesso, utilizzare il nome utente e la password ottenuti nel passaggio precedente.



Accesso al database

È possibile accedere al database di Zabbix utilizzando l'utente `root` senza inserire una password. Le credenziali sono memorizzate nel file di configurazione `/root/.my.cnf`.

Guida alla distribuzione di Azure per il proxy Zabbix

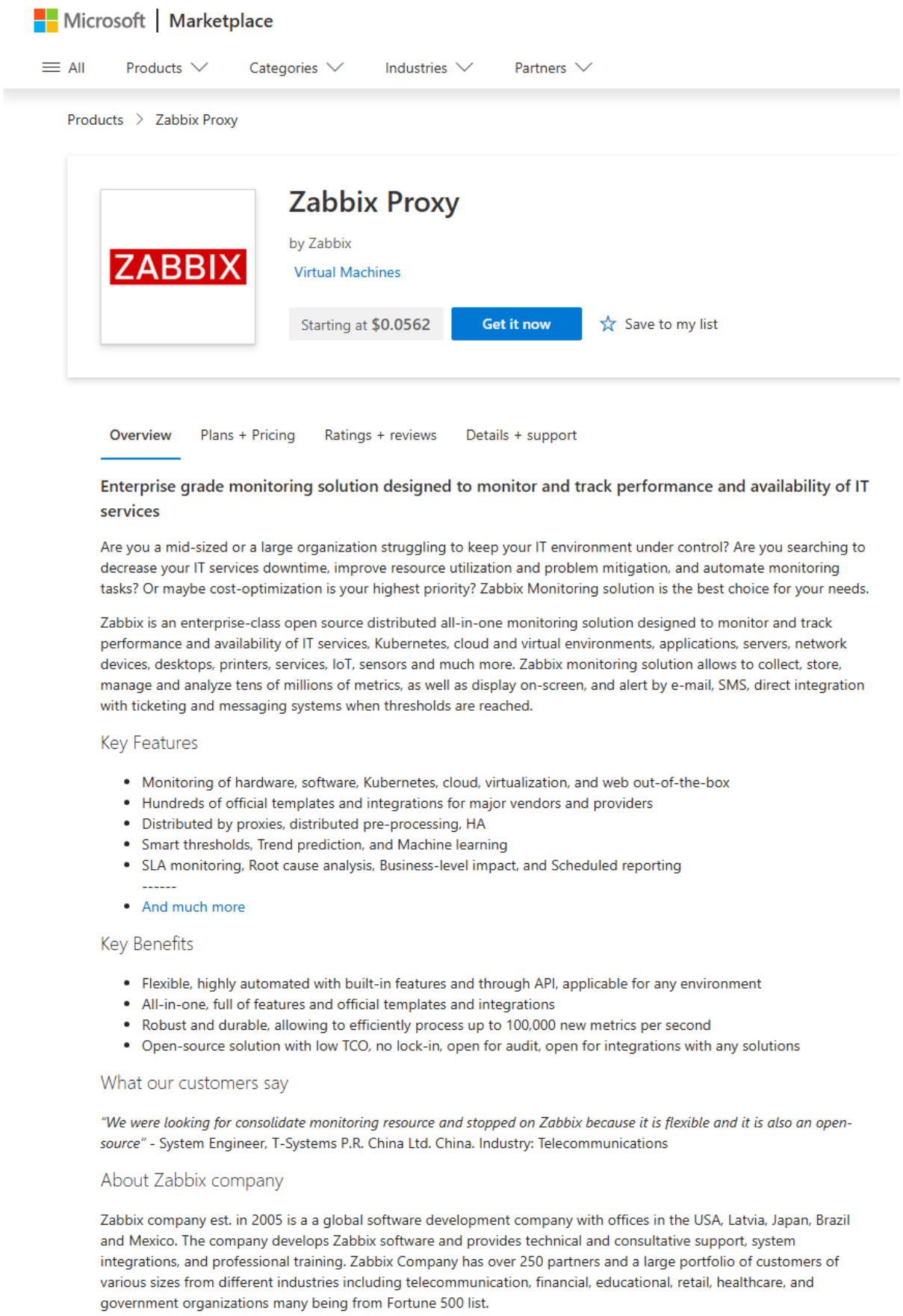
Panoramica

Questa pagina descrive come distribuire il proxy Zabbix utilizzando Azure.

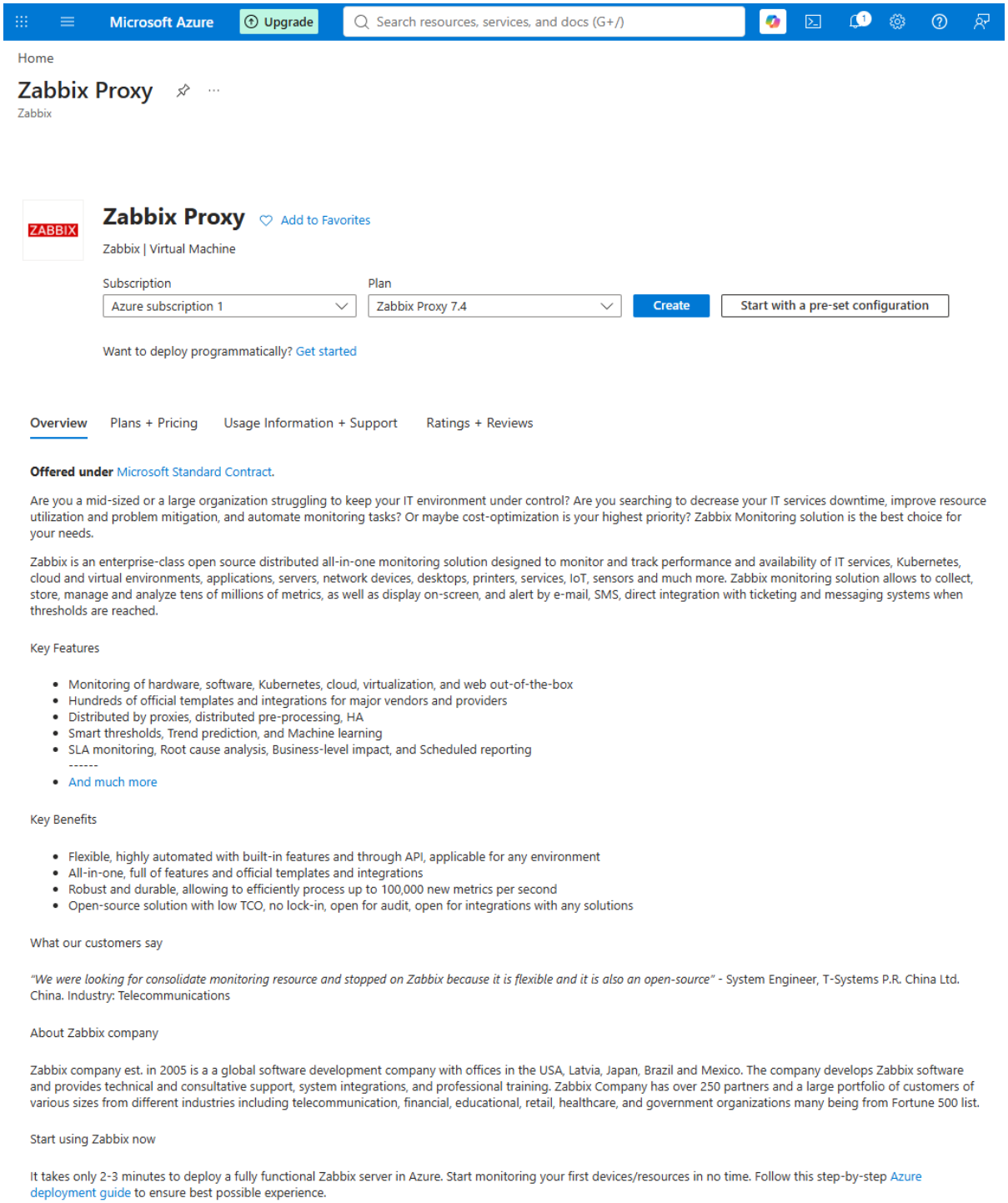
Attention:
Sono disponibili diverse versioni di Zabbix per la distribuzione sulla piattaforma. Assicurarsi di leggere la documentazione relativa alla stessa versione di Zabbix distribuita. La documentazione per le diverse versioni di Zabbix può essere selezionata dal menu a discesa delle versioni.

Distribuzione del proxy Zabbix

Per distribuire il proxy Zabbix, accedere innanzitutto alla pagina Zabbix nel marketplace Azure. Fare clic su *Get it now*.



Qui è possibile leggere ulteriori informazioni sulla distribuzione corrente, come la panoramica generale, i piani tariffari e altro. Fare clic su *Create* oppure su *Start with a pre-set configuration*.

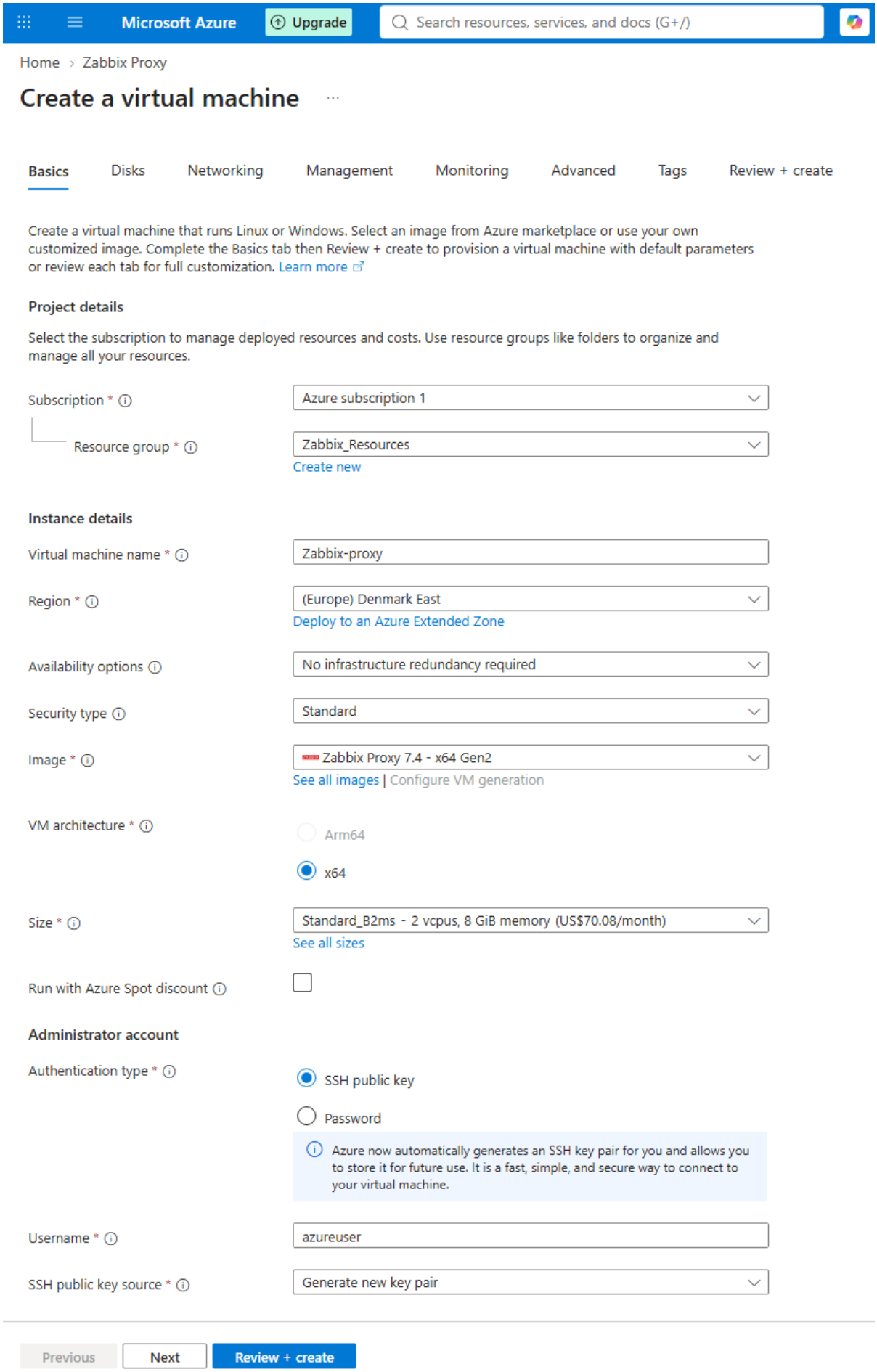


Procedere con la configurazione dettagliata della VM: selezionare le dimensioni della VM, l'account amministratore, il disco, la rete, la gestione e altre impostazioni. Queste impostazioni possono variare in base ai requisiti dell'infrastruttura.

È possibile utilizzare la seguente configurazione come esempio:

Scheda *Basics*:

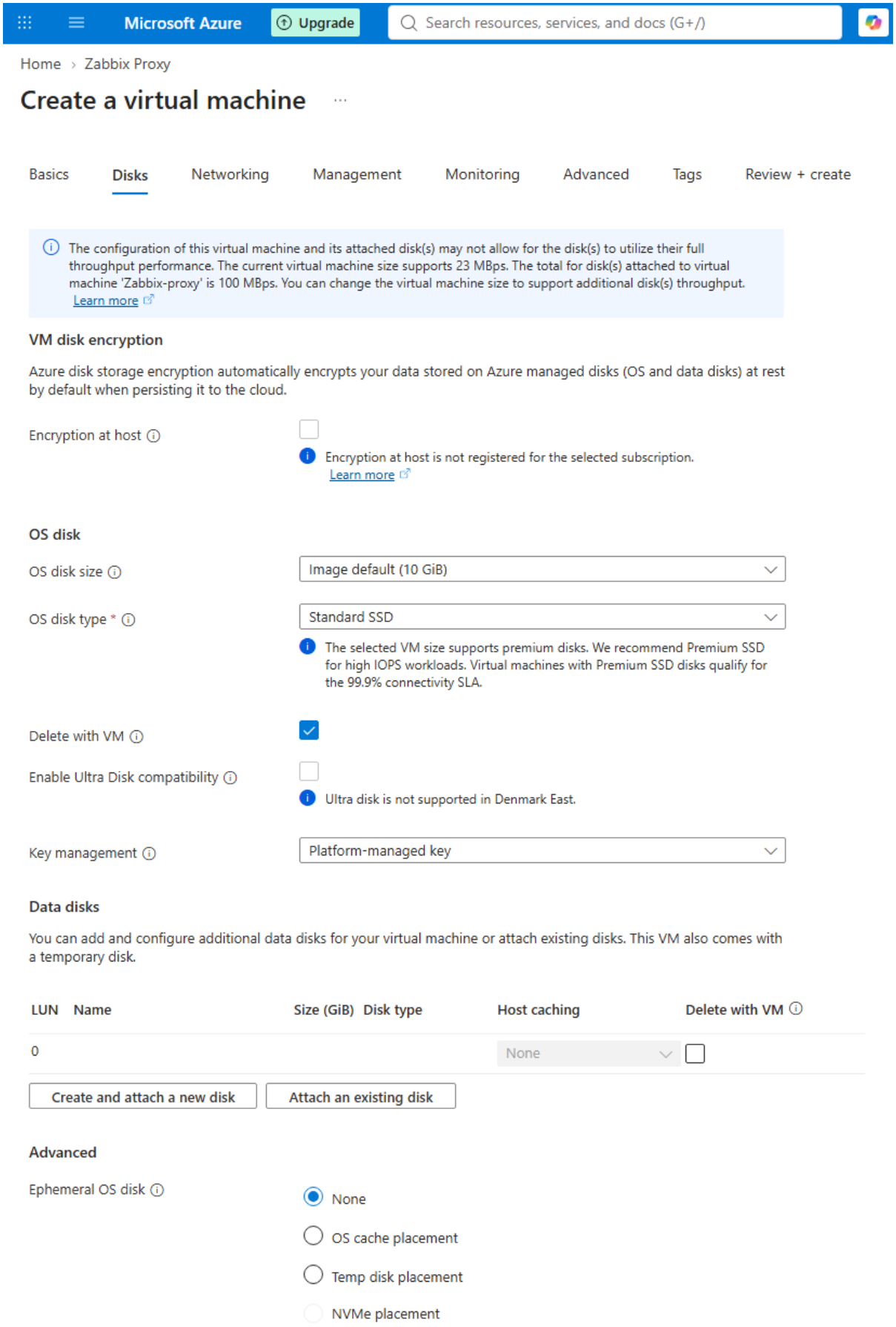
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-proxy
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Proxy 7.4 - x64 Gen2
- *Size*: Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Scheda *Disks*:

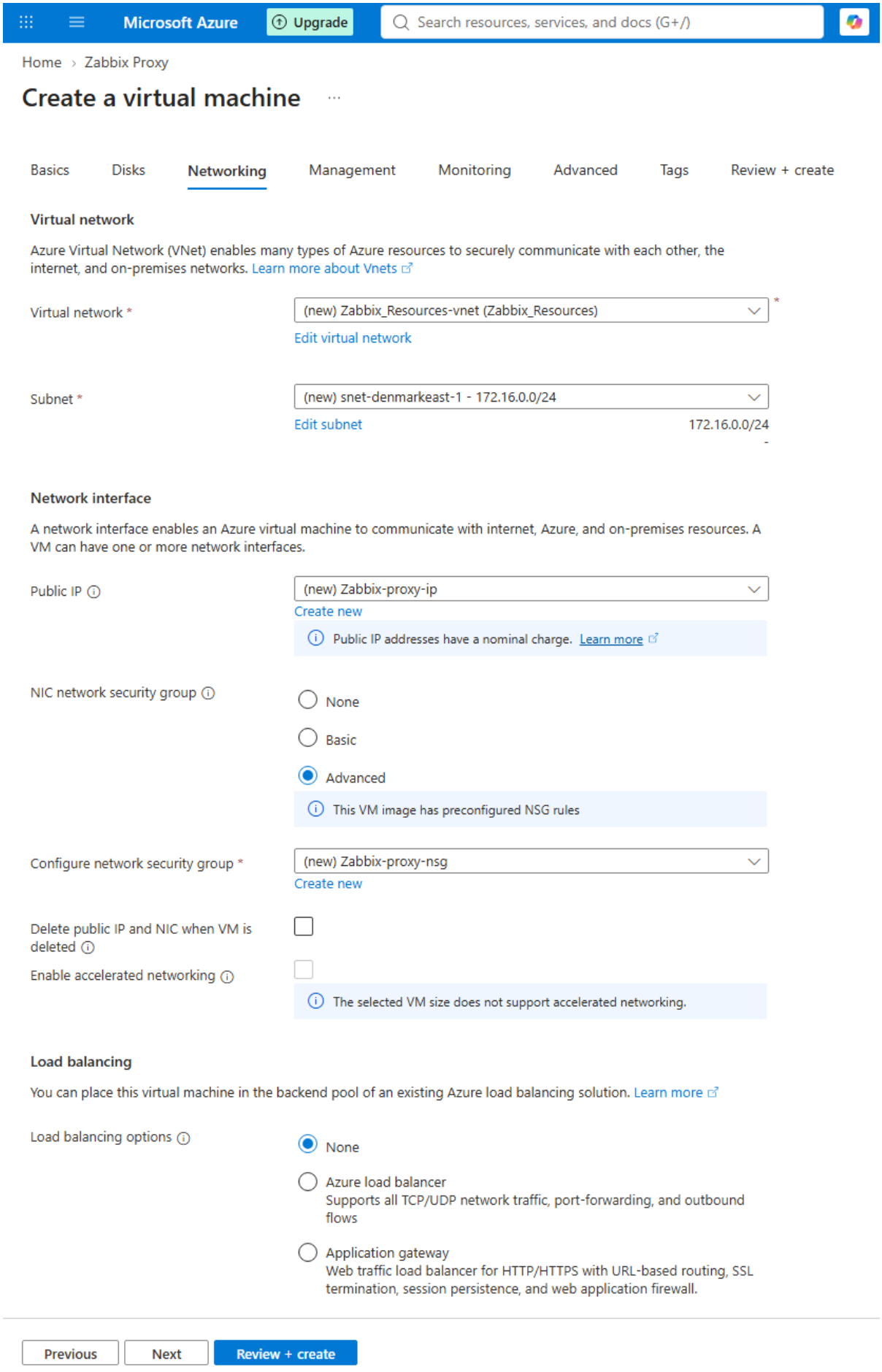
- *Encryption at host*: deselezionato

- *OS disk type*: Standard SSD
- *Delete with VM*: selezionato
- *Enable Ultra Disk compatibility*: deselezionato

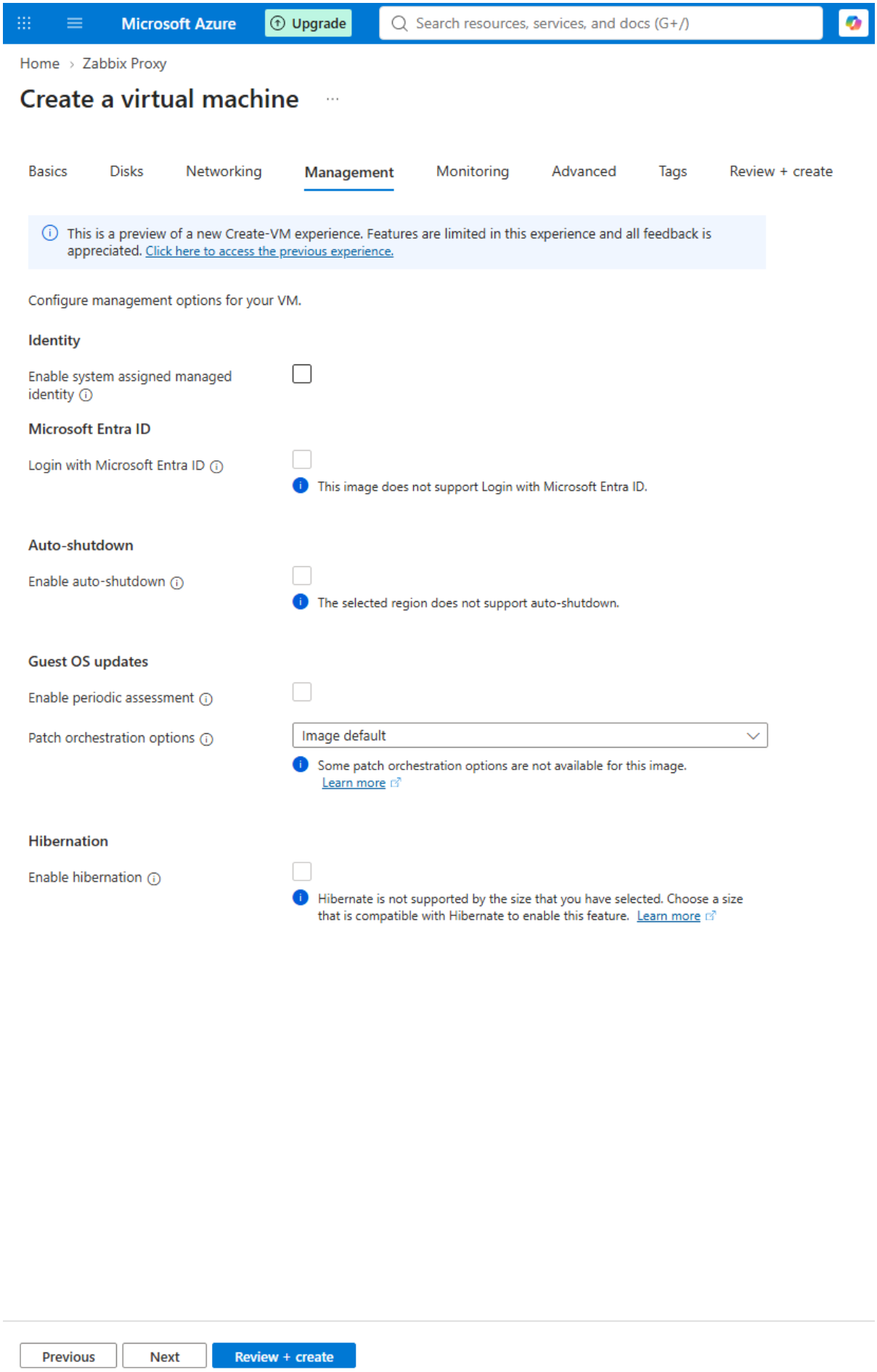


Scheda *Networking*:

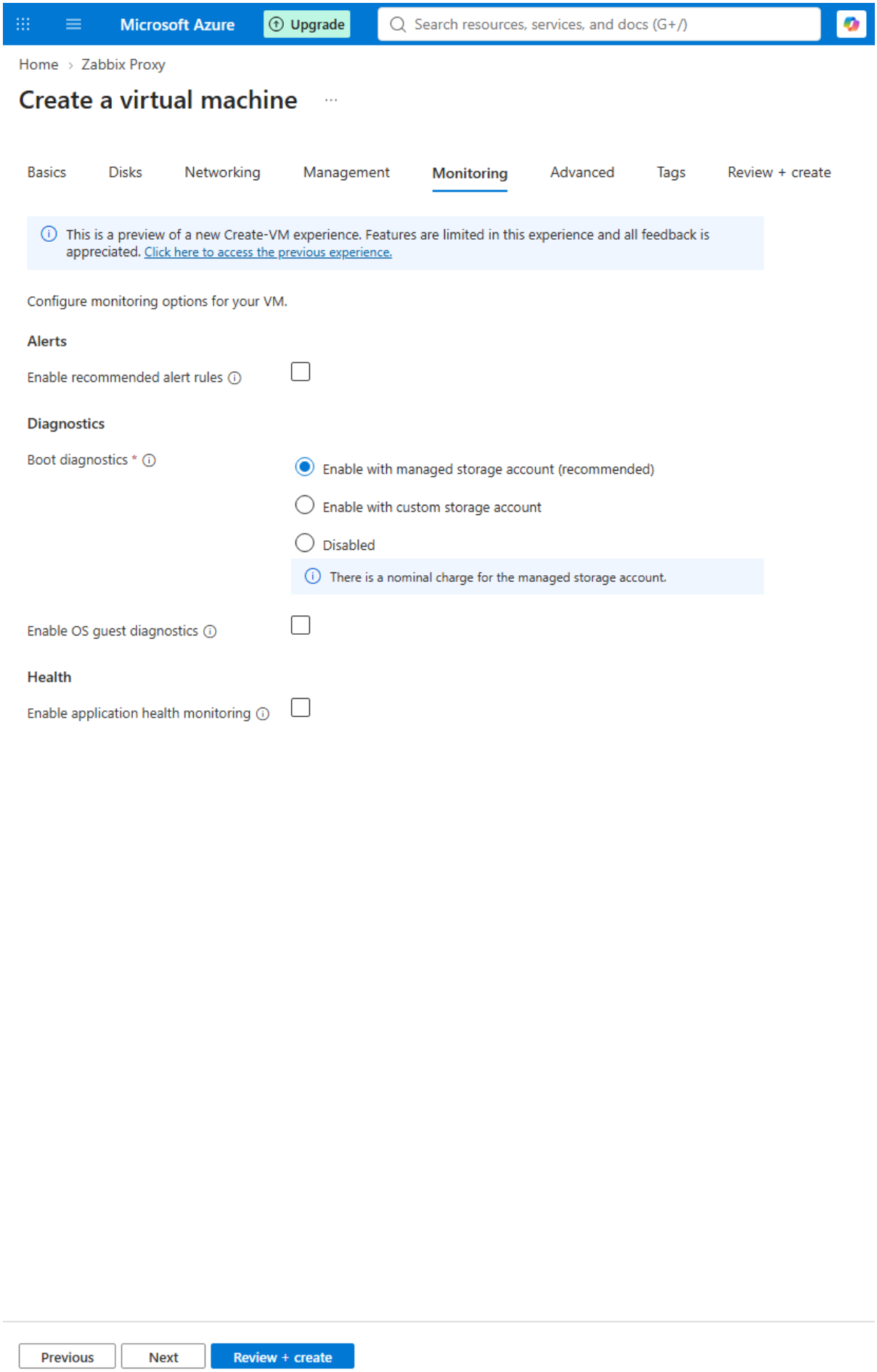
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-proxy-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-proxy-nsg



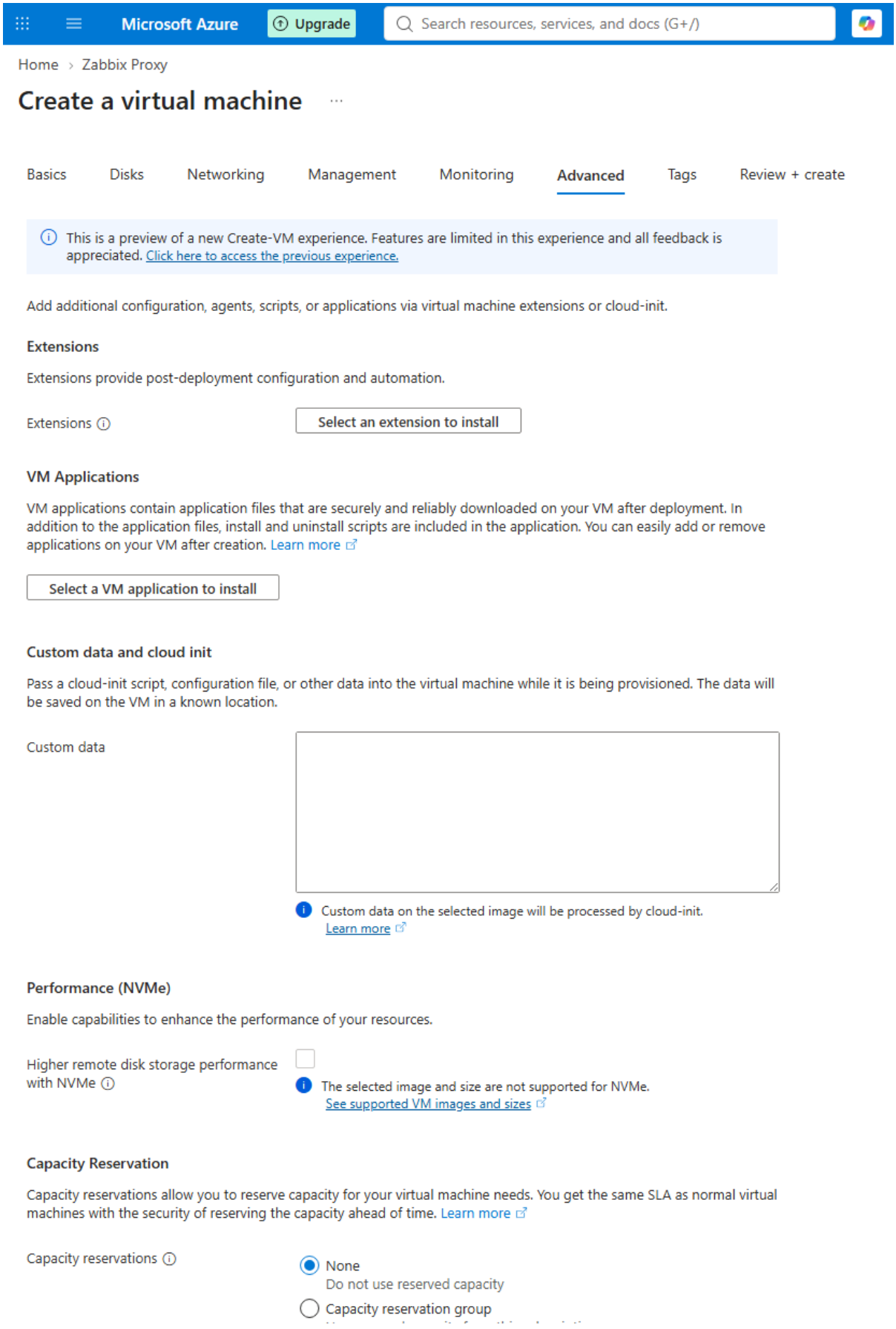
Scheda *Management*: i parametri possono essere lasciati ai valori predefiniti.



Scheda *Monitoring*: i parametri possono essere lasciati ai valori predefiniti.



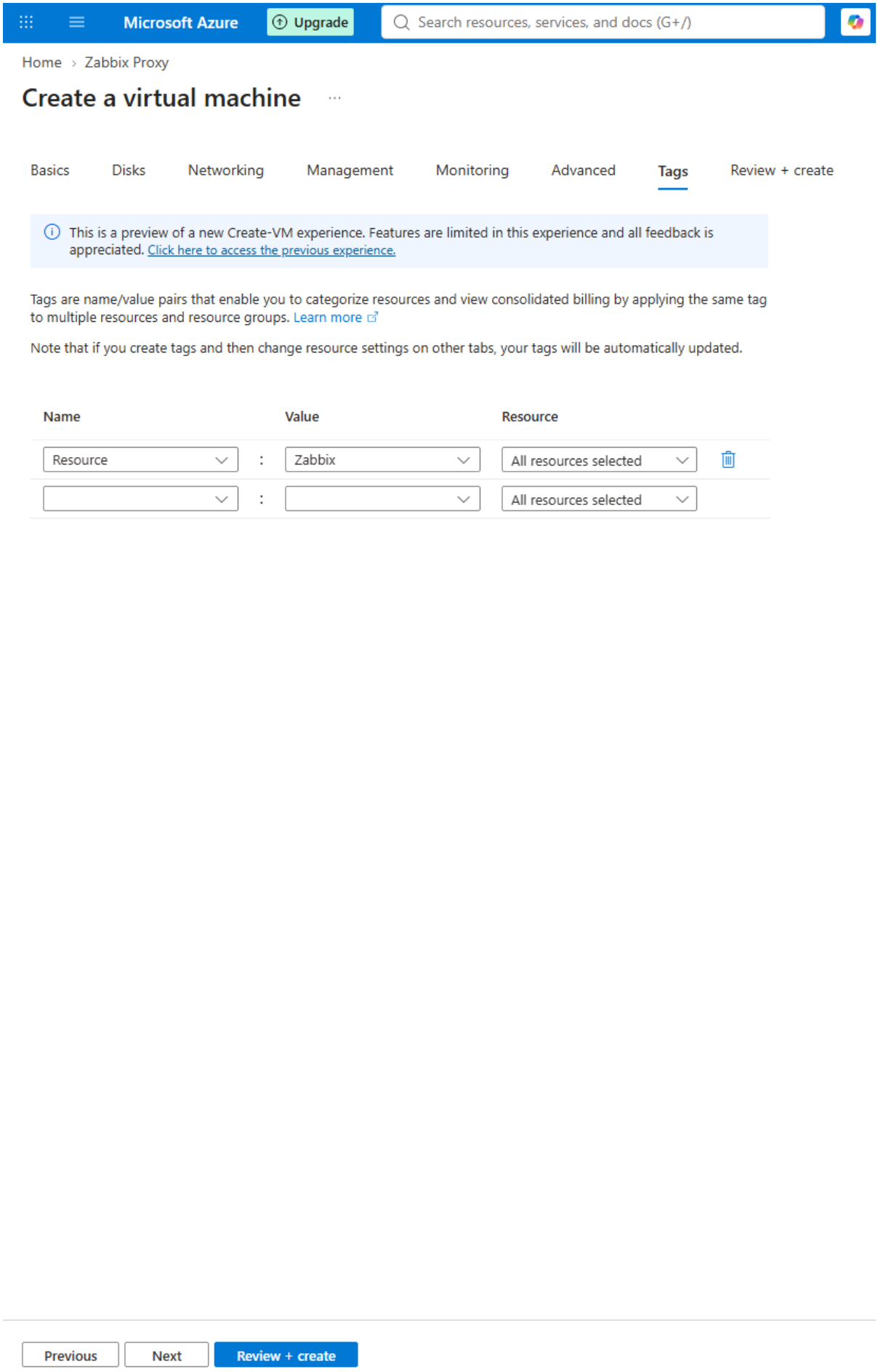
Scheda *Advanced*: i parametri possono essere lasciati ai valori predefiniti.



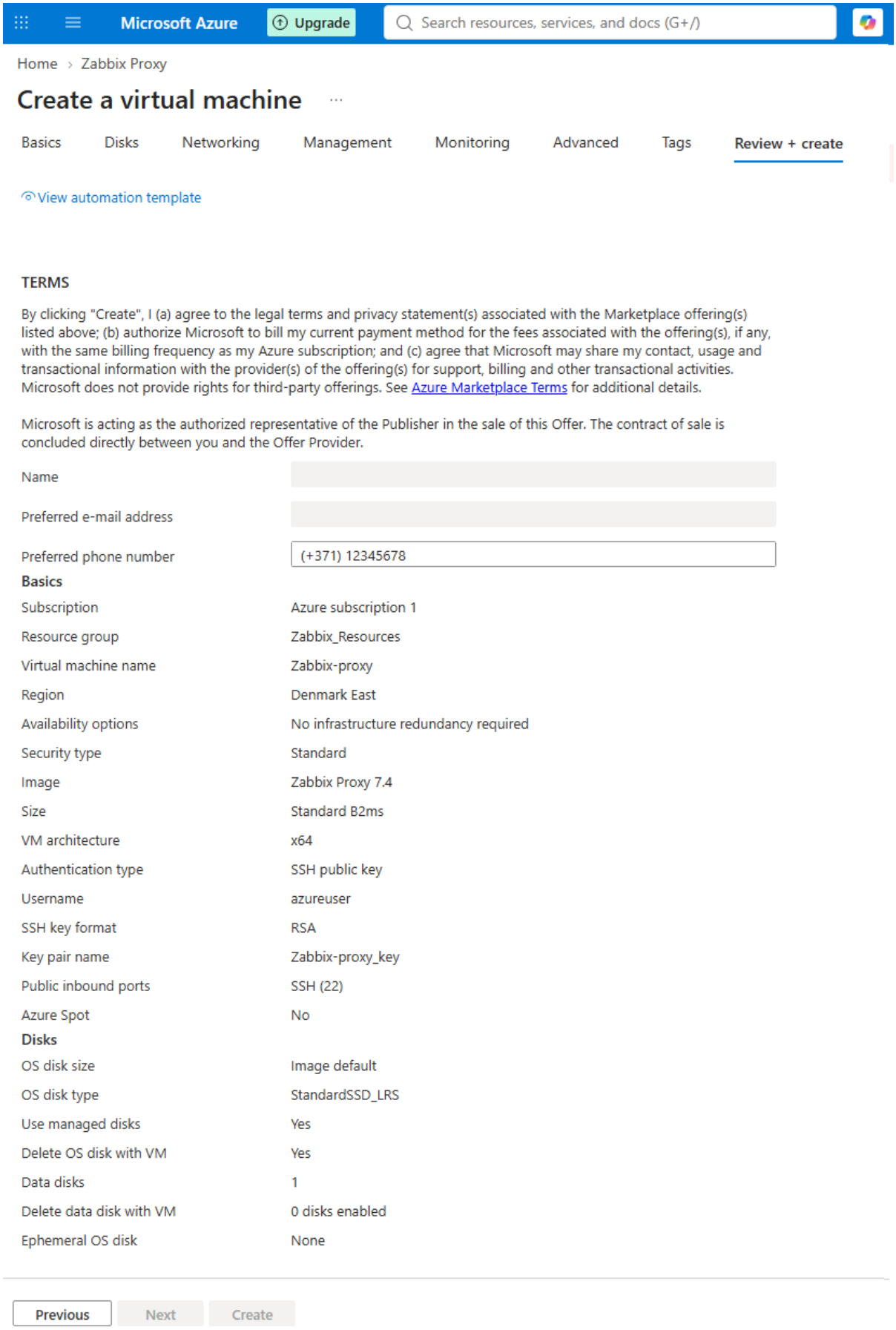
Scheda *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Confermare i dettagli della VM e creare la VM.



Si verrà reindirizzati alla pagina dello stato della distribuzione della VM. Attendere il completamento della distribuzione.

Al termine della distribuzione, aprire la risorsa e annotarne l'indirizzo IP pubblico. Connettersi al proxy Zabbix tramite SSH utilizzando le credenziali dell'amministratore create durante la distribuzione della VM per configurare il proxy.

```
********************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

********************************************************************************
[azureuser@Zabbix-proxy ~]$
```

Il file di configurazione del proxy Zabbix si trova in `/etc/zabbix/zabbix_proxy.conf`.

La configurazione iniziale del proxy consiste nel modificare i parametri `ProxyMode`, `Server` e `Hostname` nel file di configurazione del proxy Zabbix.

La modalità del proxy può essere modificata modificando il parametro `ProxyMode`. Impostare `ProxyMode=0` per la modalità attiva oppure `ProxyMode=1` per la modalità passiva.

Il parametro `Server` deve contenere l'indirizzo IP del server Zabbix.

Il parametro `Hostname` deve contenere il nome host del proxy Zabbix. Il nome host definito qui deve corrispondere al nome del proxy specificato durante la configurazione del proxy nel frontend Zabbix.

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Aprire il frontend Zabbix e accedere a *Administration* > *Proxies*. Fare clic sul pulsante *Create proxy*.

Inserire lo stesso nome del proxy utilizzato come valore del parametro *Hostname* nel file di configurazione del proxy.

Selezionare la modalità del proxy e specificare l'indirizzo IP del proxy (facoltativo per i proxy attivi).

Fare clic sul pulsante *Create proxy* per creare un proxy.

Posizione del database del proxy

Il database SQLite del proxy si trova in `/tmp/zabbix_proxy.sqlite3` ed è specificato dal parametro `DBName` nel file di configurazione del proxy Zabbix.

Guida alla distribuzione su Google Cloud

Panoramica

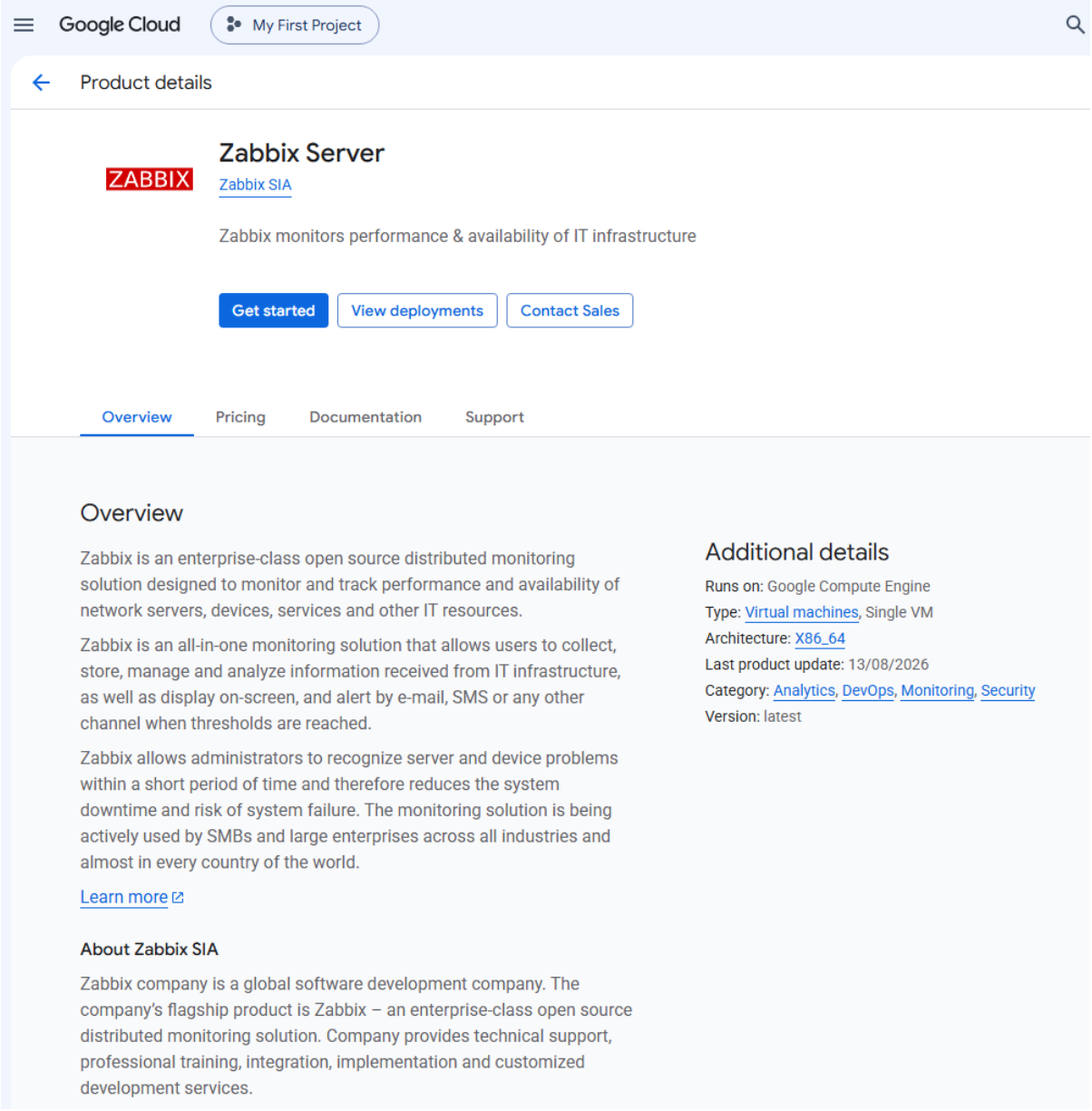
Questa pagina descrive come distribuire Zabbix utilizzando Google Cloud Marketplace.

Attention:
Sono disponibili diverse versioni di Zabbix per la distribuzione sulla piattaforma. Assicurarsi di leggere la documentazione relativa alla stessa versione di Zabbix distribuita. La documentazione per le diverse versioni di Zabbix può essere selezionata dal menu a discesa delle versioni.

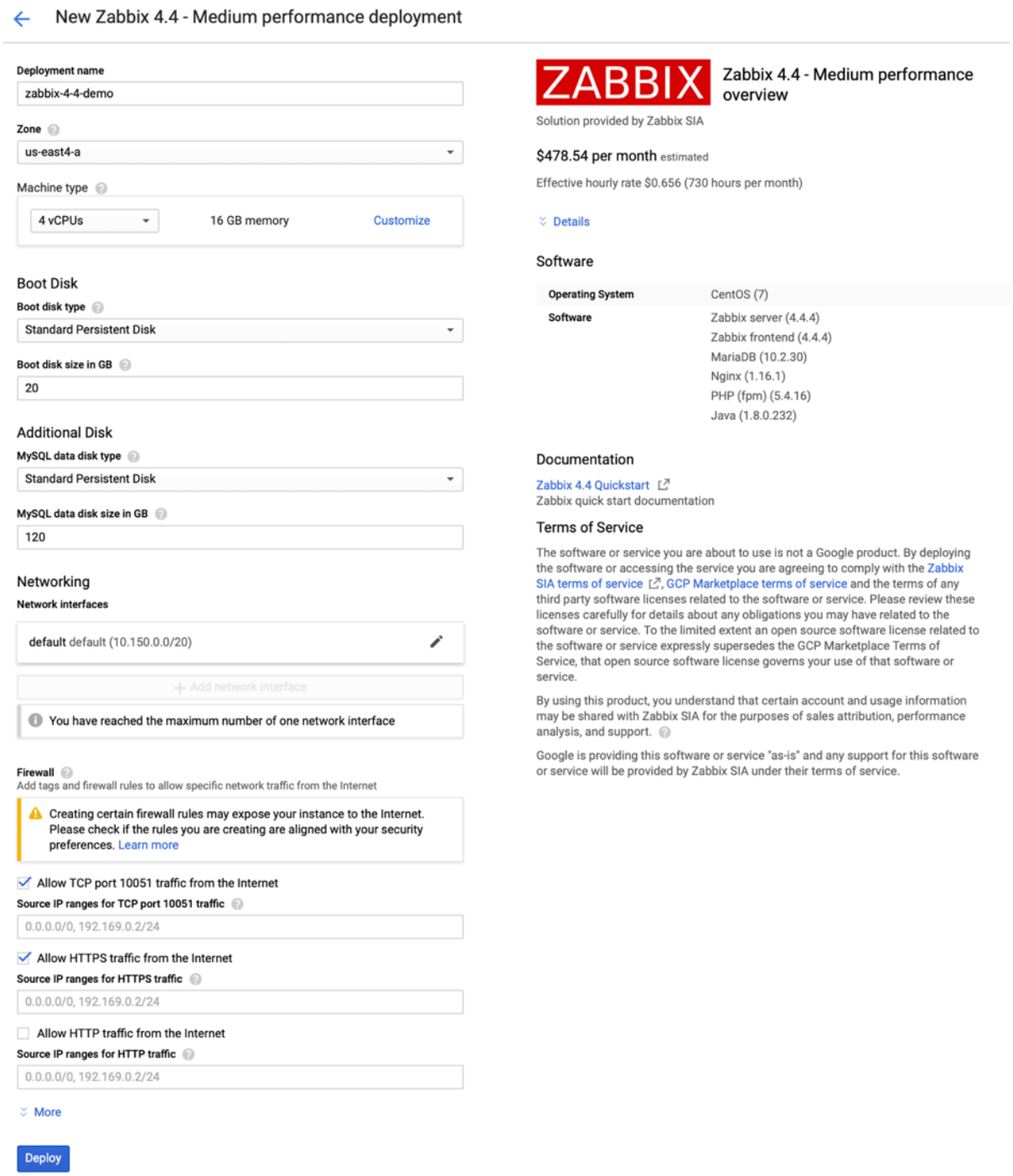
Distribuzione dell'appliance

Per prima cosa, distribuisci l'appliance Zabbix.
È davvero semplice con Google Cloud Marketplace.
Vai semplicemente al prodotto Zabbix Server nel marketplace.

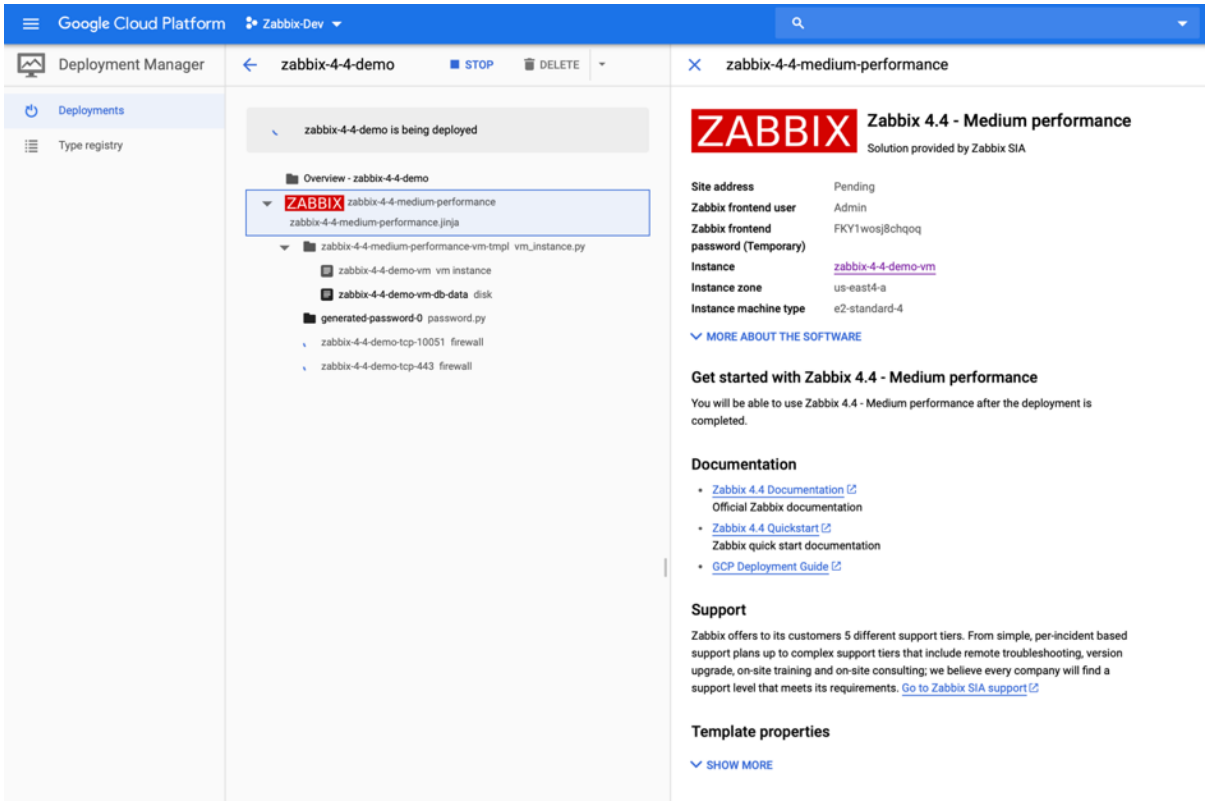
Fai clic su *Get started*.



Puoi mantenere le impostazioni predefinite oppure personalizzarle.
Al termine, fai clic su *Deploy*.



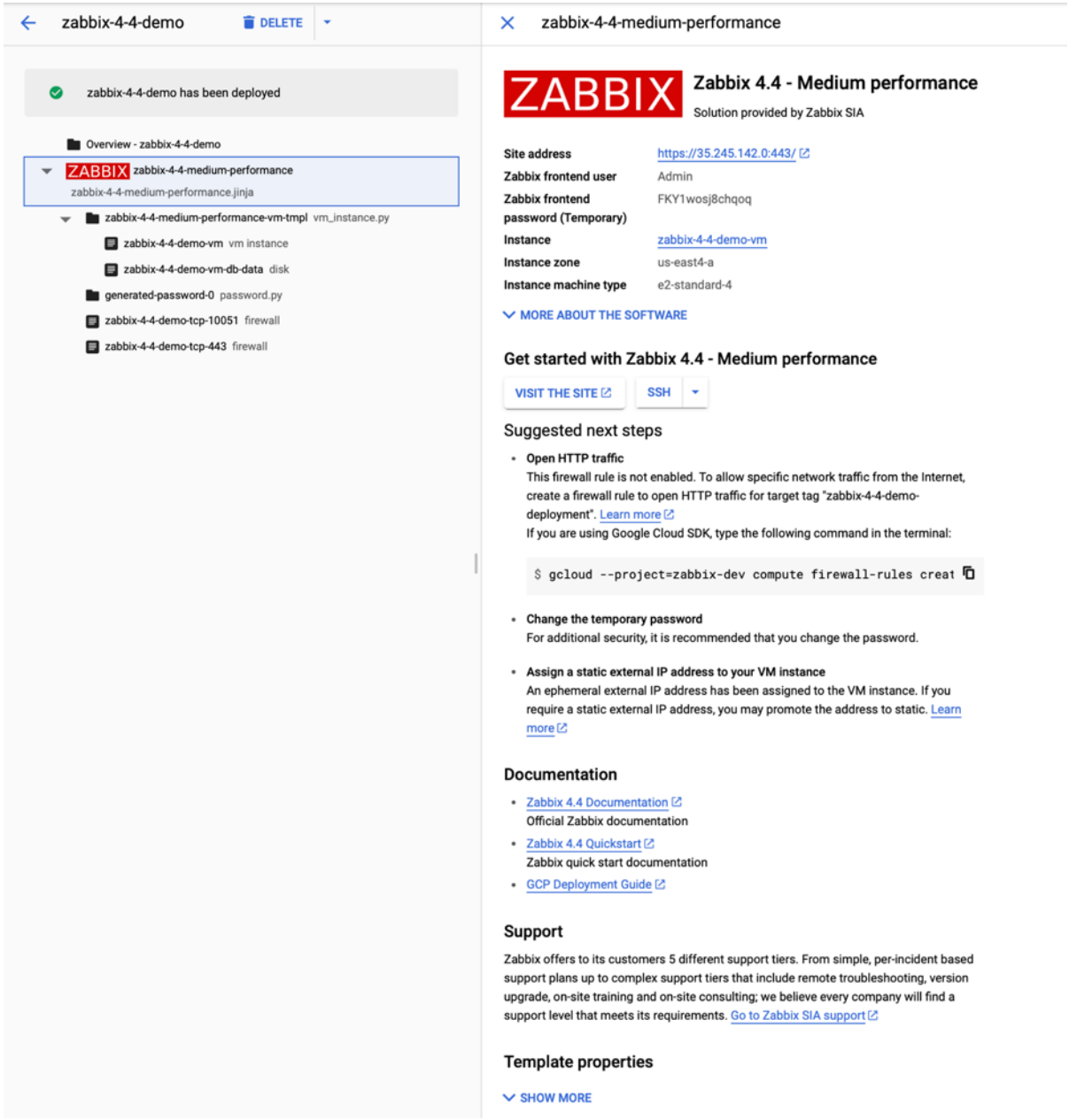
La distribuzione è ora in corso.
Attendi il completamento.



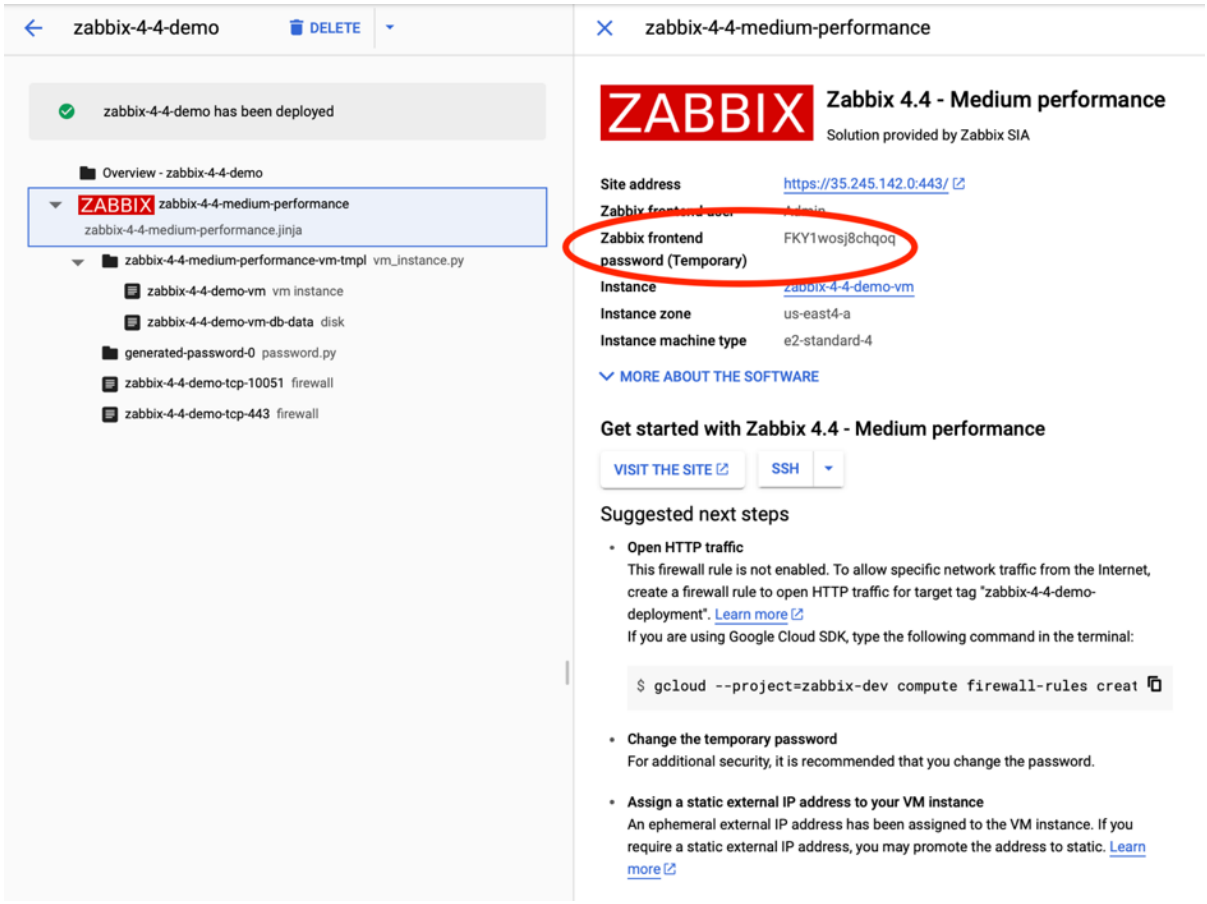
Tutto qui! La distribuzione è stata avviata.

Accesso all’appliance

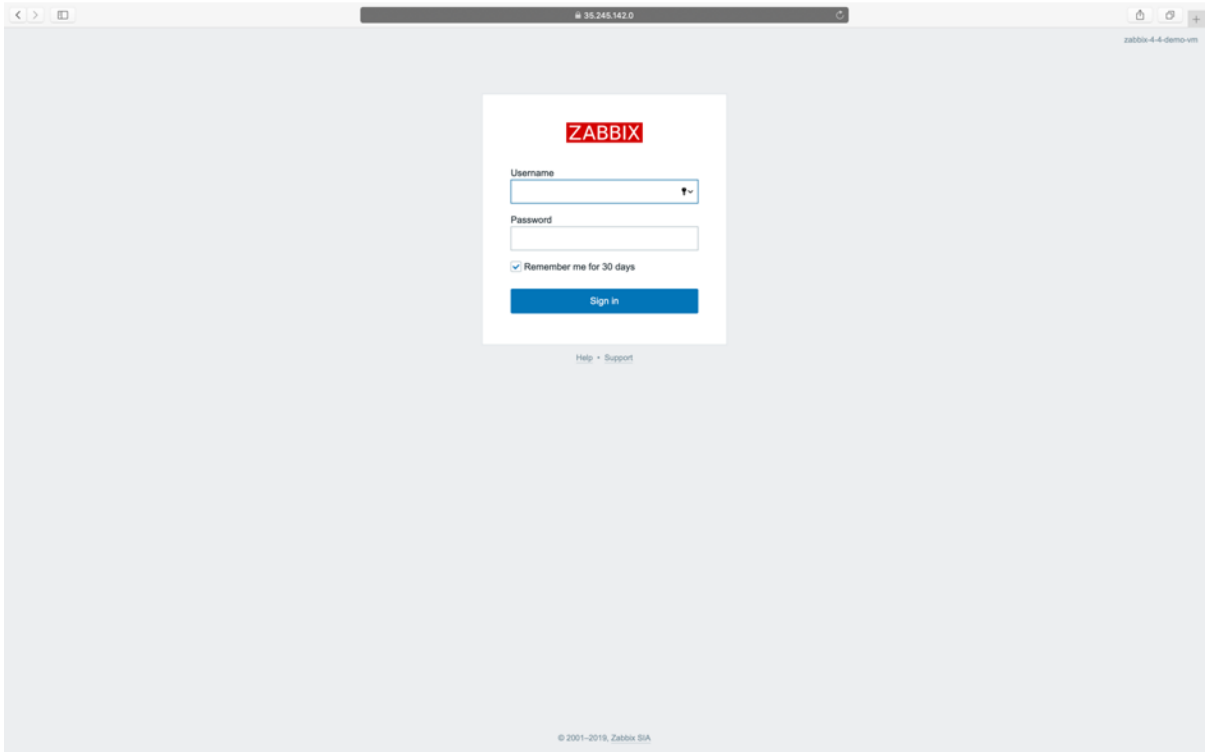
Al termine della distribuzione, dovrebbe essere visualizzato un messaggio che indica che l’appliance è in fase di distribuzione.

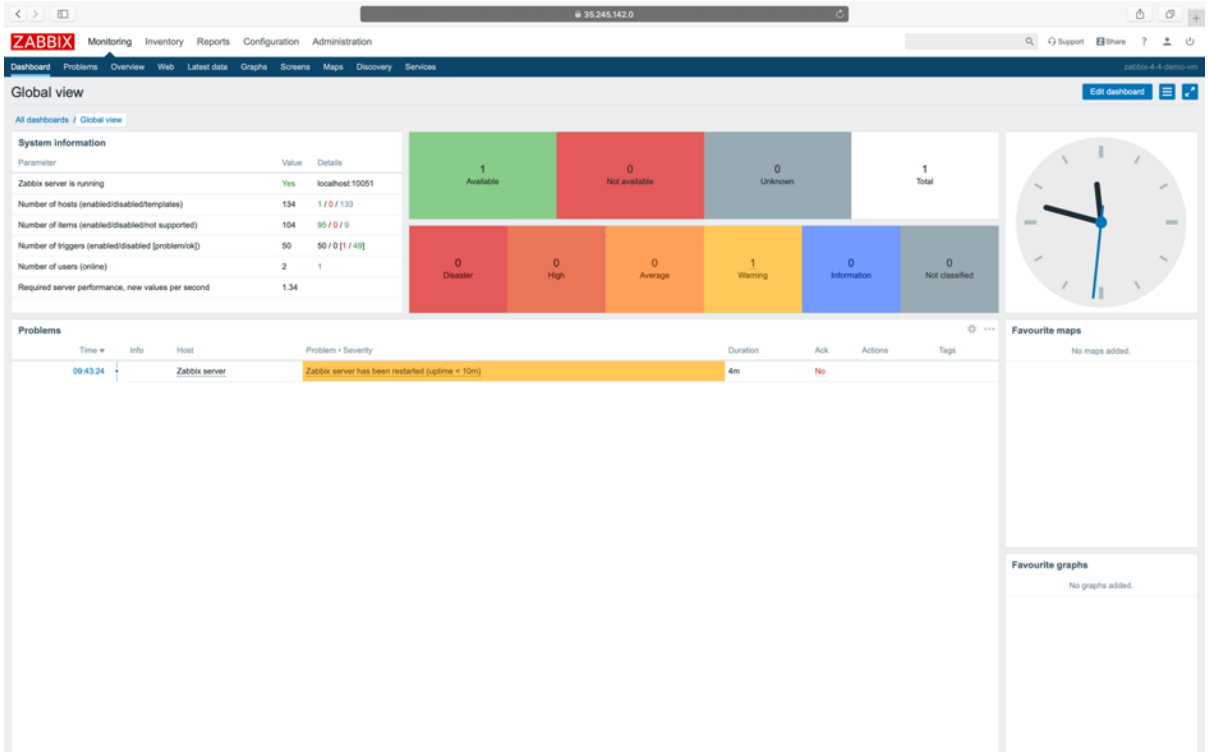


Aprire un browser web all'indirizzo *Site Address* per visualizzare il frontend di Zabbix. Prima di procedere, recuperare la password dell'utente *Admin* di Zabbix per accedere all'appliance Zabbix.

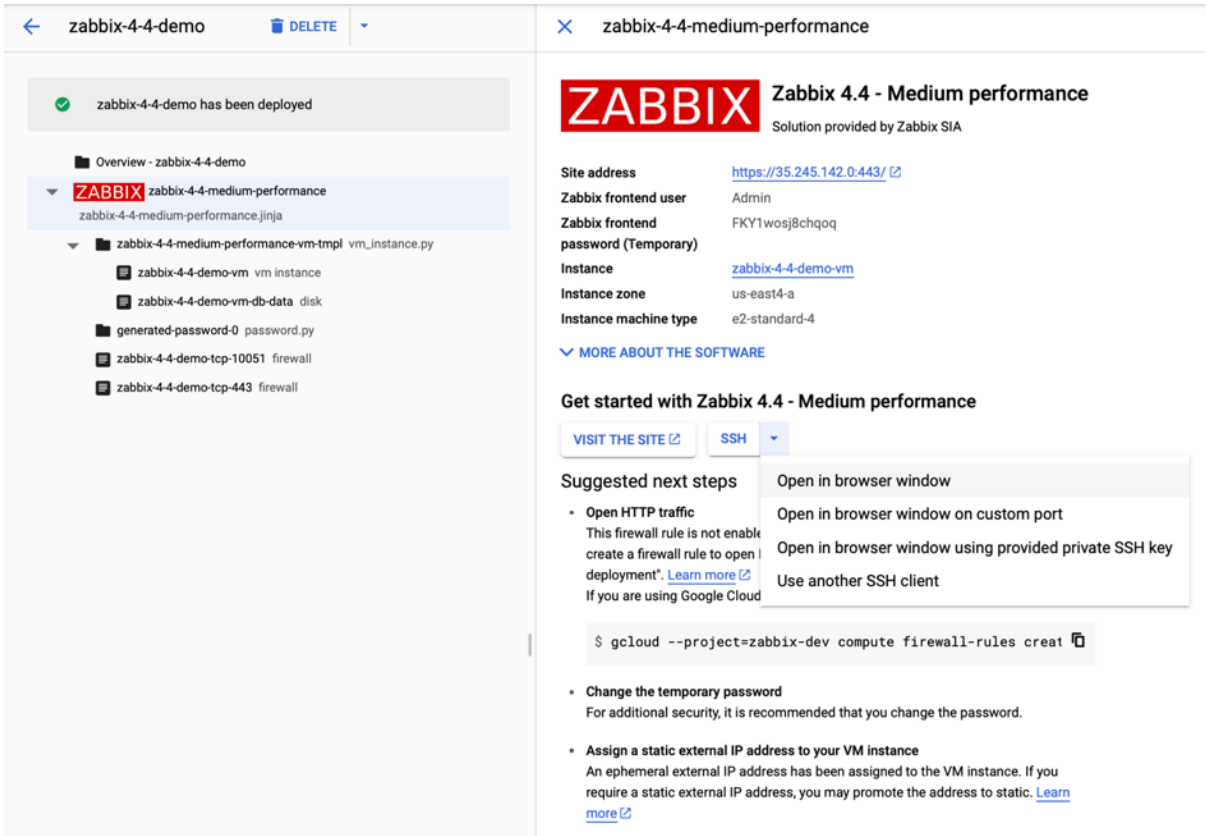


Ottimo! Ora si dispone di un’appliance Zabbix.





È inoltre possibile accedere all'istanza per interagire con il database. A tale scopo, tornare alla console Google e seguire la freccia rossa mostrata di seguito per avviare una sessione SSH utilizzando l'opzione *Open in browser window*.



MySQL è disponibile senza una password `root`. La password è memorizzata nel file di configurazione `/root/.my.cnf`. Per impostazione predefinita, l'appliance Zabbix utilizza il fuso orario UTC. Se è necessario modificarlo, modificare la direttiva di configurazione PHP `date.timezone` nel file `/etc/php.d/99-zabbix.ini` e riavviare il processo php-fpm: `systemctl restart php-fpm`.

Aggiornamento

Prima di eseguire l'aggiornamento, è fortemente consigliato consultare tutte le note di aggiornamento pertinenti.

Sarà inoltre utile sapere quali componenti è necessario aggiornare.

Panoramica

Le istruzioni dettagliate per l'aggiornamento sono disponibili per:

- Red Hat Enterprise Linux (utilizzando i pacchetti)
- Debian/Ubuntu (utilizzando i pacchetti)
- Aggiornamento utilizzando i container
- Aggiornamento utilizzando i codici sorgente

Istruzioni correlate:

- Per i server in un cluster ad alta disponibilità (HA), vedere Aggiornamento del cluster HA
- Per il database TimescaleDB, vedere Aggiornamento dello schema di TimescaleDB

Componenti da aggiornare

L'aggiornamento degli agent Zabbix è consigliato, ma non obbligatorio.

L'aggiornamento dei proxy Zabbix è fortemente consigliato.
Il server Zabbix supporta completamente i proxy della stessa versione principale del server.
Il server Zabbix supporta anche i proxy che **non sono più vecchi** della versione LTS precedente del server Zabbix, ma con funzionalità limitate (raccolta dati, esecuzione di comandi remoti, controlli immediati dei valori degli item).
Anche l'aggiornamento della configurazione è disabilitato e i proxy obsoleti funzioneranno solo con la configurazione precedente.

Attention:
I proxy più vecchi della versione LTS precedente del server Zabbix o più recenti della versione principale del server Zabbix non sono supportati. Il server Zabbix ignorerà i dati provenienti dai proxy non supportati e tutte le comunicazioni con il server Zabbix avranno esito negativo con un avviso. Per ulteriori informazioni, vedere Compatibilità delle versioni.

Per ridurre al minimo i tempi di inattività e la perdita di dati durante l'aggiornamento, si consiglia di arrestare, aggiornare e avviare il server Zabbix, quindi arrestare, aggiornare e avviare i proxy Zabbix uno alla volta. Durante il periodo di inattività del server, i proxy in esecuzione continueranno la raccolta dei dati. Quando il server sarà nuovamente operativo, i proxy obsoleti invieranno i dati al server più recente (la configurazione del proxy non verrà tuttavia aggiornata) e rimarranno parzialmente funzionanti. Eventuali notifiche relative ai problemi verificatisi durante il periodo di inattività del server Zabbix verranno generate solo dopo l'avvio del server aggiornato.

Se Zabbix proxy viene avviato per la prima volta e il file del database SQLite non è presente, il proxy lo crea automaticamente.

Nota: se Zabbix proxy utilizza SQLite3 e, all'avvio, rileva che la versione del file del database esistente è precedente a quella richiesta, **eliminerà automaticamente il file del database** e ne creerà uno nuovo. Pertanto, i dati della cronologia archiviati nel file del database SQLite andranno persi. Se la versione di Zabbix proxy è precedente a quella del file del database, Zabbix registrerà un errore e terminerà.

A seconda delle dimensioni del database, l'aggiornamento del database alla versione 8.0 potrebbe richiedere molto tempo.

Note sull'aggiornamento

L'aggiornamento diretto a Zabbix 8.0.x è supportato dalla versione **2.0**.x. Per eseguire l'aggiornamento da versioni precedenti, consultare la documentazione di Zabbix per la versione 2.0 e precedenti.

Note:
Tenere presente che, dopo l'aggiornamento, alcune integrazioni software di terze parti in Zabbix potrebbero essere interessate se il software esterno non è compatibile con la versione aggiornata di Zabbix.

Sono disponibili le seguenti note sull'aggiornamento:

Aggiornamento da	Leggi le note complete sull'aggiornamento	Modifiche più importanti tra le versioni
7.4.x	Per: Zabbix 8.0	Gli agent Zabbix delle versioni precedenti alla 1.6 non sono più supportati. Sono state aumentate le versioni minime richieste dei database. La versione minima richiesta di PHP è stata aumentata da 8.0.0 a 8.2.0. Il carattere % è stato aggiunto all'elenco `UnsafeUserParameters` per Zabbix agent e Zabbix agent 2.

Aggiornamento da	Leggi le note complete sull'aggiornamento	Modifiche più importanti tra le versioni
7.2.x	Per: Zabbix 7.4 Zabbix 8.0	Il supporto della libreria PCRE è stato rimosso.
7.0.x	Per: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Il supporto di Oracle DB è stato rimosso.
6.4.x	Per: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La versione minima richiesta di PHP è stata aumentata da 7.4.0 a 8.0.0. Poller asincroni per i controlli di agent, HTTP agent, SNMP walk[OID] e SNMP get[OID]. Tabella di database separata per i proxy. La posizione predefinita del file di configurazione dell'agent Windows è cambiata. Oracle DB è stato deprecato. Il vecchio tipo di valore numerico (float) è stato rimosso.
6.2.x	Per: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La versione minima richiesta di MySQL è stata aumentata da 8.0.0 a 8.0.30. La libreria 'libevent_pthreads' è richiesta per Zabbix server/proxy. Al primo avvio dopo un aggiornamento, il proxy Zabbix con SQLite3 elimina automaticamente la versione precedente del database (con tutta la cronologia) e ne crea una nuova.
6.0.x LTS	Per: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La versione minima richiesta di PHP è stata aumentata da 7.2.5 a 7.4.0. Il monitoraggio dei servizi è stato rielaborato in modo significativo. Durante l'aggiornamento è necessario creare trigger deterministici. Se il binary logging è abilitato per MySQL/MariaDB, sono necessari i privilegi di superuser oppure l'impostazione della variabile/parametro di configurazione *log_bin_trust_function_creators = 1*. Per istruzioni su come impostare la variabile, vedere Script per la creazione del database.
5.4.x	Per: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Sono state aumentate le versioni minime richieste dei database. server/proxy non si avvierà se il database è obsoleto. I record del registro di audit sono andati persi a causa di una modifica della struttura del database.
5.2.x	Per: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Sono state aumentate le versioni minime richieste dei database. Gli item aggregati sono stati rimossi come tipo separato.
5.0.x LTS	Per: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La versione minima richiesta di PHP è stata aumentata da 7.2.0 a 7.2.5. L'algoritmo di hashing delle password è cambiato da MD5 a bcrypt.

Aggiornamento da	Leggi le note complete sull'aggiornamento	Modifiche più importanti tra le versioni
4.4.x	Per: 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	Il supporto di IBM DB2 è stato rimosso. La versione minima richiesta di PHP è stata aumentata da 5.4.0 a 7.2.0. Sono state aumentate le versioni minime richieste dei database. La directory dei file PHP di Zabbix è cambiata.
4.2.x	Per: 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	I tipi di supporto Jabber ed Ez Texting sono stati rimossi.
4.0.x LTS	Per: 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	I proxy meno recenti non possono più inviare dati a un server aggiornato. Gli agent più recenti non potranno più funzionare con un server Zabbix meno recente.
3.4.x	Per: 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	Le librerie 'libpthread' e 'zlib' sono ora obbligatorie. Il supporto del protocollo in testo semplice è stato rimosso e l'header è obbligatorio. Gli agent Zabbix delle versioni precedenti alla 1.4 non sono più supportati. Il parametro Server nella configurazione del proxy passivo è ora obbligatorio.

Aggiornamento da	Leggi le note complete sull'aggiornamento	Modifiche più importanti tra le versioni
3.2.x	Per: 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	Il supporto di SQLite come database backend è stato rimosso per Zabbix server/frontend. Le espressioni regolari compatibili con Perl (PCRE) sono supportate al posto delle espressioni estese POSIX. Le librerie 'libpcre' e 'libevent' sono obbligatorie per Zabbix server. Sono stati aggiunti controlli del codice di uscita per i parametri utente, i comandi remoti e gli item system.run[] senza il flag 'nowait', nonché per gli script eseguiti da Zabbix server. È necessario aggiornare Zabbix Java gateway per supportare le nuove funzionalità.
3.0.x LTS	Per: 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	L'aggiornamento del database può essere lento, a seconda delle dimensioni della tabella history.
2.4.x	Per: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La versione minima richiesta di PHP è stata aumentata da 5.3.0 a 5.4.0. Il parametro agent LogFile deve essere specificato.

Aggiornamento da	Leggi le note complete sull'aggiornamento	Modifiche più importanti tra le versioni
2.2.x LTS	Per: 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	Il monitoraggio distribuito basato sui nodi è stato rimosso.
2.0.x	Per: Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La versione minima richiesta di PHP è stata aumentata da 5.1.6 a 5.3.0. Per il corretto funzionamento del server è richiesto un database MySQL con distinzione tra maiuscole e minuscole; per il corretto funzionamento di Zabbix server con il database MySQL sono richiesti il set di caratteri utf8 e la collation utf8_bin. Vedere script per la creazione del database. È richiesta l'estensione PHP 'mysqli' invece di 'mysql'.

Aggiornamento dai pacchetti

Panoramica

Questa sezione fornisce i passaggi necessari per un aggiornamento riuscito utilizzando i pacchetti RPM e DEB ufficiali forniti da Zabbix per:

- Red Hat Enterprise Linux
- Debian/Ubuntu

Pacchetti Zabbix dai repository del sistema operativo

Alcune distribuzioni del sistema operativo (in particolare quelle basate su Debian) forniscono i propri pacchetti Zabbix. Questi pacchetti **non sono supportati da Zabbix** e potrebbero essere obsoleti o non includere le funzionalità più recenti e le correzioni di bug. Si consiglia di utilizzare solo i pacchetti ufficiali dal repository ufficiale di Zabbix.

Se si esegue l'aggiornamento da pacchetti forniti dalle distribuzioni del sistema operativo (o se sono stati installati in passato), seguire questa procedura per passare ai pacchetti ufficiali di Zabbix:

1. Disinstallare i vecchi pacchetti.
2. Verificare la presenza di eventuali file residui della vecchia installazione e rimuoverli.
3. Installare i pacchetti ufficiali di Zabbix seguendo le istruzioni di installazione fornite da Zabbix.

Non eseguire un aggiornamento diretto, poiché ciò potrebbe causare un'installazione non funzionante.

Debian/Ubuntu

Panoramica

Questa sezione fornisce istruzioni su come eseguire l'aggiornamento da Zabbix **7.4.x** all'ultima versione di Zabbix **8.0.x** utilizzando i pacchetti ufficiali di Zabbix per Debian/Ubuntu.

Warning:
Prima di eseguire l'aggiornamento, consulta le note di aggiornamento pertinenti e assicurati che il tuo sistema soddisfi i requisiti per Zabbix 8.0.

Note:
Valuta l'esecuzione di due sessioni SSH parallele durante l'aggiornamento: una per eseguire i passaggi dell'aggiornamento e un'altra per monitorare i log di server/proxy. Ad esempio, esegui `tail -f zabbix_server.log` oppure `tail -f zabbix_proxy.log` nella seconda sessione per visualizzare in tempo reale le voci di log più recenti e i possibili errori. Questo può essere fondamentale per le istanze di produzione.

Per istruzioni su come eseguire l'aggiornamento tra versioni secondarie di Zabbix 8.0.x (ad esempio, dalla 8.0.1 alla 8.0.3), consulta Aggiornamento tra versioni secondarie.

Procedura di aggiornamento

1 Arrestare i processi di Zabbix

Arrestare il server Zabbix per assicurarsi che nessun nuovo dato venga inserito nel database:

```
systemctl stop zabbix-server
```

Se si sta aggiornando Zabbix proxy, agent o agent 2, arrestare anche questi componenti:

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Eseguire il backup del database Zabbix

Eseguire il backup del database Zabbix esistente per proteggersi da eventuali errori durante l'aggiornamento (ad esempio, problemi di spazio su disco, interruzioni di corrente o problemi imprevisti).

3 Eseguire il backup dei file di configurazione di Zabbix, dei file PHP e dei binari di Zabbix

Eseguire il backup dei file di configurazione di Zabbix, dei file PHP e dei binari di Zabbix esistenti.

Per i file di configurazione, eseguire:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
##### Per Apache:
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
##### Per Nginx:
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.conf /opt/zabbix-backup/
```

Per i file PHP e i binari di Zabbix, eseguire:

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Aggiornare il pacchetto di configurazione del repository

Prima di procedere con l'aggiornamento, disinstallare il pacchetto del repository Zabbix attualmente in uso:

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

Potrebbe essere necessario rimuovere manualmente eventuali vecchi pacchetti Zabbix dalla directory di lavoro (ad esempio, `rm zabbix-release_latest+debian12_all.deb`) prima di scaricare quello nuovo, per evitare che il gestore dei pacchetti riutilizzi una versione obsoleta durante il processo di aggiornamento.

Quindi, installare il pacchetto di configurazione del repository più recente per garantire la compatibilità con i pacchetti più nuovi e includere eventuali patch di sicurezza o correzioni di bug recenti.

Su **Debian 13**, eseguire:

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian13_all.deb
```

Su **Debian 12**, eseguire:

```
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:
Per le versioni Debian più datate, sostituire il link sopra con quello corretto dal repository Zabbix. Si noti, tuttavia, che i pacchetti per queste versioni potrebbero non includere tutti i componenti Zabbix e, per aggiornare tali componenti tramite pacchetti, è consigliabile aggiornare il sistema operativo. Per un elenco dei componenti inclusi, vedere Pacchetti Zabbix.

Su **Ubuntu 26.04**, eseguire:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu26.04_all.deb
```

Su **Ubuntu 24.04**, eseguire:

```
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
```

Su **Ubuntu 22.04**, eseguire:

```
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:
Per le versioni Ubuntu più datate, sostituire il link sopra con quello corretto dal repository Zabbix. Si noti, tuttavia, che i pacchetti per queste versioni potrebbero non includere tutti i componenti Zabbix e, per aggiornare tali componenti tramite pacchetti, è consigliabile aggiornare il sistema operativo. Per un elenco dei componenti inclusi, vedere Pacchetti Zabbix.

Potrebbe essere visualizzato un prompt relativo alla configurazione del repository Zabbix:

```
Configuration file '/etc/apt/sources.list.d/zabbix.list'
==> Deleted (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix.list (Y/I/N/O/D/Z) [default=N] ?
```

Inserire `Y` (o `I`) per installare la versione del manutentore del pacchetto della configurazione del repository Zabbix.

Quindi, aggiornare le informazioni del repository:

```
apt update
```

5 Aggiornamento dei componenti Zabbix

Per aggiornare i componenti Zabbix, eseguire:

```
apt install --only-upgrade zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

- Se si utilizza PostgreSQL, sostituire `mysql` con `pgsql` nel comando.
- Se si aggiorna il proxy, sostituire `server` con `proxy` nel comando.
- Se si aggiorna Zabbix agent 2, sostituire `zabbix-agent` con `zabbix-agent2 zabbix-agent2-plugin-*` nel comando.

Attention:
L'aggiornamento di Zabbix agent 2 con il comando `apt install zabbix-agent2` potrebbe causare un errore. Per ulteriori informazioni, vedere *Problemi noti*.

Potrebbe essere visualizzata una richiesta relativa alla configurazione del server Zabbix (o del proxy):

```
Configuration file '/etc/zabbix/zabbix_server.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix_server.conf (Y/I/N/O/D/Z) [default=N] ?
```

Immettere l'opzione più adatta alla propria situazione. Ad esempio, immettere `D` per confrontare la configurazione attuale con quella nuova, quindi decidere se installare la versione del responsabile del pacchetto (`Y` o `I`).

Quindi, aggiornare il frontend Zabbix e riavviare il web server.

Per Apache, eseguire:

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

Per Nginx, eseguire:

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Per Nginx, potrebbe essere visualizzata una richiesta relativa al file di configurazione `nginx.conf`. Immettere l'opzione più adatta alla propria situazione:

```
Configuration file '/etc/zabbix/nginx.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** nginx.conf (Y/I/N/O/D/Z) [default=N] ?
```

Se, dopo l'aggiornamento, è necessario installare un pacchetto Zabbix aggiuntivo dallo stesso repository, `apt` potrebbe selezionare gli aggiornamenti per i pacchetti Zabbix già installati durante la risoluzione delle dipendenze. Per evitare di aggiornare un pacchetto server o proxy Zabbix installato, bloccare temporaneamente il pacchetto prima dell'installazione e sbloccarlo successivamente:

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

6 Rivedere i parametri di configurazione del componente

Rivedere le pertinenti note di aggiornamento per verificare se sono necessarie modifiche ai parametri di configurazione.

Per i nuovi parametri opzionali, vedere la pagina Novità.

Se si utilizza Nginx come web server e il file di configurazione `/etc/zabbix/nginx.conf` è stato reinstallato durante l'aggiornamento, rivedere il file e aggiornare le direttive `listen` e `server_name` in modo che corrispondano alla configurazione del frontend di Zabbix.

7 Avviare i processi di Zabbix

Avviare i componenti Zabbix aggiornati:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Cancellare i cookie e la cache del browser web

Dopo l'aggiornamento, potrebbe essere necessario cancellare i cookie e la cache del browser web affinché l'interfaccia web di Zabbix funzioni correttamente.

Aggiornamento tra versioni minori

È possibile eseguire l'aggiornamento tra versioni minori 8.0.x di Zabbix (ad esempio, dalla 8.0.1 alla 8.0.3).

Per prima cosa, aggiornare le informazioni del repository:

```
apt update
```

Quindi, per aggiornare tutti i componenti di Zabbix, eseguire:

```
apt install --only-upgrade 'zabbix*'
```

- Per aggiornare solo Zabbix server, sostituire `'zabbix*'` con `'zabbix-server*'` nel comando.
- Per aggiornare solo Zabbix proxy, sostituire `'zabbix*'` con `'zabbix-proxy*'` nel comando.
- Per aggiornare solo Zabbix agent, sostituire `'zabbix*'` con `'zabbix-agent*'` nel comando.
- Per aggiornare solo Zabbix agent 2, sostituire `'zabbix*'` con `'zabbix-agent2*'` nel comando.

Red Hat Enterprise Linux

Panoramica

Questa sezione fornisce istruzioni su come eseguire l'aggiornamento da Zabbix **7.4.x** all'ultima versione di Zabbix **8.0.x** utilizzando i pacchetti ufficiali di Zabbix per Red Hat Enterprise Linux o le sue derivate - AlmaLinux, CentOS Stream, Oracle Linux e Rocky Linux.

Warning:
Prima di eseguire l'aggiornamento, consulta le relative note di aggiornamento e assicurati che il tuo sistema soddisfi i requisiti per Zabbix 8.0.

Note:
Valuta l'esecuzione di due sessioni SSH parallele durante l'aggiornamento: una per eseguire i passaggi dell'aggiornamento e un'altra per monitorare i log del server/proxy. Ad esempio, esegui `tail -f zabbix_server.log` oppure `tail -f zabbix_proxy.log` nella seconda sessione per visualizzare in tempo reale le voci di log più recenti e i possibili errori. Questo può essere fondamentale per le istanze di produzione.

Per istruzioni sull'aggiornamento tra versioni secondarie di Zabbix 8.0.x (ad esempio, dalla 8.0.1 alla 8.0.3), consulta Aggiornamento tra versioni secondarie.

Procedura di aggiornamento

1 Arrestare i processi di Zabbix

Arrestare il server Zabbix per assicurarsi che nessun nuovo dato venga inserito nel database:

```
systemctl stop zabbix-server
```

Se si sta aggiornando Zabbix proxy, agent o agent 2, arrestare anche questi componenti:

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Esegui il backup del database Zabbix

Esegui il backup del database Zabbix esistente per proteggerti da errori di aggiornamento (ad esempio, problemi di spazio su disco, interruzioni di corrente o problemi imprevisti).

3 Eseguire il backup dei file di configurazione di Zabbix, dei file PHP e dei binari di Zabbix

Eseguire il backup dei file di configurazione di Zabbix, dei file PHP e dei binari di Zabbix esistenti.

Per i file di configurazione, eseguire:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/httpd/conf.d/zabbix.conf  /opt/zabbix-backup/
```

Per i file PHP e i binari di Zabbix, eseguire:

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Aggiornamento del pacchetto di configurazione del repository

Prima di procedere con l'aggiornamento, aggiorna il pacchetto del repository attualmente in uso alla versione più recente per garantire la compatibilità con i pacchetti più recenti e includere eventuali patch di sicurezza o correzioni di bug recenti.

Su **RHEL 10**, esegui:

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

Se stai aggiornando il proxy Zabbix su RHEL 10 e vuoi utilizzare APM, installa anche il repository EPEL e abilita il repository CRB, che forniscono entrambi le dipendenze per il pacchetto APM del proxy Zabbix:

```
dnf -y install epel-release || dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(
crb enable
```

Su **RHEL 9**, esegui:

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

> **Note:**
> Per le versioni precedenti di RHEL o per le relative derivate, sostituisci il link precedente con quello corretto disponibile nel repository Zabbix. Tieni tuttavia presente che i pacchetti per queste versioni potrebbero non includere tutti i componenti Zabbix; per aggiornare tali componenti dai pacchetti, valuta la possibilità di aggiornare il sistema operativo. Per un elenco dei componenti inclusi, consulta Pacchetti Zabbix.

Quindi, svuota la cache del gestore pacchetti `dnf` (inclusi gli header, i metadati e i file dei pacchetti scaricati durante le installazioni o gli aggiornamenti precedenti):

```
dnf clean all
```

Alla successiva operazione `dnf`, `dnf` scaricherà metadati aggiornati dai repository, poiché quelli precedenti sono stati rimossi.

Vedi anche: Problemi noti per aggiornare il pacchetto di configurazione del repository su RHEL.

5 Aggiornamento dei componenti Zabbix

Per aggiornare i componenti Zabbix, eseguire:

```
dnf install zabbix-server-mysql zabbix-web-mysql zabbix-selinux-policy zabbix-agent
```

- Se si utilizza PostgreSQL, sostituire `mysql` con `pgsql` nel comando.
- Se si sta aggiornando il proxy, sostituire `server` con `proxy` nel comando.
- Se si dispone di RHEL 10 e si desidera utilizzare APM, aggiungere `-apm` al nome del pacchetto proxy (ad esempio, `zabbix-proxy-mysql-apm`).
- Se si sta aggiornando agent 2, sostituire `zabbix-agent` con `zabbix-agent2 zabbix-agent2-plugin-*` nel comando.
- Se sono installati altri pacchetti Zabbix, valutare la possibilità di aggiornarli includendoli nel comando.

> **Attention:**
> L'aggiornamento di Zabbix agent 2 con il comando `dnf install zabbix-agent2` potrebbe causare un errore. Per ulteriori informazioni, vedere *Problemi noti*.

Quindi, per aggiornare il frontend Zabbix con Apache e riavviare Apache, eseguire:

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

6 Rivedere i parametri di configurazione del componente

Esaminare le note di aggiornamento pertinenti per verificare se sono necessarie modifiche ai parametri di configurazione.

Per i nuovi parametri opzionali, vedere la pagina Novità.

7 Avviare i processi di Zabbix

Avviare i componenti Zabbix aggiornati:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
```

```
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Cancellare i cookie e la cache del browser web

Dopo l'aggiornamento, potrebbe essere necessario cancellare i cookie e la cache del browser web affinché l'interfaccia web di Zabbix funzioni correttamente.

Aggiornamento tra versioni minori

È possibile eseguire l'aggiornamento tra versioni minori di Zabbix 8.0.x (ad esempio, da 8.0.1 a 8.0.3).

Per aggiornare tutti i componenti di Zabbix, eseguire:

```
dnf upgrade 'zabbix-*'
```

- Per aggiornare solo Zabbix server, sostituire `'zabbix-*'` con `'zabbix-server-*'` nel comando.
- Per aggiornare solo Zabbix proxy, sostituire `'zabbix-*'` con `'zabbix-proxy-*'` nel comando.
- Per aggiornare solo Zabbix agent, sostituire `'zabbix-*'` con `'zabbix-agent-*'` nel comando.
- Per aggiornare solo Zabbix agent 2, sostituire `'zabbix-*'` con `'zabbix-agent2-*'` nel comando.

Aggiornamento dai container

Panoramica

Questa sezione descrive i passaggi necessari per un aggiornamento riuscito ai container di Zabbix **8.0**.x.

Sono disponibili insiemi separati di istruzioni per:

- Aggiornamento dell'immagine dei componenti Zabbix
- Aggiornamento del file Docker Compose

Warning:
Prima dell'aggiornamento, assicurarsi di leggere le **note di aggiornamento** pertinenti!

Attention:
Prima di avviare l'aggiornamento, verificare che gli utenti dispongano delle autorizzazioni necessarie sul database ai fini dell'aggiornamento.

 Per gli aggiornamenti da Zabbix 6.0 o versioni precedenti, durante l'aggiornamento dovranno essere creati trigger deterministici. Se la registrazione binaria è abilitata per MySQL/MariaDB, ciò richiede privilegi di superutente oppure l'impostazione della variabile/del parametro di configurazione `log_bin_trust_function_creators = 1`. Per istruzioni su come impostare la variabile, vedere Script di creazione del database.

 Si noti che, se eseguita da console, la variabile verrà impostata solo temporaneamente e verrà rimossa al riavvio di un Docker. In questo caso, mantenere in esecuzione il servizio SQL, arrestare solo il servizio `zabbix-server` eseguendo `docker compose down zabbix-server` e quindi `docker compose up -d zabbix-server`.

 In alternativa, è possibile impostare questa variabile nel file di configurazione.

A seconda delle dimensioni del database, l'aggiornamento del database alla versione 8.0 potrebbe richiedere molto tempo.

Aggiornamento dell'immagine del componente Zabbix

Queste istruzioni aggiornano le immagini Docker usate come base per i container dei componenti Zabbix.

I passaggi seguenti usano un proxy Zabbix come esempio. Sostituisci `zabbix-proxy-sqlite3` con il nome del container del tuo componente.

1. Controlla tutti i container per identificarne i nomi e le versioni correnti delle immagini:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### 5a0c2e3d02b4   zabbix/zabbix-proxy-sqlite3:alpine-7.4-latest   (...) zabbix-proxy-sqlite3
```

2. Scarica la nuova versione dell'immagine:

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Arresta il container del componente Zabbix:

```
docker stop zabbix-proxy-sqlite3
```

4. Rimuovi il container del componente Zabbix:

> **Attention:**
> Assicurati di aver annotato tutti i flag di configurazione e i mount dei volumi usati nel comando `docker run` originale prima di procedere, poiché saranno necessari per ricreare il container.

```
docker rm zabbix-proxy-sqlite3
```

5. Avvia un nuovo container usando l'immagine aggiornata:

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

> **Note:**
> Per ulteriori informazioni sull'esecuzione dei container Zabbix, consulta Docker (manual deployment).

6. Verifica che l'aggiornamento sia andato a buon fine controllando i log del container:

```
docker logs -f zabbix-proxy-sqlite3
```

Aggiornamento del file Docker Compose

Queste istruzioni aggiornano i file Docker Compose utilizzati per distribuire i componenti di Zabbix. Segui queste istruzioni se hai installato Zabbix usando Docker Compose.

1. Controlla tutti i container per identificarne i nomi e le versioni correnti delle immagini:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest    (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest (...) zabbix-docker-zabbix-web-ngina
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest    (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                               (...) zabbix-docker-mysql-server-1
```

2. Vai nella repository Docker di Zabbix che hai clonato in precedenza, scarica gli aggiornamenti più recenti e passa al branch 8.0:

> **Attention:**
> Se hai modificato `.env`, `compose.yaml` o altri file nella tua repository locale, un `git pull` potrebbe sovrascriverli. Esegui un backup di eventuali modifiche locali prima di procedere con questo passaggio.

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. Distribuisci i componenti di Zabbix usando il file Docker Compose aggiornato:

```
#### Con MySQL come database:
docker compose -f ./compose.yaml up -d

#### Con PostgreSQL come database:
docker compose -f ./compose_pgsql.yaml up -d
```

> **Note:**
> Dopo che tutti i container sono stati ricreati e avviati, potrebbero essere necessari 1-3 minuti prima che la raccolta dei dati riprenda. Per ulteriori informazioni sull'esecuzione dei container Zabbix, vedi Docker Compose.

4. Verifica che l'aggiornamento sia andato a buon fine controllando i log del container (sostituisci `zabbix-docker-zabbix-server-1` con il nome del container del tuo componente):

```
docker logs -f zabbix-docker-zabbix-server-1
```

Aggiornamento dai sorgenti

Panoramica

Questa sezione fornisce i passaggi necessari per eseguire correttamente l'upgrade da Zabbix **7.4**.x a Zabbix **8.0**.x utilizzando le fonti ufficiali di Zabbix.

> **Warning:**
> Prima dell'upgrade, assicurarsi di leggere le **note sull'upgrade**!

È inoltre consigliabile consultare i requisiti per la versione 8.0.

> **Note:**
> Durante l'upgrade può essere utile eseguire due sessioni SSH parallele: in una eseguire i passaggi dell'upgrade e nell'altra monitorare i log del server/proxy. Ad esempio, nella seconda sessione SSH eseguire `tail -f zabbix_server.log` o `tail -f zabbix_proxy.log` per visualizzare in tempo reale le voci più recenti del file di log e gli eventuali errori. Questo può essere fondamentale per le istanze in produzione.

Processo di aggiornamento del server

1 Arrestare il server

Arrestare il server Zabbix per assicurarsi che non vengano inseriti nuovi dati nel database.

2 Eseguire il backup del database Zabbix esistente

Questo è un passaggio molto importante. Assicurarsi di disporre di un backup del database. Sarà utile nel caso in cui la procedura di aggiornamento non vada a buon fine (spazio su disco insufficiente, spegnimento, qualsiasi problema imprevisto).

3 Eseguire il backup dei file di configurazione, dei file PHP e dei binari di Zabbix

Creare una copia di backup dei binari di Zabbix, dei file di configurazione e della directory dei file PHP.

4 Installare i nuovi binari del server

Utilizzare queste istruzioni per compilare il server Zabbix dai sorgenti.

5 Rivedere i parametri di configurazione del server

Assicurarsi di consultare le Note di aggiornamento per verificare se sono necessarie modifiche ai parametri di configurazione.

Per i nuovi parametri opzionali, vedere la pagina Novità.

6 Avviare i nuovi binari di Zabbix

Avviare i nuovi binari. Controllare i file di log per verificare che i binari siano stati avviati correttamente.

Zabbix server aggiornerà automaticamente il database. All'avvio, Zabbix server segnala le versioni attuali (obbligatoria e facoltativa) e quelle richieste del database. Se la versione obbligatoria attuale è precedente a quella richiesta, Zabbix server esegue automaticamente le patch di aggiornamento del database necessarie. Il livello di avvio e di avanzamento (percentuale) dell'aggiornamento del database viene scritto nel file di log di Zabbix server. Al termine dell'aggiornamento, nel file di log viene scritto il messaggio "database upgrade fully completed". Se una delle patch di aggiornamento non viene applicata correttamente, Zabbix server non si avvia. Zabbix server non si avvia neanche se la versione obbligatoria attuale del database è più recente di quella richiesta. Zabbix server si avvia solo se la versione obbligatoria attuale del database corrisponde alla versione obbligatoria richiesta.

```
8673:20161117:104750.259 current database version (mandatory/optional): 03040000/03040000
8673:20161117:104750.259 required mandatory version: 03040000
```

Prima di avviare il server:

- Assicurarsi che l'utente del database disponga di autorizzazioni sufficienti (creazione di tabelle, eliminazione di tabelle, creazione di indici, eliminazione di indici)
- Assicurarsi di avere spazio libero sufficiente sul disco.

7 Installare la nuova interfaccia web di Zabbix

La versione minima richiesta di PHP è la 8.2.0.
Aggiornare se necessario e seguire le istruzioni per l'installazione.

8 Cancellare i cookie e la cache del browser web

Dopo l'aggiornamento potrebbe essere necessario cancellare i cookie e la cache del browser web affinché l'interfaccia web di Zabbix funzioni correttamente.

Processo di aggiornamento del proxy

1 Arresta il proxy

Arresta il proxy Zabbix.

2 Eseguire il backup dei file di configurazione e dei file binari del proxy Zabbix

Crea una copia di backup del binario del proxy Zabbix e del file di configurazione.

3 Installare i nuovi binari del proxy

Utilizzare queste istruzioni per compilare il proxy Zabbix dai sorgenti.

4 Rivedi i parametri di configurazione del proxy

Assicurati di consultare le Note di aggiornamento per verificare se sono necessarie modifiche ai parametri di configurazione.

Per i nuovi parametri opzionali, consulta la pagina Novità.

5 Avviare il nuovo proxy Zabbix

Avviare il nuovo proxy Zabbix.
Controllare i file di log per verificare che il proxy sia stato avviato correttamente.

Il proxy Zabbix aggiornerà automaticamente il database.
L'aggiornamento del database avviene in modo analogo all'avvio del server Zabbix.

Processo di aggiornamento dell'agent

> **Attention:**
> L'aggiornamento degli agent non è obbligatorio. È necessario aggiornare gli agent solo se ciò è richiesto per accedere alle nuove funzionalità.

La procedura di aggiornamento descritta in questa sezione può essere utilizzata per aggiornare sia l'agent Zabbix sia Zabbix agent 2.

1 Arrestare l'agent

Ferma Zabbix agent.

2 Eseguire il backup dei file di configurazione e dei file binari dell'agente Zabbix

Crea una copia di backup del binario dell'agente Zabbix e del file di configurazione.

3 Installare i nuovi binari dell'agent

Utilizzare queste istruzioni per compilare l'agent Zabbix dai sorgenti.

In alternativa, è possibile scaricare gli agent Zabbix precompilati dalla pagina di download di Zabbix.

4 Rivedi i parametri di configurazione dell'agent

Assicurati di consultare le Note di aggiornamento per verificare se sono necessarie modifiche ai parametri di configurazione.

Per i nuovi parametri opzionali, consulta la pagina Novità.

5 Avviare il nuovo Zabbix agent

Avviare il nuovo Zabbix agent.
Controllare i file di log per verificare che l'agent sia stato avviato correttamente.

Aggiornamento tra versioni secondarie

Quando si esegue l'aggiornamento tra versioni secondarie di 8.0.x (ad esempio, dalla 8.0.1 alla 8.0.3), è necessario eseguire le stesse operazioni per server/proxy/agent previste durante l'aggiornamento tra versioni principali. L'unica differenza è che, durante l'aggiornamento tra versioni secondarie, non vengono apportate modifiche al database.

Problemi di aggiornamento

Relativo all'escaping Superato il limite di caratteri per i parametri delle funzioni dopo l'aggiornamento

In Zabbix 7.0 è stato aggiunto l'escaping corretto delle barre rovesciate nei parametri stringa delle funzioni dello storico.

Poiché durante l'aggiornamento dalle versioni di Zabbix precedenti alla 7.0 vengono aggiunte barre rovesciate supplementari, i parametri diventano più lunghi e questo può causare il malfunzionamento delle funzioni dei trigger se la lunghezza del parametro supera la dimensione massima dei dati di 255 caratteri.

Per illustrare quali problemi possono sorgere durante l'aggiornamento del database, quale sia il loro impatto e come riconoscerli e risolverli, prendiamo come esempio una configurazione con 3 item e 2 trigger.

Configurazione

1. Versione Zabbix 6.0.
2. File `/tmp/ONE` che contiene 114 barre rovesciate con una 'a' alla fine:

```
\\\
```

```
utilizzata sul host: 'Zabbix server'
  nel trigger: 'TRIG_232'.
Valore corrente del parametro:
'$,1m:now,"regexp","\\\
```

Impostazione della modalità SQL per un aggiornamento riuscito

L'impostazione `sql_mode` in MySQL/MariaDB deve avere la modalità "STRICT_TRANS_TABLES" impostata. Se è assente, l'aggiornamento del database Zabbix non riuscirà (vedere anche ZBX-19435).

Aggiornamento con MariaDB 10.2.1 e versioni precedenti

L'aggiornamento di Zabbix potrebbe non riuscire se le tabelle del database sono state create con MariaDB 10.2.1 o versioni precedenti, perché in tali versioni il formato di riga predefinito è compact. Questo problema può essere risolto modificando il formato di riga in dynamic (vedere anche ZBX-17690).

Set di caratteri e collation del database MySQL

Le installazioni di Zabbix che utilizzano un database MySQL creato con il set di caratteri utf8mb3 e la collation utf8mb3_bin potrebbero riscontrare problemi. Ad esempio, un problema noto di MySQL può causare errori `Row size too large (> 8126)` durante gli aggiornamenti dello schema del database o altre operazioni DDL.
 Si consiglia vivamente di convertire i database che utilizzano il set di caratteri utf8mb3 e la collation utf8mb3_bin nel set di caratteri utf8mb4 e nella collation utf8mb4_bin, come descritto in Riparazione del set di caratteri e della collation del database Zabbix.

Perdita di memoria e di descrittori di file con libcurl 8.21.0

Un problema noto in libcurl 8.21.0 può causare perdite di memoria e di descrittori di file nelle funzionalità di Zabbix che utilizzano libcurl durante la gestione di connessioni protette da TLS.
 Una soluzione temporanea per questo problema consiste nel eseguire il downgrade di libcurl alla versione 8.20.0 fino a quando non sarà disponibile una release che includa la correzione.

Template

Compatibilità dei template in ambienti dual-stack (IPv4/IPv6)

Negli ambienti dual-stack (sistemi configurati per supportare sia IPv4 che IPv6), il nome host `localhost` viene generalmente risolto sia in indirizzi IPv4 che IPv6. A causa della comune priorità assegnata a IPv6 rispetto a IPv4 da molti sistemi operativi e resolver DNS, i template di Zabbix potrebbero non funzionare correttamente se il servizio monitorato è configurato per ascoltare solo su IPv4.

I servizi che non sono configurati per ascoltare su indirizzi IPv6 possono diventare inaccessibili, causando errori di monitoraggio. Gli utenti potrebbero configurare correttamente l'accesso per IPv4 ma continuare a riscontrare problemi di connettività a causa del comportamento predefinito che privilegia IPv6.

Una soluzione alternativa consiste nell'assicurarsi che i servizi (Nginx, Apache, PostgreSQL, ecc.) siano configurati per ascoltare sia su indirizzi IPv4 che IPv6 e che al server/agent Zabbix sia consentito l'accesso tramite IPv6. Inoltre, nei template e nelle configurazioni di Zabbix, utilizzare esplicitamente `localhost` invece di `127.0.0.1` per garantire la compatibilità sia con IPv4 che con IPv6.

Ad esempio, quando si monitora PostgreSQL con il template PostgreSQL by Zabbix agent 2, potrebbe essere necessario modificare il file `pg_hba.conf` per consentire le connessioni all'utente `zbx_monitor`. Se l'ambiente dual-stack assegna priorità a IPv6 (il sistema risolve localhost in `::1`) e si configura `localhost` ma si aggiunge solo una voce IPv4 (`127.0.0.1/32`), la connessione non riuscirà perché non esiste una voce IPv6 corrispondente.

Il seguente esempio di file `pg_hba.conf` garantisce che l'utente `zbx_monitor` possa connettersi a qualsiasi database dalla macchina locale utilizzando sia indirizzi IPv4 che IPv6 con diversi metodi di autenticazione:

```
### 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
```

Se necessario, è anche possibile utilizzare direttamente l'indirizzo IPv4 (`127.0.0.1`) quando si configura la macro del template PostgreSQL by Zabbix agent 2 per la stringa di connessione.

Installazione accidentale dei pacchetti Zabbix di EPEL

Se il repository EPEL è installato e abilitato, l'installazione dei pacchetti Zabbix potrebbe scaricare le versioni EPEL invece dei pacchetti Zabbix ufficiali. Per risolvere il problema:

1. Rimuovere eventuali pacchetti Zabbix installati da EPEL:

```
dnf remove zabbix-server-mysql
```

2. Escludere i pacchetti Zabbix da EPEL aggiungendo la seguente riga al file `/etc/yum.repos.d/epel.repo`:

```
[epel]
...
excludepkgs=zabbix*
```

3. Reinstallare il pacchetto ufficiale del server Zabbix:

```
dnf install zabbix-server-mysql
```

Durante l'installazione, i pacchetti Zabbix ufficiali includono la parola `release` nella loro stringa di versione (ad esempio, `7.0.0-release1.el8`), distinguendosi così dai pacchetti EPEL.

Pacchetti Zabbix per RHEL negli ambienti Red Hat UBI

Quando si installa Zabbix dai pacchetti Red Hat Enterprise Linux negli ambienti Red Hat Universal Base Image, assicurarsi di avere accesso ai repository richiesti e alle dipendenze necessarie. I pacchetti Zabbix dipendono dalle librerie `libOpenIPMI.so` e `libOpenIPMIposix.so`, che non sono fornite da alcun pacchetto nei repository predefiniti del gestore pacchetti abilitati sui sistemi UBI e ciò comporterà errori di installazione.

Le librerie `libOpenIPMI.so` e `libOpenIPMIposix.so` sono disponibili nel pacchetto `OpenIPMI-libs`, fornito dal repository `redhat-#-for-<arch>-appstream-rpms`. L'accesso a questo repository è regolato dalle sottoscrizioni che, nel caso degli ambienti UBI, vengono propagate montando nel namespace del file system del container la configurazione del repository e le directory dei segreti dell'host RHEL.

Per ulteriori informazioni, vedere ZBX-24291.

Chiave di firma scaduta per i pacchetti RHEL

Durante l'aggiornamento di Zabbix su Red Hat Enterprise Linux o sulle sue derivate, potresti riscontrare un problema di chiave di firma scaduta per i pacchetti nel repository Zabbix. Quando una chiave di firma scade, i tentativi di verificare le firme dei pacchetti generano un errore che indica che il certificato o la chiave non sono più validi. Per esempio:

```
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
```

Per risolvere questi problemi, reinstalla manualmente l'ultima versione del pacchetto `zabbix-release` per la tua specifica variante di RHEL (sostituisci il link seguente con quello corretto dal repository Zabbix).

Per esempio, su **RHEL 10**, esegui:

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

Quindi aggiorna le informazioni del repository:

```
dnf update
```

Per ulteriori informazioni, vedi ZBX-24761.

Connessione TLS al database con MariaDB

La connessione TLS al database non è supportata con l'opzione `verify_ca` per il parametro DBTLSConnect se viene utilizzato MariaDB.

Possibili deadlock con MySQL/MariaDB

Quando si opera con un carico elevato e con più di un worker LLD coinvolto, è possibile incorrere in un deadlock causato da un errore di InnoDB relativo alla strategia di blocco delle righe (vedere bug upstream). L'errore è stato corretto in MySQL a partire dalla versione 8.0.29, ma non in MariaDB. Per maggiori dettagli, vedere ZBX-21506.

Correlazione globale degli eventi

Gli eventi potrebbero non essere correlati correttamente se l'intervallo di tempo tra il primo e il secondo evento è molto piccolo, cioè mezzo secondo o meno.

NetBSD 8.0 e versioni successive

Vari processi di Zabbix potrebbero arrestarsi in modo casuale all'avvio nelle versioni 8.X e 9.X di NetBSD.
Ciò è dovuto alla dimensione predefinita troppo ridotta dello stack (4 MB), che deve essere aumentata eseguendo:

```
ulimit -s 10240
```

Per ulteriori informazioni, consultare la segnalazione del problema correlata: ZBX-18275.

Limitazioni delle espressioni regolari in Zabbix agent 2

Zabbix agent 2 non supporta i lookahead e i lookbehind nelle espressioni regolari a causa delle limitazioni della libreria standard regexp di Go.

Controlli IPMI

I controlli IPMI non funzionano con il pacchetto standard della libreria OpenIPMI su Debian precedenti alla 9 (stretch) e Ubuntu precedenti alla 16.04 (xenial). Per risolvere il problema, ricompilare la libreria OpenIPMI con OpenSSL abilitato, come descritto in ZBX-6139.

IPMI — gli host non attendibili possono causare l'arresto anomalo di OpenIPMI

È presente un bug nella libreria OpenIPMI utilizzata da Zabbix per il polling dei dati IPMI, che può essere attivato da risposte appositamente predisposte provenienti da un dispositivo non attendibile. Un dispositivo IPMI non attendibile può inviare dati appositamente predisposti che causano l'arresto anomalo della libreria OpenIPMI, provocando a sua volta la terminazione del processo del server Zabbix che esegue il polling IPMI.

Controlli SSH

- Alcune distribuzioni Linux come Debian e Ubuntu non supportano le chiavi private cifrate (con passphrase) se la libreria libssh2 è installata dai pacchetti. Per maggiori dettagli, vedere ZBX-4850.
- Quando si utilizza libssh 0.9.x su alcune distribuzioni Linux con OpenSSH 8, i controlli SSH possono occasionalmente segnalare "Cannot read data from SSH server". Ciò è causato da un problema di libssh (rapporto più dettagliato). Si prevede che l'errore sia stato corretto con una release stabile di libssh 0.9.5. Per dettagli, vedere anche ZBX-17756.
- L'uso della pipe "|" nello script SSH può causare l'errore "Cannot read data from SSH server". In questo caso si consiglia di aggiornare la versione della libreria libssh. Per dettagli, vedere anche ZBX-21337.

Controlli ODBC

- Il driver MySQL unixODBC non dovrebbe essere utilizzato con Zabbix server o Zabbix proxy compilati con la libreria connettore MariaDB e viceversa; se possibile, è inoltre preferibile evitare di usare lo stesso connettore come driver a causa di un bug upstream. Configurazione consigliata:

  Connettore PostgreSQL, SQLite o Oracle → driver MariaDB o MySQL unixODBC
Connettore MariaDB → driver MariaDB unixODBC
Connettore MySQL → driver MySQL unixODBC

  Per ulteriori informazioni e le soluzioni alternative disponibili, vedere ZBX-7665.

- I dati XML interrogati da Microsoft SQL Server possono essere troncati in vari modi sui sistemi Linux e UNIX.

- È stato osservato che l'utilizzo dei controlli ODBC per il monitoraggio di database Oracle con varie versioni di Oracle Instant Client per Linux provoca l'arresto anomalo di Zabbix server.
 Vedere anche: ZBX-18402, ZBX-20803.

- Se si utilizza il driver FreeTDS UnixODBC, è necessario anteporre un'istruzione 'SET NOCOUNT ON' a una query SQL (ad esempio, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). In caso contrario, il database monitor item in Zabbix non riuscirà a recuperare le informazioni e restituirà l'errore "La query SQL ha restituito un risultato vuoto".
 Per ulteriori informazioni, vedere ZBX-19917.

Parametro del metodo di richiesta errato negli item

Il parametro del metodo di richiesta, utilizzato solo nei controlli HTTP, potrebbe essere impostato in modo errato su '1', un valore non predefinito per tutti gli item come risultato di un aggiornamento da una versione di Zabbix precedente alla 4.0. Per i dettagli su come correggere questa situazione, vedere ZBX-19308.

Monitoraggio web e HTTP agent

Zabbix server presenta perdite di memoria su alcune distribuzioni Linux a causa di un bug upstream quando "SSL verify peer" è abilitato negli scenari web o in HTTP agent. Per ulteriori informazioni e per le soluzioni alternative disponibili, vedere ZBX-10486.

Controlli semplici

Esiste un bug nelle versioni di **fping** precedenti alla v3.10 che gestisce in modo errato i pacchetti di risposta echo duplicati. Ciò può causare risultati inattesi per gli item `icmpping`, `icmppingloss`, `icmppingsec`. Si consiglia di utilizzare la versione più recente di **fping**. Per maggiori dettagli, vedere ZBX-11726.

Errori con l'esecuzione di fping in container rootless

Quando i container sono in esecuzione in modalità rootless o in un ambiente con restrizioni specifiche, durante l'esecuzione di controlli ICMP potresti riscontrare errori relativi all'esecuzione di fping, come `fping: Operation not permitted` oppure la perdita di tutti i pacchetti verso tutte le risorse.

Per risolvere questo problema, aggiungi `--cap-add=net_raw` ai comandi "docker run" o "podman run".

Inoltre, l'esecuzione di fping in ambienti non root può richiedere una modifica di sysctl, ad esempio:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

dove "1995" è il GID di zabbix. Per maggiori dettagli, vedi ZBX-22833.

Controlli SNMP

Se viene utilizzato il sistema operativo OpenBSD, un bug di tipo use-after-free nella libreria Net-SNMP fino alla versione 5.7.3 può causare un arresto anomalo di Zabbix server se il parametro SourceIP è impostato nel file di configurazione di Zabbix server. Come soluzione alternativa, non impostare il parametro SourceIP. Lo stesso problema si applica anche a Linux, ma non causa l'interruzione del funzionamento di Zabbix server. Una patch locale per il pacchetto net-snmp su OpenBSD è stata applicata e verrà rilasciata con OpenBSD 6.3.

Picchi nei dati SNMP

Sono stati osservati picchi nei dati SNMP che potrebbero essere correlati a determinati fattori fisici, come sbalzi di tensione nella rete elettrica. Per maggiori dettagli, vedere ZBX-14318.

Trap SNMP

Il pacchetto "net-snmp-perl", necessario per le trap SNMP, è stato rimosso in RHEL 8.0-8.2; reintrodotto in RHEL 8.3.

Quindi, se si utilizza RHEL 8.0-8.2, la soluzione migliore è eseguire l'aggiornamento a RHEL 8.3.

Per ulteriori informazioni, consultare anche ZBX-17192.

Arresto anomalo del processo alerter in RHEL 7

Sono stati riscontrati casi di arresto anomalo di un processo alerter di Zabbix server in RHEL 7. Per i dettagli, vedere ZBX-10461.

Aggiornamento di Zabbix agent 2 (6.0.5 o versioni precedenti)

Durante l'aggiornamento di Zabbix agent 2 (versione 6.0.5 o precedente) dai pacchetti, potrebbe verificarsi un errore di conflitto dei file relativo ai plugin. Per risolvere l'errore, eseguire un backup della configurazione di agent 2 (se necessario), disinstallare agent 2 e installarlo nuovamente.

Sui sistemi basati su RHEL, eseguire:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Sui sistemi basati su Debian, eseguire:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Per ulteriori informazioni, vedere ZBX-23250.

Cambio delle impostazioni locali del frontend

È stato osservato che le impostazioni locali del frontend possono cambiare senza una logica apparente, ovvero alcune pagine (o parti di pagine) vengono visualizzate in una lingua, mentre altre pagine (o parti di pagine) vengono visualizzate in una lingua diversa. In genere, il problema può verificarsi quando sono presenti diversi utenti, alcuni dei quali utilizzano un'impostazione locale, mentre altri ne utilizzano un'altra.

Una soluzione alternativa nota consiste nel disabilitare il multithreading in PHP e Apache.

Il problema è correlato al funzionamento dell'impostazione delle impostazioni locali in PHP: le informazioni sulle impostazioni locali vengono mantenute per processo, non per thread. Pertanto, in un ambiente multithread, quando sono presenti diversi progetti eseguiti dallo stesso processo Apache, è possibile che le impostazioni locali vengano modificate in un altro thread, modificando il modo in cui i dati possono essere elaborati nel thread di Zabbix.

Per ulteriori informazioni, consultare i report dei problemi correlati:

- ZBX-10911 (Problema con il cambio delle impostazioni locali del frontend)
- ZBX-16297 (Problema nell'elaborazione dei numeri nei grafici che utilizzano la funzione `bcdiv` delle funzioni BC Math)

Grafici

Problemi con i grafici (classici)

Se riscontri problemi con i grafici classici, si consiglia di aggiornare la libreria GD (libgd) alla versione 2.3.3-13 o successiva, e PHP alla versione 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 o successiva.

Ora legale

Le modifiche all'ora legale (DST) comportano irregolarità nella visualizzazione delle etichette dell'asse X (duplicazione della data, data mancante, ecc.).

Aggregazione sum

Quando si utilizza l'aggregazione sum in un grafico per un periodo inferiore a un'ora, i grafici mostrano valori errati (moltiplicati) quando i dati provengono dai trend.

Sovrapposizione del testo

Per alcune lingue del frontend (ad esempio il giapponese), i font locali possono causare la sovrapposizione del testo nella legenda del grafico. Per evitare questo problema, utilizzare la versione 2.3.0 (o successiva) dell'estensione PHP GD.

Monitoraggio dei file di log

Gli item `log[]` e `logrt[]` rileggono ripetutamente il file di log dall'inizio se il file system è pieno al 100% e il file di log viene aggiornato in append (vedere ZBX-10884 per ulteriori informazioni).

Query MySQL lente

Il server Zabbix genera query `SELECT` lente in caso di valori inesistenti per gli item. Questo problema è noto nelle versioni MySQL 5.6/5.7 (per una discussione più estesa, vedere ZBX-10652) e, in casi specifici, può verificarsi anche nelle versioni MySQL successive. Una soluzione alternativa consiste nel disabilitare l'ottimizzatore `index_condition_pushdown` o `prefer_ordering_index` in MySQL. Si noti, tuttavia, che questa soluzione alternativa potrebbe non risolvere tutti i problemi relativi alle query lente.

Impostazioni persistenti del filtro dai link

Quando si apre un link a una pagina del frontend di Zabbix che contiene impostazioni del filtro, incluso il selettore dell'ora, il filtro viene salvato automaticamente nel database per l'utente, sostituendo il filtro precedentemente salvato e/o le impostazioni del selettore dell'ora per quella pagina. Queste impostazioni rimangono attive finché l'utente non le aggiorna o reimposta manualmente.

Problema con l'indirizzo IPv6 nelle trap SNMPv3

A causa di un bug di net-snmp, l'indirizzo IPv6 potrebbe non essere visualizzato correttamente quando si utilizza SNMPv3 nelle trap SNMP. Per maggiori dettagli e una possibile soluzione alternativa, vedere ZBX-14541.

Indirizzo IP IPv6 lungo troncato nelle informazioni di accesso non riuscito

Un messaggio di tentativo di accesso non riuscito visualizzerà solo i primi 39 caratteri di un indirizzo IP memorizzato, poiché questo è il limite di caratteri del campo nel database. Ciò significa che gli indirizzi IP IPv6 più lunghi di 39 caratteri verranno mostrati in modo incompleto.

Controlli dell'agent Zabbix su Windows

Voci DNS inesistenti in un parametro `Server` del file di configurazione di Zabbix agent (`zabbix_agentd.conf`) possono aumentare il tempo di risposta di Zabbix agent su Windows. Ciò accade perché il daemon di caching DNS di Windows non memorizza nella cache le risposte negative per gli indirizzi IPv4. Tuttavia, per gli indirizzi IPv6 le risposte negative vengono memorizzate nella cache, quindi una possibile soluzione alternativa consiste nel disabilitare IPv4 sull'host.

Esportazione/importazione YAML

Ci sono alcuni problemi noti con esportazione/importazione YAML:

- I messaggi di errore non sono traducibili;
- A volte non è possibile importare JSON valido con estensione di file .yaml;
- Le date leggibili dall'uomo non racchiuse tra virgolette vengono convertite automaticamente in timestamp Unix.

Procedura guidata di configurazione su SUSE con NGINX e php-fpm

La procedura guidata di configurazione del frontend non può salvare il file di configurazione su SUSE con NGINX + php-fpm. Ciò è causato da un'impostazione nell'unità /usr/lib/systemd/system/php-fpm.service, che impedisce a Zabbix di scrivere in /etc. (introdotta in PHP 7.4).

Sono disponibili due opzioni di workaround:

- Impostare l'opzione ProtectSystem su 'true' invece di 'full' nell'unità systemd di php-fpm.
- Salvare manualmente il file `/etc/zabbix/web/zabbix.conf.php`.

Inoltro dell'header Authorization

In alcuni casi, Apache o NGINX possono impedire che l'header Authorization nelle richieste API raggiunga Zabbix. Questo può causare problemi di autenticazione quando si utilizza la API di Zabbix o servizi di single sign-on (SSO), come SAML con Okta.

Per risolvere questo problema, aggiorna la configurazione del tuo web server.

Per **Apache**, se lo stai utilizzando come reverse proxy (configurazione non CGI), aggiungi la seguente direttiva a `/etc/httpd/conf/httpd.c` (sui sistemi basati su RHEL) o `/etc/apache2/apache2.conf` (su Debian/Ubuntu):

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Se Apache esegue direttamente gli script per gestire le richieste (ad esempio, utilizzando mod_cgi), aggiungi invece la seguente direttiva:

```
CGIPassAuth On
```

Al contrario, **NGINX** gestisce automaticamente l'header Authorization. Tuttavia, se NGINX agisce come reverse proxy, puoi inoltrare esplicitamente l'header Authorization aggiungendo le seguenti direttive a `/etc/nginx/nginx.conf` (per la posizione del tuo frontend Zabbix):

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

Dopo aver aggiornato la configurazione, riavvia il tuo web server.

Per maggiori dettagli, vedi:

- ZBX-22952
- Apache 2.4 + PHP-FPM and Authorization headers
- Direttive SetEnvIfNoCase e CGIPassAuth
- NGINX Reverse Proxy

Chromium per il servizio web Zabbix su Ubuntu 20

Sebbene nella maggior parte dei casi il servizio web Zabbix possa essere eseguito con Chromium, su Ubuntu 20.04 l'uso di Chromium provoca il seguente errore:

```
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
```

Questo errore si verifica perché `/var/lib/zabbix` viene utilizzata come directory home dell'utente 'zabbix'.

Codici di errore personalizzati MySQL

Quando Zabbix rileva che il database di backend non è accessibile, invia una notifica e continua a tentare la connessione. Per alcuni motori di database, vengono riconosciuti codici di errore specifici. In MySQL, questi codici di errore riconosciuti includono:

- 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

Inoltre, quando si utilizza Zabbix con un'installazione MySQL su Azure, nei log di Zabbix può comparire il messaggio di errore generico *[9002] Some errors occurred*. Questo messaggio viene inviato dal database al server o al proxy Zabbix. Per determinare la causa dell'errore, consultare i log di Azure.

Espressioni regolari non valide dopo il passaggio a PCRE2

In Zabbix 6.0 è stato aggiunto il supporto per PCRE2. Anche se PCRE è ancora supportato, i pacchetti di installazione di Zabbix per RHEL 7 e versioni successive, SLES (tutte le versioni), Debian 9 e versioni successive, Ubuntu 16.04 e versioni successive sono stati aggiornati per usare PCRE2. Pur offrendo molti vantaggi, il passaggio a PCRE2 può causare la non validità di alcuni pattern regexp PCRE esistenti o un loro comportamento diverso. In particolare, questo interessa il pattern *[\w-\.]*. Per rendere di nuovo valida questa regexp senza modificarne la semantica, cambiare l'espressione in *[-\w\.]* . Ciò accade perché PCRE2 considera il trattino come delimitatore, creando un intervallo all'interno di una classe di caratteri.

Errore del widget Geomap

Le mappe nel widget Geomap potrebbero non caricarsi correttamente se avete eseguito l'upgrade da una versione precedente di Zabbix con NGINX e non siete passati al nuovo file di configurazione NGINX durante l'aggiornamento.

Per risolvere il problema, potete eliminare il vecchio file di configurazione, usare il file di configurazione del pacchetto della versione corrente e riconfigurarlo come descritto nelle istruzioni di download nella sezione *e. Configure PHP for Zabbix frontend*.

In alternativa, potete modificare manualmente un file di configurazione NGINX esistente (in genere, `/etc/zabbix/nginx.conf`). Per farlo, aprite il file e individuate il seguente blocco:

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

Quindi, sostituite questo blocco con:

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

Preprocessing — le variabili globali non sono sicure

Il codice JavaScript di preprocessing viene eseguito per ogni richiesta, ma le assegnazioni a identificatori non dichiarati (ad esempio `secret = value`) creano variabili globali implicite che possono persistere oltre l'esecuzione corrente. Memorizzare dati sensibili (token, password, ecc.) in variabili globali implicite aumenta il rischio di esposizione accidentale o di riutilizzo da parte di successive esecuzioni del preprocessing o di altre integrazioni eseguite nello stesso ambiente.

Non fare affidamento sulle variabili globali implicite. Dichiara sempre le variabili con `var` o `const` ed evita di collegare segreti a oggetti globali (ad esempio `globalThis` o `window`). Non esiste alcun modo supportato per sovrascrivere gli oggetti globali integrati dall'interno del preprocessing.

Esempio sicuro:

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

Gruppi di processori in Windows

La documentazione Microsoft afferma che i sistemi con meno di 64 processori logici hanno sempre un solo gruppo di processori, il Gruppo 0. Tuttavia, gli utenti di Zabbix hanno segnalato un raro bug ZBX-20260, in cui sono presenti due gruppi di processori su sistemi con 64 o meno processori logici. Questo ha comportato la presenza dei contatori delle prestazioni "\Processor(n)" solo per uno dei due gruppi di processori. La causa principale effettiva di questo bug non è nota. Tuttavia, un caso simile è stato descritto su stackoverflow.com, e in quel caso la causa principale era nell'interazione tra BIOS e Windows.

Limiti del filtraggio con collations utf8mb4

I filtri (ad esempio, in *Raccolta dati* > *Manutenzione*) potrebbero non funzionare correttamente quando vengono applicati a entità che contengono determinati caratteri Unicode (ad esempio, 𝒸, 𝓅). Questo problema è dovuto al modo in cui la collation predefinita utf8mb4_bin per i database MySQL o MariaDB gestisce l'ordinamento e il confronto dei caratteri Unicode.

Per risolvere questa limitazione, gli utenti possono modificare la collation delle colonne del database scegliendo alternative come utf8mb4_0900_bin, utf8mb4_0900_ai_ci o utf8mb4_unicode_520_ci. Si noti, tuttavia, che la modifica della collation può causare comportamenti imprevisti nella gestione degli spazi vuoti, nonché nell'ordinamento e nel filtraggio di altri caratteri.

Per ulteriori informazioni sulla modifica delle collations, vedere la documentazione MySQL o la documentazione MariaDB. Per i dettagli sulle differenze tra collations, vedere Unicode Character Sets nella documentazione MySQL.

Profilazione dell'uso eccessivo di memoria con tcmalloc

Se sospetti che la tua installazione di Zabbix stia usando troppa memoria, puoi usare la funzionalità di profilazione della memoria di tcmalloc per analizzare il consumo di memoria di Zabbix server/proxy.

1. Quando installi Zabbix dai sorgenti, configura i flag aggiuntivi:

```
export CFLAGS="-std=gnu99 -g -O0"
```

Il flag -std=gnu99 è richiesto per compilare Zabbix server, Zabbix proxy o Zabbix agent. Il flag -g aggiunge informazioni di debug supplementari, mentre -O0 disabilita le ottimizzazioni, che possono interferire con la profilazione di tcmalloc.

2. Imposta le seguenti variabili di ambiente prima di avviare Zabbix server. Queste variabili indicano a tcmalloc come tracciare e segnalare l'uso della memoria:

```
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. Genera un dump del profilo inviando il segnale 5 al processo di destinazione. Sostituisci 1234 con l'ID effettivo del processo (PID):

```
kill -5 1234
```

4. Stampa il profilo generato:

```
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 questo esempio, zbx_malloc2 è responsabile di quasi tutte le allocazioni di memoria.

Vedi anche:

- ZBX-25050 e ZBX-25584 per i relativi report dei problemi.
- GCC Option Summary per le opzioni di compilazione (-std=gnu99, -g, -O0, ecc.).
- la documentazione di Gperftools Heap Profiler sulle variabili di ambiente per la profilazione di tcmalloc.

Replica di gruppo MySQL 8.0 in modalità multi-primary

Quando si utilizza la replica di gruppo MySQL 8.0 in modalità multi-primary, è possibile riscontrare un errore durante i commit delle transazioni simile al seguente:

```
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
```

Questo errore sembra essere causato da problemi con le operazioni di rollback che coinvolgono vincoli di chiave esterna.

Vedere anche:

- ZBX-26060 per la segnalazione del problema correlato.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" per il problema upstream.

Accesso e systemd

Si consiglia di creare un utente *zabbix* come utente di sistema, ovvero senza possibilità di effettuare l'accesso. Alcuni utenti ignorano questa raccomandazione e utilizzano lo stesso account per effettuare l'accesso (ad esempio tramite SSH) al host che esegue Zabbix. Ciò potrebbe causare l'arresto anomalo del demone Zabbix al termine della sessione. In questo caso, nel log del server Zabbix verrà visualizzato un messaggio simile al seguente:

```
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
```

e nel log dell'agent Zabbix:

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Questo accade a causa dell'impostazione predefinita di systemd `RemoveIPC=yes`, configurata in `/etc/systemd/logind.conf`. Quando si esegue il logout dal sistema, i semafori creati in precedenza da Zabbix vengono rimossi, causando l'arresto anomalo.

Una citazione dalla documentazione di systemd:

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

Esistono 2 soluzioni a questo problema:

1. (consigliata) Smettere di utilizzare l'account *zabbix* per attività diverse dai processi Zabbix e creare un account dedicato per le altre attività.
2. (non consigliata) Impostare `RemoveIPC=no` in `/etc/systemd/logind.conf` e riavviare il sistema. Si noti che `RemoveIPC` è un parametro a livello di sistema; modificarlo avrà effetto sull'intero sistema.

Requisiti

Hardware

Memoria

Zabbix richiede sia memoria fisica che memoria su disco. La quantità di memoria su disco necessaria dipende ovviamente dal numero di host e di parametri che vengono monitorati. Se si prevede di conservare una lunga cronologia dei parametri monitorati, è opportuno considerare almeno un paio di gigabyte per avere spazio sufficiente a memorizzare la cronologia nel database. Ogni processo daemon di Zabbix richiede diverse connessioni a un server di database. La quantità di memoria allocata per la connessione dipende dalla configurazione del motore di database.

> **Note:**
> Maggiore è la memoria fisica disponibile, più velocemente funzionerà il database (e quindi Zabbix).

CPU

Zabbix e in particolare il database di Zabbix possono richiedere risorse CPU significative a seconda del numero di parametri monitorati e del motore di database scelto.

Altro hardware

Per utilizzare il supporto alle notifiche SMS in Zabbix sono necessari una porta di comunicazione seriale e un modem GSM seriale. Funzionerà anche un convertitore da USB a seriale.

Esempi di configurazione hardware

La tabella fornisce esempi di configurazione hardware, presupponendo una piattaforma **Linux/BSD/Unix**.

Questi sono esempi di dimensionamento e configurazione hardware da cui iniziare. Ogni installazione di Zabbix è unica. Assicurarsi di eseguire un benchmark delle prestazioni del proprio sistema Zabbix in un ambiente di staging o di sviluppo, in modo da comprendere appieno i requisiti prima di distribuire l'installazione di Zabbix nell'ambiente di produzione.

Dimensione dell'installazione	Metriche monitorate[1]	Core CPU/vCPU	Memoria (GiB)	Database	Amazon EC2[2]
Piccola	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Media	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Grande	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Molto grande	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] Ai fini del dimensionamento, 1 metrica monitorata è rappresentata approssimativamente dalla formula 1 metrica = 1 item + 1 trigger + 1 grafico. Questa è solo una regola di stima, non una formula per calcolare le metriche monitorate.
 [2] Esempio con istanze EC2 general purpose di Amazon, che utilizzano l'architettura ARM64 o x86_64; durante la valutazione e i test dell'installazione di Zabbix, prima dell'installazione nell'ambiente di produzione, è necessario selezionare un tipo di istanza appropriato, ad esempio ottimizzato per il calcolo, la memoria o lo storage.

Note:
La configurazione effettiva dipende in larga misura dal numero di item attivi e dalle frequenze di aggiornamento (per i dettagli, vedere la sezione dimensioni del database di questa pagina). Per le installazioni di grandi dimensioni, è altamente consigliato eseguire il database su un server separato.

Piattaforme supportate

A causa dei requisiti di sicurezza e della natura mission-critical del server di monitoraggio, UNIX è l'unico sistema operativo in grado di garantire costantemente le prestazioni, la tolleranza ai guasti e la resilienza necessarie. Zabbix funziona sulle versioni leader di mercato.

I componenti di Zabbix sono disponibili e testati per le seguenti piattaforme:

Piattaforma	Server	Agent	Agent 2	Commenti
Linux	x	x	x	
Windows	-	x	x	Zabbix agent è supportato su tutte le versioni desktop e server a partire da Windows XP (64-bit)/Server 2003. Zabbix agent 2 è supportato su tutte le versioni desktop e server a partire da Windows 10 (32-bit)/Server 2016, poiché è compilato solo con una versione Go supportata per prevenire vulnerabilità critiche di sicurezza. Da Go 1.21, le versioni minime richieste di Windows sono state innalzate, rendendo Windows 10/Server 2016 la versione minima per Zabbix agent 2.
macOS	x	x	-	
IBM AIX	x	x	-	Zabbix agent non funziona sulle piattaforme AIX con versioni inferiori a 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 può funzionare anche su altri sistemi operativi di tipo Unix.

Attention:
Zabbix disabilita i core dump se compilato con crittografia e non si avvia se il sistema non consente la disabilitazione dei core dump.

Software richiesto

Zabbix è basato su moderni server web, principali motori di database e sul linguaggio di scripting PHP.

Software esterno di terze parti circostante

Se indicato come obbligatorio, il software/la libreria richiesto è strettamente necessario. Quelli facoltativi sono necessari per supportare alcune funzioni specifiche.

Software	Stato di obbligatorietà	Versioni supportate	Commenti
*MySQL/Percona	Uno dei due	8.4.0-9.7.X	Richiesto se MySQL (o Percona) viene utilizzato come database backend di Zabbix. È richiesto il motore InnoDB. Si consiglia di utilizzare la libreria C API (libmysqlclient) per compilare server/proxy. Si consiglia vivamente di utilizzare negli ambienti di produzione solo release LTS. Le release contrassegnate come ”rolling release”, ”early access” o simili non sono consigliate.
MariaDB		10.11.00-12.3.X	È richiesto il motore InnoDB. Si consiglia di utilizzare la libreria MariaDB Connector/C per compilare server/proxy. Vedere anche: Possibili deadlock con MariaDB e Accesso agli elementi dell’interfaccia utente con MariaDB 10.5.1-10.5.9. Si consiglia vivamente di utilizzare negli ambienti di produzione solo release LTS. Le release contrassegnate come ”rolling release”, ”early access” o simili non sono consigliate.
PostgreSQL		15.0-18.X	Richiesto se PostgreSQL viene utilizzato come database backend di Zabbix. A seconda delle dimensioni dell’installazione, potrebbe essere necessario aumentare la proprietà di configurazione *work_mem* di PostgreSQL (il valore predefinito è 4MB), in modo che la quantità di memoria utilizzata dal database per una determinata operazione sia sufficiente e l’esecuzione delle query non richieda troppo tempo.
TimescaleDB per PostgreSQL		2.20.X-2.29.X	Richiesto se TimescaleDB viene utilizzato come estensione del database PostgreSQL. Assicurarsi di installare la Community Edition di TimescaleDB, che supporta la compressione. Per informazioni dettagliate sulla compatibilità delle versioni di PostgreSQL e TimescaleDB, consultare la documentazione di TimescaleDB.
Elasticsearch		7.X	Richiesto se Elasticsearch viene utilizzato come database backend di Zabbix. Elasticsearch può essere utilizzato solo per memorizzare i dati della cronologia ed è supportato solo con i server Zabbix. Il supporto di Elasticsearch è attualmente sperimentale. Vedere anche il software richiesto per server/proxy.

Software	Stato di obbligatorietà	Versioni supportate	Commenti
ClickHouse		26.3.0.0-26.9.X.X	Richiesto se ClickHouse viene utilizzato come database backend di Zabbix per memorizzare i dati della cronologia (supportato solo con i server Zabbix) o come origine dati di telemetria. Per utilizzare ClickHouse come origine dati di telemetria, è richiesto ClickHouse 26.8.17.4 o versione successiva. Anche l'ora di sistema sul server Zabbix e sul server ClickHouse deve essere sincronizzata; in caso contrario, le query di telemetria potrebbero non includere alcuni dati o restituire gli stessi dati due volte.
SQLite	Facoltativo	3.3.5-3.53.X	SQLite è supportato solo con i proxy Zabbix. Richiesto se SQLite viene utilizzato come database del proxy Zabbix.
smartmontools		7.1 o versione successiva	Richiesto per Zabbix agent 2.
who			Richiesto per il plugin del conteggio degli utenti.
dpkg			Richiesto per il plugin system.sw.packages.
pkgtool			Richiesto per il plugin system.sw.packages.
rpm			Richiesto per il plugin system.sw.packages.
pacman			Richiesto per il plugin system.sw.packages.
q applets			`qlist` e `qsize`, inclusi negli applet q, sono richiesti per il plugin system.sw.packages su Gentoo Linux.

Note:
Sebbene Zabbix possa funzionare con i database disponibili nei sistemi operativi, per un'esperienza ottimale si consiglia di utilizzare database installati dai repository ufficiali degli sviluppatori del database.

frontend

La larghezza minima dello schermo supportata dal frontend di Zabbix è di 1200px.

Se indicato come obbligatorio, il software/la libreria richiesto è strettamente necessario. Quelli opzionali sono necessari per supportare alcune funzioni specifiche.

Software	Stato obbligatorio	Versioni supportate	Commenti
PHP	Sì	8.2.0 - 8.5.X	
Apache	Uno dei seguenti	2.4 o successiva	
Nginx		1.20 o successiva	
MySQL	Uno dei seguenti	Vedere Software esterno di terze parti circostante	
PostgreSQL			
Estensioni PHP			
mysqli	Sì		Necessaria se MySQL viene utilizzato come database backend di Zabbix.
pgsql			Necessaria se PostgreSQL viene utilizzato come database backend di Zabbix.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); necessaria per il supporto agli script utente.

Software	Stato obbligatorio	Versioni supportate	Commenti
gd		2.0.28 o successiva	php-gd (se fornita come pacchetto separato dal distributore); l'estensione PHP GD deve supportare immagini PNG (*--with-png-dir*), immagini JPEG (*--with-jpeg-dir*) e FreeType 2 (*--with-freetype-dir*). Potrebbe essere richiesta la versione 2.3.0 o successiva per evitare possibili sovrapposizioni di testo nei grafici per alcune lingue del frontend.
libxml		2.6.15 o successiva	php-xml (se fornita come pacchetto separato dal distributore)
xmlwriter			php-xmlwriter (se fornita come pacchetto separato dal distributore)
xmlreader			php-xmlreader (se fornita come pacchetto separato dal distributore)
ctype			php-ctype (*--enable-ctype*)
session			php-session (se fornita come pacchetto separato dal distributore)
ldap	No		php-ldap; necessaria per l'autenticazione LDAP.
openssl			php-openssl; necessaria per l'autenticazione SAML.
gettext			php-gettext (*--with-gettext*); necessaria per le traduzioni.
cURL		7.19.4 o successiva	php-curl; necessaria per Duo Universal Prompt MFA, l'autenticazione SMTP e per testare la fonte dati di telemetria.

Librerie frontend di terze parti fornite con Zabbix:

Libreria	Stato obbligatorio	Versione fornita	Commenti
Libreria JavaScript jQuery	Sì	3.6.0	Libreria JavaScript che semplifica il processo di sviluppo cross-browser.
jQuery UI		1.12.1	Un insieme di interazioni, effetti, widget e temi dell'interfaccia utente basati su jQuery.
SAML PHP Toolkit		4.3.1	Un toolkit PHP che aggiunge il supporto all'autenticazione SAML 2.0 per consentire l'accesso a Zabbix.
Duo Universal PHP library		1.1.2	Una libreria PHP che aggiunge il supporto all'autenticazione a più fattori Duo Universal Prompt per Zabbix.
Symfony Yaml Component		5.1.0	Aggiunge il supporto all'esportazione e all'importazione degli elementi di configurazione di Zabbix nel formato YAML.

Note:
Zabbix potrebbe funzionare anche con versioni precedenti di Apache, MySQL e PostgreSQL.

Attention:
Per font diversi da DejaVu, quello predefinito, potrebbe essere richiesta la funzione PHP imagerotate. Se manca, questi font potrebbero essere visualizzati in modo errato quando viene mostrato un grafico. Questa funzione è disponibile solo se PHP viene compilato con GD integrato, cosa che non avviene in Debian e in altre distribuzioni.

Librerie di terze parti utilizzate per scrivere ed eseguire il debug del codice del frontend di Zabbix:

Libreria	Stato obbligatorio	Versione minima	Descrizione
Composer	No	2.4.1	Un gestore di pacchetti a livello applicativo per PHP che fornisce un formato standard per la gestione delle dipendenze del software PHP e delle librerie richieste.
PHPUnit		8.5.29	Un framework per i test unitari PHP utilizzato per testare il frontend di Zabbix.
SASS		3.4.22	Un linguaggio di scripting preprocessor che viene interpretato e compilato in Cascading Style Sheets (CSS).

Browser web lato client

I cookie e JavaScript devono essere abilitati.

Sono supportate le versioni stabili più recenti di Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari e Opera.

Warning:
È implementata la policy same-origin per gli IFrame, il che significa che Zabbix non può essere inserito in frame su un dominio diverso.

 Tuttavia, le pagine inserite in un frame di Zabbix avranno accesso al frontend di Zabbix (tramite JavaScript) se la pagina inserita nel frame e il frontend di Zabbix si trovano sullo stesso dominio. Una pagina come `http://secure-zabbix.com/cms/page.html`, se inserita nelle dashboard su `http://secure-zabbix.com/zabbix/`, avrà pieno accesso JS a Zabbix.

Server/proxy

Se indicato come obbligatorio, il software/la libreria richiesto è strettamente necessario. Quelli facoltativi sono necessari per supportare alcune funzioni specifiche.

Requisito	Stato obbligatorio	Descrizione
libpcre2	Sì	La libreria PCRE2 è necessaria per il supporto alle espressioni regolari compatibili con Perl (PCRE). È supportata PCRE2 v10.x.
libevent		Necessaria per la comunicazione tra processi. Versione 2.0.10 o successiva.
libevent-pthreads		Necessaria per la comunicazione tra processi.
libpthread		Necessaria per il supporto a mutex e lock di lettura-scrittura (potrebbe essere parte di libc).
libresolv		Necessaria per la risoluzione DNS (potrebbe essere parte di libc).
libiconv		Necessaria per la conversione della codifica/formattazione del testo (potrebbe essere parte di libc). Obbligatoria per Zabbix server su Linux.
libz		Necessaria per il supporto alla compressione.
libm		Libreria matematica. Necessaria solo per Zabbix server.
libmysqlclient	Una delle	Necessaria se viene utilizzato MySQL.
libmariadb		Necessaria se viene utilizzato MariaDB.
libpq5		Necessaria se viene utilizzato PostgreSQL; la versione di *libpq5* deve corrispondere alla versione del database PostgreSQL utilizzato o essere superiore.
libsqlite3		Necessaria se viene utilizzato SQLite. Necessaria solo per Zabbix proxy.
libOpenIPMI	No	Necessaria per il supporto IPMI. Necessaria solo per Zabbix server.
libssh2 o *libssh*		Necessaria per i controlli SSH. Versione 1.8.0 o successiva (libssh2); 0.9.0 o successiva (libssh).
libcurl		Necessaria per le seguenti funzionalità: - item di monitoraggio web, monitoraggio VMware e HTTP agent (per tutti: versione 7.19.1 o successiva); - item web.page.* di Zabbix agent (versione 7.19.1 o successiva; vedere anche i requisiti di agent); - autenticazione SMTP (Basic: versione 7.20.0 o successiva; OAuth: versione 7.33 o successiva; vedere anche i requisiti di frontend); - Elasticsearch (versione 7.28.0 o successiva); - item Telemetry query (versione 7.28.0 o successiva). È consigliata la versione 7.28.0 o successiva per tutte le funzionalità. Per utilizzare le funzionalità cURL aggiornate per gli item `web.page.*`, riavviare Zabbix server/proxy. Per l'autenticazione SMTP, utilizzare il pacchetto `libcurl-full` in fase di esecuzione.
libxml2		Necessaria per il monitoraggio VMware e la pre-elaborazione XML XPath.
net-snmp		Necessaria per il supporto SNMP. Versione 5.3.0 o successiva. Il supporto per i protocolli di crittografia avanzata (AES192/AES192C, AES256/AES256C) è disponibile a partire dalla libreria net-snmp 5.8; sui sistemi basati su RHEL 8+ è consigliato utilizzare net-snmp 5.8.15 o successiva.
libunixodbc		Necessaria per il monitoraggio dei database.
libgnutls o *libopenssl*		Necessaria quando si utilizza la crittografia. Versioni minime: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Necessaria per il supporto LDAP.
fping		Necessaria per gli item ping ICMP.

Requisito	Stato obbligatorio	Descrizione
c-ares		Necessaria per la risoluzione DNS asincrona se Zabbix è configurato con l'opzione `--with-ares`. In caso contrario, verrà utilizzata *libevent*. Versione minima: 1.16.0. La funzionalità di memorizzazione nella cache delle query DNS e di failover (cache delle query abilitata per impostazione predefinita) richiede c-ares 1.26.0 o successiva. Sui sistemi in cui c-ares è fornita dal pacchetto del sistema operativo, assicurarsi che la libreria runtime installata esponga le funzionalità del resolver necessarie per la memorizzazione nella cache DNS e il failover.
g++		Compilatore C++. Necessario per compilare Zabbix proxy con l'opzione `--with-apm`.
gRPC		Necessario per il supporto APM su Zabbix proxy. Librerie necessarie: *grpc++*, *grpc*, *gpr*. L'utility `grpc_cpp_plugin` è inoltre necessaria per compilare Zabbix proxy. La versione richiesta è disponibile nei repository dei pacchetti standard dei seguenti sistemi operativi: - Ubuntu 23.04 o successivo; - Debian 12 (Bookworm) o successivo; - Alpine Linux 3.15 o successivo; - RHEL, Rocky Linux, CentOS Stream 9 o successivo (solo dal repository EPEL); - Fedora X o successivo. Nelle versioni precedenti dei sistemi operativi, *gRPC* deve essere compilato dal codice sorgente.
protobuf		Necessario per il supporto APM su Zabbix proxy. Il compilatore `protoc` è inoltre necessario per compilare Zabbix proxy.
Abseil		Necessario per il supporto APM su Zabbix proxy. Librerie necessarie: *absl_synchronization*, *absl_base*. Di solito viene installato insieme a *gRPC*.

Agent

Requirement	Mandatory status	Description
libpcre2	Yes	La libreria PCRE2 è richiesta per il supporto alle Perl Compatible Regular Expression (PCRE). È supportata la versione PCRE2 v10.x. Richiesta per il monitoraggio dei log. Richiesta anche su Windows.
libpthread		Richiesta per il supporto a mutex e read-write lock (può far parte di libc). Non richiesta su Windows.
libresolv		Richiesta per la risoluzione DNS (può far parte di libc). Non richiesta su Windows.
libiconv		Richiesta per la conversione di codifica/formato del testo in UTF-8 negli item di log, contenuto di file, regex di file e item regmatch (può far parte di libc). Non richiesta su Windows.
libgnutls or *libopenssl*	No	Richiesta se si usa la crittografia. Versioni minime: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Su Microsoft Windows è richiesta OpenSSL 1.1.1 o successiva.
libldap		Richiesta se viene usato LDAP. Non supportata su Windows.
libcurl		Richiesta per il supporto esteso degli item web.page.* di Zabbix agent. Senza libcurl, è disponibile la funzionalità di base (ad esempio, web.page.get[http://example.com]). Con libcurl, l'agent supporta funzionalità aggiuntive, come URL HTTP con credenziali (ad esempio, http://user:password@example.com) e URL HTTPS. È richiesta la versione 7.19.1 o superiore (si consiglia la 7.28.0 o superiore). Per usare le funzionalità aggiornate di cURL, riavviare Zabbix agent.
libmodbus		Richiesta solo se viene usato il monitoraggio Modbus. Versione 3.0 o superiore.

Agent 2

Requirement	Mandatory status	Description
Go	Yes	Necessario per compilare Zabbix agent 2 e i relativi plugin dai sorgenti. È supportato Go 1.25 o versioni successive. Vedere go.dev per le istruzioni di installazione. Le librerie Go utilizzate da Zabbix agent 2 e dai relativi plugin sono elencate nel repository Git di Zabbix (le librerie contrassegnate come `indirect` nel repository sono dipendenze di altre librerie richieste): - Zabbix agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Yes	La libreria PCRE2 è richiesta per il supporto alle Perl Compatible Regular Expression (PCRE). È supportata la versione PCRE2 v10.x. Richiesta per il monitoraggio dei log. Richiesta anche su Windows.
libopenssl	No	Richiesta quando si utilizza la crittografia. Su piattaforme UNIX è richiesta OpenSSL 1.0.1 o versioni successive. La libreria OpenSSL deve avere il supporto PSK abilitato. LibreSSL non è supportata. Sui sistemi Microsoft Windows è richiesta OpenSSL 1.1.1 o versioni successive.

Servizio web

L'ultima versione stabile di Google Chrome è supportata per la generazione di report pianificati tramite il servizio web di Zabbix.

La versione di Go richiesta per compilare il servizio web corrisponde a quella utilizzata per Zabbix agent 2.

Java gateway

Se avete ottenuto Zabbix dal repository dei sorgenti o da un archivio, le dipendenze necessarie sono già incluse nell'albero dei sorgenti.

Se avete ottenuto Zabbix dal pacchetto della vostra distribuzione, le dipendenze necessarie sono già fornite dal sistema di packaging.

In entrambi i casi descritti sopra, il software è pronto per essere utilizzato e non sono necessari download aggiuntivi.

Se, tuttavia, desiderate fornire le vostre versioni di queste dipendenze, ad esempio se state preparando un pacchetto per una distribuzione Linux, di seguito è riportato l'elenco delle versioni delle librerie con cui è noto che Java gateway funziona. Zabbix potrebbe funzionare anche con altre versioni di queste librerie.

La tabella seguente elenca i file JAR attualmente inclusi con Java gateway nel codice originale:

Libreria	Stato obbligatorio	Versione inclusa	Commenti
android-json	Sì	4.3r1	JSON (JavaScript Object Notation) è un formato leggero per lo scambio di dati. Questa è l'implementazione Android compatibile con `org.json`, estratta dall'SDK Android.
logback-classic		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

Java gateway può essere compilato utilizzando Oracle Java oppure l'open source OpenJDK (versione 8 o successiva). I pacchetti forniti da Zabbix vengono compilati utilizzando OpenJDK. La tabella seguente elenca i pacchetti OpenJDK utilizzati per compilare i pacchetti Zabbix dalle varie distribuzioni:

Distribuzione	Pacchetto OpenJDK
AlmaLinux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)

Distribuzione	Pacchetto OpenJDK
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

Adattatore bridge

L’adattatore bridge di Zabbix non presenta requisiti particolari.

Porte predefinite

Il seguente elenco contiene le porte che i componenti Zabbix possono utilizzare in una configurazione predefinita:

Componente Zabbix	Protocollo e porta	Connessione avviata da
Zabbix agent	TCP 10050	Zabbix server o Zabbix proxy (solo controlli passivi)
Zabbix agent 2	TCP 10050	Zabbix server o Zabbix proxy (solo controlli passivi)
Zabbix server	TCP 10051	Zabbix agent attivi, Zabbix proxy attivo, Zabbix sender, Zabbix frontend
Zabbix proxy	TCP 10051	Zabbix agent attivi, Zabbix sender, Zabbix server (solo proxy passivo)
Zabbix Java gateway	TCP 10052	Zabbix server o proxy
Zabbix web service	TCP 10053	Zabbix server
Zabbix MCP server	TCP 8443	Client MCP (agent AI)
Zabbix frontend	TCP 80	Browser web, Zabbix web service (report pianificati), client API
	TCP 443	

Componente Zabbix	Protocollo e porta	Connessione avviata da
Trap SNMP (snmptrapd)	UDP 162	Dispositivi SNMP
Adattatore bridge Zabbix	TCP 10054	Zabbix server

Warning:
Aprire solo le porte richieste dai componenti effettivamente in esecuzione e limitare ogni regola del firewall agli host specifici che devono connettersi (ad esempio, consentire la porta 10050 solo dal Zabbix server o dal proxy, non da qualsiasi indirizzo). Non esporre direttamente le porte Zabbix a Internet.
 Prima di aprire le porte, proteggere l'installazione: abilitare la crittografia tra i componenti, limitare gli host consentiti nei file di configurazione dei componenti e consultare le best practice di sicurezza.

Dimensione del database

I dati di configurazione di Zabbix richiedono una quantità fissa di spazio su disco e non crescono molto.

La dimensione del database di Zabbix dipende principalmente da queste variabili, che definiscono la quantità di dati storici memorizzati:

- Numero di valori elaborati al secondo

Questo è il numero medio di nuovi valori che il server Zabbix riceve ogni secondo. Ad esempio, se abbiamo 3000 item da monitorare con un intervallo di aggiornamento di 60 secondi, il numero di valori al secondo viene calcolato come 3000/60 = **50**.

Ciò significa che ogni secondo vengono aggiunti 50 nuovi valori al database di Zabbix.

- Impostazioni dell'housekeeper per la history

Zabbix conserva i valori per un periodo di tempo fisso, normalmente diverse settimane o mesi. Ogni nuovo valore richiede una certa quantità di spazio su disco per i dati e l'indice.

Quindi, se volessimo conservare 30 giorni di history e ricevessimo 50 valori al secondo, il numero totale di valori sarebbe circa (**30***24*3600)* **50** = 129.600.000, ovvero circa 130M di valori.

A seconda del motore di database utilizzato, del tipo di valori ricevuti (float, interi, stringhe, file di log, ecc.), lo spazio su disco necessario per conservare un singolo valore può variare da 40 byte a centinaia di byte. Normalmente è di circa 90 byte per valore per gli item numerici[2]. Nel nostro caso, ciò significa che 130M di valori richiederanno 130M * 90 byte = **10.9GB** di spazio su disco.

Note:
La dimensione dei valori di item di testo/log è impossibile da prevedere con precisione, ma si può stimare circa 500 byte per valore.

- Impostazione dell'housekeeper per i trends

Zabbix conserva un insieme di valori max/min/avg/count su 1 ora per ciascun item nella tabella **trends**. I dati vengono utilizzati per i grafici di trend e di lungo periodo. Il periodo di un'ora non può essere personalizzato.

Il database di Zabbix, a seconda del tipo di database, richiede circa 90 byte per ciascun total. Supponiamo di voler conservare i dati di trend per 5 anni. I valori per 3000 item richiederanno 3000*24*365* **90** = **2.2GB** all'anno, oppure **11GB** per 5 anni.

- Impostazioni dell'housekeeper per gli eventi

Ogni evento di Zabbix richiede approssimativamente 250 byte di spazio su disco[1]. È difficile stimare il numero di eventi generati quotidianamente da Zabbix. Nel caso peggiore, possiamo assumere che Zabbix generi un evento al secondo.

Per ogni evento recuperato, viene creato un record event_recovery. Normalmente la maggior parte degli eventi verrà recuperata, quindi possiamo assumere un record event_recovery per evento. Ciò significa 80 byte aggiuntivi per evento.

Facoltativamente, gli eventi possono avere tag; ogni record di tag richiede approssimativamente 100 byte di spazio su disco[1]. Il numero di tag per evento (#tags) dipende dalla configurazione. Quindi ciascuno richiederà ulteriori #tags * 100 byte di spazio su disco.

Ciò significa che, se vogliamo conservare 3 anni di eventi, saranno necessari 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B di spazio su disco[2].

Note:
[1] Maggiore in presenza di nomi evento, tag e valori non ASCII.
 [2] Le stime delle dimensioni si basano su MySQL e potrebbero essere diverse per altri database.

La tabella contiene formule che possono essere utilizzate per calcolare lo spazio su disco richiesto dal sistema Zabbix:

Parameter	Formula for required disk space (in bytes)
Zabbix configuration	Fixed size. Normally 10MB or less.
History	days*(items/refresh rate)*24*3600*bytes items : number of items days : number of days to keep history refresh rate : average refresh rate of items bytes : number of bytes required to keep single value, depends on database engine, normally ~90 bytes.
Trends	days*(items/3600)*24*3600*bytes items : number of items days : number of days to keep history bytes : number of bytes required to keep single trend, depends on the database engine, normally ~90 bytes.
Events	days*events*24*3600*bytes events : number of event per second. One (1) event per second in worst-case scenario. days : number of days to keep history bytes : number of bytes required to keep single event, depends on the database engine, normally ~330 + average number of tags per event * 100 bytes.

Quindi, lo spazio totale richiesto su disco può essere calcolato come:

Configuration + History + Trends + Events

Lo spazio su disco NON verrà utilizzato immediatamente dopo l'installazione di Zabbix. La dimensione del database crescerà e poi smetterà di crescere a un certo punto, che dipende dalle impostazioni dell'housekeeper.

Sincronizzazione dell'ora

È molto importante avere un'ora di sistema precisa sul server su cui è in esecuzione Zabbix. ntpd è il demone più diffuso che sincronizza l'ora del host con quella di altre macchine. Si raccomanda vivamente di mantenere sincronizzata l'ora di sistema su tutti i sistemi su cui sono in esecuzione i componenti di Zabbix.

Novità e note sull'aggiornamento: 8.0

Zabbix 8.0.0 prosegue il percorso iniziato con Zabbix 7.4.0, aggiungendo nuove funzionalità e miglioramenti.

Inoltre, in questa pagina:

- Novità nei template, inclusi i nuovi template e le modifiche ai template esistenti.
- Note sull'aggiornamento, incluse le modifiche incompatibili.

Vedere anche: Modifiche all'API.

Note di aggiornamento

Queste note sono destinate all'aggiornamento da Zabbix 7.4.x a Zabbix 8.0.0.

Tutte le note sono raggruppate in:

- **Modifiche incompatibili** - modifiche che potrebbero compromettere le installazioni esistenti e altre informazioni critiche relative al processo di aggiornamento
- **Altro** - tutte le restanti informazioni che descrivono le modifiche nelle funzionalità di Zabbix

Vedi anche:

- Procedura di aggiornamento per tutte le informazioni pertinenti sull'aggiornamento da versioni precedenti a Zabbix 8.0.0;
- Aggiornamento del cluster HA per le istruzioni sull'aggiornamento dei server in un cluster ad **alta disponibilità** (HA).

Modifiche incompatibili

Altro

Novità e note sull'aggiornamento: 8.0.x

Questa pagina presenta tutte le nuove funzionalità incluse nelle versioni secondarie della versione principale di Zabbix. Sono incluse anche le modifiche ai template e le note sull'aggiornamento.

Vedere anche Novità della versione principale.

Novità di Zabbix 8.0.1 Questa versione non è ancora stata rilasciata.

Vedere le note di aggiornamento per questa versione.

Note di aggiornamento

Appliance Zabbix

Panoramica L'appliance Zabbix consente di distribuire immediatamente il server Zabbix e il frontend, invece di configurarli manualmente o riutilizzare un server esistente per Zabbix.

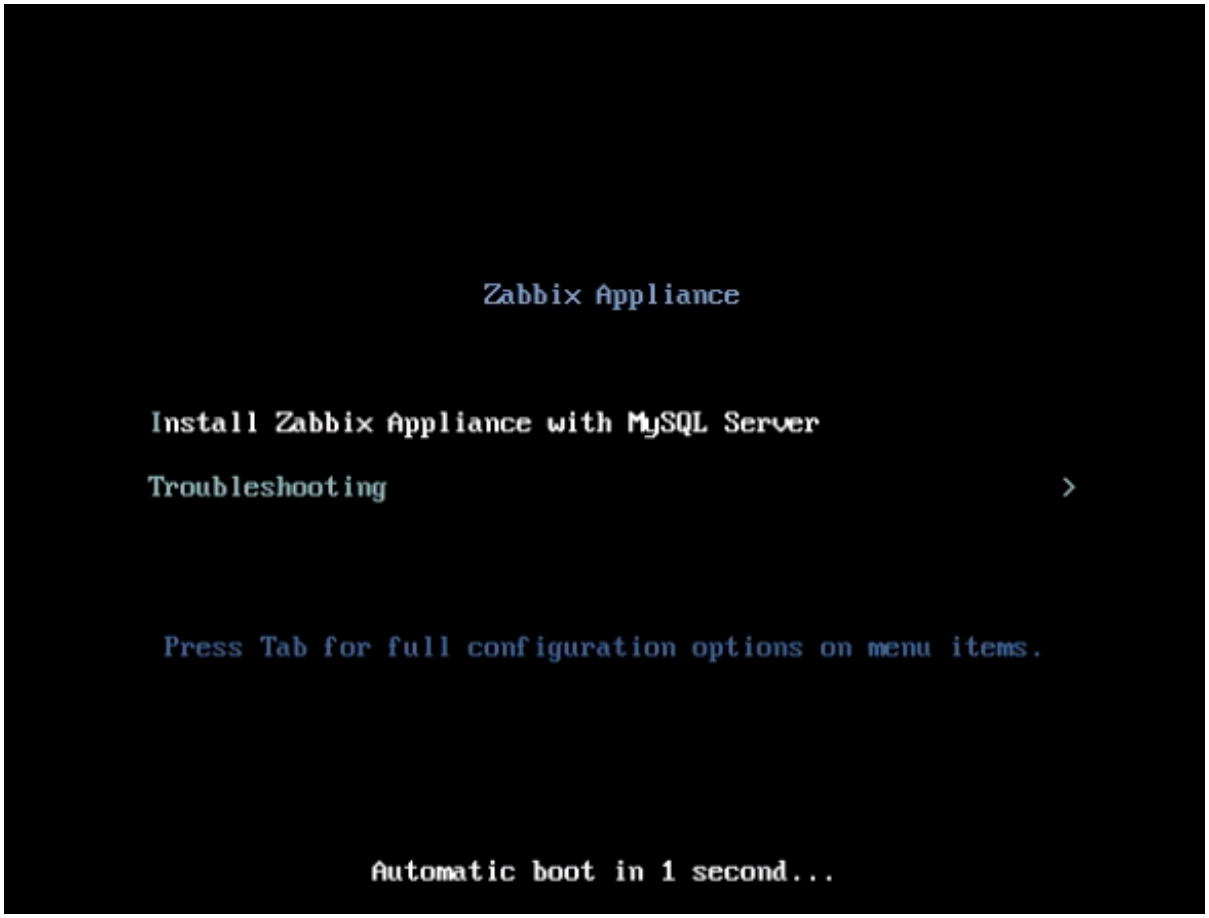
L'appliance per Zabbix 8.0 è basata su derivati di Red Hat Enterprise Linux 10, come AlmaLinux o Rocky Linux, e contiene un server Zabbix preconfigurato in esecuzione su MySQL e il frontend in esecuzione su un web server NGINX.

> **Attention:**
> Questa appliance è progettata per la valutazione di Zabbix. Se ne sconsiglia l'utilizzo in ambienti di produzione critici.

Le immagini dell'appliance sono disponibili per il download nei seguenti formati:

- CD/DVD di installazione (.iso)
- VMware (.vmx) - vedere le note
- Formato di virtualizzazione aperto (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - vedere le note
- KVM, Parallels, QEMU, chiavetta USB, VirtualBox, Xen (.raw) - vedere le note
- KVM, QEMU (.qcow2)

Menu di avvio del CD/DVD di installazione di Zabbix:



Avvio rapido Prerequisiti

1. Assicurarsi che la macchina host disponga di risorse sufficienti per soddisfare i requisiti di sistema della macchina virtuale:

- *RAM*: 4 GB
- *Spazio su disco*: devono essere allocati almeno 8 GB per la macchina virtuale
- *CPU*: minimo 2 core

2. Se non è ancora installato, installare il software di virtualizzazione per avviare l'immagine dell'appliance (ad esempio, VirtualBox).
3. Scaricare l'appliance nel formato supportato dal software di virtualizzazione in uso.
4. Verificare le impostazioni di rete per assicurarsi che DHCP sia abilitato sulla macchina host.

Installazione

1. Avvia la macchina virtuale dell'appliance dall'immagine scaricata. Per le istruzioni, consulta la documentazione del tuo software di virtualizzazione, ad esempio, documentazione di VirtualBox.
2. Configura le impostazioni di rete della macchina virtuale per consentire l'accesso da un browser sull'host. Questo può essere ottenuto abilitando la *modalità Bridged*.
3. Accedi alla macchina virtuale usando le credenziali di sistema predefinite.
4. Per recuperare l'indirizzo IP, esegui il seguente comando sulla macchina virtuale:

```
ip addr show
```

5. Apri un browser sull'host e punta all'indirizzo IP che l'appliance ha ricevuto tramite DHCP.
6. Accedi a Zabbix usando le credenziali predefinite e inizia il monitoraggio.

Configurazione Questa sezione descrive le impostazioni di configurazione predefinite richieste più di frequente, insieme alle opzioni di personalizzazione disponibili.

Credenziali

Sistema

- Nome utente: `root`

- Password: `zabbix`

frontend di Zabbix

- Nome utente: `Admin`
- Password: `zabbix`

Dopo aver effettuato l'accesso, è possibile modificare la password predefinita nelle impostazioni del profilo utente oppure creare nuovi utenti ed eliminare quello predefinito.

Database

Le password per tutti gli utenti del database vengono generate casualmente durante il processo di installazione. Per il database sono definiti i seguenti utenti:

Root:

- Nome utente: `root`
- Password: la password è memorizzata nel file `/root/.my.cnf` Non è necessario inserire una password con l'account root.

Zabbix server:

- Nome utente: `zabbix_srv`
- Password: la password è memorizzata nel file di configurazione del server

Zabbix frontend:

- Nome utente: `zabbix_web`
- Password: la password è memorizzata nel file di configurazione del frontend

Per modificare la password di un utente del database, modificarla in MySQL e nel file di configurazione corrispondente.

Accesso al frontend

È possibile accedere al frontend di Zabbix all'indirizzo *http://<virtual machine's IP>*.

Per impostazione predefinita, l'accesso è consentito da qualsiasi posizione. Per limitare l'accesso, modificare `/etc/nginx/conf.d/zabbix.`
Dopo aver salvato il file modificato, riavviare NGINX accedendo tramite SSH come utente root ed eseguendo:

```
systemctl restart nginx
```

Indirizzo IP statico

Per impostazione predefinita, l'appliance usa DHCP per ottenere l'indirizzo IP. Per impostare un indirizzo IP statico:

1. Accedi come utente root.
2. Esegui i seguenti comandi, sostituendo i valori con il tuo indirizzo IP personalizzato:

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # Indirizzo IP/CIDR prefix dell'appliance
nmcli con mod eth0 ipv4.gateway 192.168.1.1     # Indirizzo IP del gateway
nmcli con mod eth0 ipv4.dns 8.8.8.8    # Indirizzo IP del server DNS
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

In alternativa, puoi aprire il file `/etc/NetworkManager/system-connections/eth0.nmconnection` e aggiungere le seguenti righe:

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

Dopo aver salvato il file modificato, esegui i comandi:

```
nmcli con reload
nmcli con up eth0
```

Per verificare che le modifiche alla configurazione di rete siano state applicate, esegui il seguente comando:

```
nmcli dev show eth0
```

Configurazione del firewall

Per gestire le impostazioni del firewall, l'appliance utilizza iptables con regole predefinite:

- Apertura della porta SSH (22 TCP)
- Apertura delle porte di Zabbix agent (10050 TCP) e Zabbix trapper (10051 TCP)
- Apertura delle porte HTTP (80 TCP) e HTTPS (443 TCP)
- Apertura della porta SNMP trap (162 UDP)
- Apertura delle connessioni in uscita verso la porta NTP (123 UDP)
- Limitazione dei pacchetti ICMP a 5 pacchetti al secondo
- Blocco di tutte le altre connessioni in ingresso

Per aprire porte aggiuntive, modificare il file `/etc/sysconfig/iptables` e ricaricare le regole del firewall:

```
systemctl reload iptables
```

Repository

L'appliance Zabbix utilizza il pacchetto *zabbix-release* dal repository di Zabbix. I repository sono configurati nella directory `/etc/yum.repos.d/*`.

Fuso orario

Per impostazione predefinita, l'appliance utilizza UTC per l'orologio di sistema. Per cambiare il fuso orario, copiare il file appropriato da `/usr/share/zoneinfo` a `/etc/localtime`, ad esempio:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

> **Note:**
> Il fuso orario del frontend di Zabbix è impostato separatamente e può essere modificato nelle impostazioni del frontend. Il fuso orario predefinito per il frontend di Zabbix è Europe/Riga.

Posizioni dei file

- I file di configurazione del server e dell'agent Zabbix si trovano in `/etc/zabbix`.
- I file di log del server e dell'agent Zabbix si trovano in `/var/log/zabbix`.
- Il file di configurazione del frontend Zabbix si trova in `/etc/zabbix/web`.
- Il frontend Zabbix si trova in `/usr/share/zabbix`.
- La directory home dell'utente *zabbix* è `/var/lib/zabbix`.

Servizi di sistema

I servizi systemd sono disponibili. Per visualizzare l'elenco dei servizi Zabbix, eseguire il seguente comando sulla macchina virtuale:

```
systemctl list-units zabbix*
```

Note specifiche sul formato dell'immagine VMware

Le immagini in formato *vmdk* possono essere utilizzate direttamente nei prodotti VMware Player, Server e Workstation. Per utilizzarle in ESX, ESXi e vSphere, devono essere convertite utilizzando VMware vCenter Converter (è richiesta l'autenticazione per il download). Se si utilizza VMware vCenter Converter, potrebbero verificarsi problemi con l'adattatore di rete ibrido. In tal caso, è possibile provare a specificare l'adattatore E1000 durante il processo di conversione. In alternativa, al termine della conversione, è possibile eliminare l'adattatore esistente e aggiungere un adattatore E1000.

Immagine HDD/flash (raw)

Per avviare l'immagine, eseguire:

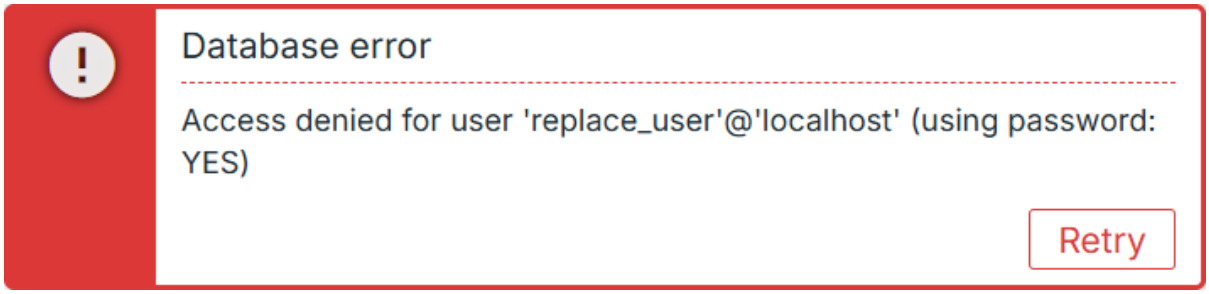
```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

Sostituire `/dev/sdc` con il percorso del dispositivo del disco Flash/HDD.

Hyper-V

Se l'appliance non riesce ad avviarsi in Hyper-V, provare a premere **Ctrl+Alt+F2** per passare a una sessione TTY.

Risoluzione dei problemi Se durante il tentativo di accesso al frontend viene visualizzato il messaggio di errore *Access denied for user 'replace_user'@'localhost' (using password: YES)*, potrebbe indicare che l'installazione è ancora in corso.



Se l'errore persiste dopo aver atteso alcuni minuti, oppure se noti altri comportamenti imprevisti, è probabile che il processo di installazione non sia stato completato correttamente. In questo caso, consigliamo di eliminare l'appliance corrente e distribuirla nuovamente seguendo le stesse istruzioni di installazione. Questo passaggio in genere risolve il problema.

Tieni presente che non è consigliato tentare di correggere manualmente un'installazione danneggiata, poiché potrebbe causare ulteriori complicazioni.

Processi

Panoramica Zabbix è composto da diversi processi che lavorano insieme per monitorare i sistemi e inviare notifiche. Ogni processo ha un ruolo specifico nell'infrastruttura di monitoraggio.

Processi principali

- Server - il processo centrale che esegue il polling e la ricezione dei dati, calcola i trigger e invia notifiche agli utenti.
- Agent - raccoglie dati dagli host monitorati.
- Agent 2 - un agent di nuova generazione con architettura a plugin e funzionalità avanzate.
- Proxy - raccoglie i dati di monitoraggio dai dispositivi monitorati per conto del server Zabbix e invia le informazioni a un server Zabbix centralizzato in una configurazione distribuita.
- Java gateway - consente il monitoraggio delle applicazioni Java.
- MCP server - consente agli agenti AI di lavorare con Zabbix utilizzando il Model Context Protocol.

Per accedere facilmente a Zabbix da qualsiasi luogo e piattaforma, è disponibile un'interfaccia basata sul web. L'interfaccia fa parte del server Zabbix e generalmente (ma non necessariamente) viene eseguita sulla stessa macchina fisica del server.

Tutte le informazioni di configurazione e i dati raccolti da Zabbix vengono archiviati in un database.

Intestazione L'intestazione è presente in tutti i messaggi di richiesta e risposta tra i componenti di Zabbix.
È necessaria per determinare la lunghezza del messaggio, se è compresso o meno, e se si tratta o meno di un pacchetto di grandi dimensioni.

Il protocollo di comunicazione di Zabbix ha un limite di dimensione del pacchetto di 1 GB per connessione. Il limite di 1 GB si applica sia alla lunghezza dei dati del pacchetto ricevuto sia alla lunghezza dei dati non compressi.

Quando si invia la configurazione a Zabbix proxy, il limite di dimensione del pacchetto viene aumentato a 4 GB per consentire la sincronizzazione di configurazioni di grandi dimensioni. Quando la lunghezza dei dati prima della compressione supera i 4 GB, Zabbix server avvia automaticamente
l'uso del formato di pacchetto di grandi dimensioni (flag 0x04), che aumenta il limite di dimensione del pacchetto a 16 GB.

Si noti che, sebbene il formato di pacchetto di grandi dimensioni possa essere usato per inviare qualsiasi dato, attualmente solo il sincronizzatore di configurazione di Zabbix proxy
è in grado di gestire pacchetti più grandi di 1 GB.

Struttura dell'intestazione L'intestazione è composta da quattro campi. Tutti i numeri nell'intestazione sono formattati in little-endian.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" o 5A 42 58 44

Field	Size	Size (large packet)	Description
<FLAGS>	1	1	Flag del protocollo: 0x01 - protocollo di comunicazione Zabbix 0x02 - compressione 0x04 - large packet
<DATALEN>	4	8	Lunghezza dei dati.
<RESERVED>	4	8	Quando viene usata la compressione (flag 0x02) - la lunghezza dei dati non compressi Quando la compressione non viene usata - 00 00 00 00

Esempi di header Ecco alcuni frammenti di codice che mostrano come aggiungere l'header del protocollo Zabbix ai dati che si desidera inviare, in modo da ottenere il pacchetto da inviare a Zabbix affinché venga interpretato correttamente. Questi frammenti di codice presuppongono che i dati non siano più grandi di 1 GB, quindi non viene usato il formato per pacchetti di grandi dimensioni.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

oppure

```
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;
```

oppure

```
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;
```

oppure

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

server

Panoramica Zabbix server è il processo centrale del software Zabbix.

Il server esegue il polling e il trapping dei dati, calcola i trigger e invia notifiche agli utenti. È il componente centrale a cui gli agent e i proxy di Zabbix segnalano i dati sulla disponibilità e sull'integrità dei sistemi. Il server può anche verificare da remoto i servizi in rete (come web server e mail server) usando semplici controlli di servizio.

Il server è il repository centrale in cui vengono archiviati tutti i dati di configurazione, statistici e operativi, ed è l'entità in Zabbix che avvisa attivamente gli amministratori quando si verificano problemi in uno qualsiasi dei sistemi monitorati.

Il funzionamento di un server Zabbix di base è suddiviso in tre componenti distinti; essi sono: Zabbix server, frontend e archiviazione del database.

Tutte le informazioni di configurazione di Zabbix sono memorizzate nel database, con cui interagiscono sia il server sia il frontend. Ad esempio, quando si crea un nuovo item usando il frontend (o l'API), questo viene aggiunto alla tabella items nel database. Poi, circa una volta al minuto, Zabbix server interrogherà la tabella items per ottenere un elenco degli item attivi, che viene quindi memorizzato in una cache all'interno di Zabbix server. Per questo motivo possono essere necessari fino a due minuti affinché eventuali modifiche apportate nel frontend di Zabbix vengano visualizzate nella sezione dei dati più recenti.

Vedi anche:

- Configurazione ad alta disponibilità per Zabbix server
- Parametri di configurazione: Zabbix server

Esecuzione del server Se installato come pacchetto

Zabbix server viene eseguito come processo daemon. Il server può essere avviato eseguendo:

```
systemctl start zabbix-server
```

Questo funzionerà sulla maggior parte dei sistemi GNU/Linux. Su altri sistemi potrebbe essere necessario eseguire:

```
/etc/init.d/zabbix-server start
```

Allo stesso modo, per arrestare/riavviare/visualizzare lo stato, usare i seguenti comandi:

```
systemctl stop zabbix-server
systemctl restart zabbix-server
systemctl status zabbix-server
```

Avvio manuale

Se quanto sopra non funziona, devi avviarlo manualmente. Trova il percorso del binario `zabbix_server` ed esegui:

```
zabbix_server
```

Puoi usare i seguenti parametri da riga di comando con Zabbix server:

Parameter	Description
`-c --config <config-file>`	Percorso del file di configurazione.
`-f --foreground`	Esegui Zabbix server in primo piano.
`-R --runtime-control <option>`	Esegui funzioni amministrative.
`-T --test-config`	Valida il file di configurazione ed esci.
`-h --help`	Mostra le informazioni di aiuto ed esci.
`-V --version`	Mostra il numero di versione.

Esempi di esecuzione di Zabbix server con parametri da riga di comando:

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

Controllo runtime

Opzioni di controllo runtime:

Opzione	Descrizione	Destinazione
`config_cache_reload`	Ricarica la cache di configurazione. Ignorata se la cache è attualmente in fase di caricamento.	
`history_cache_clear=<target>`	Cancella la cache della cronologia per l'item specificato dal relativo ID. Influisce su tutti i valori dell'item, ad eccezione del primo e dell'ultimo valore.	**target** - ID dell'item
`diaginfo[=<section>]`	Raccoglie informazioni diagnostiche nel file di log del server.	`historycache` - statistiche della cache della cronologia; `valuecache` - statistiche della cache dei valori; `preprocessing` - statistiche del gestore della pre-elaborazione; `alerting` - statistiche del gestore degli avvisi; `lld` - statistiche del gestore LLD; `locks` - elenco dei mutex (vuoto sui sistemi *BSD*); `connector` - statistiche per i connector con la coda più grande.
`ha_status`	Registra lo stato del cluster ad alta disponibilità (HA).	
`ha_remove_node=<target>`	Rimuove il nodo ad alta disponibilità (HA) specificato dal nome o dall'ID. I nodi attivi/in standby non possono essere rimossi.	**target** - nome o ID del nodo (ottenibile eseguendo `ha_status`).
`ha_set_failover_delay=<delay>`	Imposta l'intervallo di failover dell'alta disponibilità (HA). Sono supportati i suffissi temporali, ad esempio 10s, 1m.	

Opzione	Descrizione	Destinazione
`proxy_config_cache_reload [=<target>]`	Ricarica la cache di configurazione del proxy.	**target** - elenco di nomi di proxy separati da virgole. Se non viene specificata alcuna destinazione, ricarica la configurazione per tutti i proxy.
`secrets_reload`	Ricarica i segreti da Vault.	
`service_cache_reload`	Ricarica la cache del gestore dei servizi.	
`snmp_cache_reload`	Ricarica la cache SNMP, cancellando le proprietà del motore SNMP (tempo del motore, avvii del motore, ID del motore, credenziali) per tutti gli host. Utilizzare questa opzione per forzare la cancellazione globale della cache durante la risoluzione dei problemi SNMP.	
`housekeeper_execute`	Avvia la procedura di housekeeping. Ignorata se la procedura di housekeeping è attualmente in corso.	
`trigger_housekeeper_execute`	Avvia la procedura di housekeeping dei trigger. Ignorata se la procedura di housekeeping dei trigger è attualmente in corso.	
`log_level_increase [=<target>]`	Aumenta il livello di log; influisce su tutti i processi se la destinazione non è specificata. Non supportata sui sistemi *BSD*.	**process type** - tutti i processi del tipo specificato (ad esempio, `poller`). Vedere tutti i tipi di processo del server. **process type,N** - tipo e numero del processo (ad esempio, `poller,3`). **pid** - identificatore del processo (1 a 65535). Per valori maggiori, specificare la destinazione nel formato 'process type,N'.
`log_level_decrease [=<target>]`	Diminuisce il livello di log; influisce su tutti i processi se la destinazione non è specificata. Non supportata sui sistemi *BSD*.	
`prof_enable [=<target>]`	Abilita la profilazione. Influisce su tutti i processi se la destinazione non è specificata. La profilazione abilitata fornisce dettagli su tutti i rwlock/mutex in base al nome della funzione.	**process type** - tutti i processi del tipo specificato (ad esempio, `history syncer`) Tipi di processo supportati come destinazioni della profilazione: `alerter`, `alert manager`, `availability manager`, `configuration syncer`, `discovery manager`, `escalator`, `history poller`, `history syncer`, `housekeeper`, `http poller`, `icmp pinger`, `ipmi manager`, `ipmi poller`, `java poller`, `lld manager`, `lld worker`, `odbc poller`, `poller`, `preprocessing manager`, `proxy poller`, `self-monitoring`, `service manager`, `snmp trapper`, `task manager`, `timer`, `trapper`, `unreachable poller`, `vmware collector`. **process type,N** - tipo e numero del processo (ad esempio, `history syncer,1`). **pid** - identificatore del processo (1 a 65535). Per valori maggiori, specificare la destinazione nel formato 'process type,N'. **scope** - è possibile utilizzare `rwlock`, `mutex`, `processing` con il tipo e il numero del processo (ad esempio, `history syncer,1,processing`) oppure con tutti i processi del tipo (ad esempio, `history syncer,rwlock`).

Opzione	Descrizione	Destinazione
`prof_disable[=<destinazione>]`	Disabilita la profilazione. Influisce su tutti i processi se la destinazione non è specificata.	**process type** - tutti i processi del tipo specificato (ad esempio, `history syncer`). Tipi di processo supportati come destinazioni della profilazione: vedere `prof_enable`. **process type,N** - tipo e numero del processo (ad esempio, `history syncer,1`). **pid** - identificatore del processo (1 a 65535). Per valori maggiori, specificare la destinazione nel formato 'process type,N'.

Quando un tipo di processo è specificato come destinazione per il controllo runtime, il comando si applica solo ai processi in esecuzione di quel tipo. Se al momento non è in esecuzione alcun processo del tipo specificato, il comando restituisce un errore:

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

Esempio di utilizzo del controllo runtime per ricaricare la cache di configurazione del server:

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

Esempi di utilizzo del controllo runtime per ricaricare la configurazione del proxy:

```
### Ricarica la configurazione di tutti i proxy:
zabbix_server -R proxy_config_cache_reload

### Ricarica la configurazione di Proxy1 e Proxy2:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Esempio di utilizzo del controllo runtime per svuotare la cache della cronologia di un item:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R history_cache_clear=42243
```

Esempi di utilizzo del controllo runtime per raccogliere informazioni diagnostiche:

```
### Raccogli tutte le informazioni diagnostiche disponibili nel file di log del server:
zabbix_server -R diaginfo

### Raccogli le statistiche della cache della cronologia nel file di log del server:
zabbix_server -R diaginfo=historycache
```

Esempio di utilizzo del controllo runtime per ricaricare la cache SNMP:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Quando un'interfaccia SNMPv3 viene aggiornata tramite la Zabbix UI, Zabbix ricaricherà automaticamente, nella maggior parte dei casi, le nuove credenziali SNMPv3 per tale interfaccia; usa `-R snmp_cache_reload` solo se il polling continua a fallire dopo la modifica delle credenziali (ad esempio, a causa di incoerenze engineBoots/engineID o di dispositivi non conformi agli RFC), oppure quando devi forzare una pulizia globale della cache SNMP per la risoluzione dei problemi.

Esempio di utilizzo del controllo runtime per avviare l'esecuzione di housekeeper:

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

Esempi di utilizzo del controllo runtime per modificare il livello di log:

```
### Aumenta il livello di log di tutti i processi:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Aumenta il livello di log del secondo processo poller:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Aumenta il livello di log del processo con PID 1234:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234
```

```
### Diminuisce il livello di log di tutti i processi http poller:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Esempio di impostazione del ritardo di failover HA al minimo di 10 secondi:

```
zabbix_server -R ha_set_failover_delay=10s
```

Processo utente

Zabbix server è progettato per essere eseguito come utente non root. Verrà eseguito come l'utente non root con cui viene avviato. Quindi puoi eseguire server come qualsiasi utente non root senza problemi.

Se provi a eseguirlo come 'root', passerà a un utente 'zabbix' hardcoded, che deve essere presente nel tuo sistema. Puoi eseguire server come 'root' solo se modifichi di conseguenza il parametro 'AllowRoot' nel file di configurazione del server.

Se Zabbix server e agent vengono eseguiti sulla stessa macchina, è consigliato usare un utente diverso per l'esecuzione del server rispetto a quello usato per l'esecuzione dell'agent. Altrimenti, se entrambi vengono eseguiti come lo stesso utente, l'agent può accedere al file di configurazione del server e qualsiasi utente con livello Admin in Zabbix può facilmente recuperare, ad esempio, la password del database.

File di configurazione

Il percorso predefinito del file di configurazione del server dipende dal sistema operativo e dal metodo di installazione di Zabbix:

- Per l'installazione dai sorgenti: `/usr/local/etc/zabbix_server.conf`, oppure come impostato dalle variabili in fase di compilazione `--sysconfdir` o `--prefix`.
- Per l'installazione dai pacchetti e dall'appliance: `/etc/zabbix/zabbix_server.conf`.

Per informazioni dettagliate sulla configurazione di `zabbix_server`, vedere le opzioni del file di configurazione.

Script di avvio

Gli script vengono utilizzati per avviare/arrestare automaticamente i processi di Zabbix durante l'avvio/arresto del sistema. Gli script si trovano nella directory misc/init.d.

Tipi di processi e thread del server

- `agent poller` - processo poller asincrono per i controlli passivi con un thread worker;
- `alert manager` - gestore della coda degli avvisi;
- `alert syncer` - writer del database degli avvisi;
- `alerter` - processo per l'invio delle notifiche;
- `availability manager` - processo per l'aggiornamento della disponibilità degli host;
- `browser poller` - poller per i controlli degli item browser;
- `configuration syncer` - processo per la gestione della cache in memoria dei dati di configurazione;
- `configuration syncer worker` - processo per la risoluzione e la sincronizzazione dei valori delle macro utente nei nomi degli item;
- `connector manager` - processo manager per i connector;
- `connector worker` - processo per la gestione delle richieste del connector manager;
- `discovery manager` - processo manager per la discovery dei dispositivi;
- `discovery worker` - processo per la gestione delle attività di discovery del discovery manager;
- `escalator` - processo per l'escalation delle azioni;
- `ha manager` - processo per la gestione dell'alta disponibilità;
- `history poller` - processo per la gestione dei controlli calcolati che richiedono una connessione al database;
- `history syncer` - writer del database della cronologia;
- `housekeeper` - processo per la rimozione dei dati obsoleti (cronologia e trend degli item, sessioni utente, eventi ecc.), nonché dei dati lasciati dagli oggetti eliminati;
- `http agent poller` - processo poller asincrono per i controlli HTTP con un thread worker;
- `http poller` - poller per il monitoraggio web;
- `icmp pinger` - poller per i controlli icmpping;
- `internal poller` - poller per i controlli interni;
- `ipmi manager` - manager del poller IPMI;
- `ipmi poller` - poller per i controlli IPMI;
- `java poller` - poller per i controlli Java;
- `lld manager` - processo manager delle attività di low-level discovery;
- `lld worker` - processo worker delle attività di low-level discovery;
- `odbc poller` - poller per i controlli ODBC;
- `poller` - poller normale per i controlli passivi;
- `preprocessing manager` - manager delle attività di preprocessing con thread worker di preprocessing;

- `preprocessing worker` - thread per il preprocessing dei dati;
- `proxy poller` - poller per i proxy passivi;
- `proxy group manager` - manager del bilanciamento del carico e dell'alta disponibilità dei proxy;
- `report manager`- manager delle attività di generazione dei report pianificati;
- `report writer` - processo per la generazione dei report pianificati;
- `self-monitoring` - processo per la raccolta delle statistiche interne del server;
- `service manager` - processo per la gestione dei servizi mediante la ricezione delle informazioni sui problemi, sui tag dei problemi e sul ripristino dei problemi da history syncer, task manager e alert manager;
- `snmp poller` - processo poller asincrono per i controlli SNMP con un thread worker (solo item `walk[OID]` e `get[OID]`);
- `snmp trapper` - trapper per i trap SNMP;
- `supervisor` - coordina l'avvio e l'arresto dei processi e dei thread del server e ne monitora la disponibilità;
- `task manager` - processo per l'esecuzione remota delle attività richieste da altri componenti (ad esempio, chiusura di un problema, riconoscimento di un problema, controllo immediato del valore di un item, funzionalità dei comandi remoti);
- `telemetry query poller` - poller per i controlli degli item di tipo query di telemetria;
- `timer` - timer per l'elaborazione delle manutenzioni;
- `trapper` - trapper per i controlli attivi, i trap e la comunicazione con i proxy;
- `trigger housekeeper` - processo per la rimozione dei problemi e degli eventi generati da trigger che sono stati successivamente eliminati;
- `unreachable poller` - poller per i dispositivi non raggiungibili;
- `vmware collector` - collector dei dati VMware responsabile della raccolta dei dati dai servizi VMware.

Il file di log del server può essere utilizzato per osservare questi tipi di processi.

Il file di log del server viene creato con autorizzazioni di lettura e scrittura esclusivamente per il proprietario del file. Inoltre, il file è leggibile dal gruppo proprietario. Tutte le altre autorizzazioni sono negate.

È possibile monitorare vari tipi di processi del server Zabbix utilizzando l'item interno `zabbix[process,<type>,<mode>,<state>]`.

Statistiche delle transazioni del history syncer

Il titolo del processo history syncer mostra statistiche dettagliate sulle transazioni del history syncer:

```
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 triggers":

- A - trigger elaborati a causa dei valori di history;
- B - trigger elaborati a causa dei timer.

I tempi, in `processed...in N (<timings>) sec`, sono:

- Tempo impiegato per scrivere i valori degli item nel database;
- Tempo impiegato per aggiornare i dati degli item (stato, errori, inventario dell'host, ecc.);
- Tempo impiegato per svuotare le trends nel database;
- Tempo impiegato per calcolare i trigger;
- Tempo impiegato per elaborare eventi e azioni.

Procedura di housekeeping

Il processo `housekeeper` rimuove periodicamente i dati obsoleti (storico degli item e trend, sessioni utente, eventi, ecc.), oltre ai dati residui lasciati dagli oggetti eliminati. Viene eseguito in cicli, con frequenza e limite di eliminazione per ciclo determinati da `HousekeepingFrequency` e `MaxHousekeeperDelete`. I dati non rimossi in un ciclo vengono trasferiti al successivo. L'housekeeping automatico può essere abilitato e configurato in *Administration* > *Housekeeping*.

Quando il processo `housekeeper` rimuove i dati residui lasciati dagli oggetti eliminati, si basa sui task presenti nella tabella `housekeeper`, che viene popolata ogni volta che un oggetto viene eliminato. Ad esempio, quando elimini un host, Zabbix elimina i relativi item, ma non il loro storico, i trend o i problemi. Al loro posto, i trigger del database popolano la tabella `housekeeper` con task composti dai seguenti campi:

- `housekeeperid` - ID del task
- `object` - tipo di oggetto (0 - item; 1 - trigger; 2 - service; 3 - host rilevato; 4 - service rilevato)
- `objectid` - ID dell'oggetto (aiuta il housekeeper a trovare i dati associati all'oggetto)

Ad esempio, l'eliminazione di un host con due item e un trigger popola la tabella `housekeeper` come segue:

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
+---------------+--------+----------+
|             1 |      1 |    28724 |
```

```
|              2 |       0 |    59396 |
|              3 |       0 |    59397 |
+----------------+---------+----------+
```

I trigger del database popolano la tabella `housekeeper` con task senza verificare la presenza di dati associati all'oggetto; tale verifica viene eseguita dal processo `housekeeper` quando rimuove i dati:

- Per gli item eliminati (incluse le regole LLD) - rimuove i relativi dati da tutte le tabelle di history e trends (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`). Rimuove anche i relativi eventi interni dalla tabella `problem`.
- Per i trigger eliminati - rimuove i relativi eventi dalle tabelle correlate agli eventi (`problem`, `event_symptom`, `events`; il database stesso rimuove anche gli eventi dalla tabella `event_recovery`). Inoltre notifica il processo `service manager` degli eventi rimossi.

Note:
Un processo separato `trigger housekeeper` gestisce un'attività più limitata: la rimozione di problemi ed eventi che non hanno un trigger sorgente noto. La frequenza di esecuzione è controllata da `ProblemHousekeepingFrequency`. Fino all'avvio della procedura di housekeeping dei trigger, i problemi creati da trigger successivamente eliminati potrebbero ancora generare problemi di service e assegnarli ai service. Se la tua configurazione prevede molte regole di calcolo dello stato dei service basate su trigger rilevati/non rilevati frequentemente, considera di aumentare la frequenza della procedura di housekeeping regolando il parametro di configurazione del server `ProblemHousekeepingFrequency`.

- Per i service eliminati - rimuove i relativi problemi dalla tabella `problem`; gli eventi dalla tabella `events` vengono rimossi solo dopo il periodo di conservazione dei dati configurato.
- Per gli host/service rilevati eliminati (network discovery) - rimuove i relativi eventi dalla tabella `events`.

Il processo `housekeeper` rimuove solo gli eventi non associati a problemi. Ad esempio, un evento obsoleto di problema/ripristino non verrà rimosso se è associato a un problema aperto. Quando `housekeeper` rimuove entità obsolete, rimuove prima i problemi e poi gli eventi.

Note:
Le tabelle che usano la modalità `partition` (tabelle partizionate di TimescaleDB) vengono ignorate; vengono elaborate solo le tabelle che usano la modalità `regular`.

Piattaforme supportate A causa dei requisiti di sicurezza e della natura mission-critical del funzionamento del server, UNIX è l'unico sistema operativo in grado di garantire in modo costante le prestazioni, la tolleranza ai guasti e la resilienza necessarie. Zabbix funziona sulle versioni leader di mercato.

Il server Zabbix è testato sulle seguenti piattaforme:

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server

Note:
Zabbix potrebbe funzionare anche su altri sistemi operativi simili a Unix.

Locale Si noti che il server richiede una locale UTF-8 affinché alcuni elementi testuali possano essere interpretati correttamente. La maggior parte dei moderni sistemi Unix-like ha una locale UTF-8 impostata come predefinita; tuttavia, in alcuni sistemi potrebbe essere necessario configurarla esplicitamente.

Configurazione ad alta disponibilità per i server Zabbix

Panoramica

L'alta disponibilità (HA) è in genere richiesta nelle infrastrutture critiche che non possono permettersi praticamente alcun downtime. Quindi, per qualsiasi servizio che possa guastarsi, deve essere previsto un'opzione di failover per subentrare nel caso in cui il servizio corrente si guasti.

Zabbix offre una soluzione di alta disponibilità **nativa** facile da configurare e che non richiede alcuna esperienza pregressa in ambito HA. La HA nativa di Zabbix può essere utile come ulteriore livello di protezione contro guasti software/hardware di Zabbix server oppure per ridurre il downtime dovuto alla manutenzione.

Nella modalità di alta disponibilità di Zabbix, più Zabbix server vengono eseguiti come nodi in un cluster. Mentre uno Zabbix server nel cluster è attivo, gli altri sono in standby, pronti a subentrare se necessario.



Il passaggio a Zabbix HA non è vincolante. È possibile tornare in qualsiasi momento al funzionamento autonomo.

Vedi anche: Dettagli di implementazione

Abilitazione dell'alta disponibilità

Avvio del server Zabbix come nodo del cluster

Per avviare un server Zabbix come nodo del cluster, sono necessari due parametri nella configurazione del server:

- Il parametro `HANodeName` deve essere specificato per ogni server Zabbix che sarà un nodo del cluster HA.

Si tratta di un identificatore univoco del nodo (ad esempio, `zabbix-node-01`) con cui il server verrà indicato nelle configurazioni di agent e proxy. Se non si specifica `HANodeName`, il server verrà avviato in modalità standalone.

- Il parametro `NodeAddress` deve essere specificato per ogni nodo.

Il parametro `NodeAddress` (`address:port`) verrà utilizzato dal frontend Zabbix per connettersi al nodo server attivo. `NodeAddress` deve corrispondere all'indirizzo IP o al nome FQDN del rispettivo server Zabbix.

Riavviare tutti i server Zabbix dopo aver apportato modifiche ai file di configurazione. Verranno ora avviati come nodi del cluster. Il nuovo stato dei server può essere visualizzato in *Report* > *Informazioni di sistema* e anche eseguendo:

```
zabbix_server -R ha_status
```

Questo comando runtime registrerà lo stato corrente del cluster HA nel log del server Zabbix (e nello 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
```

Preparazione del frontend

Assicurarsi che `address:port` del server Zabbix non sia definito nel file di configurazione del frontend di Zabbix (`zabbix.conf.php`).

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

Il frontend di Zabbix rileverà automaticamente il nodo attivo leggendo le impostazioni dalla tabella dei nodi nel database di Zabbix. L'indirizzo del nodo attivo verrà utilizzato come indirizzo del server Zabbix.

Configurazione del proxy

I nodi del cluster HA (server) devono essere elencati nella configurazione di un proxy Zabbix passivo o attivo.

Per un proxy passivo, i nomi dei nodi devono essere elencati nel parametro Server parameter del proxy, separati da una **virgola**.

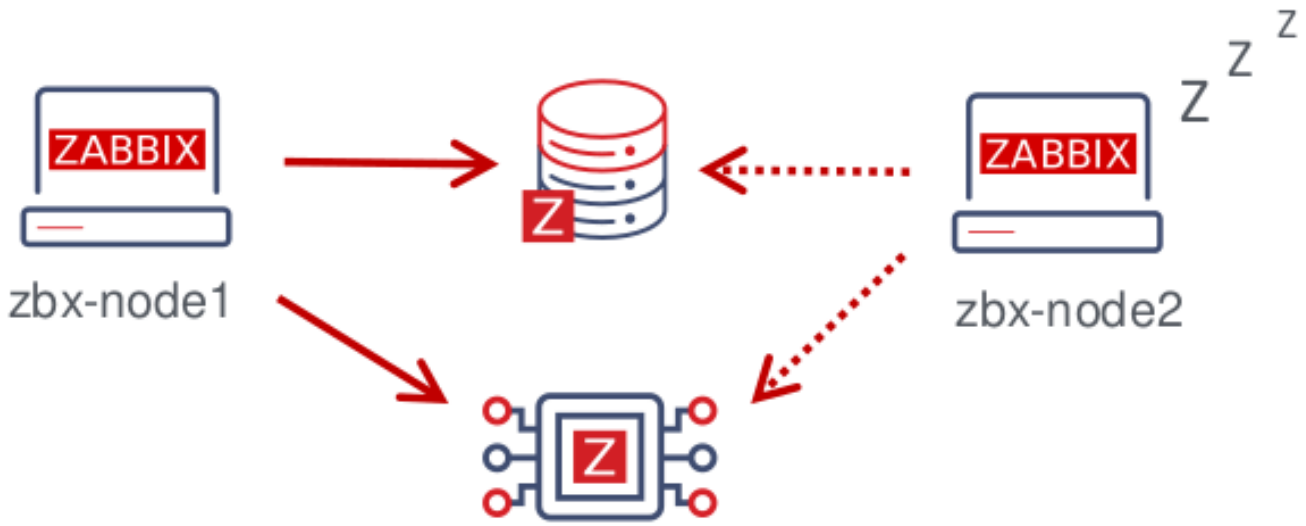
```
Server=zabbix-node-01,zabbix-node-02
```

Per un proxy attivo, i nomi dei nodi devono essere elencati nel parametro Server parameter del proxy, separati da un **punto e virgola**.

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

Configurazione dell'agent

I nodi del cluster HA (server) devono essere elencati nella configurazione di Zabbix agent o Zabbix agent 2.



Per abilitare i controlli passivi, i nomi dei nodi devono essere elencati nel parametro Server parameter, separati da una **virgola**.

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

Per abilitare i controlli attivi, i nomi dei nodi devono essere elencati nel parametro ServerActive parameter. Si noti che per i controlli attivi i nodi devono essere separati da una virgola rispetto a eventuali altri server, mentre i nodi stessi devono essere separati da un **punto e virgola**, ad esempio:

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

Failover al nodo di standby

Zabbix eseguirà automaticamente il failover su un altro nodo se il nodo attivo si arresta. Perché il failover avvenga, deve esserci almeno un nodo in stato di standby.

Quanto sarà rapido il failover? Tutti i nodi aggiornano il proprio ultimo tempo di accesso (e lo stato, se cambia) ogni 5 secondi. Quindi:

- Se il nodo attivo si arresta e riesce a segnalare il proprio stato come "stopped", un altro nodo subentrerà entro **5 secondi**.
- Se il nodo attivo si arresta/diventa non disponibile senza riuscire ad aggiornare il proprio stato, i nodi di standby attenderanno il **ritardo di failover** + 5 secondi prima di subentrare.

Il ritardo di failover è configurabile, con intervallo supportato tra 10 secondi e 15 minuti (un minuto per impostazione predefinita). Per modificare il ritardo di failover, puoi eseguire:

```
zabbix_server -R ha_set_failover_delay=5m
```

Gestione del cluster HA

Lo stato attuale del cluster HA può essere gestito utilizzando le opzioni dedicate di controllo runtime:

- `ha_status` - registra lo stato del cluster HA nel log del server Zabbix (e nello stdout);
- `ha_remove_node=target` - rimuove un nodo HA identificato dal relativo `<target>` - nome o ID del nodo (il nome/l'ID può essere ottenuto dall'output dell'esecuzione di `ha_status`), ad esempio:

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

I nodi attivi/standby non possono essere rimossi.

- `ha_set_failover_delay=delay` - imposta il ritardo di failover HA (tra 10 secondi e 15 minuti; sono supportati i suffissi temporali (ad esempio, `30s`, `1m`)).

Lo stato del nodo può essere monitorato:

- In *Report* > *Informazioni di sistema*.
- Nel widget del dashboard *Informazioni di sistema*.
- Utilizzando l'opzione di controllo runtime `ha_status` del server (vedere sopra).

L'item interno `zabbix[cluster,discovery,nodes]` può essere utilizzato per il rilevamento dei nodi, poiché restituisce un JSON con le informazioni sui nodi ad alta disponibilità.

Disabilitazione del cluster HA

Per disabilitare un cluster ad alta disponibilità:

1. Creare copie di backup dei file di configurazione.
2. Arrestare i nodi in standby.
3. Rimuovere il parametro HANodeName dal server primario attivo.
4. Riavviare il server primario (si avvierà in modalità standalone).

Aggiornamento del cluster HA

Per eseguire un aggiornamento di versione principale per i nodi HA:

1. Arrestare tutti i nodi.
2. Creare un backup completo del database.
3. Se il database usa la replica, assicurarsi che tutti i nodi siano sincronizzati e non presentino problemi. Non eseguire l'aggiornamento se la replica è interrotta.
4. Selezionare un singolo nodo che eseguirà l'aggiornamento del database, modificare la sua configurazione in modalità standalone commentando `HANodeName` e aggiornarlo.
5. Assicurarsi che l'aggiornamento del database sia completamente terminato (*System information* deve mostrare che Zabbix server è in esecuzione).
6. Riavviare il nodo in modalità HA.
7. Aggiornare e avviare il resto dei nodi (non è necessario modificarli in modalità standalone, poiché a questo punto il database è già stato aggiornato).

In un aggiornamento di versione minore è sufficiente aggiornare il primo nodo, assicurarsi che sia stato aggiornato e sia in esecuzione, quindi avviare l'aggiornamento sul nodo successivo.

Dettagli di implementazione

Il cluster ad alta disponibilità (HA) è una soluzione opzionale ed è supportato per Zabbix server. La soluzione HA nativa è progettata per essere semplice da usare, funziona tra siti diversi e non ha requisiti specifici per i database riconosciuti da Zabbix. Gli utenti sono liberi di usare la soluzione HA nativa di Zabbix oppure una soluzione HA di terze parti, a seconda di ciò che si adatta meglio ai requisiti di alta disponibilità nel loro ambiente.

La soluzione è composta da più istanze o nodi `zabbix_server`. Ogni nodo:

- È configurato separatamente.
- Usa lo stesso database.
- Può avere diverse modalità: active, standby, unavailable, stopped.

Solo un nodo può essere active (in esecuzione) alla volta. Un nodo standby esegue solo un processo - l'HA manager. Un nodo standby non esegue raccolta dati, elaborazione o altre attività normali del server; non ascolta sulle porte; ha un numero minimo di connessioni al database.

Sia i nodi active sia quelli standby aggiornano il proprio ultimo tempo di accesso ogni 5 secondi. Ogni nodo standby monitora l'ultimo tempo di accesso del nodo active. Se l'ultimo tempo di accesso del nodo active supera i secondi di 'failover delay', il nodo standby passa a essere il nodo active e assegna lo stato `unavailable` al nodo precedentemente active.

Il nodo active monitora la propria connettività al database - se viene persa per più di `failover delay`−5 secondi, deve interrompere tutta l'elaborazione e passare alla modalità standby. Il nodo active monitora anche lo stato dei nodi standby - se l'ultimo tempo di accesso di un nodo standby supera i secondi di 'failover delay', al nodo standby viene assegnato lo stato `unavailable`.

I nodi sono progettati per essere compatibili tra le versioni minori di Zabbix.

Parametri di configurazione: Zabbix server

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del server Zabbix.

Specificare i valori dei parametri nel file di configurazione del server Zabbix (`zabbix_server.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
AlertScriptsPath	Specificare il percorso degli script di avviso.
AllowRoot	Consentire l'esecuzione del server come `root`.
AllowSoftwareUpdateCheck	Consentire la ricezione di informazioni sugli aggiornamenti software da `zabbix.com`.
AllowUnsupportedDBVersions	Consentire al server di funzionare con versioni del database non supportate.
BridgeAdapterURL	Impostare l'URL HTTP(S) dell'adattatore bridge di Zabbix.
CacheSize	Impostare la dimensione della cache di configurazione (in byte).
CacheUpdateFrequency	Impostare la frequenza di aggiornamento della cache di configurazione (in secondi).
DBHost	Specificare l'host del database (o la directory del socket).
DBName	Specificare il nome del database. **Obbligatorio**.
DBPassword	Specificare la password del database.
DBPort	Specificare la porta dell'host del database.
DBSchema	Specificare il nome dello schema del database. Utilizzato per PostgreSQL.
DBSocket	Specificare il percorso del file socket MySQL.
DBUser	Specificare l'utente del database.
DBTLSConnect	Specificare un valore per imporre una connessione TLS al database.
DBTLSCAFile	Specificare il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del database.
DBTLSCertFile	Specificare il percorso completo del file contenente il certificato del server Zabbix per l'autenticazione al database.
DBTLSKeyFile	Specificare il percorso completo del file contenente la chiave privata per l'autenticazione al database.
DBTLSCipher	Specificare l'elenco dei cifrari di crittografia consentiti dal server Zabbix per i protocolli TLS fino a TLS v1.2. Supportato solo per MySQL.
DBTLSCipher13	Specificare l'elenco delle suite di cifratura consentite dal server Zabbix per il protocollo TLS v1.3. Supportato solo per MySQL, a partire dalla versione 8.0.16.
DebugLevel	Impostare il livello di debug.
EnableGlobalScripts	Abilitare gli script globali sul server Zabbix.
EnableMobileDevices	Abilitare il supporto dei dispositivi mobili sul server Zabbix.
ExportDir	Specificare una directory per abilitare l'esportazione in tempo reale di eventi, dati storici e trend.
ExportFileSize	Impostare la dimensione massima del file di esportazione (in byte).
ExportType	Specificare i tipi di entità per l'esportazione in tempo reale (eventi, dati storici, trend).
ExternalScripts	Specificare il percorso degli script esterni.
Fping6Location	Specificare il percorso di fping6.
FpingLocation	Specificare il percorso di fping.
FrontendAllowedIP	Specificare gli indirizzi IP o gli intervalli CIDR autorizzati a connettersi dal frontend.
HANodeName	Specificare il nome del nodo in un cluster ad alta disponibilità.
HistoryCacheSize	Impostare la dimensione della cache dei dati storici.
HistoryIndexCacheSize	Impostare la dimensione della cache degli indici dei dati storici.
HistoryProvider	Impostare il provider di archiviazione dei dati storici (Elasticsearch, ClickHouse o SQL).
HistoryStorageDateIndex	Questo parametro è **deprecato** (utilizzare invece HistoryProvider). Abilitare l'archiviazione dei valori storici in indici diversi in base alla data.
HistoryStorageURL	Questo parametro è **deprecato** (utilizzare invece HistoryProvider). Specificare l'URL dell'archiviazione dei dati storici.
HistoryStorageTypes	Questo parametro è **deprecato** (utilizzare invece HistoryProvider). Specificare i tipi di valore da inviare all'archiviazione dei dati storici.
HousekeepingFrequency	Impostare la frequenza della procedura di housekeeping (in ore).
Include	Specificare una directory o singoli file da includere nel file di configurazione.
JavaGateway	Specificare l'indirizzo IP (o il nome host) del Java gateway Zabbix.
JavaGatewayPort	Specificare la porta del Java gateway Zabbix.
ListenBacklog	Impostare il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Specificare l'indirizzo IP di ascolto del trapper.
ListenPort	Specificare la porta di ascolto del trapper.
LoadModule	Specificare il modulo da caricare all'avvio del server.
LoadModulePath	Specificare il percorso completo della posizione dei moduli del server.

Parametro	Descrizione
LogFile	Specificare il file di log del server.
LogFileSize	Impostare la dimensione massima del file di log.
LogSlowQueries	Impostare la durata massima che una query del database può raggiungere prima di essere registrata (in millisecondi).
LogType	Specificare il tipo di output del log.
MaxConcurrentChecksPerPoller	Impostare il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun poller HTTP agent, agent o SNMP.
MaxHousekeeperDelete	Impostare il numero massimo di righe da eliminare per attività durante un ciclo di housekeeping.
NodeAddress	Specificare l'IP o il nome host, con porta opzionale, per sostituire il modo in cui il frontend si connette al server.
PidFile	Specificare il file PID.
ProblemHousekeepingFrequency	Impostare la frequenza di eliminazione dei problemi relativi ai trigger eliminati.
ProxyConfigFrequency	Impostare la frequenza con cui il server invia i dati di configurazione al proxy Zabbix.
ProxyDataFrequency	Impostare la frequenza con cui il server richiede i dati storici al proxy Zabbix.
ServiceManagerSyncFrequency	Impostare la frequenza di sincronizzazione della configurazione di un gestore dei servizi.
SMSDevices	Specificare i file dei modem che possono essere utilizzati dal server Zabbix.
SNMPTrapperFile	Specificare il file temporaneo utilizzato per trasferire i dati dal demone trap SNMP al server.
SocketDir	Specificare la directory in cui archiviare i socket IPC utilizzati dai servizi interni di Zabbix.
SourceIP	Specificare l'indirizzo IP di origine.
SSHKeyLocation	Specificare la posizione delle chiavi pubbliche e private per i controlli e le azioni SSH.
SSLCertLocation	Specificare la posizione dei file dei certificati del client SSL per l'autenticazione del client.
SSLKeyLocation	Specificare la posizione dei file delle chiavi private SSL per l'autenticazione del client.
SSLCALocation	Specificare la posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato del server SSL.
StartAgentPollers	Impostare il numero di istanze pre-fork dei poller asincroni dell'agent Zabbix.
StartAlerters	Impostare il numero di istanze pre-fork degli alerter.
StartBrowserPollers	Impostare il numero di istanze pre-fork dei poller degli item browser.
StartConnectors	Impostare il numero di istanze pre-fork dei worker connector.
StartDBSyncers	Impostare il numero di istanze pre-fork dei sincronizzatori dei dati storici.
StartDiscoverers	Impostare il numero di istanze pre-fork dei worker di discovery.
StartEscalators	Impostare il numero di istanze pre-fork degli escalator.
StartHistoryPollers	Impostare il numero di istanze pre-fork dei poller dei dati storici.
StartHTTPAgentPollers	Impostare il numero di istanze pre-fork dei poller asincroni HTTP agent.
StartHTTPPollers	Impostare il numero di istanze pre-fork dei poller HTTP.
StartIPMIPollers	Impostare il numero di istanze pre-fork dei poller IPMI.
StartJavaPollers	Impostare il numero di istanze pre-fork dei poller Java.
StartLLDProcessors	Impostare il numero di istanze pre-fork dei worker di low-level discovery (LLD).
StartODBCPollers	Impostare il numero di istanze pre-fork dei poller ODBC.
StartPingers	Impostare il numero di istanze pre-fork dei pinger ICMP.
StartPollersUnreachable	Impostare il numero di istanze pre-fork dei poller per gli host non raggiungibili (inclusi IPMI e Java).
StartPollers	Impostare il numero di istanze pre-fork dei poller.
StartPreprocessors	Impostare il numero di istanze pre-avviate dei worker di preprocessing.
StartProxyPollers	Impostare il numero di istanze pre-fork dei poller per i proxy passivi.
StartReportWriters	Impostare il numero di istanze pre-fork dei writer dei report.
StartSNMPPollers	Impostare il numero di istanze pre-fork dei poller SNMP asincroni.
StartSNMPTrapper	Impostare su "1" per avviare un processo trapper SNMP.
StartTelemetryQueryPollers	Impostare il numero di istanze pre-fork dei poller asincroni delle query di Telemetry.
StartTimers	Impostare il numero di istanze pre-fork dei timer.
StartTrappers	Impostare il numero di istanze pre-fork dei trapper.
StartVMwareCollectors	Impostare il numero di istanze pre-fork dei collector VMware.
StatsAllowedIP	Specificare gli indirizzi IP o i nomi DNS delle istanze Zabbix esterne autorizzate a richiedere le statistiche.
TelemetryProvider	Impostare l'origine dei dati di telemetria.
Timeout	Specificare per quanto tempo attendere (in secondi) la connessione e lo scambio di dati con il proxy Zabbix, l'agent, il web service e per i controlli SNMP (ad eccezione degli item SNMP `walk[OID]` e `get[OID]`).
TLSCAFile	Specificare il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato peer, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCertFile	Specificare il percorso completo del file contenente il certificato del server o la catena di certificati, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.

Parametro	Descrizione
TLSCipherAll	Specificare la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituire i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherAll13	Specificare la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituire i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert	Specificare la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituire i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato.
TLSCipherCert13	Specificare la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituire i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK	Specificare la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituire i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su PSK.
TLSCipherPSK13	Specificare la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituire i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su PSK.
TLSCRLFile	Specificare il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti Zabbix.
TLSFrontendAccept	Specificare il livello di crittografia per le connessioni in ingresso dal frontend.
TLSFrontendCertIssuer	Specificare l'emittente consentito del certificato del frontend.
TLSFrontendCertSubject	Specificare il subject consentito del certificato del frontend.
TLSKeyFile	Specificare il percorso completo del file contenente la chiave privata del server, utilizzata per le comunicazioni crittografate tra i componenti Zabbix.
TLSListen	Richiedere connessioni esclusivamente TLS sul socket del trapper.
TmpDir	Specificare la directory temporanea.
TrapperTimeout	Specificare il timeout del trapper in secondi per: - il recupero dei dati storici dal proxy Zabbix; - l'invio dei dati di configurazione al proxy Zabbix; - l'esecuzione di script globali o di comandi remoti sul server Zabbix.
TrendCacheSize	Impostare la dimensione della cache dei trend.
TrendFunctionCacheSize	Impostare la dimensione della cache delle funzioni dei trend.
UnavailableDelay	Impostare la frequenza di verifica della disponibilità dell'host durante il periodo di indisponibilità.
UnreachableDelay	Impostare la frequenza di verifica della disponibilità dell'host durante il periodo di irraggiungibilità.
UnreachablePeriod	Impostare dopo quanti secondi di irraggiungibilità considerare l'host non disponibile.
User	Abbandonare i privilegi passando a un utente specifico ed esistente nel sistema.
ValueCacheSize	Impostare la dimensione della cache dei valori storici.
Vault	Specificare il provider vault.
VaultAppRoleID	Specificare l'ID AppRole del vault HashiCorp.
VaultAppSecretID	Specificare l'ID segreto AppRole del vault HashiCorp.
VaultDBPath	Specificare la posizione da cui recuperare le credenziali del database tramite chiavi.
VaultPrefix	Specificare un prefisso personalizzato per il percorso o la query del vault.
VaultTLSCertFile	Specificare il nome del file del certificato SSL utilizzato per l'autenticazione del client.
VaultTLSKeyFile	Specificare il nome del file della chiave privata SSL utilizzata per l'autenticazione del client.
VaultToken	Specificare il token di autenticazione del vault HashiCorp.
VaultURL	Specificare l'URL del server vault.
VMwareCacheSize	Impostare la dimensione della memoria condivisa per l'archiviazione dei dati VMware.
VMwareFrequency	Impostare la frequenza di raccolta dei dati da un singolo servizio VMware, in secondi.
VMwarePerfFrequency	Impostare la frequenza di recupero delle statistiche dei contatori delle prestazioni da un singolo servizio VMware, in secondi.
VMwareTimeout	Impostare il numero massimo di secondi per cui un `vmware collector` attenderà una risposta dal servizio VMware.
WebDriverURL	Specificare l'URL dell'interfaccia WebDriver.
WebServiceURL	Specificare l'URL del web service nel formato `http[s]://host:port/report`.

Tenere presente quanto segue:

- I valori predefiniti riflettono i valori predefiniti del demone, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta i file di configurazione solo con codifica UTF-8 senza BOM.
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Warning:
Un numero eccessivo di processi può esaurire il limite di descrittori di file per processo del gestore di preprocessing.

Tali processi includono la raccolta dei dati tramite poller, poller per host non raggiungibili, poller ODBC, poller HTTP, poller Java, pinger, trapper e proxypoller, insieme al gestore IPMI, al trapper SNMP, ai worker di preprocessing e ai worker di discovery. L'esaurimento del limite dei descrittori di file farà **arrestare** il server Zabbix, in genere poco dopo l'avvio, ma talvolta dopo un periodo più lungo.

Per evitare questi problemi, esaminare il file di configurazione del server Zabbix per ottimizzare il numero di controlli e processi simultanei. Inoltre, se necessario, assicurarsi che il limite dei descrittori di file sia sufficientemente elevato verificando e modificando i limiti del sistema.

Dettagli del parametro

AlertScriptsPath

Specifica il percorso degli script di avviso. Dipende dalla variabile di installazione in fase di compilazione `datadir`.

Predefinito: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Consente al server di essere eseguito come `root`. Se disabilitato e il server viene avviato da `root`, il server tenterà invece di passare all'utente `zabbix`. Non ha effetto se avviato da un utente normale.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

AllowSoftwareUpdateCheck

Consente alla UI di Zabbix di ricevere informazioni sugli aggiornamenti software da `zabbix.com`.

Predefinito: 1
 Valori: 0 - non consentire; 1 - consentire

AllowUnsupportedDBVersions

Consente al server di funzionare con versioni del database non supportate.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

BridgeAdapterURL

Imposta l'URL HTTP(S) dell'adattatore bridge di Zabbix.

Esempio:

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

Imposta la dimensione della cache di configurazione (in byte). La dimensione della memoria condivisa per memorizzare i dati di host, item e trigger.

Predefinito: 32M
 Intervallo: 128K-64G

CacheUpdateFrequency

Imposta la frequenza di aggiornamento della cache di configurazione (in secondi). Vedi anche le opzioni di runtime control.

Predefinito: 10
 Intervallo: 1-3600

DBHost

Specifica l'host del database (o la directory del socket).

Per MySQL:

- `localhost` o una stringa vuota utilizza il socket UNIX di default.

Per PostgreSQL:

- `localhost` viene risolto tramite DNS (in genere in 127.0.0.1).
- Una stringa vuota utilizza il socket UNIX di default.
- Un percorso (ad esempio, `/var/run/pgbouncer`) comporta l'utilizzo del socket UNIX in quel percorso.
- Un elenco separato da virgole può contenere più valori `host:port`. Ogni host viene provato in ordine finché non viene stabilita una connessione in lettura-scrittura.

Esempio: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]`

Predefinito: `localhost`

DBName

Specificare il nome del database.

Obbligatorio: **Sì**

DBPassword

Specifica la password del database. Commenta questa riga se non viene utilizzata alcuna password.

DBPort

Specifica la porta dell'host del database Zabbix (vedi DBHost).
 DBSocket e DBPort si escludono a vicenda nella configurazione del server. Specifica solo uno dei due, oppure lascia entrambi non definiti.

Predefinito per MySQL: 3306

Predefinito per PostgreSQL: 5432

Intervallo: 1024-65535

DBSchema

Specifica il nome dello schema del database. Utilizzato per PostgreSQL.

DBSocket

Specificare il percorso del file socket di MySQL.
 DBSocket e DBPort si escludono a vicenda nella configurazione del server. Specificarne solo uno, oppure lasciare entrambi non definiti.

DBUser

Specificare l'utente del database.

DBTLSConnect

Specifica un valore per imporre una connessione TLS al database: - *required* - connessione tramite TLS; - *verify_ca* - connessione tramite TLS e verifica del certificato; - *verify_full* - connessione tramite TLS, verifica del certificato e verifica che l'identità del database specificata da DBHost corrisponda al suo certificato.

Con MySQL, a partire dalla versione 5.7.11, e PostgreSQL sono supportati i seguenti valori: required, verify_ca, verify_full.
 Con MariaDB, a partire dalla versione 10.2.6, sono supportati i valori required e verify_full.
Per impostazione predefinita non è impostato alcun valore e il comportamento dipende dalla configurazione del database.

DBTLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del database.

Obbligatorio: no (sì, se DBTLSConnect è impostato su verify_ca o verify_full)

DBTLSCertFile

Specifica il percorso completo del file contenente il certificato del server Zabbix per l'autenticazione al database.

DBTLSKeyFile

Specifica il percorso completo del file contenente la chiave privata per l'autenticazione al database.

DBTLSCipher

Specifica l'elenco delle cifre di crittografia che il server Zabbix consente per i protocolli TLS fino a TLS v1.2. Supportato solo per MySQL.

DBTLSCipher13

Specifica l'elenco delle suite di cifratura che il server Zabbix consente per il protocollo TLS v1.3. Supportato solo per MySQL, a partire dalla versione 8.0.16.

DebugLevel

Imposta il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (genera molte informazioni); - *5* - debug esteso (genera ancora più informazioni).
 Vedi anche le opzioni di runtime control.

Predefinito: 3
 Intervallo: 0-5

EnableGlobalScripts

Abilita gli script globali sul server Zabbix.
 Nota: l'esecuzione degli script globali è abilitata per impostazione predefinita. Tuttavia, per le nuove installazioni a partire da Zabbix 7.0, EnableGlobalScripts è impostato esplicitamente su 0 (disabilitato).

Predefinito: 1
 Valori: 0 - disabilita; 1 - abilita

EnableMobileDevices

Abilita il supporto per i dispositivi mobili sul server Zabbix.

Quando questa opzione è abilitata, il server Zabbix elaborerà le richieste per aggiungere o rimuovere dispositivi mobili e invierà notifiche push a tali dispositivi.

Predefinito: 1
 Valori: 0 - disabilita; 1 - abilita

ExportDir

Specifica una directory per abilitare l'esportazione in tempo reale di eventi, history e trends.

ExportFileSize

Imposta la dimensione massima del file di esportazione (in byte). Utilizzato per la rotazione se `ExportDir` è impostato.

Predefinito: 1G
 Intervallo: 1M-1G

ExportType

Specifica i tipi di entità per l'esportazione in tempo reale (eventi, history, trends). Un elenco delimitato da virgole. Per impostazione predefinita, vengono esportati tutti i tipi. Valido solo se `ExportDir` è impostato.
 Nota che se `ExportType` è specificato, ma `ExportDir` non lo è, si tratta di un errore di configurazione e il server non si avvierà.

Esempio per l'esportazione di history e trends:

```
ExportType=history,trends
```

Esempio per la sola esportazione degli eventi:

```
ExportType=events
```

ExternalScripts

Specifica la posizione dello script esterno. Dipende dalla variabile di installazione definita in fase di compilazione `datadir`.

Predefinito: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Specifica la posizione di fping6. Assicurati che il binario fping6 abbia come proprietario root e che il flag SUID sia impostato. Lascia vuoto (`Fping6Location=`) se la tua utility fping è in grado di elaborare indirizzi IPv6.

Predefinito: `/usr/sbin/fping6`

FpingLocation

Specifica la posizione di fping. Assicurati che il binario fping abbia come proprietario root e che sia impostato il flag SUID.

Predefinito: `/usr/sbin/fping`

FrontendAllowedIP

Specifica gli indirizzi IP o gli intervalli CIDR autorizzati a connettersi dal frontend. Elenco separato da virgole. Per impostazione predefinita, tutte le connessioni sono accettate per le richieste del frontend. Se il supporto IPv6 è abilitato, allora `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4.

Esempio:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Specifica il nome del nodo in un cluster ad alta disponibilità. Quando è vuoto, il server funziona in modalità standalone e viene creato un nodo con nome vuoto.

HistoryCacheSize

Imposta la dimensione della cache della history (in byte). La dimensione della memoria condivisa per l'archiviazione dei dati di history.

Predefinito: 16M
 Intervallo: 128K-16G

HistoryIndexCacheSize

Imposta la dimensione della cache dell'indice della history (in byte). La dimensione della memoria condivisa per indicizzare i dati della history memorizzati nella cache della history. La dimensione della cache dell'indice richiede approssimativamente 100 byte per memorizzare un item.

Predefinito: 4M
 Intervallo: 128K-16G

HistoryProvider

Imposta il provider di archiviazione della history usando il formato `<provider>[;<options>]`, dove:

- `<provider>` - uno dei provider di archiviazione della history supportati: `sql` (predefinito), `elasticsearch` o `clickhouse`.
- `<options>` - coppie `key=value` separate da virgole. I valori possono essere racchiusi tra virgolette.

Opzioni per tutti i provider di archiviazione della history:

- `value_types` - imposta i tipi di valore da memorizzare (`uint`, `dbl`, `str`, `log`, `text`, `json`). Ognuno può essere usato una sola volta e solo per un singolo provider.
- `precache` - imposta 1 (predefinito) per precaricare in blocco i dati dal provider di archiviazione della history nella value cache. Questo può migliorare significativamente il riscaldamento della value cache dopo il riavvio.

Opzioni per Elasticsearch e ClickHouse:

- `url` - imposta l'URL HTTP o HTTPS usato per connettersi al provider di archiviazione della history.
- `log_slow_queries` - imposta la durata massima (in millisecondi) che una query al provider di archiviazione della history può impiegare prima di essere registrata nel log. Questa opzione diventa disponibile a partire da `DebugLevel=3`.
- `source_ip` - imposta l'indirizzo IP sorgente usato per le connessioni in uscita verso il provider di archiviazione della history.

Opzioni per Elasticsearch:

- `date_index` - imposta 1 per abilitare la memorizzazione dei valori di history in indici diversi in base alla data.

Opzioni per ClickHouse:

- `db` - imposta il nome del database.
- `username` - imposta il nome utente usato per accedere al database.
- `password` - imposta la password usata per accedere al database.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - imposta queste opzioni per configurare una connessione sicura (TLS/SSL) al database.

Esempio per ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Esempio per Elasticsearch:

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

Esempio per ClickHouse ed Elasticsearch:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

Questo parametro è deprecato (usa invece HistoryProvider).
 Abilita l'archiviazione dei valori storici in indici diversi in base alla data. Questo parametro viene utilizzato per la configurazione di Elasticsearch.

Predefinito: 0
 Valori: 0 - disabilitato; 1 - abilitato

HistoryStorageURL

Questo parametro è deprecato (utilizzare invece HistoryProvider).
 Specifica l'URL di archiviazione della history. Questo parametro viene utilizzato per la configurazione di Elasticsearch.

HistoryStorageTypes

Questo parametro è deprecato (utilizzare invece HistoryProvider).
 Specifica i tipi di valore da inviare all'archiviazione della cronologia. Un elenco delimitato da virgole. Questo parametro viene utilizzato per la configurazione di Elasticsearch.

Predefinito: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Imposta la frequenza della procedura di housekeeping (in ore). L'housekeeping rimuove le informazioni obsolete dal database.
Nota: per evitare che il processo di housekeeping venga sovraccaricato (ad esempio, quando i periodi di history e trend vengono notevolmente ridotti), in un singolo ciclo di housekeeping, per ogni item, vengono eliminate non più di 4 volte le ore di `HousekeepingFrequency` di informazioni obsolete. Quindi, se `HousekeepingFrequency` è 1, per ciclo verranno eliminate non più di 4 ore di informazioni obsolete (a partire dalla voce più vecchia).
Nota: per ridurre il carico all'avvio del server, l'housekeeping viene posticipato di 30 minuti dopo l'avvio del server. Quindi, se `HousekeepingFrequency` è 1, la

primissima procedura di housekeeping dopo l'avvio del server verrà eseguita dopo 30 minuti e successivamente si ripeterà con un intervallo di un'ora.
È possibile disabilitare l'housekeeping automatico impostando `HousekeepingFrequency` su `0`. In questo caso, la procedura di housekeeping può essere avviata solo tramite l'opzione di controllo runtime `housekeeper_execute` e il periodo di informazioni obsolete eliminato in un singolo ciclo di housekeeping è pari a 4 volte il periodo trascorso dall'ultimo ciclo di housekeeping, ma non inferiore a 4 ore e non superiore a 4 giorni.
Vedi anche le opzioni di runtime control e i dettagli sulla housekeeping procedure.

Predefinito: 1
 Intervallo: 0-24

Include

Specifica una directory o singoli file da includere nel file di configurazione.
 Durante l'installazione di Zabbix, questo parametro è impostato su `/usr/local/etc`, a meno che tu non lo modifichi durante la compilazione.

Regole generali per il parametro `Include`:

- Il percorso può essere relativo alla posizione del file `zabbix_server.conf`.
- Puoi specificare più parametri `Include` in un singolo file di configurazione.
- Zabbix elabora il parametro `Include` non appena lo legge, quindi considera di posizionarlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non si avvierà.

Regole per l'inclusione di directory o pattern di file:

- Non dare per scontato alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non puoi prevedere quale valore verrà utilizzato.
- Fai attenzione alle copie di backup che alcuni editor di testo creano automaticamente (ad esempio, modificando `include/my_specific.conf` potrebbe essere creato `include/my_specific.conf.BAK`), poiché entrambi i file verranno inclusi.

Per includere solo i file rilevanti da una directory, usa il carattere jolly (`*`) per la corrispondenza dei pattern.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Specificare l'indirizzo IP o il nome host di Zabbix Java gateway. Richiesto solo se i poller Java sono avviati.

JavaGatewayPort

Specifica la porta del Zabbix Java gateway.

Predefinito: 10052
 Intervallo: 1024-32767

ListenBacklog

Imposta il numero massimo di connessioni in sospeso nella coda TCP.
Il valore predefinito è una costante codificata, che dipende dal sistema.
Anche il valore massimo supportato dipende dal sistema; valori troppo elevati possono essere troncati silenziosamente al "massimo specificato dall'implementazione".

Predefinito: `SOMAXCONN`
 Intervallo: 0 - INT_MAX

ListenIP

Specifica l'indirizzo IP di ascolto del trapper. Un elenco delimitato da virgole.
 Il trapper ascolterà su tutte le interfacce di rete se questo parametro non viene specificato.

Predefinito: `0.0.0.0`

ListenPort

Specifica la porta di ascolto del trapper.

Predefinito: 10051
 Intervallo: 1024-32767

LoadModule

Specificare il modulo da caricare all'avvio del server. I moduli vengono utilizzati per estendere le funzionalità del server. Il modulo deve trovarsi nella directory specificata da `LoadModulePath` oppure il percorso deve precedere il nome del modulo. Se il percorso precedente è assoluto (inizia con `/`), `LoadModulePath` viene ignorato.
 Formati:

- LoadModule=<module.so>
- LoadModule=<path/module.so>
- LoadModule=</abs_path/module.so>

È possibile specificare più parametri LoadModule.

LoadModulePath

Specifica il percorso completo della posizione dei moduli del server. Il valore predefinito dipende dalle opzioni di compilazione.

LogFile

Specifica il file di log del server.

Obbligatorio: Sì, se LogType è impostato su file; altrimenti no

LogFileSize

Imposta la dimensione massima del file di log (in MB).
0 - disabilita la rotazione automatica dei log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file non riesce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024
 Obbligatorio: Sì, se LogType è impostato su file; altrimenti no

LogSlowQueries

Imposta la durata massima che una query del database può impiegare prima di essere registrata (in millisecondi).
0 - non registrare le query lente.
Questa opzione diventa disponibile a partire da DebugLevel=3.

Predefinito: 0
 Intervallo: 0-3600000

LogType

Specifica il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro LogFile; - *system* - scrive il log nel syslog; - *console* - scrive il log nell'output standard.

Predefinito: file

MaxConcurrentChecksPerPoller

Imposta il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun HTTP agent poller, agent poller o SNMP poller. Vedere StartHTTPAgentPollers, StartAgentPollers e StartSNMPPollers.

Predefinito: 1000
 Intervallo: 1-1000

MaxHousekeeperDelete

Imposta il numero massimo di righe da eliminare per task in un singolo ciclo di housekeeping dalle tabelle history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint e problem.
I task di housekeeping esistono nella tabella housekeeper nel formato: [housekeeperid], [object], [objectid].
 Questo parametro si applica solo all'eliminazione dei dati residui lasciati da item già eliminati.
 Se impostato a 0, non viene applicato alcun limite.
 In questo caso è necessario sapere bene cosa si sta facendo, per non sovraccaricare il database.
 Quando vengono eliminati molti item, il carico sul database aumenta, perché l'housekeeper dovrà rimuovere tutti i dati storici che questi item avevano.
 Ad esempio, se dobbiamo rimuovere solo 1 prototipo di item dal template, ma questo template è collegato a 50 host e per ogni host il prototipo viene espanso in 100 item reali, in totale devono essere rimossi 5000 item (1*50*100).
 Se per MaxHousekeeperDelete è impostato 500 (MaxHousekeeperDelete=500), il processo di housekeeper dovrà rimuovere fino a 2500000 valori (5000*500) per gli item eliminati dalle tabelle history e trends in un singolo ciclo.

Vedi anche i dettagli sulla procedura di housekeeping.

Predefinito: 5000
 Intervallo: 0-1000000

NodeAddress

Specifica un indirizzo IP o un nome host con porta opzionale per sostituire il modo in cui il frontend si connette al server.
Formato: <address>[:<port>]

Se IP o nome host non è impostato, verrà usato il valore di ListenIP. Se ListenIP non è impostato, verrà usato il valore localhost.
Se la porta non è impostata, verrà usato il valore di ListenPort. Se ListenPort non è impostato, verrà usato il valore 10051.

Questa opzione può essere sostituita dall'indirizzo specificato nella configurazione del frontend.

Vedi anche: parametro HANodeName; Abilitazione dell'alta disponibilità.

Predefinito: localhost:10051

PidFile

Specifica il file PID.

Predefinito: /tmp/zabbix_server.pid

ProblemHousekeepingFrequency

Imposta la frequenza di eliminazione dei problemi per i trigger eliminati (in secondi).
 Vedi anche i dettagli sulla procedura di housekeeping.

Predefinito: 60
 Intervallo: 1-3600

ProxyConfigFrequency

Imposta la frequenza di invio dei dati di configurazione al proxy Zabbix da parte del server. Usato solo per i proxy in modalità passiva.

Predefinito: 10
 Intervallo: 1-604800

ProxyDataFrequency

Imposta la frequenza con cui il server richiede i dati storici dal proxy Zabbix. Utilizzato solo per i proxy in modalità passiva.

Predefinito: 1
 Intervallo: 1-3600

ServiceManagerSyncFrequency

Imposta la frequenza di sincronizzazione della configurazione di un service manager (in secondi).

Predefinito: 60
 Intervallo: 1-3600

SMSDevices

Specifica i file del modem che sono autorizzati a essere usati dal server Zabbix. Un elenco delimitato da virgole.
 L'invio di SMS non è possibile se questo parametro non è impostato.

Esempio:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Specifica il file temporaneo utilizzato per passare i dati dal demone SNMP trap al server.
Deve essere lo stesso indicato in `zabbix_trap_receiver.pl` oppure nel file di configurazione SNMPTT.

Predefinito: `/tmp/zabbix_traps.tmp`

SocketDir

Specifica la directory in cui archiviare i socket IPC utilizzati dai servizi interni di Zabbix.

Predefinito: `/tmp`

SourceIP

Specifica l'indirizzo IP di origine per:

- le connessioni in uscita verso Zabbix proxy e Zabbix agent
- le connessioni agentless (VMware, SSH, JMX, SNMP, Telnet e controlli semplici)
- le connessioni HTTP agent
- le richieste HTTP JavaScript degli item script
- le richieste HTTP JavaScript del preprocessing
- l'invio delle email di notifica (connessioni al server SMTP)
- le notifiche webhook (connessioni HTTP JavaScript)
- le connessioni al Vault

SSHKeyLocation

Specifica il percorso delle chiavi pubbliche e private per i controlli e le azioni SSH.

SSLCertLocation

Specificare il percorso dei file del certificato client SSL per l'autenticazione del client.
 Questo parametro viene utilizzato nel monitoraggio web, per la comunicazione con Vault, per il provider di archiviazione della cronologia ClickHouse e per l'origine dati di telemetria ClickHouse.

SSLKeyLocation

Specificare la posizione dei file delle chiavi private SSL per l'autenticazione del client.
 Questo parametro viene utilizzato nel monitoraggio web, per la comunicazione con Vault, per il provider di archiviazione della cronologia ClickHouse e per la fonte di dati di telemetria ClickHouse.

SSLCALocation

Specifica la posizione dei file dell'autorità di certificazione (CA) per la verifica dei certificati del server SSL. Se non impostato, verrà utilizzata la directory a livello di sistema.
 Il valore di questo parametro verrà impostato come opzione CURLOPT_CAPATH di libcurl. Per le versioni di libcurl precedenti alla 7.42.0, questa impostazione ha effetto solo se libcurl è stato compilato per utilizzare OpenSSL. Per ulteriori informazioni, vedere la pagina web di cURL.
 Questo parametro viene utilizzato nel monitoraggio web, nell'autenticazione SMTP, negli item HTTP agent, per la comunicazione con Vault, per il provider di archiviazione della cronologia ClickHouse e per la sorgente dei dati di telemetria ClickHouse.

StartAgentPollers

Imposta il numero di istanze pre-forked di Zabbix agent pollers. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartAlerters

Imposta il numero di istanze pre-forked di alerters.

Predefinito: 3
 Intervallo: 1-100

StartBrowserPollers

Imposta il numero di istanze pre-forked dei pollers degli item browser.

Predefinito: 1
 Intervallo: 0-1000

StartConnectors

Imposta il numero di istanze pre-forked di connector workers. Il processo di gestione dei connector viene avviato automaticamente quando viene avviato un connector worker.

Predefinito: 0
 Intervallo: 0-1000

StartDBSyncers

Imposta il numero di istanze pre-forked di history syncers.
Nota: fai attenzione quando modifichi questo valore; aumentarlo potrebbe fare più male che bene. Indicativamente, il valore predefinito dovrebbe essere sufficiente per gestire fino a 4000 NVPS.

Predefinito: 4
 Intervallo: 1-100

StartDiscoverers

Imposta il numero di istanze pre-forked di discovery workers[1].

Predefinito: 5
 Intervallo: 0-1000

StartEscalators

Imposta il numero di istanze pre-forked di escalators.

Predefinito: 1
 Intervallo: 1-100

StartHistoryPollers

Imposta il numero di istanze pre-forked di history pollers.
Richiesto solo per i controlli calcolati.

Predefinito: 5
 Intervallo: 0-1000

StartHTTPAgentPollers

Imposta il numero di istanze pre-forked di HTTP agent pollers. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartHTTPPollers

Imposta il numero di istanze pre-forked di HTTP pollers[1].

Predefinito: 1
 Intervallo: 0-1000

StartIPMIPollers

Imposta il numero di istanze pre-forked di IPMI pollers.

Predefinito: 0
 Intervallo: 0-1000

StartJavaPollers

Imposta il numero di istanze pre-forked di Java pollers[1].

Predefinito: 0
 Intervallo: 0-1000

StartLLDProcessors

Imposta il numero di istanze pre-forked dei worker di discovery a basso livello (LLD)[1].
 Il processo manager LLD viene avviato automaticamente quando viene avviato un worker LLD.

Predefinito: 2
 Intervallo: 1-100

StartODBCPollers

Imposta il numero di istanze pre-forked di ODBC pollers[1].

Predefinito: 1
 Intervallo: 0-1000

StartPingers

Imposta il numero di istanze pre-forked di ICMP pingers[1].

Predefinito: 1
 Intervallo: 0-1000

StartPollersUnreachable

Imposta il numero di istanze pre-forked di pollers per host non raggiungibili (inclusi IPMI e Java)[1].
 Almeno un poller per host non raggiungibili deve essere in esecuzione se vengono avviati poller regolari, IPMI o Java.

Predefinito: 1
 Intervallo: 0-1000

StartPollers

Imposta il numero di istanze pre-forked di pollers[1].

Predefinito: 5
 Intervallo: 0-1000

StartPreprocessors

Imposta il numero di istanze pre-avviate dei thread di worker[1] di preprocessing a non meno del numero di core CPU disponibili. È opportuno impostare un numero maggiore di worker se il preprocessing non è vincolato dalla CPU e comporta molte richieste di rete.

Predefinito: 16
 Intervallo: 1-1000

StartProxyPollers

Imposta il numero di istanze pre-forked di pollers per proxy passivi[1].

Predefinito: 1
 Intervallo: 0-250

StartReportWriters

Imposta il numero di istanze pre-forked di report writers.
Se impostato su 0, la generazione pianificata dei report viene disabilitata.
Il processo report manager viene avviato automaticamente quando viene avviato un report writer.

Predefinito: 0
 Intervallo: 0-100

StartSNMPPollers

Imposta il numero di istanze pre-fork di SNMP pollers. Vedi `MaxConcurrentChecksPerPoller`.

Predefinito: 1
 Intervallo: 0-1000

StartSNMPTrapper

Impostare su 1 per avviare un processo SNMP trapper.

Predefinito: 0
 Intervallo: 0-1

StartTelemetryQueryPollers

Imposta il numero di istanze pre-forked dei poller di query Telemetry asincrone. Vedi MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartTimers

Imposta il numero di istanze pre-forked di timers.
I timers elaborano i periodi di manutenzione.

Predefinito: 1
 Intervallo: 1-1000

StartTrappers

Imposta il numero di istanze pre-forked di trappers[1].
 I trappers accettano connessioni in ingresso da Zabbix sender, agent attivi e proxy attivi.

Predefinito: 5
 Intervallo: 0-1000

StartVMwareCollectors

Imposta il numero di istanze pre-forked di VMware collector.

Predefinito: 0
 Intervallo: 0-250

StatsAllowedIP

Specifica gli indirizzi IP (facoltativamente in notazione CIDR) o i nomi DNS delle istanze Zabbix esterne autorizzate a richiedere le statistiche. Un elenco delimitato da virgole.
 Se questo parametro non è impostato, nessuna richiesta di statistiche verrà accettata.
 Se il supporto IPv6 è abilitato, allora `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4.

Esempio:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Impostare la fonte dei dati di telemetria utilizzando il formato `<provider>[;<options>]` dove:

- `<provider>` - una delle fonti di dati di telemetria supportate: `clickhouse`
- `<options>` - coppie `key=value` delimitate da virgole. I valori possono essere racchiusi tra virgolette.

Opzioni per ClickHouse:

- `url` - impostare l'URL HTTP o HTTPS utilizzato per connettersi alla fonte dei dati di telemetria.
- `db` - impostare il nome del database.
- `username` - impostare il nome utente utilizzato per accedere al database.
- `password` - impostare la password utilizzata per accedere al database.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - impostare queste opzioni per configurare una connessione sicura (TLS/SSL) al database.
- `source_ip` - impostare l'indirizzo IP di origine utilizzato per le connessioni in uscita verso la fonte dei dati di telemetria.
- `vault_path` - impostare il percorso di un segreto Vault contenente i valori di `username` e `password`. Non utilizzare insieme alle opzioni `username` e `password`.

Esempio per ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Esempio per ClickHouse con Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix proxy, agent, web service e i controlli SNMP legacy (numero OID singolo o stringa).

Questo parametro definisce la durata di varie operazioni di comunicazione:

- esecuzione di comandi remoti su Zabbix agent
- esecuzione di comandi SSH/Telnet
- richieste a Zabbix web service
- timeout di comunicazione per la richiesta di test del mediatype e il metodo `history.push`
- ripianificazione degli item quando un'interfaccia IPMI diventa non disponibile
- invio della risposta a Zabbix proxy quando non è possibile scambiare dati a causa di problemi di autorizzazione o crittografia
- scadenza per i socket asincroni IPC e le opzioni di controllo runtime
- connessioni JMX
- recupero delle statistiche da un Zabbix proxy o server remoto
- invio delle risposte a Zabbix frontend
- richieste DNS del poller asincrono
- risposta per l'heartbeat dei controlli attivi
- recupero dei dati (valori) di Zabbix agent dagli agent attivi
- recupero dei dati da Zabbix sender
- risposta quando il trapper non riesce a elaborare una richiesta
- invio dell'elenco dei controlli attivi a Zabbix agent

Questo timeout **non** verrà utilizzato per i controlli che hanno configurazioni di timeout flessibile impostate nel frontend (a livello globale, di proxy o per singolo item). Ad esempio, gli item SNMP `walk[OID]` e `get[OID]` usano il timeout configurato nel frontend; i controlli SNMP legacy continuano a usare il valore di timeout del server.

Predefinito: 3
 Intervallo: 1-30

TLSCAFile

Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Specifica il percorso completo del file contenente il certificato del server o la catena di certificati, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCipherAll

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificati e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificati e PSK.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Esempio per OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificati.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Esempio per OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificato.

TLSCipherPSK

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Esempio per OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSFrontendAccept

Specifica il livello di crittografia per le connessioni in ingresso dal frontend.

È possibile specificare più valori, separati da virgole:

- unencrypted - accetta connessioni senza crittografia.
- cert - accetta connessioni protette con TLS e un certificato.

Predefinito: `unencrypted`

TLSFrontendCertIssuer

Specifica l'issuer del certificato frontend consentito.

TLSFrontendCertSubject

Specifica il subject del certificato frontend consentito.

TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata del server, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSListen

Richiedi connessioni solo TLS sul socket trapper.

Valori supportati:

- required - accetta solo connessioni TLS

TmpDir

Specifica la directory temporanea.

Predefinito: `/tmp`

TrapperTimeout

Specifica il timeout del trapper in secondi per:

- il recupero dei dati storici da Zabbix proxy
- l'invio dei dati di configurazione a Zabbix proxy
- l'esecuzione di script globali o l'esecuzione di comandi remoti su Zabbix server

Predefinito: 300
 Intervallo: 1-300

TrendCacheSize

Imposta la dimensione della cache delle tendenze, in byte.
La dimensione della memoria condivisa per l'archiviazione delle tendenze.

Predefinito: `4M`
 Intervallo: 128K-16G

TrendFunctionCacheSize

Imposta la dimensione della cache delle trend-function, in byte.
La dimensione della memoria condivisa per memorizzare nella cache i dati calcolati delle trend function.

Predefinito: `4M`
 Intervallo: 128K-2G

UnavailableDelay

Imposta la frequenza di controllo della disponibilità dell'host durante il periodo di indisponibilità in secondi.

Predefinito: 60
 Intervallo: 1-3600

UnreachableDelay

Imposta la frequenza di controllo della disponibilità dell'host durante il periodo di irraggiungibilità in secondi.

Predefinito: 15
 Intervallo: 1-3600

UnreachablePeriod

Imposta dopo quanti secondi di unreachability considerare l'host non disponibile.

Predefinito: 45
 Intervallo: 1-3600

Utente

Riduce i privilegi a un utente specifico già esistente nel sistema.
Ha effetto solo se eseguito come 'root' e AllowRoot è disabilitato.

Predefinito: zabbix

ValueCacheSize

Imposta la dimensione della cache dei valori della cronologia (in byte).
 La dimensione della memoria condivisa per la memorizzazione nella cache delle richieste di dati della cronologia degli item.
 Impostando il valore su 0 si disabilita la cache dei valori (non consigliato).
 Quando la cache dei valori esaurisce la memoria condivisa, viene scritto un messaggio di avviso nel log del server ogni 5 minuti.

Predefinito: 8M
 Intervallo: 0, 128K-64G

Vault

Specificare il provider del vault: - *HashiCorp* - HashiCorp KV Secrets Engine versione 2; - *CyberArk* - CyberArk Central Credential Provider
Deve corrispondere al provider del vault impostato nel frontend.

Predefinito: HashiCorp

VaultAppRoleID

Specifica l'AppRole ID di HashiCorp vault.
 Se impostato, abilita l'autenticazione AppRole. Non può essere usato con VaultToken.
 Supportato solo per HashiCorp vault.

VaultAppSecretID

Specifica l'ID segreto AppRole di HashiCorp vault.
 Supportato solo per HashiCorp vault.

VaultDBPath

Specifica la posizione da cui recuperare le credenziali del database tramite chiavi. Si tratta di un percorso o di una query di Vault, a seconda del Vault.

Le chiavi utilizzate per HashiCorp sono password e username.

Esempio di percorso con VaultPrefix=/v1/secret/data/zabbix/:

```
database
```

Esempio di percorso senza VaultPrefix:

```
secret/zabbix/database
```

Le chiavi utilizzate per CyberArk sono Content e UserName.

Esempio:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Questa opzione può essere utilizzata solo se DBUser e DBPassword non sono specificati.

VaultPrefix

Specifica un prefisso personalizzato per il percorso o la query del vault, a seconda del Vault. Se non specificato, verranno utilizzati i valori predefiniti più adatti.
 Nota che data viene aggiunto automaticamente dopo il punto di mount per HashiCorp se VaultPrefix non è specificato.

Esempio di prefisso per HashiCorp:

```
/v1/secret/data/zabbix/
```

Esempio di prefisso per Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Specifica il nome del file del certificato SSL utilizzato per l'autenticazione del client. Il file del certificato deve essere in formato PEM1.
 Se il file del certificato contiene anche la chiave privata, lascia vuoto il campo del file della chiave SSL.
 La directory contenente questo file è specificata dal parametro di configurazione SSLCertLocation.
 Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultTLSKeyFile

Specifica il nome del file della chiave privata SSL utilizzato per l'autenticazione del client. Il file della chiave privata deve essere in formato PEM1.
La directory contenente questo file è specificata dal parametro di configurazione SSLKeyLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultToken

Specifica il token di autenticazione di HashiCorp Vault. Il token di autenticazione di HashiCorp Vault deve essere stato generato esclusivamente per il server Zabbix con autorizzazione di sola lettura ai percorsi specificati in Vault macros e autorizzazione di sola lettura al percorso specificato nel parametro di configurazione facoltativo VaultDBPath.
 Si verifica un errore se `VaultToken` e la variabile di ambiente `VAULT_TOKEN` sono definiti contemporaneamente.
 Non può essere usato con VaultAppRoleID.
 Supportato solo per HashiCorp Vault.

VaultURL

Specifica l'URL del server vault. La directory dei certificati CA a livello di sistema verrà utilizzata se SSLCALocation non è specificato.

Predefinito: `https://127.0.0.1:8200`

VMwareCacheSize

Imposta la dimensione della memoria condivisa per l'archiviazione dei dati VMware. È possibile usare un controllo interno VMware `zabbix[vmware,buffer,...]` per monitorare l'utilizzo della cache VMware (vedere Controlli interni).
Nota che la memoria condivisa non viene allocata se non sono configurate istanze di `vmware collector` da avviare.

Predefinito: 8M
 Intervallo: 256K-2G

VMwareFrequency

Imposta la frequenza di raccolta dei dati da un singolo servizio VMware (in secondi). Questa frequenza dovrebbe essere impostata sul più basso intervallo di aggiornamento di qualsiasi item di monitoraggio VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwarePerfFrequency

Imposta la frequenza di recupero delle statistiche dei contatori di prestazioni da un singolo servizio VMware (in secondi).
Questa frequenza dovrebbe essere impostata sull'intervallo di aggiornamento minimo di qualsiasi item di monitoraggio VMware che utilizza i contatori di prestazioni VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwareTimeout

Imposta il numero massimo di secondi che un `vmware collector` attenderà una risposta dal servizio VMware (vCenter o hypervisor ESX).

Predefinito: 10
 Intervallo: 1-300

WebDriverURL

Specifica l'URL dell'interfaccia WebDriver.

Esempio (usato con il server standalone di Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Specifica l'URL del servizio web nel formato `http[s]://host:port/report`.

Esempio:

```
WebServiceURL=http://localhost:10053/report
```

Lo schema `http://` può essere omesso solo per connessioni non TLS (HTTP); con TLS configurato, è necessario usare `https://`.

Note a piè di pagina

[1] Si noti che un numero eccessivo di processi di raccolta dati (poller, unreachable poller, ODBC poller, HTTP poller, Java poller, pinger, trapper, proxypoller) insieme a IPMI manager, SNMP trapper, preprocessing worker e discovery worker può esaurire il limite di descrittori di file per processo per il preprocessing manager.

Warning:
L'esaurimento del limite di descrittori di file causerà l'arresto di Zabbix server, in genere poco dopo l'avvio, ma talvolta dopo più tempo. Per evitare tali problemi, esaminare il file di configurazione di Zabbix server per ottimizzare il numero di controlli e processi concorrenti. Inoltre, se necessario, assicurarsi che il limite di descrittori di file sia impostato sufficientemente in alto verificando e regolando i limiti di sistema.

proxy

Panoramica Il proxy Zabbix è un processo che può raccogliere dati di monitoraggio da uno o più dispositivi monitorati e inviare le informazioni al server Zabbix, operando essenzialmente per conto del server. Tutti i dati raccolti vengono memorizzati localmente in un buffer e quindi trasferiti al server Zabbix a cui appartiene il proxy.

La distribuzione di un proxy è facoltativa, ma può essere molto utile per distribuire il carico di un singolo server Zabbix. Se i dati vengono raccolti esclusivamente dai proxy, l'elaborazione sul server richiede meno risorse CPU e I/O del disco.

Un proxy Zabbix è la soluzione ideale per il monitoraggio centralizzato di sedi remote, filiali e reti prive di amministratori locali.

Il proxy Zabbix richiede un database separato.

Attention:
Si noti che i database supportati dal proxy Zabbix sono SQLite, MySQL e PostgreSQL.

Vedere anche:

- Monitoraggio distribuito tramite proxy
- Bilanciamento del carico e alta disponibilità del proxy
- Sincronizzazione della configurazione di monitoraggio
- Protocollo di scambio dati tra server e proxy
- Parametri di configurazione: proxy Zabbix
- Configurazione di un proxy Zabbix Cloud

Esecuzione del proxy Se installato come pacchetto

Zabbix proxy viene eseguito come processo daemon. Il proxy può essere avviato eseguendo:

```
systemctl start zabbix-proxy
```

Questo funzionerà sulla maggior parte dei sistemi GNU/Linux. Su altri sistemi potrebbe essere necessario eseguire:

```
/etc/init.d/zabbix-proxy start
```

Analogamente, per arrestare/riavviare/visualizzare lo stato di Zabbix proxy, utilizzare i seguenti comandi:

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

Avvio manuale

Se quanto sopra non funziona, devi avviarlo manualmente. Trova il percorso del binario `zabbix_proxy` ed esegui:

```
zabbix_proxy
```

Puoi usare i seguenti parametri da riga di comando con Zabbix proxy:

Parameter	Description
`-c --config <config-file>`	Percorso del file di configurazione.
`-f --foreground`	Esegui Zabbix proxy in primo piano.
`-R --runtime-control <option>`	Esegui funzioni amministrative.
`-T --test-config`	Valida il file di configurazione ed esci.
`-h --help`	Mostra le informazioni di aiuto ed esci.
`-V --version`	Mostra il numero di versione.

Esempi di esecuzione di Zabbix proxy con parametri da riga di comando:

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

Controllo runtime

Opzioni di controllo runtime:

Option	Description	Target
`config_cache_reload`	Ricarica la cache di configurazione. Ignorato se la cache è attualmente in fase di caricamento. Il proxy Zabbix attivo si connetterà al server Zabbix e richiederà i dati di configurazione. Il proxy Zabbix passivo richiederà i dati di configurazione dal server Zabbix alla successiva connessione del server al proxy.	
`history_cache_clear[<target>]`	Svuota la cache della cronologia per l'item specificato dal relativo ID. Influisce su tutti i valori dell'item, eccetto il primo e l'ultimo valore.	**target** - ID dell'item.
`diaginfo[=<section>]`	Raccoglie informazioni diagnostiche nel file di log del proxy.	`historycache` - statistiche della cache della cronologia; `preprocessing` - statistiche del manager di preprocessing; `locks` - elenco dei mutex (è vuoto sui sistemi *BSD*).
`snmp_cache_reload`	Ricarica la cache SNMP - svuota le proprietà del motore SNMP (engine time, engine boots, engine id, credentials) per tutti gli host. Da usare per forzare una pulizia globale della cache durante la risoluzione dei problemi SNMP.	
`housekeeper_execute`	Avvia la procedura di housekeeping. Ignorato se la procedura di housekeeping è attualmente in corso.	
`log_level_increase[=<target>]`	Aumenta il livello di log, interessa tutti i processi se non viene specificato un target. Non supportato sui sistemi *BSD*.	**process type** - tutti i processi del tipo specificato (ad esempio, `poller`). Vedere tutti i tipi di processo del proxy. **process type,N** - tipo di processo e numero (ad esempio, `poller,3`) **pid** - identificatore del processo (1 to 65535). Per valori maggiori specificare il target come 'process type,N'.
`log_level_decrease[=<target>]`	Diminuisce il livello di log, interessa tutti i processi se non viene specificato un target. Non supportato sui sistemi *BSD*.	
`prof_enable[=<target>]`	Abilita il profiling. Interessa tutti i processi se non viene specificato un target. Il profiling abilitato fornisce dettagli di tutti i rwlock/mutex per nome funzione.	**process type** - tutti i processi del tipo specificato (ad esempio, `history syncer`). Vedere tutti i tipi di processo del proxy. **process type,N** - tipo di processo e numero (ad esempio, `history syncer,1`). **pid** - identificatore del processo (1 to 65535). Per valori maggiori specificare il target come 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` possono essere usati con il tipo di processo e il numero (ad esempio, `history syncer,1,processing`) oppure con tutti i processi di un tipo (ad esempio, `history syncer,rwlock`).

Option	Description	Target
`prof_disable[=<target>]`	Disabilita il profiling. Interessa tutti i processi se non viene specificato un target.	**process type** - tutti i processi del tipo specificato (ad esempio, `history syncer`). Vedere tutti i tipi di processo del proxy. **process type,N** - tipo di processo e numero (ad esempio, `history syncer,1`). **pid** - identificatore del processo (1 to 65535). Per valori maggiori specificare il target come 'process type,N'.

Quando un tipo di processo viene specificato come target per il controllo runtime, il comando si applica solo ai processi in esecuzione di quel tipo. Se al momento non è in esecuzione alcun processo del tipo specificato, il comando restituisce un errore:

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

Esempio di utilizzo del controllo runtime per ricaricare la cache di configurazione del proxy:

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

Esempio di utilizzo del controllo runtime per svuotare la cache della cronologia di un item:

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R history_cache_clear=42243
```

Esempi di utilizzo del controllo runtime per raccogliere informazioni diagnostiche:

```
### Raccogli tutte le informazioni diagnostiche disponibili nel file di log del proxy:
zabbix_proxy -R diaginfo

### Raccogli le statistiche della cache della cronologia nel file di log del proxy:
zabbix_proxy -R diaginfo=historycache
```

Esempio di utilizzo del controllo runtime per ricaricare la cache SNMP:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Quando un'interfaccia SNMPv3 viene aggiornata tramite la UI di Zabbix, nella maggior parte dei casi Zabbix ricaricherà automaticamente le nuove credenziali SNMPv3 per tale interfaccia; usare `-R snmp_cache_reload` solo se il polling continua a fallire dopo la modifica delle credenziali (ad esempio, a causa di incoerenze engineBoots/engineID o di dispositivi non RFC), oppure quando è necessario forzare una pulizia globale della cache SNMP per la risoluzione dei problemi.

Esempio di utilizzo del controllo runtime per avviare l'esecuzione di housekeeper:

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

Esempi di utilizzo del controllo runtime per modificare il livello di log:

```
### Aumenta il livello di log di tutti i processi:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Aumenta il livello di log del secondo processo poller:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Aumenta il livello di log del processo con PID 1234:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Diminuisce il livello di log di tutti i processi http poller:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Esecuzione con un utente

Il proxy Zabbix è progettato per essere eseguito come utente non root. Verrà eseguito con l'utente non root con cui è stato avviato. Pertanto, è possibile eseguire il proxy con qualsiasi utente non root senza problemi.

Se si tenta di eseguirlo come 'root', passerà a un utente 'zabbix' codificato, che deve essere presente nel sistema. È possibile eseguire il proxy come 'root' solo modificando di conseguenza il parametro `AllowRoot` nel file di configurazione del proxy.

File di configurazione

Il percorso predefinito del file di configurazione dipende dal sistema operativo e dal metodo di installazione di Zabbix:

- Per l'installazione dai sorgenti: `/usr/local/etc/zabbix_proxy.conf`, oppure come impostato dalle variabili di compilazione `--sysconfdir` o `--prefix`.
- Per l'installazione dai pacchetti: `/etc/zabbix/zabbix_proxy.conf`.

Per i dettagli sulla configurazione di `zabbix_proxy`, vedere le opzioni del file di configurazione.

Tipi di processi e thread del proxy

- `agent poller` - processo poller asincrono per i controlli passivi con un thread di lavoro;
- `apm manager` - processo manager per il monitoraggio delle prestazioni delle applicazioni (APM);
- `apm worker` - processo per la gestione delle attività di monitoraggio delle prestazioni delle applicazioni (APM) provenienti da `apm manager`;
- `availability manager` - processo per gli aggiornamenti della disponibilità degli host;
- `browser poller` - poller per i controlli degli item browser;
- `configuration syncer` - processo per la gestione della cache in memoria dei dati di configurazione;
- `data sender` - processo per l'invio dei dati del proxy;
- `discovery manager` - processo manager per il rilevamento dei dispositivi;
- `discovery worker` - processo per la gestione delle attività di rilevamento provenienti da discovery manager;
- `history syncer` - processo di scrittura nel database della cronologia;
- `housekeeper` - processo per la rimozione della cronologia obsoleta degli item;
- `http agent poller` - processo poller asincrono per i controlli HTTP con un thread di lavoro;
- `http poller` - poller per il monitoraggio web;
- `icmp pinger` - poller per i controlli icmpping;
- `internal poller` - poller per i controlli interni;
- `ipmi manager` - manager dei poller IPMI;
- `ipmi poller` - poller per i controlli IPMI;
- `java poller` - poller per i controlli Java;
- `odbc poller` - poller per i controlli ODBC;
- `poller` - poller normale per i controlli passivi;
- `preprocessing manager` - manager delle attività di pre-elaborazione con thread di lavoro per la pre-elaborazione;
- `preprocessing worker` - thread per la pre-elaborazione dei dati;
- `self-monitoring` - processo per la raccolta delle statistiche interne del server;
- `snmp poller` - processo poller asincrono per i controlli SNMP con un thread di lavoro (solo item `walk[OID]` e `get[OID]`);
- `snmp trapper` - trapper per i trap SNMP;
- `supervisor` - coordina l'avvio e l'arresto dei processi e dei thread del server e ne monitora la disponibilità;
- `task manager` - processo per l'esecuzione remota delle attività richieste da altri componenti (ad esempio, chiusura del problema, riconoscimento del problema, controllo immediato del valore dell'item, funzionalità dei comandi remoti);
- `telemetry query poller` - poller per i controlli degli item di query di telemetria;
- `trapper` - trapper per i controlli attivi, i trap e la comunicazione con il proxy;
- `unreachable poller` - poller per i dispositivi non raggiungibili;
- `vmware collector` - raccoglitore di dati VMware responsabile della raccolta dei dati dai servizi VMware.

Il file di log del proxy può essere utilizzato per osservare questi tipi di processi.

Il file di log del proxy viene creato con autorizzazioni di lettura e scrittura esclusivamente per il proprietario del file. Inoltre, il file è leggibile dal gruppo proprietario. Tutte le altre autorizzazioni sono negate.

È possibile monitorare vari tipi di processi del proxy Zabbix utilizzando l'item interno `zabbix[process,<type>,<mode>,<state>]`.

Statistiche delle transazioni del history syncer

Il titolo del processo history syncer mostra statistiche dettagliate sulle transazioni del history syncer.

```
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
```

I tempi, in `processed...in N (<timings>) sec`, sono:

- Tempo impiegato per scrivere i valori degli item nel database.
- Tempo impiegato per aggiornare i dati dell'item (stato, errori).

Procedura di housekeeping

Zabbix proxy dispone del processo `housekeeper` che rimuove la cronologia e le tendenze obsolete degli item. Viene eseguito in cicli, con una frequenza determinata da `HousekeepingFrequency`, e con un limite di eliminazione per ciclo determinato

da ProxyLocalBuffer e ProxyOfflineBuffer. A differenza della procedura di housekeeping di Zabbix server, il processo housekeeper del proxy non utilizza la tabella housekeeper: elimina tutti i dati obsoleti una volta per ogni ciclo di housekeeping.

Piattaforme supportate Zabbix proxy viene eseguito sullo stesso elenco di piattaforme supportate di Zabbix server.

Buffer di memoria Il buffer di memoria consente di archiviare nuovi dati (valori degli item, rilevamento della rete, autoregistrazione degli host) nel buffer e di caricarli nel server Zabbix senza accedere al database. Il buffer di memoria è stato introdotto per il proxy in Zabbix 7.0.

Nelle installazioni precedenti a Zabbix 7.0, i dati raccolti venivano archiviati nel database prima di essere caricati nel server Zabbix. Per queste installazioni, questo comportamento rimane quello predefinito dopo l'aggiornamento a Zabbix 7.0.

Per ottenere prestazioni ottimali, si consiglia di configurare l'uso del buffer di memoria sul proxy. A tale scopo, modificare il valore di ProxyBufferMode da disk (valore predefinito codificato per le installazioni esistenti) a hybrid (scelta consigliata) o memory. È inoltre necessario impostare la dimensione del buffer di memoria (parametro ProxyMemoryBufferSize).

In modalità ibrida, il buffer è protetto dalla perdita di dati poiché i dati non inviati vengono scaricati nel database se il proxy viene arrestato, il buffer è pieno o i dati sono troppo vecchi. Quando tutti i valori sono stati scaricati nel database, il proxy torna a utilizzare il buffer di memoria.

In modalità memoria, viene utilizzato il buffer di memoria, ma non è prevista alcuna protezione contro la perdita di dati. Se il proxy viene arrestato o la memoria si riempie, i dati non inviati vengono eliminati.

La modalità ibrida (ProxyBufferMode=hybrid) viene applicata a tutte le nuove installazioni a partire da Zabbix 7.0.

Parametri aggiuntivi come ProxyMemoryBufferSize e ProxyMemoryBufferAge definiscono rispettivamente la dimensione del buffer di memoria e l'età massima dei dati nel buffer.

In caso di configurazione in conflitto, il proxy stampa un errore e non si avvia, ad esempio se:

- ProxyBufferMode è impostato su hybrid o memory e ProxyMemoryBufferSize è 0.
- ProxyBufferMode è impostato su hybrid o memory e ProxyLocalBuffer non è 0.

Locale Si noti che il proxy richiede una locale UTF-8 affinché alcuni item testuali possano essere interpretati correttamente. La maggior parte dei sistemi Unix-like moderni ha una locale UTF-8 come impostazione predefinita; tuttavia, in alcuni sistemi potrebbe essere necessario impostarla esplicitamente.

Calcolo delle code durante la manutenzione

> **Attention:**
> Il proxy di Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calculation
> i dettagli.

Utilizzo del frontend Zabbix dietro un proxy Se il frontend Zabbix viene eseguito dietro un proxy server, il percorso dei cookie nel file di configurazione del proxy deve essere riscritto in modo da corrispondere al percorso sottoposto a reverse proxy. Vedere gli esempi riportati di seguito. Se il percorso dei cookie non viene riscritto, gli utenti potrebbero riscontrare problemi di autorizzazione quando tentano di accedere al frontend Zabbix.

Esempio di configurazione per NGINX

```
### ..
location / {
### ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
### ..
```

Esempio di configurazione per Apache

```
### ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
### ..
```

Monitoraggio distribuito tramite proxy

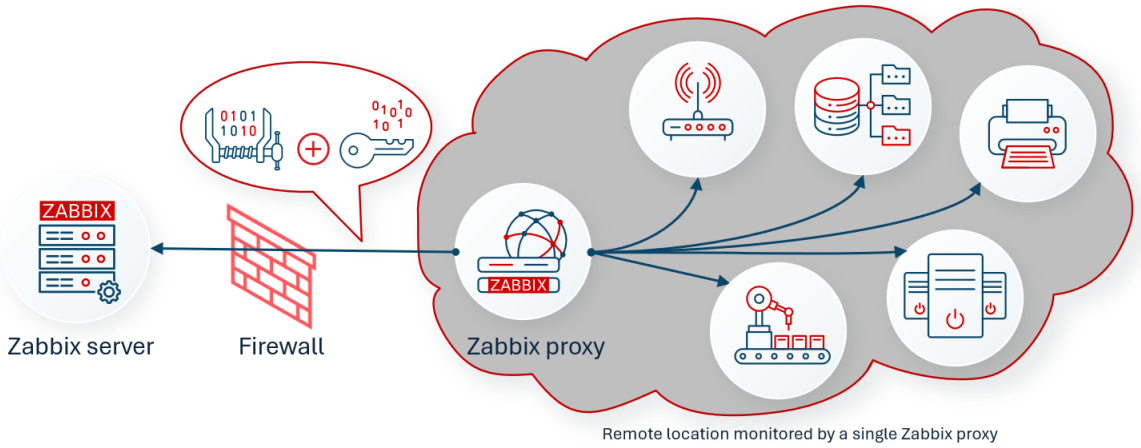
Panoramica

I proxy Zabbix possono essere utilizzati per creare una configurazione di monitoraggio distribuita efficace e affidabile.

I proxy vengono distribuiti localmente per raccogliere dati sulle prestazioni e sulla disponibilità. I dati vengono quindi inviati a un server centralizzato.

Un proxy Zabbix può essere utilizzato per:

- Monitorare sedi remote
- Monitorare sedi con comunicazioni inaffidabili
- Alleggerire il server Zabbix durante il monitoraggio di migliaia di dispositivi
- Raccogliere dati di telemetria per il monitoraggio delle prestazioni delle applicazioni (APM)



Il proxy richiede una sola connessione TCP al server Zabbix. In questo modo è più semplice aggirare un firewall, poiché è necessario configurare una sola regola del firewall.

Attention:
Il proxy Zabbix deve utilizzare un database separato. Indicarlo al database del server Zabbix interromperà la configurazione.

Tutti i dati raccolti dal proxy vengono archiviati localmente prima di essere trasmessi al server. In questo modo non vengono persi dati a causa di problemi temporanei di comunicazione con il server. I parametri `ProxyLocalBuffer` e `ProxyOfflineBuffer` nel file di configurazione del proxy controllano per quanto tempo i dati vengono conservati localmente.

Attention:
Può accadere che un proxy, che riceve direttamente dal database del server Zabbix le modifiche più recenti alla configurazione, disponga di una configurazione più aggiornata rispetto al server Zabbix, la cui configurazione potrebbe non essere aggiornata altrettanto rapidamente a causa del valore di CacheUpdateFrequency. Di conseguenza, il proxy potrebbe iniziare a raccogliere dati e inviarli al server Zabbix, che ignora tali dati.

Il proxy Zabbix è un raccoglitore di dati. Non calcola i trigger, non elabora gli eventi e non invia avvisi. Per una panoramica delle funzionalità del proxy, consultare la tabella seguente:

Funzione		Supportata dal proxy
Item		
	Controlli dell'agent Zabbix	**Sì**
	Controlli dell'agent Zabbix (attivi)	**Sì** [1]
	Controlli semplici	**Sì**
	Item trapper	**Sì**
	Controlli SNMP	**Sì**
	Trap SNMP	**Sì**
	Controlli IPMI	**Sì**
	Controlli JMX	**Sì**
	Monitoraggio dei file di log	**Sì**
	Controlli interni	**Sì**

Funzione		Supportata dal proxy
	Controlli SSH	**Sì**
	Controlli Telnet	**Sì**
	Controlli esterni	**Sì**
	Item dipendenti	**Sì**
	Item script	**Sì**
	Item browser	**Sì**
	Item query di telemetria	**Sì**
Monitoraggio web integrato		**Sì**
Pre-elaborazione dei valori degli item		**Sì**
Individuazione della rete		**Sì**
Registrazione automatica dell'agent attivo		**Sì**
Individuazione di basso livello		**Sì** [2]
Comandi remoti		**Sì**
Raccolta dei dati di telemetria (APM)		**Sì** [3]
Calcolo dei trigger		*No*
Elaborazione degli eventi		*No*
Correlazione degli eventi		*No*
Invio degli avvisi		*No*

Note:
[1] Per assicurarsi che un agent richieda i controlli attivi al proxy (e non al server), il proxy deve essere elencato nel parametro **ServerActive** nel file di configurazione dell'agent.
 [2] Per l'LLD, il proxy Zabbix raccoglie e pre-elabora i dati, quindi li invia al server Zabbix per un'ulteriore elaborazione.
 [3] I dati di telemetria vengono raccolti solo se APM è abilitato sul proxy (vedere la scheda *APM* nella configurazione del proxy); il server Zabbix non può ricevere dati di telemetria.

Funzionalità del proxy

Quando si sceglie se usare o non usare un proxy, è necessario tenere conto di diverse considerazioni.

	Proxy
Leggero	**Sì**
GUI	No
Funziona in modo indipendente	**Sì**
Facile manutenzione	**Sì**
Creazione automatica del DB	**Sì**[1]
Amministrazione locale	No
Pronto per hardware embedded	**Sì**
Connessioni TCP unidirezionali	**Sì**
Configurazione centralizzata	**Sì**
Genera notifiche	No

[1] La funzionalità di creazione automatica del DB funziona solo con SQLite. I database supportati in modo diverso richiedono una configurazione manuale.

Attention:
Il proxy di Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calcolo delle code durante la manutenzione per i dettagli.

Protezione dal sovraccarico

Se il server Zabbix è rimasto inattivo per un certo periodo e i proxy hanno raccolto molti dati, quando il server si avvia potrebbe andare in sovraccarico (l'utilizzo della history cache rimane al 95-100% per un certo tempo). Questo sovraccarico può causare un calo delle prestazioni, con l'elaborazione dei controlli più lenta del dovuto. Per evitare i problemi causati dal sovraccarico della history cache è stata implementata una protezione per questo scenario.

Quando la history cache del server Zabbix è piena, l'accesso in scrittura alla history cache viene limitato, bloccando i processi di raccolta dati del server. Il caso più comune di sovraccarico della history cache si verifica dopo un periodo di inattività del server, quando i proxy caricano i dati raccolti. Per evitare questo problema è stata aggiunta la limitazione dei proxy (attualmente non può essere disabilitata).

Quando l'utilizzo della history cache raggiunge l'80%, il server Zabbix entra in modalità di limitazione. In modalità di limitazione il server accetta i dati dei proxy solo quando l'utilizzo della history cache è inferiore al 60%, alternando i proxy accettati. Quando l'utilizzo della history cache scende sotto il 20%, il server torna alla modalità normale.

Inoltre, in modalità normale il server Zabbix limita i singoli proxy che inviano pacchetti molto grandi (10.000+ record) se l'utilizzo della history cache supera il 60%. Questa decisione viene applicata nell'istante in cui il server valuta un caricamento del proxy e quindi potrebbe non essere sempre riflessa immediatamente nei grafici dell'utilizzo della history cache (l'item interno `zabbix[wcache,history,pused]` e il relativo intervallo di aggiornamento potrebbero non rilevare picchi brevi).

Questa modalità di limitazione continuerà finché l'utilizzo della cache non raggiunge di nuovo l'80%, oppure scende al 20%, oppure la lista di limitazione è vuota. Nel primo caso il server smetterà di accettare nuovamente i dati dei proxy. Negli altri due casi il server tornerà a funzionare normalmente, accettando i dati da tutti i proxy.

Le informazioni precedenti possono essere illustrate nella seguente tabella:

Utilizzo della history write cache	Modalità del server Zabbix	Azione del server Zabbix
Raggiunge l'80%	Attesa	Smette di accettare i dati dei proxy, ma mantiene una *lista di limitazione* (lista prioritaria di proxy da contattare in seguito).
Raggiunge il 60%	Normale, ma pronta a essere limitata	Può rifiutare caricamenti molto grandi dai proxy (più di 10k record) al momento di decidere se accettare i dati; continua ad accettare gli altri dati dei proxy.
Scende al 20%	Normale	Elimina la lista di limitazione e inizia ad accettare normalmente i dati dei proxy.

È possibile usare l'item interno zabbix[wcache,history,pused] per correlare questo comportamento del server Zabbix con una metrica.

Configurazione

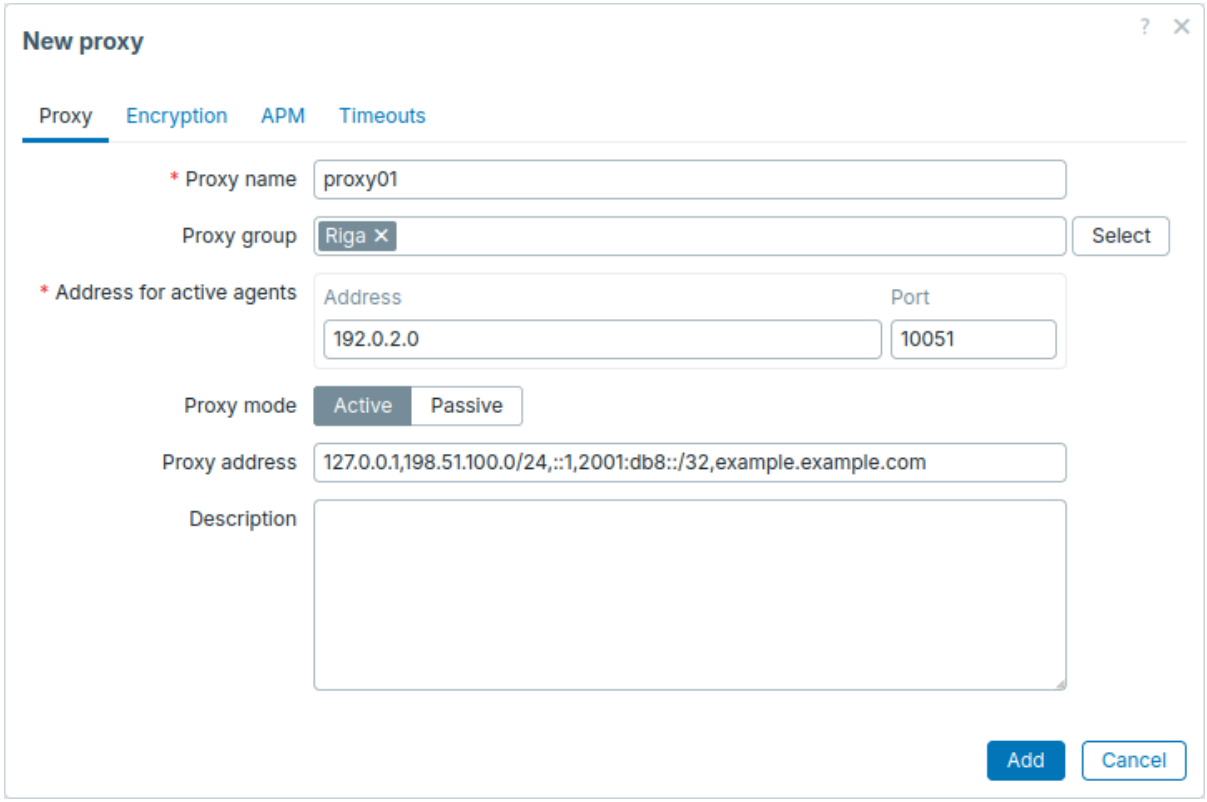
Una volta installato e configurato un proxy, è il momento di configurarlo nel frontend di Zabbix.

Aggiunta di proxy

> **Note:**
> Invece di installare un proxy sulla propria infrastruttura, è possibile distribuirne uno in Zabbix Cloud. A tale scopo, fare clic su *Create proxy in Cloud* in *Administration* > *Proxies*; Zabbix Cloud verrà aperto in una nuova scheda del browser. Dopo aver creato un proxy, configurarlo nel frontend di Zabbix come descritto di seguito. Per ulteriori informazioni, vedere Configurazione di un proxy Zabbix Cloud.

Per configurare un proxy nel frontend di Zabbix:

1. Andare in *Administration* > *Proxies*.
2. Fare clic su *Create proxy*.
3. Inserire i parametri del proxy nel modulo.

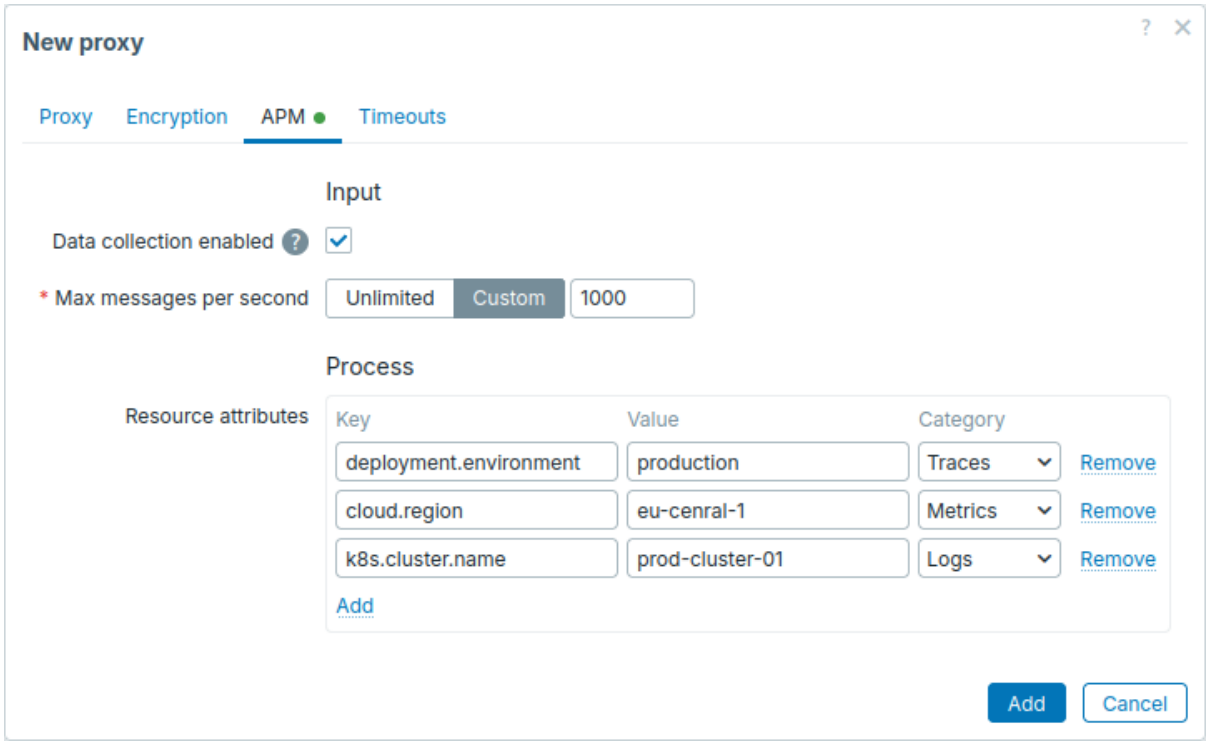


Parametro	Descrizione
Proxy name	Inserire il nome del proxy. Deve corrispondere al nome indicato nel parametro *Hostname* nel file di configurazione del proxy.
Proxy group	Selezionare un gruppo di proxy per il bilanciamento del carico/alta disponibilità del proxy.
Address for active agents	Inserire l'indirizzo a cui devono connettersi gli agent attivi o i sender monitorati. Supportato **solo** dagli agent Zabbix 7.0 o versioni successive. Questo indirizzo viene utilizzato per connettersi sia ai proxy attivi sia a quelli passivi. Questo campo è disponibile solo se nel campo *Proxy group* è selezionato un gruppo di proxy.
Address	Indirizzo IP/nome DNS a cui connettersi.
Port	Numero della porta TCP (10051 per impostazione predefinita) a cui connettersi. Sono supportate le macro utente.
Proxy mode	Selezionare la modalità del proxy. **Active** - il proxy si connetterà al server Zabbix e richiederà i dati di configurazione **Passive** - il server Zabbix si connette al proxy *Nota*: senza comunicazioni crittografate, i dati di configurazione del proxy (sensibili) potrebbero diventare disponibili a soggetti che hanno accesso alla porta trapper del server Zabbix quando si utilizza un proxy attivo. Ciò è possibile perché chiunque può fingere di essere un proxy attivo e richiedere i dati di configurazione se non viene eseguita l'autenticazione o se gli indirizzi dei proxy non sono limitati nel campo *Proxy address*.
Proxy address	Se specificato, le richieste del proxy attivo vengono accettate solo dall'elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, o dai nomi DNS del proxy Zabbix attivo. Questo campo è disponibile solo se nel campo *Proxy mode* è selezionato un proxy attivo. Le macro non sono supportate.
Interface	Inserire i dettagli dell'interfaccia per un proxy passivo. Questo campo è disponibile solo se nel campo Proxy mode è selezionato un proxy passivo.
Address	Indirizzo IP/nome DNS del proxy passivo.
Port	Numero della porta TCP del proxy passivo (10051 per impostazione predefinita). Sono supportate le macro utente.
Description	Inserire la descrizione del proxy.

La scheda **Encryption** consente di richiedere connessioni crittografate con il proxy.

Parametro	Descrizione
Connections to proxy	Modalità con cui il server si connette al proxy passivo: nessuna crittografia (impostazione predefinita), tramite PSK (chiave precondivisa) o certificato.
Connections from proxy	Selezionare i tipi di connessione consentiti dal proxy attivo. È possibile selezionare contemporaneamente diversi tipi di connessione (utile per i test e per il passaggio a un altro tipo di connessione). L'impostazione predefinita è "No encryption".
Issuer	Emittente consentito del certificato. Il certificato viene prima convalidato con la CA (autorità di certificazione). Se è valido e firmato dalla CA, il campo *Issuer* può essere utilizzato per limitare ulteriormente le CA consentite. Questo campo è facoltativo ed è destinato all'uso quando l'installazione di Zabbix utilizza certificati di più CA.
Subject	Subject consentito del certificato. Il certificato viene prima convalidato con la CA. Se è valido e firmato dalla CA, il campo *Subject* può essere utilizzato per consentire un solo valore della stringa *Subject*. Se questo campo è vuoto, viene accettato qualsiasi certificato valido firmato dalla CA configurata.
PSK identity	Stringa dell'identità della chiave precondivisa. Non inserire informazioni sensibili nell'identità PSK, poiché viene trasmessa non crittografata sulla rete per informare il destinatario su quale PSK utilizzare.
PSK	Chiave precondivisa (stringa esadecimale). Lunghezza massima: 512 cifre esadecimali (PSK da 256 byte) se Zabbix utilizza la libreria GnuTLS o OpenSSL, 64 cifre esadecimali (PSK da 32 byte) se Zabbix utilizza la libreria mbed TLS (PolarSSL). Esempio: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

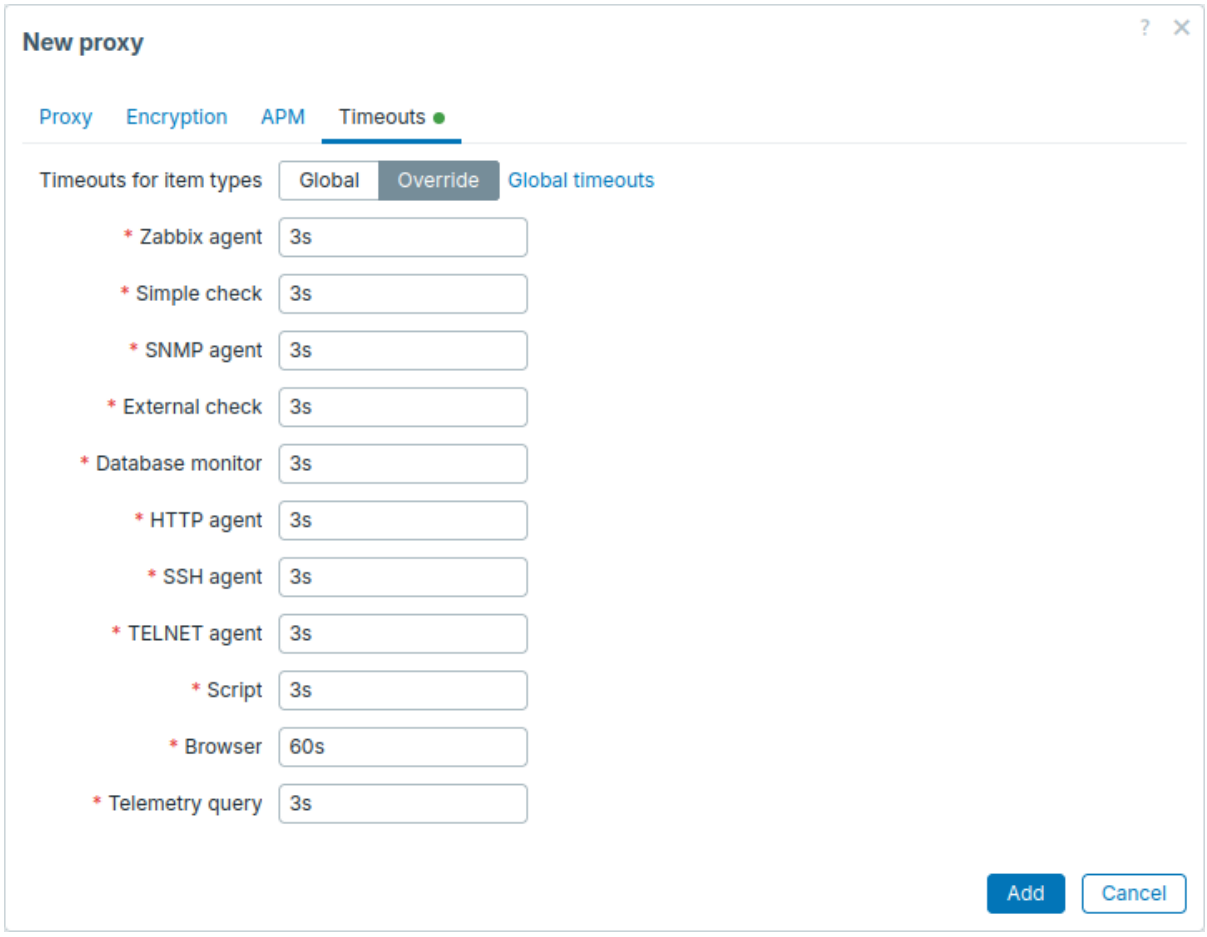
La scheda **APM** consente di configurare il monitoraggio delle prestazioni delle applicazioni (APM) sul proxy. Il proxy può raccogliere dati di telemetria (tracce, metriche e log) inviati dalle applicazioni o dai collector OpenTelemetry tramite il protocollo OpenTelemetry (OTLP/gRPC) e scrivere i dati nella sorgente dati di telemetria.



Parametro	Descrizione
Data collection enabled	Selezionare la casella per abilitare la raccolta dei dati di telemetria sul proxy. Prima di abilitare la raccolta dei dati di telemetria, impostare il parametro `StartAPMCollector` su 1 nel file di configurazione del proxy (è necessario un proxy Zabbix con supporto APM).

Parametro	Descrizione
Max messages per second	Impostare il numero massimo di messaggi al secondo accettati dal proxy: **Unlimited** - accetta tutti i messaggi in ingresso. **Custom** - accetta fino al numero specificato di messaggi al secondo. Un messaggio è una richiesta OTLP inviata da un'applicazione, da un collector OpenTelemetry o da un'altra sorgente di telemetria. Le sue dimensioni dipendono dal mittente; ad esempio, un collector OpenTelemetry può raggruppare migliaia di span in un unico messaggio, mentre una semplice applicazione può inviare messaggi più piccoli (ad esempio, uno per traccia). Se il limite viene superato, il proxy risponde con il codice di stato gRPC `UNAVAILABLE` (errore temporaneo), quindi il mittente può ritentare l'invio del messaggio in un secondo momento, a seconda della propria configurazione.
Add resource attributes	Specificare gli attributi della risorsa da aggiungere ai dati di telemetria in ingresso del tipo selezionato (ad esempio, per contrassegnare tutti i dati ricevuti da questo proxy con il relativo ambiente o la relativa regione). È quindi possibile utilizzare questi attributi per filtrare i dati di telemetria, ad esempio nelle condizioni degli item di telemetry query. Se i dati in ingresso contengono già un attributo della risorsa con lo stesso nome, viene mantenuto l'attributo in ingresso invece di quello specificato qui.
Name	Inserire il nome dell'attributo (ad esempio, `deployment.environment`, `cloud.region`). Deve essere univoco per il *Type* selezionato.
Value	Inserire il valore dell'attributo (ad esempio, `production`, `eu-central-1`). Può essere vuoto.
Type	Selezionare il tipo di dati in ingresso a cui si applica l'attributo: *Traces*, *Metrics* o *Logs*.

La scheda **Timeouts** consente di sostituire i timeout globali per i tipi di item che lo supportano.



Parametro	Descrizione
Timeouts for item types	Impostare il timeout dell'item (in base al relativo tipo): **Global** - utilizza il timeout globale (visualizzato nel campo *Timeout* disabilitato per ogni tipo di item); **Override** - imposta un timeout personalizzato (nel campo *Timeout* per ogni tipo di item). Intervallo consentito: 1 - 600s (impostazione predefinita: ereditato dai timeout globali). Sono supportati i suffissi temporali, ad esempio 30s e 1m, e le macro utente. Facendo clic sul collegamento *Global timeouts* è possibile configurare i timeout globali. Il collegamento *Global timeouts* è visibile solo agli utenti di tipo *Super admin* con autorizzazioni per la sezione frontend *Administration* > *General*. Si noti che, sebbene i timeout a livello di proxy sostituiscano quelli globali, questi ultimi vengono sostituiti dai timeout dei singoli item, se configurati.

Note:

Se la versione principale del proxy non corrisponde alla versione principale del server, l'icona verrà visualizzata accanto a *Timeouts for item types*, con il messaggio al passaggio del mouse "Timeouts disabled because the proxy and server versions do not match". In questi casi, il proxy utilizzerà il parametro `Timeout` dal file di configurazione del proxy.

Il modulo di modifica di un proxy esistente presenta i seguenti pulsanti aggiuntivi:

- *Refresh configuration* - aggiorna la configurazione del proxy;
- *Clone* - crea un nuovo proxy basato sulle proprietà del proxy esistente;
- *Delete* - elimina il proxy.

Configurazione dell'host

È possibile specificare che un singolo host debba essere monitorato da un proxy o da un gruppo di proxy nel modulo di configurazione dell'host, usando il campo *Monitored by*.



L'aggiornamento di massa degli host è un altro modo per specificare che gli host debbano essere monitorati da un proxy o da un gruppo di proxy.

Bilanciamento del carico e alta disponibilità del proxy

Panoramica

I proxy Zabbix possono essere organizzati in gruppi di proxy per abilitare il bilanciamento del carico e l'alta disponibilità dei proxy.

Il bilanciamento del carico e l'alta disponibilità dei proxy consistono nella ridistribuzione automatica degli host tra i proxy all'interno di un gruppo di proxy:

- Se un proxy va offline, i suoi host verranno spostati su altri proxy, mantenendo così un'elevata disponibilità dei proxy.
- Se un proxy ha un numero di host molto maggiore o minore rispetto agli altri proxy, i suoi host verranno spostati su altri proxy per bilanciare il carico dei proxy.

La ridistribuzione degli host funziona solo tra i proxy di un gruppo che soddisfano le seguenti condizioni:

- I proxy eseguono Zabbix 7.0 o versioni successive.
- La versione del proxy corrisponde alla versione del server Zabbix. Se si utilizza Zabbix agent (passive), la versione del proxy deve corrispondere alla versione dell'agent. Gli agent attivi richiedono solo Zabbix 7.0 o versioni successive.
- Il gruppo di proxy ha uno stato online.
- Gli host sono configurati per essere monitorati da un gruppo di proxy anziché da singoli proxy.

Note:
Lo stato del gruppo di proxy può essere monitorato con i controlli interni da qualsiasi host assegnato a un gruppo di proxy. Tuttavia, per monitorare lo stato di un singolo proxy in un gruppo, assegnare l'host a quel proxy; in caso contrario, i risultati potrebbero essere incoerenti.

Redistribuzione degli host

Il bilanciamento del carico dei proxy e l'alta disponibilità sono gestiti da Zabbix server tramite il proxy group manager, che monitora continuamente lo stato di tutti i proxy in ciascun gruppo di proxy e la relativa distribuzione degli host.

L'alta disponibilità dei proxy all'interno di un gruppo è garantita tramite il failover del proxy: quando un proxy va offline, i relativi host vengono immediatamente ridistribuiti ad altri proxy. Si verifica anche il bilanciamento del carico dei proxy, poiché gli host vengono riassegnati ai proxy con il minor numero di host assegnati.

Inoltre, il bilanciamento del carico dei proxy viene attivato quando il numero di host di un proxy differisce dalla media del gruppo di almeno 10 host e di un fattore pari a 2 (eccesso di host o deficit di host). Se lo squilibrio persiste dopo un periodo di tolleranza (10 x failover delay), il gruppo di proxy viene messo in coda per la ridistribuzione degli host.

Il proxy group manager ridistribuisce gli host usando la seguente logica:

1. Calcolare il numero medio di host per proxy.
2. Per i proxy con eccesso di host, spostare gli host in eccesso nel pool di proxy non assegnati.
3. Per i proxy con deficit di host, calcolare quanti host sono necessari per raggiungere l'equilibrio.
4. Rimuovere il numero richiesto di host dai proxy con il maggior numero di host.
5. Spostare gli host non assegnati ai proxy con il minor numero di host.

Esempi di ridistribuzione degli host:

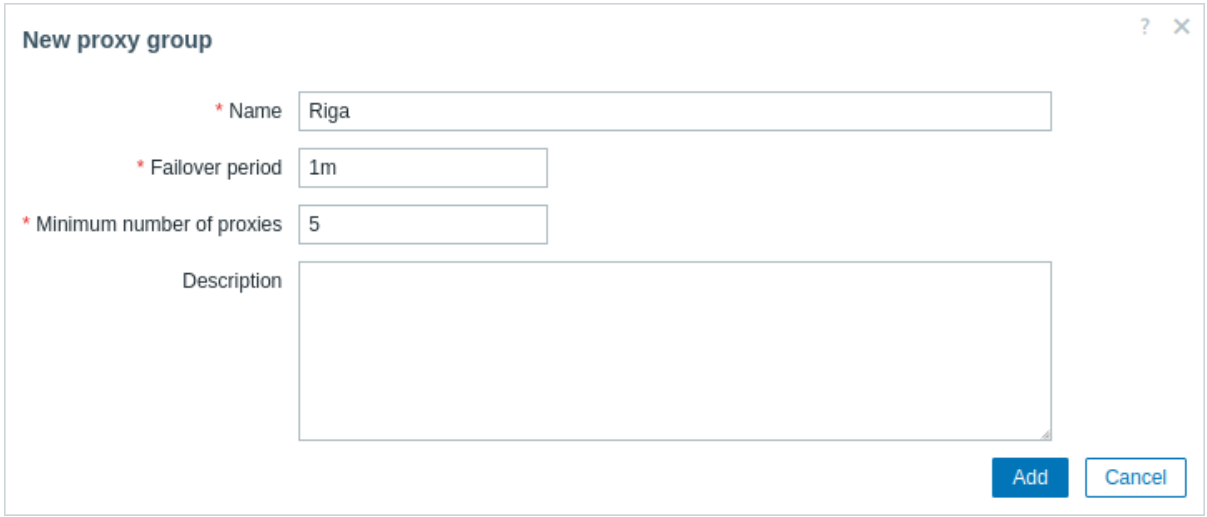
Host sul proxy	Media del gruppo	Riassegnazione degli host
100	50	Sì
60	50	No
40	50	No
25	50	Sì
15	5	Sì
10	5	No

Attention:
Avere meno di 10 host monitorati da un proxy group può portare a una distribuzione non uniforme degli host tra i proxy del gruppo.

Configurazione di un gruppo di proxy

Per configurare un gruppo di proxy nel frontend di Zabbix:

1. Andare su *Administration* > *Proxy groups*.
2. Fare clic su *Create proxy group*.
3. Inserire i parametri del gruppo di proxy nel modulo.



Parametro	Descrizione
Name	Nome del gruppo di proxy.
Failover period	Periodo, espresso in secondi, durante il quale un proxy nel gruppo di proxy deve comunicare con il server Zabbix per essere considerato online (valore predefinito: 1m; intervallo: 10s-15m). Se il proxy non comunica entro questo periodo, il suo stato viene modificato in *Offline* e i relativi host vengono immediatamente ridistribuiti ad altri proxy. Il bilanciamento del carico dei proxy inizia dopo 10 volte questo periodo. Supporta i suffissi temporali (ad esempio, 30s, 1m) e le macro utente.
Minimum number of proxies	Numero minimo di proxy online richiesto per mantenere il gruppo di proxy online (valore predefinito: 1; intervallo: 1-1000). Supporta le macro utente. Questo valore deve essere inferiore al numero totale di proxy nel gruppo. Ad esempio, in un gruppo di 10 proxy, impostare il minimo su 10 farà sì che il gruppo passi offline se un proxy qualsiasi si guasta. I proxy online in un gruppo offline continuano a funzionare normalmente, ma il bilanciamento del carico e l'alta disponibilità non saranno attivi.
Description	Descrizione del gruppo di proxy.
Proxies	Visualizza un elenco di un massimo di cinque proxy (come collegamenti o in testo normale, a seconda delle autorizzazioni dell'utente sui proxy) durante la modifica di un gruppo che contiene proxy.

Configurazione del bilanciamento del carico dei proxy

Per utilizzare il bilanciamento del carico dei proxy, è necessario configurare un gruppo di proxy nel frontend di Zabbix (vedere sopra) e assicurarsi che gli host siano monitorati da un gruppo di proxy, non da proxy singoli (è possibile utilizzare l'aggiornamento di massa degli host per spostare gli host dai proxy al gruppo di proxy).

Se si utilizza Zabbix agent, configurarlo anche come segue:

- Per i controlli passivi, elencare tutti i proxy del gruppo di proxy nel parametro Server.
- Per i controlli attivi, è consigliabile elencare tutti i proxy del gruppo di proxy o il server Zabbix nel parametro ServerActive. Tenere presente che solo Zabbix agent 7.0 (o versione successiva) funzionerà con i gruppi di proxy in modalità attiva.

Se il parametro `ServerActive` include un solo proxy di un gruppo di proxy (o il server Zabbix), l'agent sarà comunque in grado di connettersi al proxy corretto. Quando il servizio dell'agent viene avviato e si connette al proxy specificato, l'agent riceverà e memorizzerà nella cache l'elenco completo degli indirizzi IP dei proxy e il relativo carico corrente all'interno del gruppo. Successivamente, i controlli attivi verranno reindirizzati al proxy online corretto per l'host, in base all'assegnazione corrente di proxy-host all'interno del gruppo di proxy.

Warning:
La specifica di un solo proxy nel parametro `ServerActive` di Zabbix agent può causare la perdita dei dati di monitoraggio se l'agent viene avviato o riavviato mentre il proxy specificato è offline.

Note:
Quando si utilizza Zabbix sender, anche le richieste di dati vengono reindirizzate al proxy online corretto per l'host, in base all'assegnazione corrente di proxy-host all'interno del gruppo di proxy. Tuttavia, se si inviano valori relativi a più host da un file di input, utilizzare l'opzione `-g` per evitare di inviare i dati al proxy errato.

Zabbix agent deve inoltre essere in grado di connettersi a tutti i proxy del gruppo di proxy attraverso il firewall. In caso contrario, i controlli attivi potrebbero bloccarsi o non riuscire durante il reindirizzamento o il failover. Ad esempio:

- Durante i controlli attivi, un proxy può reindirizzare l'agent a un altro proxy. Se tale proxy è bloccato da un firewall, la comunicazione si bloccherà in attesa di una risposta.
- In configurazioni ad alta disponibilità stabili senza un recente ribilanciamento, gli agent potrebbero non contattare mai i proxy di backup. Se le regole del firewall sono cambiate e non sono state testate, il failover potrebbe non riuscire.

Test del bilanciamento del carico del proxy

Per testare il bilanciamento del carico del proxy:

1. Configurare un gruppo di proxy.
2. Assicurarsi che il gruppo di proxy abbia uno stato online.
3. Assicurarsi che gli host siano monitorati da un gruppo di proxy, non da proxy singoli (è possibile utilizzare l'aggiornamento di massa degli host per spostare gli host dai proxy al gruppo di proxy).

4. Attendere alcuni secondi per l'aggiornamento della configurazione e la distribuzione degli host tra i proxy nel gruppo di proxy. Osservare la modifica aggiornando l'elenco degli host in *Amministrazione* > *Proxy*.

Note importanti

- Le trappole SNMP non sono supportate dai proxy nel gruppo di proxy.
- I controlli che dipendono da una configurazione esterna (ad esempio, gli script per i controlli esterni o la configurazione ODBC per i controlli del database) devono avere la stessa configurazione su tutti i proxy nel gruppo di proxy.
- I controlli del database richiedono autorizzazioni estese sull'oggetto/server del database.
- Gli host VMware monitorati da un gruppo di proxy vengono distribuiti casualmente tra i proxy del gruppo. Questo fa sì che ogni proxy memorizzi nella cache tutti i dati VMware, causando un carico aggiuntivo su vCenter.
- Gli host creati in base ai dati di autoregistrazione provenienti da un proxy in un gruppo di proxy vengono impostati per essere monitorati dal gruppo di proxy. Tuttavia, gli host creati in base ai dati di rilevamento della rete provenienti da un proxy in un gruppo di proxy vengono impostati per essere monitorati da quel proxy.

Sincronizzazione della configurazione di monitoraggio

Panoramica

Questa pagina fornisce dettagli sull'aggiornamento della configurazione di monitoraggio per il proxy, ovvero su come le modifiche apportate alla configurazione di monitoraggio sul server vengono sincronizzate con il proxy.

Aggiornamento incrementale

L'aggiornamento della configurazione del proxy è incrementale.
Durante una sincronizzazione della configurazione vengono aggiornate solo le entità modificate (pertanto, se non è stata modificata alcuna entità, non verrà inviato nulla).
Questo approccio consente di risparmiare risorse e di impostare un intervallo più breve (quasi istantaneo) per l'aggiornamento della configurazione del proxy.

Le modifiche alla configurazione del proxy vengono monitorate utilizzando numeri di revisione. Nei dati di configurazione inviati al proxy vengono incluse solo le entità con revisioni superiori a quella della configurazione del proxy.

Le entità incluse in una sincronizzazione della configurazione sono le seguenti:

Entità	Dettagli
dati TLS di autoregistrazione	Tutti i dati TLS di autoregistrazione.
espressioni	Tutte le espressioni (espressioni regolari, tabelle delle espressioni).
configurazione globale	Configurazione globale definita nella tabella 'config'
host	Tutte le proprietà, interfacce, inventario, item, pre-elaborazione degli item, parametri degli item e scenari web di un host.
macro host	Tutte le macro definite su un host e tutti gli ID dei template collegati a esso.
regola di discovery del proxy	Regole di discovery e controlli assegnati a un proxy.

Ciò significa che:

- Se un item viene modificato su un **host**, verrà sincronizzata l'intera configurazione di tale host.
- Se viene modificata un'**espressione regolare**, verranno sincronizzate tutte le espressioni regolari.

Un'eccezione è rappresentata dalle macro host, che vengono inviate anche se è stato modificato qualsiasi elemento dell'host.

Il comando `-R config_cache_reload` sul proxy avvierà anch'esso un aggiornamento incrementale.

Si noti che una sincronizzazione completa della configurazione verrà eseguita all'avvio/riavvio di un proxy, durante un failover HA, se il token di sessione è cambiato oppure se l'aggiornamento della configurazione non è riuscito sul proxy, ad esempio se la connessione è stata interrotta durante la ricezione dei dati di configurazione.

Parametri di configurazione

Il parametro **ProxyConfigFrequency** determina la frequenza con cui la configurazione del proxy viene sincronizzata con il server (10 secondi per impostazione predefinita).

Si noti che ProxyConfigFrequency è:

- parametro del server per i proxy passivi
- parametro del proxy per i proxy attivi

Sui proxy attivi, ProxyConfigFrequency è un nuovo parametro a partire da Zabbix 6.4 e deve essere usato al posto del parametro ConfigFrequency, ora deprecato.

Attention:
Se vengono usati sia ProxyConfigFrequency sia ConfigFrequency, il proxy registrerà un errore e terminerà.

Calcolo delle code durante la manutenzione

Attention:
Il proxy di Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calculation of queues during maintenance per i dettagli.

Protocollo di scambio dati server-proxy

Server - lo scambio di dati con il proxy si basa sul formato JSON.

I messaggi di richiesta e risposta devono iniziare con header and data length.

proxy passivo

Richiesta di configurazione

Il server invia innanzitutto una richiesta `proxy config` vuota. Questa richiesta viene inviata ogni `ProxyConfigFrequency` (parametro di configurazione del server) secondi.

Il proxy risponde con la versione corrente del proxy, il token di sessione e la revisione della configurazione. Il server risponde con i dati di configurazione che devono essere aggiornati.

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	Versione del proxy (<major>.<minor>.<build>).
session	*string*	Token di sessione della configurazione del proxy.
config_revision	*number*	Revisione della configurazione del proxy.
server→proxy:		
full_sync	*number*	1 - se vengono inviati tutti i dati di configurazione; assente altrimenti (facoltativo).
data	*array*	Oggetto contenente i dati delle tabelle. Assente se la configurazione non è stata modificata (facoltativo).
<table>	*object*	Uno o più oggetti con i dati di <table> (facoltativo, a seconda delle modifiche).
fields	*array*	Array di nomi dei campi.
-	*string*	Nome del campo.
data	*array*	Array di righe.
-	*array*	Array di colonne.
-	*string*,*number*	Valore della colonna con il tipo che dipende dal tipo di colonna nello schema del database.
macro.secrets	*object*	Informazioni sulle macro segrete, assenti se non ci sono modifiche nelle macro del vault (facoltativo).
config_revision	*number*	Revisione della cache di configurazione, inviata con i dati di configurazione (facoltativo).
del_hostids	*array*	Array degli hostid rimossi (facoltativo).
-	*number*	Identificatore dell'host.
del_macro_hostids	*array*	Array degli hostid con tutte le macro rimosse (facoltativo).
-	*number*	Identificatore dell'host.
proxy→server:		
response	*string*	Informazioni sull'esito della richiesta ('success' o 'failed').
version	*string*	Versione del proxy (<major>.<minor>.<build>).

Esempio:

server→proxy:

```
{
  "request":"proxy config"
}
```

proxy→server:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

server→proxy:

```
{
    "full_sync": 1,
    "data": {
        "hosts": {
            "fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
                [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
            ]
        },
        "item_rtdata": {
            "fields": ["itemid", "lastlogsize", "mtime"],
            "data": [
                [44161, 0, 0],
                [44162, 0, 0]
            ]
        },
        "item_preproc": {
            "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
            "data": []
        },
        "item_parameter": {
            "fields": ["item_parameterid", "itemid", "name", "value"],
            "data": []
        },
        "globalmacro": {
```

```
            "fields": ["globalmacroid", "macro", "value", "type"],
            "data": [
                [2, "{$SNMP_COMMUNITY}", "public", 0]
            ]
        },
        "hosts_templates": {
            "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
            "data": []
        },
        "hostmacro": {
            "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
            "data": [
                [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
            ]
        },
        "drules": {
            "fields": ["druleid", "name", "iprange", "delay"],
            "data": [
                [2, "Local network", "127.0.0.1", "10s"]
            ]
        },
        "dchecks": {
            "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
            "data": [
                [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
            ]
        },
        "regexps": {
            "fields": ["regexpid", "name"],
            "data": [
                [1, "File systems for discovery"],
                [2, "Network interfaces for discovery"],
                [3, "Storage devices for SNMP discovery"],
                [4, "Windows service names for discovery"],
                [5, "Windows service startup states for discovery"]
            ]
        },
        "expressions": {
            "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
            "data": [
                [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
                [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
                [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
                [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
                [7, 2, "^Software Loopback Interface", 4, ",", 1],
                [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
                [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
                [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
                [11, 2, "^[Ss]ystem$", 4, ",", 1],
                [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
            ]
        },
        "settings": {
            "fields": ["name", "type", "value_str", "value_int"],
            "data": [
                ["autoreg_tls_accept", 2, "", 1],
                ["hk_history_global", 2, "", 0],
                ["snmptrap_logging", 2, "", 1],
                ["proxy_secrets_provider", 2, "", 0],
                ["hk_history", 1, "31d", 0],
                ["timeout_db_monitor", 1, "3s", 0],
                ["timeout_external_check", 1, "3s", 0],
```

```
                ["timeout_http_agent", 1, "3s", 0],
                ["timeout_simple_check", 1, "3s", 0],
                ["timeout_snmp_agent", 1, "3s", 0],
                ["timeout_ssh_agent", 1, "3s", 0],
                ["timeout_telnet_agent", 1, "3s", 0],
                ["timeout_zabbix_agent", 1, "3s", 0],
                ["timeout_browser", 1, "30s", 0]
            ]
        },
        "httptest": {
            "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
            "data": []
        },
        "httptestitem": {
            "fields": ["httptestitemid", "httptestid", "itemid", "type"],
            "data": []
        },
        "httptest_field": {
            "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
            "data": []
        },
        "httpstep": {
            "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
            "data": []
        },
        "httpstepitem": {
            "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
            "data": []
        },
        "httpstep_field": {
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
        }
    },
    "config_revision": 2
}
```


proxy→server:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Richiesta dati

La richiesta `proxy data` viene utilizzata per ottenere la disponibilità delle interfacce host, i dati storici, di discovery e di autoregistrazione dal proxy. Questa richiesta viene inviata ogni `ProxyDataFrequency` (parametro di configurazione del server) secondi.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		

name	value type	description
session	*string*	Token della sessione dati.
interface availability	*array*	*(optional)* Array di oggetti dati sulla disponibilità delle interfacce.
interfaceid	*number*	Identificatore dell'interfaccia.
available	*number*	Disponibilità dell'interfaccia: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - sconosciuta **1**, *INTERFACE_AVAILABLE_TRUE* - disponibile **2**, *INTERFACE_AVAILABLE_FALSE* - non disponibile
error	*string*	Messaggio di errore dell'interfaccia o stringa vuota.
history data	*array*	*(optional)* Array di oggetti dati storici.
itemid	*number*	Identificatore dell'item.
clock	*number*	Timestamp del valore dell'item (secondi).
ns	*number*	Timestamp del valore dell'item (nanosecondi).
value	*string*	*(optional)* Valore dell'item.
id	*number*	Identificatore del valore (contatore crescente, univoco all'interno di una sessione dati).
timestamp	*number*	*(optional)* Timestamp degli item di tipo log.
source	*string*	*(optional)* Valore source dell'item eventlog.
severity	*number*	*(optional)* Valore severity dell'item eventlog.
eventid	*number*	*(optional)* Valore eventid dell'item eventlog.
state	*string*	*(optional)* Stato dell'item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Ultima dimensione del log per gli item di tipo log.
mtime	*number*	*(optional)* Tempo di modifica per gli item di tipo log.
discovery data	*array*	*(optional)* Array di oggetti dati di discovery.
clock	*number*	Timestamp dei dati di discovery.
druleid	*number*	Identificatore della regola di discovery.
dcheckid	*number*	Identificatore del controllo di discovery o null per i dati della regola di discovery.
type	*number*	Tipo di controllo di discovery: **-1** dati della regola di discovery **0**, *SVC_SSH* - controllo del servizio SSH **1**, *SVC_LDAP* - controllo del servizio LDAP **2**, *SVC_SMTP* - controllo del servizio SMTP **3**, *SVC_FTP* - controllo del servizio FTP **4**, *SVC_HTTP* - controllo del servizio HTTP **5**, *SVC_POP* - controllo del servizio POP **6**, *SVC_NNTP* - controllo del servizio NNTP **7**, *SVC_IMAP* - controllo del servizio IMAP **8**, *SVC_TCP* - controllo della disponibilità della porta TCP **9**, *SVC_AGENT* - Zabbix agent **10**, *SVC_SNMPv1* - agent SNMPv1 **11**, *SVC_SNMPv2* - agent SNMPv2 **12**, *SVC_ICMPPING* - ping ICMP **13**, *SVC_SNMPv3* - agent SNMPv3 **14**, *SVC_HTTPS* - controllo del servizio HTTPS **15**, *SVC_TELNET* - controllo della disponibilità Telnet
ip	*string*	Indirizzo IP dell'host.
dns	*string*	Nome DNS dell'host.
port	*number*	*(optional)* Numero di porta del servizio.
key_	*string*	*(optional)* Chiave dell'item per il controllo di discovery di tipo **9** *SVC_AGENT*
value	*string*	*(optional)* Valore ricevuto dal servizio, può essere vuoto per la maggior parte dei servizi.

name	value type	description
status	*number*	*(optional)* Stato del servizio: **0**, *DOBJECT_STATUS_UP* - servizio attivo **1**, *DOBJECT_STATUS_DOWN* - servizio non attivo
auto reg- is- tra- tion	*array*	*(optional)* Array di oggetti dati di autoregistrazione.
clock	*number*	Timestamp dei dati di autoregistrazione.
host	*string*	Nome host.
ip	*string*	*(optional)* Indirizzo IP dell'host.
dns	*string*	*(optional)* Nome DNS risolto dall'indirizzo IP.
port	*string*	*(optional)* Porta dell'host.
host_metadata	*string*	*(optional)* Metadati dell'host inviati dall'agent (in base al parametro di configurazione dell'agent HostMetadata o HostMetadataItem).
tasks	*array*	*(optional)* Array di task.
type	*number*	Tipo di task: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - risultato del comando remoto
status	*number*	Stato di esecuzione del comando remoto: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - comando remoto completato con successo **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - comando remoto non riuscito
error	*string*	*(optional)* Messaggio di errore.
parent_taskid	*number*	ID del task padre.
more	*number*	*(optional)* 1 - ci sono altri dati storici da inviare.
clock	*number*	*(optional)* Timestamp del trasferimento dati (secondi).
ns	*number*	*(optional)* Timestamp del trasferimento dati (nanosecondi).
version	*string*	Versione del proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informazione di successo della richiesta ('success' o 'failed').
tasks	*array*	*(optional)* Array di task.
type	*number*	Tipo di task: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - comando remoto
clock	*number*	Ora di creazione del task.
ttl	*number*	Tempo in secondi dopo il quale il task scade.
commandtype	*number*	Tipo di comando remoto: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - usa script personalizzato **1**, *ZBX_SCRIPT_TYPE_IPMI* - usa IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - usa SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - usa Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - usa script globale (attualmente funzionalmente equivalente a uno script personalizzato)
command	*string*	Comando remoto da eseguire.
execute_on	*number*	Destinazione di esecuzione per gli script personalizzati: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - esegui lo script sull'agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - esegui lo script sul server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - esegui lo script sul proxy
port	*number*	*(optional)* Porta per i comandi Telnet e SSH.
authtype	*number*	*(optional)* Tipo di autenticazione per i comandi SSH.
username	*string*	*(optional)* Nome utente per i comandi Telnet e SSH.
password	*string*	*(optional)* Password per i comandi Telnet e SSH.
publickey	*string*	*(optional)* Chiave pubblica per i comandi SSH.
privatekey	*string*	*(optional)* Chiave privata per i comandi SSH.
parent_taskid	*number*	ID del task padre.

name	value type	description
hostid	*number*	ID dell'host di destinazione.

Esempio:

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
        }
    ]
}
```

Richiesta di task

Lo scambio `proxy tasks` gestisce la consegna e la conferma dei task di comandi remoti tra il server e un proxy passivo. Quando il server interroga il proxy per i risultati dei task, invia una richiesta vuota `proxy tasks` all'intervallo definito da `ProxyDataFrequency`. Il proxy risponde con la propria versione corrente e con eventuali risultati dei task in sospeso (inclusi i timestamp). Il server conferma quindi la ricezione restituendo una `response` e può includere nuovi task da eseguire per il proxy.

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Versione del proxy (<major>.<minor>.<build>).
clock	*number*	*(optional)* Timestamp del trasferimento dati (secondi).
ns	*number*	*(optional)* Timestamp del trasferimento dati (nanosecondi).
server→proxy:		
response	*string*	Informazioni di esito della richiesta ('success' or 'failed').

Esempio:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

Proxy attivo

Richiesta di configurazione

La richiesta `proxy config` viene inviata dal proxy attivo per ottenere i dati di configurazione del proxy. Questa richiesta viene inviata ogni `ProxyConfigFrequency` (parametro di configurazione del proxy) secondi.

name	value type	description
proxy→server:		
request	*string*	'proxy config'
host	*string *	Nome del proxy.
version	*string*	Versione del proxy (<major>.<minor>.<build>).
session	*string*	Token della sessione di configurazione del proxy.
config_revision	*number*	Revisione della configurazione del proxy.
hostmap_revision	*number*	Revisione corrente della mappatura host-proxy.
server→proxy:		
full_sync	*number*	1 - se vengono inviati tutti i dati di configurazione; altrimenti assente (facoltativo).
data	*array*	Oggetto contenente i dati delle tabelle. Assente se la configurazione non è stata modificata (facoltativo).
<table>	*object*	Uno o più oggetti con i dati di <table> (facoltativo, a seconda delle modifiche).
fields	*array*	Array di nomi dei campi.
-	*string*	Nome del campo.
data	*array*	Array di righe.
-	*array*	Array di colonne.
-	*string,number*	Valore della colonna con tipo dipendente dal tipo di colonna nello schema del database.
macro.secrets	*object*	Informazioni sulle macro segrete, assenti se non ci sono modifiche nelle macro del vault (facoltativo).
proxy_group	*string*	Nome del gruppo di proxy a cui appartiene il proxy.
config_revision	*number*	Revisione della cache di configurazione, inviata con i dati di configurazione (facoltativo).
del_hostids	*array*	Array degli hostid rimossi (facoltativo).
-	*number*	Identificatore dell'host.
del_macro_hostids	*array*	Array degli hostid con tutte le macro rimosse (facoltativo).
-	*number*	Identificatore dell'host.

Esempio:

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
```

```
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
```

```
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
```

```
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Richiesta dati

La richiesta `proxy data` viene inviata dal proxy per fornire i dati di disponibilità delle interfacce host, cronologia, discovery e autoregistrazione. Questa richiesta viene inviata ogni `DataSenderFrequency` (parametro di configurazione del proxy) secondi. Nota che il proxy attivo continuerà comunque a interrogare il server Zabbix ogni secondo per le attività di comandi remoti (con una richiesta `proxy data` vuota).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Nome del proxy.
session	*string*	Token della sessione dati.
interface availability	*array*	*(optional)* Array di oggetti di disponibilità delle interfacce.
interfaceid	*number*	Identificatore dell'interfaccia.
available	*number*	Disponibilità dell'interfaccia: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - sconosciuta **1**, *INTERFACE_AVAILABLE_TRUE* - disponibile **2**, *INTERFACE_AVAILABLE_FALSE* - non disponibile
error	*string*	Messaggio di errore dell'interfaccia oppure stringa vuota.
history data	*array*	*(optional)* Array di oggetti di dati storici.
itemid	*number*	Identificatore dell'item.
clock	*number*	Timestamp del valore dell'item (secondi).
ns	*number*	Timestamp del valore dell'item (nanosecondi).
value	*string*	*(optional)* Valore dell'item.
id	*number*	Identificatore del valore (contatore crescente, univoco all'interno di una sessione dati).
timestamp	*number*	*(optional)* Timestamp degli item di tipo log.
source	*string*	*(optional)* Valore source dell'item eventlog.
severity	*number*	*(optional)* Valore severity dell'item eventlog.
eventid	*number*	*(optional)* Valore eventid dell'item eventlog.
state	*string*	*(optional)* Stato dell'item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Ultima dimensione del log per gli item di tipo log.
mtime	*number*	*(optional)* Ora di modifica degli item di tipo log.
discovery data	*array*	*(optional)* Array di oggetti di discovery data.
clock	*number*	Timestamp dei dati di discovery.
druleid	*number*	Identificatore della regola di discovery.

name	value type	description
dcheckid	*number*	Identificatore del controllo di discovery oppure null per i dati della regola di discovery.
type	*number*	Tipo di controllo di discovery: **-1** dati della regola di discovery **0**, *SVC_SSH* - controllo del servizio SSH **1**, *SVC_LDAP* - controllo del servizio LDAP **2**, *SVC_SMTP* - controllo del servizio SMTP **3**, *SVC_FTP* - controllo del servizio FTP **4**, *SVC_HTTP* - controllo del servizio HTTP **5**, *SVC_POP* - controllo del servizio POP **6**, *SVC_NNTP* - controllo del servizio NNTP **7**, *SVC_IMAP* - controllo del servizio IMAP **8**, *SVC_TCP* - controllo della disponibilità della porta TCP **9**, *SVC_AGENT* - agent Zabbix **10**, *SVC_SNMPv1* - agent SNMPv1 **11**, *SVC_SNMPv2* - agent SNMPv2 **12**, *SVC_ICMPPING* - ping ICMP **13**, *SVC_SNMPv3* - agent SNMPv3 **14**, *SVC_HTTPS* - controllo del servizio HTTPS **15**, *SVC_TELNET* - controllo della disponibilità Telnet
ip	*string*	Indirizzo IP dell'host.
dns	*string*	Nome DNS dell'host.
port	*number*	*(optional)* Numero di porta del servizio.
key_	*string*	*(optional)* Chiave dell'item per il controllo di discovery di tipo **9** *SVC_AGENT*
value	*string*	*(optional)* Valore ricevuto dal servizio, può essere vuoto per la maggior parte dei servizi.
status	*number*	*(optional)* Stato del servizio: **0**, *DOBJECT_STATUS_UP* - servizio attivo **1**, *DOBJECT_STATUS_DOWN* - servizio non attivo
autoregistration	*array*	*(optional)* Array di oggetti di autoregistrazione.
clock	*number*	Timestamp dei dati di autoregistrazione.
host	*string*	Nome dell'host.
ip	*string*	*(optional)* Indirizzo IP dell'host.
dns	*string*	*(optional)* Nome DNS risolto dall'indirizzo IP.
port	*string*	*(optional)* Porta dell'host.
host_metadata	*string*	*(optional)* Metadati dell'host inviati dall'agent (in base al parametro di configurazione dell'agent HostMetadata o HostMetadataItem).
tasks	*array*	*(optional)* Array di attività.
type	*number*	Tipo di attività: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - risultato del comando remoto
status	*number*	Stato di esecuzione del comando remoto: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - comando remoto completato con successo **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - comando remoto non riuscito
error	*string*	*(optional)* Messaggio di errore.
parent_taskid	*number*	ID dell'attività padre.
more	*number*	*(optional)* 1 - ci sono altri dati storici da inviare
clock	*number*	*(optional)* Timestamp del trasferimento dati (secondi).
ns	*number*	*(optional)* Timestamp del trasferimento dati (nanosecondi).
version	*string*	Versione del proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informazione di esito della richiesta ('success' o 'failed').

name	value type	description
upload	*string*	Controllo del caricamento per i dati storici (history, autoregistration, disponibilità host, network discovery). Valori possibili: **enabled** - funzionamento normale **disabled** - il server non accetta dati (possibilmente a causa del superamento della cache interna)
tasks	*array*	*(optional)* Array di attività.
type	*number*	Tipo di attività: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - comando remoto
clock	*number*	Ora di creazione dell'attività.
ttl	*number*	Tempo in secondi dopo il quale l'attività scade.
commandtype	*number*	Tipo di comando remoto: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - usa script personalizzato **1**, *ZBX_SCRIPT_TYPE_IPMI* - usa IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - usa SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - usa Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - usa script globale (attualmente funzionalmente equivalente a uno script personalizzato)
command	*string*	Comando remoto da eseguire.
execute_on	*number*	Destinazione di esecuzione per gli script personalizzati: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - esegui lo script sull'agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - esegui lo script sul server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - esegui lo script sul proxy
port	*number*	*(optional)* Porta per i comandi Telnet e SSH.
authtype	*number*	*(optional)* Tipo di autenticazione per i comandi SSH.
username	*string*	*(optional)* Nome utente per i comandi Telnet e SSH.
password	*string*	*(optional)* Password per i comandi Telnet e SSH.
publickey	*string*	*(optional)* Chiave pubblica per i comandi SSH.
privatekey	*string*	*(optional)* Chiave privata per i comandi SSH.
parent_taskid	*number*	ID dell'attività padre.
hostid	*number*	ID dell'host di destinazione.

Example:

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
```

```
}]
}
```

Parametri di configurazione: Zabbix proxy

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del proxy Zabbix.

Specificare i valori dei parametri nel file di configurazione del proxy di Zabbix (`zabbix_proxy.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
AllowRoot	Consente al proxy di essere eseguito come `root`.
AllowUnsupportedDBVersions	Consente al proxy di funzionare con versioni del database non supportate.
APMListenIP	Specifica l'indirizzo IP su cui il collector APM deve rimanere in ascolto per i dati di telemetria in ingresso.
APMListenPort	Specifica la porta su cui il collector APM deve rimanere in ascolto per i dati di telemetria in ingresso.
APMTLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per verificare i certificati delle applicazioni o dei collector OpenTelemetry che inviano dati al collector APM.
APMTLSCertFile	Specifica il percorso completo del file contenente il certificato o la catena di certificati del collector APM.
APMTLSKeyFile	Specifica il percorso completo del file contenente la chiave privata del collector APM.
CacheSize	Imposta la dimensione della cache di configurazione.
ConfigFrequency	Questo parametro è **deprecato** (utilizzare invece ProxyConfigFrequency). Imposta la frequenza di recupero dei dati di configurazione dal server Zabbix.
DataSenderFrequency	Imposta la frequenza di invio dei dati raccolti al server Zabbix.
DBHost	Specifica l'host del database (o la directory del socket).
DBName	Specifica il nome del database o il percorso del file del database per SQLite3.
DBPassword	Specifica la password del database.
DBPort	Specifica la porta dell'host del database Zabbix.
DBSchema	Specifica il nome dello schema del database. Utilizzato per PostgreSQL.
DBSocket	Specifica il percorso del file socket MySQL.
DBUser	Specifica l'utente del database.
DBTLSConnect	Specifica un valore per imporre una connessione TLS al database.
DBTLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del database.
DBTLSCertFile	Specifica il percorso completo del file contenente il certificato del proxy Zabbix per l'autenticazione al database.
DBTLSKeyFile	Specifica il percorso completo del file contenente la chiave privata per l'autenticazione al database.
DBTLSCipher	Specifica l'elenco dei cifrari di crittografia consentiti dal proxy Zabbix per i protocolli TLS fino a TLS v1.2. Supportato solo per MySQL.
DBTLSCipher13	Specifica l'elenco delle suite di cifratura consentite dal proxy Zabbix per il protocollo TLS v1.3. Supportato solo per MySQL, a partire dalla versione 8.0.16.
DebugLevel	Imposta il livello di debug.
EnableRemoteCommands	Consente i comandi remoti dal server Zabbix.
ExternalScripts	Specifica la posizione degli script esterni.
Fping6Location	Specifica la posizione di fping6.
FpingLocation	Specifica la posizione di fping.
HistoryCacheSize	Imposta la dimensione della cache della cronologia.
HistoryIndexCacheSize	Imposta la dimensione della cache degli indici della cronologia.
Hostname	Specifica il nome del proxy.
HostnameItem	Specifica l'item utilizzato per impostare Hostname (se non definito).
HousekeepingFrequency	Imposta la frequenza della procedura di housekeeping (in ore).
Include	Specifica una directory o singoli file da includere nel file di configurazione.
JavaGateway	Specifica l'indirizzo IP (o il nome host) del Java gateway Zabbix.

Parametro	Descrizione
JavaGatewayPort	Specifica la porta del Java gateway Zabbix.
ListenBacklog	Imposta il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Specifica l'indirizzo IP di ascolto del trapper.
ListenPort	Specifica la porta di ascolto del trapper.
LoadModule	Specifica il modulo da caricare all'avvio del proxy.
LoadModulePath	Specifica il percorso completo della posizione dei moduli del proxy.
LogFile	Specifica il file di log del proxy.
LogFileSize	Imposta la dimensione massima del file di log.
LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
LogSlowQueries	Imposta la durata massima che una query del database può impiegare prima di essere registrata (in millisecondi).
LogType	Specifica il tipo di output del log.
MaxConcurrentChecksPerPoll	Imposta il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun poller HTTP agent, poller agent o poller SNMP.
PidFile	Specifica il file PID.
ProxyBufferMode	Specifica il meccanismo di archiviazione per i dati di cronologia, discovery e autoregistrazione (disco/memoria/ibrido).
ProxyConfigFrequency	Imposta la frequenza di recupero dei dati di configurazione dal server Zabbix, in secondi.
ProxyLocalBuffer	Imposta la durata di conservazione locale dei dati (in ore), anche se i dati sono già stati sincronizzati con il server.
ProxyMemoryBufferAge	Imposta l'età massima dei dati nel buffer di memoria del proxy, in secondi.
ProxyMemoryBufferSize	Imposta la dimensione della cache di memoria condivisa per i dati raccolti di cronologia, discovery e registrazione automatica.
ProxyMode	Imposta la modalità operativa del proxy (attiva/passiva).
ProxyOfflineBuffer	Imposta la durata di conservazione dei dati (in ore) in caso di assenza di connettività con il server Zabbix.
Server	Se ProxyMode è impostato su `active`: specifica l'indirizzo IP o il nome DNS del server Zabbix (address:port) oppure il cluster (address:port;address2:port) da cui ottenere i dati di configurazione e a cui inviare i dati. Se ProxyMode è impostato su `passive`: specifica gli indirizzi IP o i nomi DNS del server Zabbix.
SNMPTrapperFile	Specifica il file temporaneo utilizzato per trasferire i dati dal demone trap SNMP al proxy.
SocketDir	Specifica la directory per l'archiviazione dei socket IPC utilizzati dai servizi Zabbix interni.
SourceIP	Specifica l'indirizzo IP di origine.
SSHKeyLocation	Specifica la posizione delle chiavi pubbliche e private per i controlli e le azioni SSH.
SSLCertLocation	Specifica la posizione dei file dei certificati client SSL per l'autenticazione del client.
SSLKeyLocation	Specifica la posizione dei file delle chiavi private SSL per l'autenticazione del client.
SSLCALocation	Specifica la posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato del server SSL.
StartAgentPollers	Imposta il numero di istanze pre-fork dei poller agent asincroni Zabbix.
StartAPMCollector	Impostare su 1 per avviare il collector APM.
StartBrowserPollers	Imposta il numero di istanze pre-fork dei poller degli item browser.
StartDBSyncers	Imposta il numero di istanze pre-fork dei sincronizzatori della cronologia.
StartDiscoverers	Imposta il numero di istanze pre-fork dei worker di discovery.
StartHTTPAgentPollers	Imposta il numero di istanze pre-fork dei poller HTTP agent asincroni.
StartHTTPPollers	Imposta il numero di istanze pre-fork dei poller HTTP.
StartIPMIPollers	Imposta il numero di istanze pre-fork dei poller IPMI.
StartJavaPollers	Imposta il numero di istanze pre-fork dei poller Java.
StartODBCPollers	Imposta il numero di istanze pre-fork dei poller ODBC.
StartPingers	Imposta il numero di istanze pre-fork dei pinger ICMP.
StartPollersUnreachable	Imposta il numero di istanze pre-fork dei poller per gli host non raggiungibili (inclusi IPMI e Java).
StartPollers	Imposta il numero di istanze pre-fork dei poller.
StartPreprocessors	Imposta il numero di istanze pre-avviate dei worker di preprocessing.
StartSNMPPollers	Imposta il numero di istanze pre-fork dei poller SNMP asincroni.
StartSNMPTrapper	Impostare su "1" per avviare un processo trapper SNMP.
StartTelemetryQueryPollers	Imposta il numero di istanze pre-fork dei poller asincroni per le query di telemetria.
StartTrappers	Imposta il numero di istanze pre-fork dei trapper.
StartVMwareCollectors	Imposta il numero di istanze pre-fork dei collector VMware.
StatsAllowedIP	Specifica gli indirizzi IP o i nomi DNS delle istanze Zabbix esterne autorizzate a richiedere statistiche.
TelemetryProvider	Imposta l'origine dei dati di telemetria.

Parametro	Descrizione
Timeout	Specifica per quanto tempo attendere (in secondi) la stabilizzazione della connessione e lo scambio di dati con il proxy Zabbix, l'agent, il servizio web e per i controlli SNMP (ad eccezione degli item SNMP `walk[OID]` e `get[OID]`).
TLSAccept	Specifica il livello di crittografia richiesto per le connessioni in ingresso dal server Zabbix.
TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica dei certificati peer, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCertFile	Specifica il percorso completo del file contenente il certificato del server o la catena di certificati, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCipherAll	Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherAll13	Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert	Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato.
TLSCipherCert13	Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK	Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su PSK.
TLSCipherPSK13	Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su PSK.
TLSConnect	Specifica il livello di crittografia richiesto per le connessioni al server Zabbix.
TLSCRLFile	Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti Zabbix.
TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata del proxy, utilizzata per le comunicazioni crittografate tra i componenti Zabbix.
TLSListen	Richiede connessioni esclusivamente TLS sul socket del trapper.
TLSPSKFile	Specifica il percorso completo del file contenente la chiave precondivisa del proxy, utilizzata per le comunicazioni crittografate con il server Zabbix.
TLSPSKIdentity	Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server Zabbix.
TLSServerCertIssuer	Specifica l'emittente del certificato del server consentito.
TLSServerCertSubject	Specifica il subject del certificato del server consentito.
TmpDir	Specifica la directory temporanea.
TrapperTimeout	Specifica il timeout in secondi per: - il recupero dei dati di configurazione dal server Zabbix; - l'esecuzione di script globali o di comandi remoti.
UnavailableDelay	Imposta la frequenza di verifica della disponibilità dell'host durante il periodo di indisponibilità.
UnreachableDelay	Imposta la frequenza di verifica della disponibilità dell'host durante il periodo di irraggiungibilità.
UnreachablePeriod	Imposta dopo quanti secondi di irraggiungibilità l'host deve essere considerato non disponibile.
User	Riduce i privilegi a quelli di un utente specifico ed esistente nel sistema.
Vault	Specifica il provider del vault.
VaultAppRoleID	Specifica l'ID AppRole del vault HashiCorp.
VaultAppSecretID	Specifica l'ID segreto AppRole del vault HashiCorp.
VaultDBPath	Specifica la posizione da cui recuperare le credenziali del database tramite chiavi.
VaultPrefix	Specifica un prefisso personalizzato per il percorso o la query del vault.
VaultTLSCertFile	Specifica il nome del file del certificato SSL utilizzato per l'autenticazione del client.
VaultTLSKeyFile	Specifica il nome del file della chiave privata SSL utilizzata per l'autenticazione del client.
VaultToken	Specifica il token di autenticazione del vault HashiCorp.
VaultURL	Specifica l'URL del server vault.
VMwareCacheSize	Imposta la dimensione della memoria condivisa per l'archiviazione dei dati VMware.
VMwareFrequency	Imposta la frequenza di raccolta dei dati da un singolo servizio VMware, in secondi.
VMwarePerfFrequency	Imposta la frequenza di recupero delle statistiche dei contatori delle prestazioni da un singolo servizio VMware, in secondi.
VMwareTimeout	Imposta il numero massimo di secondi per cui un `vmware collector` attende una risposta dal servizio VMware.
WebDriverURL	Specifica l'URL dell'interfaccia WebDriver.

Si noti quanto segue:

- I valori predefiniti riflettono i valori predefiniti del demone, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili d'ambiente.

- Zabbix supporta i file di configurazione solo con codifica UTF-8 senza BOM.
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli del parametro

AllowRoot

Consente al proxy di essere eseguito come `root`. Se disabilitato e il proxy viene avviato da `root`, il proxy tenterà invece di passare all'utente `zabbix`. Non ha effetto se avviato da un utente normale.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

AllowUnsupportedDBVersions

Consente al proxy di funzionare con versioni del database non supportate.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

APMListenIP

Specificare l'indirizzo IP su cui l'APM collector deve rimanere in ascolto per i dati di telemetria in ingresso.
 Se non specificato, l'APM collector rimarrà in ascolto su tutte le interfacce di rete.

Predefinito: `0.0.0.0`

APMListenPort

Specificare la porta su cui l'APM collector deve rimanere in ascolto per i dati di telemetria in ingresso.
 La porta predefinita è la porta standard di OpenTelemetry (OTLP/gRPC), utilizzata per impostazione predefinita dagli SDK e dai collector di OpenTelemetry.

Predefinito: 4317
 Intervallo: 1024-32767

APMTLSCAFile

Specificare il percorso completo del file contenente i certificati delle CA di livello superiore per verificare i certificati delle applicazioni o dei collector OpenTelemetry che inviano dati all'APM collector.
 Se questo parametro è impostato, l'APM collector accetta solo connessioni crittografate; in caso contrario, accetta solo connessioni non crittografate.

APMTLSCertFile

Specificare il percorso completo del file contenente il certificato o la catena di certificati del collector APM.
 Questo parametro deve essere impostato insieme a `APMTLSKeyFile` e solo se `APMTLSCAFile` è impostato.

APMTLSKeyFile

Specificare il percorso completo del file contenente la chiave privata del collector APM.
 Questo parametro deve essere impostato insieme a `APMTLSCertFile` e solo se `APMTLSCAFile` è impostato.

CacheSize

Imposta la dimensione della cache di configurazione, in byte. La dimensione della memoria condivisa per memorizzare i dati di host e item.

Predefinito: 32M
 Intervallo: 128K-64G

ConfigFrequency

Questo parametro è **deprecato** (utilizzare invece `ProxyConfigFrequency`).
Imposta la frequenza di recupero dei dati di configurazione dal server Zabbix (in secondi).
Parametro del proxy attivo. Ignorato per i proxy passivi (vedere il parametro `ProxyMode`).

Predefinito: 3600
 Intervallo: 1-604800

DataSenderFrequency

Imposta la frequenza di invio dei dati raccolti dal proxy al server Zabbix. Si noti che un proxy attivo continuerà comunque a interrogare il server Zabbix ogni secondo per le attività dei comandi remoti.
Parametro del proxy attivo. Ignorato per i proxy passivi (vedere il parametro `ProxyMode`).

Predefinito: 1
 Intervallo: 1-3600

DBHost

Specificare l'host del database (o la directory del socket).

Per MySQL:

- `localhost` o una stringa vuota usa il socket UNIX di default.

Per PostgreSQL:

- `localhost` viene risolto tramite DNS (in genere in `127.0.0.1`).
- Una stringa vuota usa il socket UNIX di default.
- Un percorso (ad esempio, `/var/run/pgbouncer`) comporta l'uso del socket UNIX in quel percorso.
- Un elenco separato da virgole può contenere più valori `host:port`. Ogni host viene provato in ordine finché non viene stabilita una connessione in lettura-scrittura. Esempio: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:3005`

Predefinito: `localhost`

DBName

Specificare il nome del database o il percorso al file del database per SQLite3 (l'architettura multiprocesso di Zabbix non consente di utilizzare un database in memoria, ad esempio `:memory:`, `file::memory:?cache=shared` o `file:memdb1?mode=memory&cache=share` non tentare di utilizzare lo stesso database usato dal server Zabbix.

Obbligatorio: Sì

DBPassword

Specifica la password del database. Commenta questa riga se non viene utilizzata alcuna password. Ignorato per SQLite.

DBPort

Specifica la porta dell'host del database Zabbix (vedere DBHost). Ignorato per SQLite.
 DBSocket e DBPort si escludono a vicenda nella configurazione del proxy. Specificarne solo uno, oppure lasciare entrambi non definiti.

Predefinito per MySQL: 3306

Predefinito per PostgreSQL: 5432

Intervallo: 1024-65535

DBSchema

Specifica il nome dello schema del database. Usato per PostgreSQL.

DBSocket

Specifica il percorso del file socket di MySQL.
DBSocket e DBPort si escludono a vicenda nella configurazione del proxy. Specificarne solo uno, oppure lasciare entrambi non definiti.

DBUser

Specifica l'utente del database. Ignorato per SQLite.

DBTLSConnect

Specifica un valore per imporre una connessione TLS al database: - *required* - connettersi usando TLS; - *verify_ca* - connettersi usando TLS e verificare il certificato; - *verify_full* - connettersi usando TLS, verificare il certificato e verificare che l'identità del database specificata da DBHost corrisponda al suo certificato
Su MySQL a partire dalla versione 5.7.11 e su PostgreSQL sono supportati i seguenti valori: "required", "verify", "verify_full".
Su MariaDB a partire dalla versione 10.2.6 sono supportati i valori "required" e "verify_full".
Per impostazione predefinita non è impostato alcun valore e il comportamento dipende dalla configurazione del database.

DBTLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del database.

Obbligatorio: no (sì, se DBTLSConnect è impostato su *verify_ca* o *verify_full*)

DBTLSCertFile

Specifica il percorso completo del file contenente il certificato del proxy Zabbix per l'autenticazione al database.

DBTLSKeyFile

Specifica il percorso completo del file contenente la chiave privata per l'autenticazione al database.

DBTLSCipher

Specifica l'elenco delle cifre di crittografia che il proxy Zabbix consente per i protocolli TLS fino a TLS v1.2. Supportato solo per MySQL.

DBTLSCipher13

Specifica l'elenco delle suite di cifratura che il proxy Zabbix consente per il protocollo TLS v1.3. Supportato solo per MySQL, a partire dalla versione 8.0.16.

DebugLevel

Imposta il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (produce molte informazioni); - *5* - debug esteso (produce ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

EnableRemoteCommands

Consenti i comandi remoti dal server Zabbix.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

ExternalScripts

Specifica il percorso degli script esterni (dipende dalla variabile di installazione definita in fase di compilazione `datadir`).

Predefinito: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Specifica il percorso di fping6. Assicurati che il binario fping6 abbia come proprietario root e che il flag SUID sia impostato. Lascia vuoto (`Fping6Location=`) se la tua utility `fping` è in grado di elaborare indirizzi IPv6.

Predefinito: `/usr/sbin/fping6`

FpingLocation

Specifica la posizione di fping. Assicurati che il binario fping abbia come proprietario root e che sia impostato il flag SUID.

Predefinito: `/usr/sbin/fping`

HistoryCacheSize

Imposta la dimensione della cache della history, in byte. La dimensione della memoria condivisa per l'archiviazione dei dati di history.

Predefinito: `16M`
 Intervallo: 128K-16G

HistoryIndexCacheSize

Imposta la dimensione della cache dell'indice della history, in byte. La dimensione della memoria condivisa per l'indicizzazione dei dati della history memorizzati nella cache della history. La dimensione della cache dell'indice richiede circa 100 byte per memorizzare un item.

Predefinito: `4M`
 Intervallo: 128K-16G

Nome host

Specifica un nome proxy univoco, con distinzione tra maiuscole e minuscole. Assicurati che il nome proxy sia noto al server.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128

Predefinito: impostato da HostnameItem

HostnameItem

Specifica un item per impostare `Hostname` (se non definito).
 Questo item verrà eseguito sul proxy in modo analogo a quanto avviene su un agent.
 Ignorato se `Hostname` è impostato.
Non supporta parametri utente, contatori di prestazioni o alias; supporta `system.run[]`.

Predefinito: `system.hostname`

HousekeepingFrequency

Imposta la frequenza della procedura di housekeeping (in ore). L'housekeeping consiste nella rimozione delle informazioni obsolete dal database.
Nota: per ridurre il carico all'avvio del proxy, l'housekeeping viene posticipato di 30 minuti dopo l'avvio del proxy. Pertanto, se `HousekeepingFrequency` è 1, la primissima procedura di housekeeping dopo l'avvio del proxy verrà eseguita dopo 30 minuti e successivamente si ripeterà ogni ora.
È possibile disabilitare l'housekeeping automatico impostando HousekeepingFrequency su 0. In questo caso, la procedura di housekeeping può essere avviata solo tramite l'opzione di controllo runtime `housekeeper_execute`.
Vedere anche le opzioni di runtime control e i dettagli sulla housekeeping procedure.

Predefinito: 1
 Intervallo: 0-24

Include

Specifica una directory o singoli file da includere nel file di configurazione.
 Durante l'installazione di Zabbix, questo parametro è impostato su `/usr/local/etc`, a meno che non venga modificato durante la compilazione.

Regole generali per il parametro Include:

- Il percorso può essere relativo alla posizione del file zabbix_proxy.conf.
- È possibile specificare più parametri Include in un unico file di configurazione.
- Zabbix elabora il parametro Include non appena lo legge, quindi è consigliabile posizionarlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non si avvierà.

Regole per l'inclusione di directory o pattern di file:

- Non dare per scontato alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup che alcuni editor di testo creano automaticamente (ad esempio, la modifica di include/my_specific.conf può creare include/my_specific.conf.BAK), poiché entrambi i file verranno inclusi.

Per includere solo i file rilevanti da una directory, usa il carattere jolly (*) per la corrispondenza dei pattern.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Specificare l'indirizzo IP o il nome host del Java gateway di Zabbix. È richiesto solo se i poller Java sono avviati.

JavaGatewayPort

Specifica la porta del Zabbix Java gateway.

Predefinito: 10052
 Intervallo: 1024-32767

ListenBacklog

Imposta il numero massimo di connessioni in sospeso nella coda TCP.
Il valore predefinito è una costante codificata, che dipende dal sistema.
Anche il valore massimo supportato dipende dal sistema; valori troppo elevati possono essere troncati silenziosamente al "massimo specificato dall'implementazione".

Predefinito: SOMAXCONN
 Intervallo: 0 - INT_MAX

ListenIP

Specifica l'indirizzo IP di ascolto del trapper. Un elenco delimitato da virgole.
Il trapper ascolterà su tutte le interfacce di rete se questo parametro non viene specificato.

Predefinito: 0.0.0.0

ListenPort

Specifica la porta di ascolto del trapper.

Predefinito: 10051
 Intervallo: 1024-32767

LoadModule

Specificare il modulo da caricare all'avvio del proxy. I moduli vengono utilizzati per estendere le funzionalità del proxy. Il modulo deve trovarsi nella directory specificata da LoadModulePath oppure il percorso deve precedere il nome del modulo. Se il percorso precedente è assoluto (inizia con /), LoadModulePath viene ignorato.
 Formati: - LoadModule=<module.so> - LoadModule=<path/module.so> - LoadModule=</abs_path/module.so>

È possibile specificare più parametri LoadModule.

LoadModulePath

Il percorso completo della posizione dei moduli del proxy. Il valore predefinito dipende dalle opzioni di compilazione.

LogFile

Specifica il file di log del proxy.

Obbligatorio: Sì, se LogType è impostato su file; altrimenti no

LogFileSize

Imposta la dimensione massima del file di log (in MB).
0 - disabilita la rotazione automatica dei log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file non riesce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024

LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

LogSlowQueries

Imposta la durata massima che una query del database può impiegare prima di essere registrata (in millisecondi).
0 - non registrare le query lente.
Questa opzione diventa disponibile a partire da `DebugLevel=3`.

Predefinito: 0
 Intervallo: 0-3600000

LogType

Specifica il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro `LogFile`; - *system* - scrive il log in syslog; - *console* - scrive il log nell'output standard.

Predefinito: `file`

MaxConcurrentChecksPerPoller

Imposta il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun HTTP agent poller, agent poller o SNMP poller. Vedere `StartHTTPAgentPollers`, `StartAgentPollers` e `StartSNMPPollers`.

Predefinito: 1000
 Intervallo: 1-1000

PidFile

Specifica il file PID.

Predefinito: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Specifica il meccanismo di archiviazione per i dati di history, discovery e autoregistrazione: - *disk* - i dati vengono archiviati nel database e caricati dal database; - *memory* - i dati vengono archiviati in memoria e caricati dalla memoria. Se il buffer esaurisce la memoria, i dati più vecchi verranno scartati. All'arresto, il buffer viene scartato. - *hybrid* - il buffer del proxy normalmente funziona come in modalità memory finché non esaurisce la memoria o il record più vecchio supera l'età configurata. Se ciò accade, il buffer viene svuotato nel database e funziona come in modalità disk finché tutti i dati non sono stati caricati, quindi ricomincia a funzionare con la memoria. All'arresto, il buffer in memoria viene svuotato nel database.

Vedi anche: Proxy memory buffer.

Predefinito: `disk`
 Valori: disk; memory; hybrid

ProxyConfigFrequency

Imposta la frequenza di recupero dei dati di configurazione dal server Zabbix in secondi.
Parametro del proxy attivo. Ignorato per i proxy passivi (vedere il parametro `ProxyMode`).

Predefinito: 10
 Intervallo: 1-604800

ProxyLocalBuffer

Imposta la durata della conservazione dei dati in locale (in ore), anche se i dati sono già stati sincronizzati con il server..
Questo parametro può essere usato se i dati locali verranno utilizzati da applicazioni di terze parti.

Predefinito: 0
 Intervallo: 0-720

ProxyMemoryBufferAge

Imposta l'età massima dei dati nel buffer di memoria del proxy in secondi.
 Quando è abilitato (diverso da zero) e i record nel buffer di memoria del proxy sono più vecchi, forza il buffer del proxy a passare alla modalità database fino a quando tutti i record non vengono caricati sul server. Questo parametro deve essere minore o uguale al parametro `ProxyOfflineBuffer`.

Predefinito: 0
 Intervallo: 0;600-864000

ProxyMemoryBufferSize

Imposta la dimensione della cache in memoria condivisa per i dati di history, discovery e auto registration raccolti (in byte). Se abilitato (diverso da zero), il proxy manterrà in memoria i dati di history, discovery e autoregistration, a meno che la cache non sia

piena o i record memorizzati siano più vecchi del valore definito di `ProxyMemoryBufferAge`. Questo parametro non può essere usato insieme al parametro `ProxyLocalBuffer`.

Predefinito: 0
 Intervallo: 0;128K-2G

ProxyMode

Imposta la modalità operativa del proxy.
0 - proxy in modalità attiva
1 - proxy in modalità passiva
Nota che i dati di configurazione del proxy (sensibili) possono diventare disponibili a soggetti che hanno accesso alla porta trapper del server Zabbix quando si utilizza un proxy attivo. Ciò è possibile perché chiunque può fingersi un proxy attivo e richiedere i dati di configurazione; l'autenticazione non viene eseguita.

Predefinito: 0
 Intervallo: 0-1

ProxyOfflineBuffer

Imposta la durata di conservazione dei dati (in ore) in caso di assenza di connettività con il server Zabbix.
I dati più vecchi andranno persi.

Predefinito: 1
 Intervallo: 1-720

Server

Se ProxyMode è impostato su `active`: specificare l'indirizzo IP o il nome DNS del server Zabbix (address:port) oppure il cluster (address:port;address2:port) da cui ottenere i dati di configurazione e a cui inviare i dati.
Se la porta non è specificata, viene utilizzata la porta predefinita.
I nodi del cluster devono essere separati da un punto e virgola.

Se ProxyMode è impostato su `passive`: specificare gli indirizzi IP, facoltativamente in notazione CIDR, oppure i nomi DNS del server Zabbix. Un elenco delimitato da virgole.
 Le connessioni in ingresso saranno accettate solo dagli indirizzi elencati qui. Se il supporto IPv6 è abilitato, `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo.
`::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obbligatorio: sì

SNMPTrapperFile

Specifica il file temporaneo utilizzato per passare i dati dal demone SNMP trap al proxy.
Deve essere lo stesso di `zabbix_trap_receiver.pl` o del file di configurazione SNMPTT.

Predefinito: `/tmp/zabbix_traps.tmp`

SocketDir

Specifica la directory in cui archiviare i socket IPC utilizzati dai servizi interni di Zabbix.

Predefinito: `/tmp`

SourceIP

Specifica l'indirizzo IP di origine per:

- connessioni in uscita verso Zabbix server
- connessioni senza agent (VMware, SSH, JMX, SNMP, Telnet e controlli semplici)
- connessioni HTTP agent
- richieste HTTP JavaScript dell'item script
- richieste HTTP JavaScript del preprocessing
- connessioni al Vault

SSHKeyLocation

Specifica il percorso delle chiavi pubbliche e private per i controlli SSH e le azioni.

SSLCertLocation

Specificare la posizione dei file del certificato client SSL per l'autenticazione del client.
Questo parametro viene utilizzato nel monitoraggio web, per la comunicazione con Vault e per la sorgente dati di telemetria ClickHouse.

SSLKeyLocation

Specifica la posizione dei file delle chiavi private SSL per l'autenticazione del client.
Questo parametro viene utilizzato nel monitoraggio web, per la comunicazione con Vault e per la fonte dati di telemetria ClickHouse.

SSLCALocation

Specificare la posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato del server SSL.
Si noti che il valore di questo parametro verrà impostato come opzione CURLOPT_CAPATH di libcurl. Per le versioni di libcurl precedenti alla 7.42.0, questa impostazione ha effetto solo se libcurl è stato compilato per utilizzare OpenSSL. Per ulteriori informazioni, consultare la pagina web di cURL.
Questo parametro viene utilizzato nel monitoraggio web, nell'autenticazione SMTP, negli item HTTP agent, per la comunicazione con Vault e per la sorgente dati di telemetria ClickHouse.

StartAgentPollers

Imposta il numero di istanze pre-forked di Zabbix agent pollers. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartAPMCollector

Impostare su 1 per avviare un'istanza pre-forked di un collector APM (i processi apm manager e apm worker).

Predefinito: 0
 Intervallo: 0-1

StartBrowserPollers

Imposta il numero di istanze pre-forked di browser item pollers.

Predefinito: 1
 Intervallo: 0-1000

StartDBSyncers

Imposta il numero di istanze pre-forked di history syncers.
Nota: fai attenzione quando modifichi questo valore; aumentarlo potrebbe fare più male che bene.

Predefinito: 4
 Intervallo: 1-100

StartDiscoverers

Imposta il numero di istanze pre-forked di discovery workers.

Predefinito: 5
 Intervallo: 0-1000

StartHTTPAgentPollers

Imposta il numero di istanze pre-forked di HTTP agent pollers. Vedi MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartHTTPPollers

Imposta il numero di istanze pre-forked di HTTP pollers.

Predefinito: 1
 Intervallo: 0-1000

StartIPMIPollers

Imposta il numero di istanze pre-forked di IPMI pollers.

Predefinito: 0
 Intervallo: 0-1000

StartJavaPollers

Imposta il numero di istanze pre-forked di Java pollers.

Predefinito: 0
 Intervallo: 0-1000

StartODBCPollers

Imposta il numero di istanze pre-forked di ODBC pollers.

Predefinito: 1
 Intervallo: 0-1000

StartPingers

Imposta il numero di istanze pre-forked di ICMP pingers.

Predefinito: 1
 Intervallo: 0-1000

StartPollersUnreachable

Imposta il numero di istanze pre-forked di pollers per host non raggiungibili (inclusi IPMI e Java). Deve essere in esecuzione almeno un poller per host non raggiungibili se vengono avviati poller regolari, IPMI o Java.

Predefinito: 1
 Intervallo: 0-1000

StartPollers

Imposta il numero di istanze pre-forked di pollers.

Predefinito: 5
 Intervallo: 0-1000

StartPreprocessors

Imposta il numero di istanze pre-avviate dei thread di preprocessing worker in modo che sia non inferiore al numero di core CPU disponibili. È opportuno impostare un numero maggiore di worker se il preprocessing non è vincolato dalla CPU e comporta molte richieste di rete.

Predefinito: 16
 Intervallo: 1-1000

StartSNMPPollers

Imposta il numero di istanze pre-forked di SNMP pollers. Vedi MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartTelemetryQueryPollers

Imposta il numero di istanze pre-forked dei poller di query Telemetry asincrone. Vedi MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartSNMPTrapper

Impostare su "1" per avviare un processo SNMP trapper.

Predefinito: 0
 Intervallo: 0-1

StartTrappers

Imposta il numero di istanze pre-forked di trappers.
I trappers accettano connessioni in ingresso da Zabbix sender e dagli agent attivi.

Predefinito: 5
 Intervallo: 0-1000

StartVMwareCollectors

Imposta il numero di istanze di vmware collector pre-forked.

Predefinito: 0
 Intervallo: 0-250

StatsAllowedIP

Specifica gli indirizzi IP (facoltativamente in notazione CIDR) o i nomi DNS delle istanze Zabbix esterne autorizzate a richiedere le statistiche. Elenco separato da virgole.
 Se questo parametro non è impostato, non verrà accettata alcuna richiesta di statistiche.
Se il supporto IPv6 è abilitato, `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` vengono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4.

Esempio:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Impostare la fonte dei dati di telemetria utilizzando il formato `<provider>[;<options>]` dove:

- `<provider>` - una delle fonti di dati di telemetria supportate: `clickhouse`
- `<options>` - coppie `key=value` delimitate da virgole. I valori possono essere racchiusi tra virgolette.

Opzioni per ClickHouse:

- `url` - impostare l'URL HTTP o HTTPS utilizzato per connettersi alla fonte dei dati di telemetria.
- `db` - impostare il nome del database.
- `username` - impostare il nome utente utilizzato per accedere al database.
- `password` - impostare la password utilizzata per accedere al database.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - impostare queste opzioni per configurare una connessione sicura (TLS/SSL) al database.
- `source_ip` - impostare l'indirizzo IP di origine utilizzato per le connessioni in uscita verso la fonte dei dati di telemetria.
- `vault_path` - impostare il percorso di un segreto Vault contenente i valori di `username` e `password`. Non utilizzare insieme alle opzioni `username` e `password`.

Esempio per ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Esempio per ClickHouse con Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix server, agent, web service e i controlli SNMP legacy (numero OID singolo o stringa).

Questo parametro definisce la durata di varie operazioni di comunicazione:

- Esecuzione di comandi remoti su Zabbix agent
- Esecuzione di comandi SSH / Telnet
- Riprogrammazione degli item quando l'interfaccia IPMI diventa non disponibile
- Invio della risposta a Zabbix server quando non è possibile scambiare dati a causa di problemi di autorizzazioni o crittografia
- Scadenza per i socket asincroni IPC e le opzioni di controllo runtime
- Richieste DNS dell'asincrono poller
- Risposta per l'heartbeat dei controlli attivi
- Recupero dei dati (valori) di Zabbix agent dagli agent attivi
- Recupero dei dati da Zabbix sender
- Invio dell'elenco dei controlli attivi a Zabbix agent

Questo timeout **non** verrà usato per quei controlli che hanno configurazioni di flexible timeout nel frontend (a livello globale, di proxy o per singolo item). Ad esempio, gli item SNMP `walk[OID]` e `get[OID]` usano il timeout configurato nel frontend; i controlli SNMP legacy continuano a usare il valore di timeout del server.

Default: 3
 Range: 1-30

TLSAccept

Specifica il livello di crittografia richiesto per le connessioni in ingresso da Zabbix server. Usato per un proxy passivo, ignorato su un proxy attivo. È possibile specificare più valori, separati da virgole: - *unencrypted* - accetta connessioni senza crittografia (predefinito); - *psk* - accetta connessioni con TLS e una chiave precondivisa (PSK); - *cert* - accetta connessioni con TLS e un certificato.

Obbligatorio: sì per un proxy passivo, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCAFile

Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Specifica il percorso completo del file contenente il certificato del proxy o la catena di certificati, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCipherAll

Specifica la stringa di priorità GnuTLS o la stringa dei cipher OpenSSL (TLS 1.2). Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificati e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificati e PSK.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Esempio per OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificato.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Esempio per OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione della ciphersuite per la crittografia basata su certificato.

TLSCipherPSK

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Esempio per OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Specifica il livello di crittografia richiesto per le connessioni del proxy al server Zabbix. Usato per un proxy attivo, ignorato per un proxy passivo. Può essere specificato un solo valore: - *unencrypted* - connessione senza crittografia (predefinito); - *psk* - connessione tramite TLS e una chiave precondivisa (PSK); - *cert* - connessione tramite TLS e un certificato.

Obbligatorio: sì per un proxy attivo, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCRLFile

Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata del proxy, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSListen

Richiedi connessioni solo TLS sul socket trapper.

Valori supportati:

- required - accetta solo connessioni TLS

TLSPSKFile

Specifica il percorso completo del file contenente la chiave pre-condivisa del proxy, utilizzata per le comunicazioni crittografate con il server Zabbix.

TLSPSKIdentity

Specifica la stringa di identità della chiave pre-condivisa, utilizzata per le comunicazioni crittografate con Zabbix server.

TLSServerCertIssuer

Specifica l'emittente del certificato del server consentito.

TLSServerCertSubject

Specifica il subject del certificato del server consentito.

TmpDir

Specifica la directory temporanea.

Predefinito: `/tmp`

TrapperTimeout

Specifica il timeout in secondi per:
 - il recupero dei dati di configurazione dal server Zabbix;
 - l'esecuzione di script globali o l'esecuzione di comandi remoti.

Predefinito: 300
 Intervallo: 1-300

UnavailableDelay

Imposta la frequenza di controllo della disponibilità dell'host durante il periodo di indisponibilità in secondi.

Predefinito: 60
 Intervallo: 1-3600

UnreachableDelay

Imposta la frequenza di controllo della disponibilità dell'host durante il periodo di irraggiungibilità in secondi.

Predefinito: 15
 Intervallo: 1-3600

UnreachablePeriod

Imposta dopo quanti secondi di unreachability considerare l'host non disponibile.

Predefinito: 45
 Intervallo: 1-3600

Utente

Riduce i privilegi a un utente specifico già esistente nel sistema.
Ha effetto solo se eseguito come `root` e `AllowRoot` è disabilitato.

Predefinito: `zabbix`

Vault

Specificare il provider del vault: - *HashiCorp* - HashiCorp KV Secrets Engine versione 2; - *CyberArk* - CyberArk Central Credential Provider
Deve corrispondere al provider del vault impostato nel frontend.

Predefinito: `HashiCorp`

VaultAppRoleID

Specifica l'AppRole ID di HashiCorp vault.
 Se impostato, abilita l'autenticazione AppRole. Non può essere usato con VaultToken.
 Supportato solo per HashiCorp vault.

VaultAppSecretID

Specifica il secret ID dell'AppRole di HashiCorp vault.
 Supportato solo per HashiCorp vault.

VaultDBPath

Specifica il percorso per recuperare le credenziali del database tramite chiavi. Si tratta di un percorso o di una query di Vault, a seconda del Vault.

Le chiavi usate per HashiCorp sono `password` e `username`.

Esempio di percorso con `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Esempio di percorso senza VaultPrefix:

```
secret/zabbix/database
```

Le chiavi usate per CyberArk sono `Content` e `UserName`.

Esempio:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

Questa opzione può essere utilizzata solo se `DBUser` e `DBPassword` non sono specificati.

VaultPrefix

Specifica un prefisso personalizzato per il percorso o la query del vault, a seconda del Vault. Se non specificato, verranno utilizzati i valori predefiniti più adatti.
Nota che 'data' viene aggiunto automaticamente dopo il mountpoint per HashiCorp se `VaultPrefix` non è specificato.
Nota che 'data' viene aggiunto automaticamente dopo il mountpoint per HashiCorp se `VaultPrefix` non è specificato.

Esempio di prefisso per HashiCorp:

```
/v1/secret/data/zabbix/
```

Esempio di prefisso per Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Specifica il nome del file del certificato SSL utilizzato per l'autenticazione del client. Il file del certificato deve essere in formato PEM1.
Se il file del certificato contiene anche la chiave privata, lascia vuoto il campo del file della chiave SSL.
La directory contenente questo file è specificata dal parametro di configurazione SSLCertLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultTLSKeyFile

Specifica il nome del file della chiave privata SSL utilizzato per l'autenticazione del client. Il file della chiave privata deve essere in formato PEM1.
La directory contenente questo file è specificata dal parametro di configurazione SSLKeyLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultToken

Specifica il token di autenticazione di HashiCorp Vault che dovrebbe essere stato generato esclusivamente per il proxy Zabbix con autorizzazione di sola lettura al percorso specificato nel parametro di configurazione opzionale VaultDBPath.
Si verifica un errore se `VaultToken` e la variabile di ambiente `VAULT_TOKEN` sono definiti contemporaneamente.
 Non può essere usato con VaultAppRoleID.
 Supportato solo per HashiCorp Vault.

VaultURL

Specifica l'URL del server vault. La directory dei certificati CA a livello di sistema verrà utilizzata se SSLCALocation non è specificato.

Predefinito: `https://127.0.0.1:8200`

VMwareCacheSize

Imposta la dimensione della memoria condivisa per memorizzare i dati VMware.
È possibile usare un controllo interno di Zabbix `zabbix[vmware,buffer,...]` per monitorare l'utilizzo della cache VMware (vedere Controlli interni).
Nota che la memoria condivisa non viene allocata se non sono configurate istanze del collector vmware da avviare.

Predefinito: 8M
 Intervallo: 256K-2G

VMwareFrequency

Imposta la frequenza di raccolta dei dati da un singolo servizio VMware (in secondi).
Questa frequenza deve essere impostata sul più breve intervallo di aggiornamento di qualsiasi item di monitoraggio VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwarePerfFrequency

Imposta la frequenza di recupero delle statistiche dei contatori di prestazioni da un singolo servizio VMware (in secondi).
Questa frequenza dovrebbe essere impostata sull'intervallo di aggiornamento minimo di qualsiasi item di monitoraggio VMware che utilizza i contatori di prestazioni VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwareTimeout

Imposta il numero massimo di secondi che un `vmware collector` attenderà una risposta dal servizio VMware (vCenter o hypervisor ESX).

Predefinito: 10
 Intervallo: 1-300

WebDriverURL

Specifica l'URL dell'interfaccia WebDriver.

Esempio (usato con il server standalone di Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

agent

Panoramica Zabbix agent viene distribuito su un target di monitoraggio per monitorare attivamente le risorse e le applicazioni locali (dischi rigidi, memoria, statistiche del processore, ecc.).

L'agent raccoglie localmente le informazioni operative e invia i dati al server Zabbix per un'ulteriore elaborazione. In caso di errori (ad esempio, un disco rigido pieno o un processo di servizio arrestato in modo anomalo), il server Zabbix può avvisare attivamente gli amministratori della macchina specifica che ha segnalato l'errore.

Gli agent Zabbix sono altamente efficienti perché utilizzano chiamate di sistema native per raccogliere informazioni statistiche.

È possibile eseguire più istanze dell'agent su un host.

Vedere anche:

- Protocolli dei controlli passivi e attivi
- Esecuzione dell'agent come root
- Parametri di configurazione: Zabbix agent
- Parametri di configurazione: Zabbix agent (Windows)
- Zabbix agent su Microsoft Windows
- Zabbix agent su IBM AIX

Controlli passivi e attivi

Gli agent Zabbix possono eseguire controlli passivi e attivi:

- Controlli passivi - l'agent Zabbix risponde a una richiesta da parte di Zabbix server (o proxy). Ad esempio, il server richiede dati (ad esempio, il carico della CPU) e l'agent restituisce il risultato.
- Controlli attivi - l'agent Zabbix raccoglie e invia i dati senza attendere una richiesta da parte di Zabbix server (o proxy). Per prima cosa, recupera un elenco di item di monitoraggio da server (carico della CPU, memoria disponibile, ecc.), quindi raccoglie i dati richiesti e invia periodicamente nuovi valori al server.

Il tipo di controllo dell'agent viene configurato selezionando il rispettivo tipo di item di monitoraggio. L'agent Zabbix elabora gli item di tipo *Zabbix agent* o *Zabbix agent (active)*.

Piattaforme supportate Per le piattaforme supportate, fare riferimento alla pagina Requisiti.

Agent sui sistemi Unix-like Zabbix agent sui sistemi Unix-like viene eseguito sull'host monitorato.

Installazione

Zabbix agent può essere installato sui sistemi basati su Linux utilizzando uno dei seguenti metodi:

- Pacchetti Zabbix - selezionare il componente Agent (dopo aver selezionato la versione di Zabbix, la distribuzione del sistema operativo e la versione del sistema operativo) e seguire le istruzioni fornite;
- Sorgenti Zabbix - scaricare i file sorgente e compilare Zabbix agent configurando i sorgenti con l'opzione `--enable-agent`.

Attention:
In generale, gli agent Zabbix a 32 bit funzionano sui sistemi a 64 bit, ma in alcuni casi potrebbero non funzionare.

Note:
I binari precompilati di Zabbix agent sono disponibili per il download per macOS, IBM AIX, FreeBSD, OpenBSD e Solaris. I binari legacy, compatibili con la versione corrente di Zabbix server/proxy, sono disponibili per NetBSD e HP-UX. Per installare ed eseguire l'agent su IBM AIX, vedere Zabbix agent su IBM AIX.

Se installato come pacchetto

Zabbix agent viene eseguito come processo daemon. L'agent può essere avviato eseguendo:

```
systemctl start zabbix-agent
```

Questo funzionerà sulla maggior parte dei sistemi GNU/Linux. Su altri sistemi potrebbe essere necessario eseguire:

```
/etc/init.d/zabbix-agent start
```

Per arrestare, riavviare o controllare lo stato di Zabbix agent, utilizzare i seguenti comandi:

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

Se installato da un archivio binario

Dopo aver estratto l'archivio, individuare il binario `zabbix_agentd` e il file di configurazione `zabbix_agentd.conf`. La loro posizione all'interno dell'archivio dipende dalla piattaforma (ad esempio, `sbin/` e `conf/`, oppure `usr/sbin/` e `etc/zabbix/`). Eseguire direttamente il binario dalla directory estratta e specificare il file di configurazione con l'opzione `-c`. Ad esempio:

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Nei sistemi simili a UNIX, Zabbix agent è progettato per essere eseguito da un utente non root. Se viene avviato come root, passa all'utente `zabbix` (oppure all'utente specificato dal parametro [User](<PARAMS>#user)), a meno che nel file di configurazione non sia impostato [AllowRoot=1](<PARAMS>#allowroot). Questo utente deve esistere nel sistema; in caso contrario, l'agent non viene avviato.

Su IBM AIX, l'agent può essere eseguito anche con il System Resource Controller (SRC) e avviato all'avvio del sistema; vedere Zabbix agent su IBM AIX.

Avvio manuale

È possibile avviare Zabbix agent eseguendo direttamente il binario `zabbix_agentd`, ad esempio:

```
zabbix_agentd
```

Questa operazione funziona quando la directory contenente `zabbix_agentd` è inclusa in `PATH`.

Agent sui sistemi Windows Zabbix agent su Windows viene eseguito come servizio Windows.

Installazione

Zabbix agent può essere installato su Windows utilizzando uno dei seguenti metodi:

- Binarî precompilati di Zabbix agent - scaricare il pacchetto di installazione MSI di Zabbix agent e seguire le istruzioni nella pagina Installazione di Zabbix agent su Windows da MSI;
- Sorgenti di Zabbix - scaricare i file sorgente e seguire le istruzioni nella pagina Compilazione di Zabbix agent su Windows.

Vedere la pagina Zabbix agent su Microsoft Windows per ulteriori dettagli sull'installazione di Zabbix agent (da un archivio ZIP) come servizio Windows.

Opzioni I seguenti parametri della riga di comando possono essere utilizzati con Zabbix agent:

Parametro	Descrizione
UNIX e Windows agent	
`-c --config <config-file>`	Percorso del file di configurazione. È possibile utilizzare questa opzione per specificare un file di configurazione diverso da quello predefinito.
`-f --foreground`	Esegue Zabbix agent in primo piano (predefinito: `true`).
`-p --print`	Stampa gli item conosciuti ed esce. Per restituire anche i risultati dei parametri utente, è necessario specificare il file di configurazione (se non si trova nella posizione predefinita).
`-t --test <item key>`	Testa l'item specificato ed esce. Per restituire anche i risultati dei parametri utente, è necessario specificare il file di configurazione (se non si trova nella posizione predefinita).
`-T --test-config`	Convalida il file di configurazione ed esce.
`-h --help`	Visualizza le informazioni della guida.
`-V --version`	Visualizza il numero di versione.
Solo UNIX agent	
`-R --runtime-control <option>`	Esegue funzioni amministrative. Vedere controllo runtime.
Solo Windows agent	
`-m --multiple-agents`	Utilizza più istanze dell'agent (con le opzioni `-i`, `-d`, `-s`, `-x`). Per distinguere i nomi dei servizi delle istanze, ogni nome del servizio includerà il valore Hostname del file di configurazione specificato.

Parametro	Descrizione
`-S --startup-type <value>`	Imposta il tipo di avvio del servizio Zabbix Windows agent. Valori consentiti: `automatic` - *(predefinito)* avvia automaticamente il servizio all'avvio di Windows; `delayed` - ritarda l'avvio del servizio finché i servizi avviati automaticamente non hanno completato l'avvio (disponibile in Windows Server 2008/Vista e versioni successive); `manual` - avvia manualmente il servizio (da parte di un utente o di un'applicazione); `disabled` - disabilita il servizio, impedendone l'avvio da parte di un utente o di un'applicazione. È possibile utilizzare questa opzione insieme all'opzione `-i` oppure separatamente per modificare il tipo di avvio di un servizio già installato.
`-i --install`	Installa Zabbix Windows agent come servizio.
`-d --uninstall`	Disinstalla il servizio Zabbix Windows agent.
`-s --start`	Avvia il servizio Zabbix Windows agent.
`-x --stop`	Arresta il servizio Zabbix Windows agent.

Esempi specifici di utilizzo dei parametri della riga di comando:

- Stampa tutti gli item integrati dell'agent con i relativi valori.
- Testa un parametro utente con la chiave `mysql.ping` definita nel file di configurazione specificato.
- Installa un servizio Zabbix Agent per Windows utilizzando il percorso predefinito del file di configurazione.
- Installa un servizio Zabbix Agent [Hostname] per Windows utilizzando il file di configurazione situato nella stessa cartella dell'eseguibile dell'agent e rende univoco il nome del servizio aggiungendovi il valore `Hostname` dal file di configurazione.
- Modifica il tipo di avvio di un servizio Zabbix Agent per Windows installato utilizzando il file di configurazione situato nella stessa cartella dell'eseguibile dell'agent.

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

Controllo in fase di esecuzione

Con le opzioni di controllo in fase di esecuzione è possibile modificare il livello di log dei processi dell'agent.

Opzione	Descrizione	Destinazione
`log_level_increase[=<target>]`	Aumenta il livello di log. Se la destinazione non è specificata, vengono interessati tutti i processi.	La destinazione può essere specificata come: **tipo di processo** - tutti i processi del tipo specificato (ad esempio, `listener`) Vedere tutti i tipi di processo dell'agent. **tipo di processo,N** - tipo e numero del processo (ad esempio, `listener,3`) **pid** - identificatore del processo (1 a 65535). Per valori maggiori, specificare la destinazione come 'tipo di processo,N'.
`log_level_decrease[=<target>]`	Riduce il livello di log. Se la destinazione non è specificata, vengono interessati tutti i processi.	
`userparameter_reload`	Ricarica i valori delle opzioni `UserParameter` e `Include` dal file di configurazione corrente.	

Esempi:

- Aumento del livello di log di tutti i processi
- Aumento del livello di log del terzo processo listener
- Aumento del livello di log del processo con PID 1234
- Riduzione del livello di log di tutti i processi di controllo attivo

```
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:
Il controllo in fase di esecuzione non è supportato su OpenBSD, NetBSD e Windows.

Tipi di processo dell'agent I tipi di processo dell'agent sono i seguenti:

- `active checks` - processo per l'esecuzione dei controlli attivi;
- `collector` - processo per la raccolta dei dati;
- `listener` - processo per l'ascolto dei controlli passivi.

Il file di log dell'agent può essere utilizzato per osservare questi tipi di processo.

Il file di log dell'agent viene creato con autorizzazioni di lettura e scrittura solo per il proprietario del file. Inoltre, il file è leggibile dal gruppo del proprietario. Tutte le altre autorizzazioni sono negate.

Esegui come utente Zabbix agent su UNIX è progettato per essere eseguito come utente non root. Verrà eseguito con l'utente non root con cui è stato avviato. Pertanto, è possibile eseguire agent con qualsiasi utente non root senza problemi.

Se si tenta di eseguirlo come "root", passerà all'utente "zabbix" codificato nel programma, che deve essere presente nel sistema. È possibile eseguire agent come "root" solo modificando di conseguenza il parametro `AllowRoot` nel file di configurazione di agent.

File di configurazione Il percorso predefinito del file di configurazione dipende dal sistema operativo e dal metodo di installazione di Zabbix:

- Su UNIX, il valore predefinito è:
  - Per l'installazione dai sorgenti: `/usr/local/etc/zabbix_agentd.conf`, oppure quello impostato dalle variabili di compilazione `--sysconfdir` o `--prefix`.
  - Per l'installazione dai pacchetti e dall'appliance: `/etc/zabbix/zabbix_agentd.conf`.
- Su Windows, il valore predefinito è `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`.

Se su un host è in esecuzione una singola istanza dell'agent, questa può utilizzare il file di configurazione predefinito oppure un file di configurazione specificato nella riga di comando. In caso di più istanze, ogni istanza dell'agent deve avere il proprio file di configurazione (una delle istanze può utilizzare il file di configurazione predefinito).

Per informazioni dettagliate sulla configurazione dell'agent Zabbix, vedere le opzioni del file di configurazione per l'agent Zabbix Unix o l'agent Zabbix Windows.

Locale Si noti che l'agent richiede una locale UTF-8 affinché alcuni item testuali dell'agent possano restituire il contenuto previsto. La maggior parte dei moderni sistemi Unix-like ha una locale UTF-8 come impostazione predefinita; tuttavia, in alcuni sistemi potrebbe essere necessario impostarla esplicitamente.

Codice di uscita L'agent Zabbix restituisce 0 in caso di uscita riuscita e 1 in caso di errore.

Protocolli di controllo passivi e attivi

Questa sezione fornisce i dettagli del protocollo relativi ai controlli passivi e attivi eseguiti da Zabbix agent e Zabbix agent 2.

Zabbix utilizza un protocollo di comunicazione basato su JSON per comunicare con gli agent.

I protocolli di Zabbix agent e Zabbix agent 2 sono stati unificati a partire da Zabbix 7.0. La differenza tra le richieste/risposte di Zabbix agent e Zabbix agent 2 è espressa dal valore del tag "variant".

Controlli passivi

Un controllo passivo è una semplice richiesta di dati. Il server o il proxy Zabbix richiede alcuni dati (ad esempio, il carico della CPU) e l'agent Zabbix invia il risultato al server.

I controlli passivi vengono eseguiti in modo asincrono: non è necessario ricevere la risposta a una richiesta prima che vengano avviati altri controlli. Anche la risoluzione DNS è asincrona.

Il poller dell'agent tenterà di connettersi a tutti gli indirizzi restituiti dalla ricerca DNS. In questo modo, se un indirizzo IP non è raggiungibile, il poller proverà il successivo indirizzo disponibile, aumentando la probabilità di una connessione riuscita. Questo miglioramento si applica sia al server Zabbix sia al proxy.

La concorrenza massima dei controlli asincroni è 1000 (definita da MaxConcurrentChecksPerPoller).

Il numero di poller asincroni dell'agent è definito dal parametro StartAgentPollers.

Richiesta del server

Per la definizione dell'intestazione e della lunghezza dei dati, fare riferimento ai dettagli del protocollo.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	"passive checks"
data		*array of object*	yes	Passive check item.
	key	*string*	yes	Item key with expanded macros.
	timeout	*number*	yes	Communication timeout.

Risposta dell'agent

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Field	Type	Mandatory	Value
version	*string*	yes	The agent version number.
variant	*number*	yes	The agent variant (*1* - Zabbix agent, *2* - Zabbix agent 2).
data	*array of object*	yes	Contains the result of the check.
value	*string*	no	The item value if the check was successful.
error	*string*	no	The error message if the check was not successful.

Ad esempio, per gli item supportati:

1. Il server apre una connessione TCP
2. Il server invia **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"agent.ping","timeout":3}]}**
3. L'agent legge la richiesta e risponde con **<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"value":1}]}**
4. Il server elabora i dati per ottenere il valore, '1' nel nostro caso
5. La connessione TCP viene chiusa

Per gli item non supportati:

1. Il server apre una connessione TCP
2. Il server invia **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"vfs.fs.size[/nono]","timeout":3}]}**
3. L'agent legge la richiesta e risponde con **<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"error":"Unsupport item key."}]}**
4. Il server elabora i dati, cambia lo stato dell'item in non supportato con il messaggio di errore specificato
5. La connessione TCP viene chiusa

Failover al vecchio protocollo

Per garantire che Zabbix server o proxy possano funzionare con agent delle versioni precedenti alla 7.2, che utilizzano il protocollo in testo semplice, è stato implementato un failover al vecchio protocollo.

I controlli passivi vengono eseguiti usando il protocollo JSON (7.0 e versioni successive) dopo il riavvio o quando la configurazione dell'interfaccia viene modificata.

Se non viene ricevuta alcuna risposta JSON valida (l'agent ha inviato "ZBX_NOTSUPPORTED"), Zabbix memorizzerà nella cache l'interfaccia come vecchio protocollo e **ritenterà** il controllo inviando solo la chiave dell'item.

Si noti che ogni ora Zabbix server/proxy proverà nuovamente a lavorare con il nuovo protocollo con tutte le interfacce, tornando al vecchio protocollo se necessario.

Controlli attivi

I controlli attivi richiedono un'elaborazione più complessa. L'agent deve prima recuperare dal server/proxy un elenco di item e/o comandi remoti da elaborare in modo indipendente.

I server/proxy da cui ottenere i controlli attivi sono elencati nel parametro `ServerActive` del file di configurazione dell'agent. La frequenza con cui richiedere questi controlli è impostata dal parametro `RefreshActiveChecks` nello stesso file di configurazione. Tuttavia, se l'aggiornamento dei controlli attivi non riesce, viene ritentato dopo 60 secondi, un intervallo codificato nel programma.

Note:
A partire da Zabbix 6.4, l'agent (in modalità attiva) non riceve più dal server/proxy una copia completa della configurazione una volta ogni due minuti (impostazione predefinita). Invece, per ridurre il traffico di rete e l'uso delle risorse, viene eseguita una sincronizzazione incrementale della configurazione ogni 5 secondi (impostazione predefinita), in base alla quale il server/proxy fornisce una copia completa della configurazione **solo** se l'agent non l'ha ancora ricevuta oppure se è cambiato qualcosa nella configurazione dell'host, nelle macro globali o nelle espressioni regolari globali.

L'agent invia quindi periodicamente i nuovi valori al server. Se l'agent ha ricevuto dei comandi remoti da eseguire, verrà inviato anche il risultato dell'esecuzione. Si noti che l'esecuzione di comandi remoti su un agent attivo è supportata a partire da Zabbix agent 7.0.

Note:
Se un agent si trova dietro il firewall, potresti prendere in considerazione l'uso esclusivo dei controlli attivi, perché in questo caso non sarebbe necessario modificare il firewall per consentire le connessioni iniziali in ingresso.

Ottenere l'elenco degli item

Richiesta dell'agent

La richiesta di active checks viene utilizzata per ottenere gli active checks da elaborare da agent.
Questa richiesta viene inviata dall'agent all'avvio e poi a intervalli di RefreshActiveChecks.

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Host name.
host_metadata	*string*	no	The configuration parameter HostMetadata or HostMetadataItem metric value.
interface	*string*	no	The configuration parameter HostInterface or HostInterfaceItem metric value.
ip	*string*	no	The configuration parameter ListenIP first IP if set.
port	*number*	no	The configuration parameter ListenPort value if set and not default agent listening port.
version	*string*	yes	The agent version number.
variant	*number*	yes	The agent variant (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Configuration identifier for incremental configuration sync.
session	*string*	no	Session identifier for incremental configuration sync.

Risposta del server

La risposta degli active checks viene inviata dal server a agent dopo l'elaborazione della richiesta di active checks.

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	Error information in case of failure.
data	*array of objects*	no	Active check items. Omitted if host configuration is unchanged.
key	*string*	no	Item key with expanded macros.
itemid	*number*	no	Item identifier.
delay	*string*	no	Item update interval. Flexible/scheduling intervals are supported by both Zabbix agent and Zabbix agent 2 since Zabbix 7.0.
lastlogsize	*number*	no	Item lastlogsize.
mtime	*number*	no	Item mtime.
timeout	*string*	no	Item timeout.
refresh_unsupported	*number*	no	Unsupported item refresh interval.
regexp	*array of objects*	no	Global regular expressions.
name	*string*	no	Global regular expression name.
expression	*string*	no	Global regular expression.
expression_type	*number*	no	Global regular expression type.
exp_delimiter	*string*	no	Global regular expression delimiter.
case_sensitive	*number*	no	Global regular expression case sensitivity setting.
commands	*array of objects*	no	Remote commands to execute. Included if remote command execution has been triggered by an action operation or manual script execution. Note that remote command execution on an active agent is supported since Zabbix agent 7.0. Older active agents will ignore any remote commands included in the active checks server response.
command	*string*	no	Remote command.
id	*number*	no	Remote command identifier.
wait	*number*	no	Remote command mode of execution ("0" (nowait) for commands from action operations; "1" (wait) for commands from manual script execution).

Field	Type	Mandatory	Value
timeout	*number*	no	Remote command execution timeout in server/proxy configuration.
config_revision	*number*	no	Configuration identifier for incremental configuration sync. Omitted if host configuration is unchanged. Incremented if host configuration is changed.

Il server deve rispondere con success.

Ad esempio:

1. Agent apre una connessione TCP
2. Agent richiede l'elenco dei check
3. Server risponde con un elenco di item e dei comandi remoti da eseguire
4. Agent analizza la risposta
5. La connessione TCP viene chiusa
6. Agent avvia la raccolta periodica dei dati ed esegue i comandi remoti (supportato da Zabbix agent 7.0)

Attention:
Si noti che i dati di configurazione (sensibili) possono diventare disponibili a soggetti che hanno accesso alla porta trapper del server Zabbix quando si utilizza un active check. Questo è possibile perché chiunque può fingersi un active agent e richiedere i dati di configurazione degli item; l'autenticazione non viene eseguita a meno che non si utilizzino opzioni di crittografia.

Invio dei dati raccolti

Agent invia

La richiesta di dati dell'agent contiene i valori degli item raccolti e i valori dei comandi remoti eseguiti (se presenti).

```
{
  "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	Valori degli item.

Field	Type	Mandatory	Value
id	*number*	yes	L'identificatore del valore (contatore incrementale usato per verificare valori duplicati in caso di problemi di rete).
itemid	*number*	yes	L'identificatore dell'item.
value	*string*	no	Il valore dell'item.
lastlogsize	*number*	no	Il lastlogsize dell'item.
mtime	*number*	no	Il mtime dell'item.
state	*number*	no	Lo stato dell'item.
source	*string*	no	La sorgente del registro eventi del valore.
eventid	*number*	no	L'eventid del registro eventi del valore.
severity	*number*	no	La severity del registro eventi del valore.
timestamp	*number*	no	Il timestamp del registro eventi del valore.
clock	*number*	yes	Il timestamp del valore (secondi dall'Epoch).
ns	*number*	yes	I nanosecondi del timestamp del valore.
commands	*array of objects*	no	Risultato dell'esecuzione dei comandi remoti. Si noti che l'esecuzione di comandi remoti su un agent attivo è supportata da Zabbix agent 7.0. Gli agent attivi precedenti ignoreranno eventuali comandi remoti inclusi nella risposta del server ai controlli attivi.
id	*number*	no	Identificatore del comando remoto.
value	*string*	no	Risultato dell'esecuzione del comando remoto se l'esecuzione è andata a buon fine.
error	*string*	no	Messaggio di errore dell'esecuzione del comando remoto se l'esecuzione non è riuscita.
session	*string*	yes	Identificatore univoco della sessione generato ogni volta che l'agent viene avviato (lungo 32 cifre esadecimali).
host	*string*	yes	Nome dell'host.
version	*string*	yes	Numero di versione dell'agent.
variant	*number*	yes	Variante dell'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

A ogni valore viene assegnato un ID virtuale. L'ID del valore è un semplice contatore crescente, univoco all'interno di una singola sessione di dati (identificata dal token di sessione). Questo ID viene usato per scartare i valori duplicati che potrebbero essere inviati in ambienti con connettività scarsa.

Risposta del server

La risposta dei dati dell'agent viene inviata dal server all'agent dopo l'elaborazione della richiesta di dati dell'agent.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	Risultati dell'elaborazione degli item.

Attention:
Se l'invio di alcuni valori non riesce sul server (ad esempio, perché host o item sono stati disabilitati o eliminati), l'agent non ritenterà l'invio di tali valori.

Ad esempio:

1. L'agent apre una connessione TCP
2. L'agent invia un elenco di valori
3. Il server elabora i dati e invia lo stato indietro
4. La connessione TCP viene chiusa

Attention:
Il messaggio di errore verrà troncato a 2048 simboli sul lato server.

Messaggio heartbeat

Invio da parte dell'agent

Il messaggio heartbeat viene inviato da un agent attivo al server/proxy Zabbix ogni `HeartbeatFrequency` secondi (configurato nel file di configurazione di Zabbix agent o Zabbix agent 2).

Viene utilizzato per monitorare la disponibilità dei controlli attivi.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

Campo	Tipo	Obbligatorio	Valore
request	*string*	sì	`active check heartbeat`
host	*string*	sì	Il nome dell'host.
heartbeat_freq	*number*	sì	La frequenza heartbeat dell'agent (parametro di configurazione HeartbeatFrequency).
version	*string*	sì	Il numero di versione dell'agent.
variant	*number*	sì	La variante dell'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Risposta di reindirizzamento

Quando un host è stato riassegnato, il server può indicare all'agent di reindirizzare il proprio heartbeat (e i successivi controlli attivi) a un'altra istanza di proxy o server.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.1:10055"
    }
  }
```

Campo		Tipo	Obbligatorio	Valore
response		*string*	sì	`success` | `failed`
redirect		*object*	sì	Istruzioni di reindirizzamento.
	revision	*number*	sì	Identificatore della revisione della configurazione.
	address	*string*	sì	Indirizzo del server/proxy di destinazione.

Protocollo XML precedente

Note:
Zabbix accetterà fino a 16 MB di dati XML codificati in Base64, ma un singolo valore decodificato non deve superare i 64 KB; in caso contrario, verrà troncato a 64 KB durante la decodifica.

Esecuzione di agent come root

A partire da Zabbix **5.0.0**, il file di servizio systemd per Zabbix agent nei pacchetti ufficiali include esplicitamente le direttive `User` e `Group`. Entrambe sono impostate su `zabbix`.

Non è più possibile configurare l'utente con cui Zabbix agent viene eseguito tramite il file `zabbix_agentd.conf`, poiché l'agent ignorerà questa configurazione ed eseguirà il processo come l'utente specificato nel file di servizio systemd. Per eseguire Zabbix agent come root è necessario apportare le modifiche descritte di seguito.

Zabbix agent

Per sostituire l'utente e il gruppo predefiniti per Zabbix agent, eseguire:

```
systemctl edit zabbix-agent
```

Quindi, aggiungere il contenuto seguente:

```
[Service]
User=root
Group=root
```

Ricaricare i daemon e riavviare il servizio zabbix-agent:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Per **Zabbix agent**, questa operazione riabilita la funzionalità di configurazione dell'utente nel file di configurazione dell'agent. Ora è necessario impostare i parametri di configurazione `User=root` e `AllowRoot=1` nel file di configurazione.

Zabbix agent 2

Per sovrascrivere l'utente e il gruppo predefiniti per Zabbix agent 2, eseguire:

```
systemctl edit zabbix-agent2
```

Quindi, aggiungere il seguente contenuto:

```
[Service]
User=root
Group=root
```

Ricaricare i daemon e riavviare il servizio zabbix-agent2:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Per **Zabbix agent2** questo determina completamente l'utente con cui viene eseguito. Non sono necessarie ulteriori modifiche.

Parametri di configurazione: Zabbix agent

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento dell'agent Zabbix.

Specificare i valori dei parametri nel file di configurazione dell'agent di Zabbix (`zabbix_agentd.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un modello.
AllowKeyRegexp	Consente l'esecuzione delle chiavi item che corrispondono a un modello basato su un'espressione regolare.
AllowRoot	Consente all'agent di essere eseguito come `root`.
BufferSend	Imposta la frequenza di invio dei valori dal buffer al server/proxy Zabbix.
BufferSize	Imposta il numero massimo di valori nel buffer di memoria.
DebugLevel	Imposta il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un modello.
DenyKeyRegexp	Nega l'esecuzione delle chiavi item che corrispondono a un modello basato su un'espressione regolare.
EnableRemoteCommands	Consente i comandi remoti dal server Zabbix. Questo parametro è deprecato; utilizzare invece `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.
HeartbeatFrequency	Imposta la frequenza dei messaggi heartbeat in secondi.
HostInterface	Specifica l'interfaccia del host.
HostInterfaceItem	Specifica un item per ottenere l'interfaccia del host.
HostMetadata	Specifica i metadati del host.
HostMetadataItem	Specifica un item dell'agent Zabbix per ottenere i metadati del host.
Hostname	Specifica il nome host.
HostnameItem	Specifica un item dell'agent Zabbix per ottenere il nome host.
Include	Specifica una directory o singoli file da includere nel file di configurazione.
ListenBacklog	Imposta il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Specifica l'indirizzo IP di ascolto dell'agent per le connessioni dal server/proxy.

Parametro	Descrizione
ListenPort	Specifica la porta di ascolto dell'agent.
LoadModule	Specifica il modulo da caricare all'avvio dell'agent.
LoadModulePath	Specifica il percorso completo della posizione dei moduli dell'agent.
LogFile	Specifica il file di log dell'agent.
LogFileSize	Imposta la dimensione massima del file di log.
LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
LogType	Specifica il tipo di output del log.
MaxLinesPerSecond	Imposta il numero massimo di nuove righe al secondo da inviare al server/proxy Zabbix durante l'elaborazione dei controlli attivi `log` o `logrt`.
PidFile	Specifica il file PID.
RefreshActiveChecks	Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi.
Server	Specifica gli indirizzi IP o i nomi DNS dei server/proxy Zabbix.
ServerActive	Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	Specifica l'indirizzo IP di origine.
StartAgents	Imposta il numero di istanze pre-fork di `zabbix_agentd` per l'elaborazione dei controlli passivi.
Timeout	Specifica per quanto tempo attendere (in secondi) la creazione della connessione e lo scambio di dati con il proxy o il server Zabbix.
TLSAccept	Specifica il livello di crittografia richiesto per le connessioni in entrata.
TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato peer, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCertFile	Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCipherAll	Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherAll13	Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert	Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherCert13	Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK	Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSCipherPSK13	Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSConnect	Specifica il livello di crittografia richiesto per le connessioni al server o al proxy Zabbix.
TLSCRLFile	Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti Zabbix.
TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti Zabbix.
TLSPSKFile	Specifica il percorso completo del file contenente la chiave precondivisa dell'agent, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSPSKIdentity	Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSServerCertIssuer	Specifica l'emittente del certificato del server (proxy) consentito.
TLSServerCertSubject	Specifica il subject del certificato del server (proxy) consentito.
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti dei parametri definiti dall'utente.
User	Riduce i privilegi a un utente specifico ed esistente nel sistema.
UserParameter	Specifica un parametro definito dall'utente da monitorare.
UserParameterDir	Specifica il percorso di ricerca predefinito per i comandi UserParameter.

Tenere presente quanto segue:

- I valori predefiniti riflettono i valori predefiniti del demone, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta i file di configurazione solo con codifica UTF-8 senza BOM.
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli del parametro

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più

breve e semplice.
 Possono essere presenti più parametri `Alias`. Non sono consentiti più parametri con la stessa chiave `Alias`.
 Chiavi `Alias` diverse possono fare riferimento alla stessa chiave item.
 Gli alias possono essere utilizzati in `HostMetadataItem`, ma non nel parametro `HostnameItem`.

Esempio 1: recupero dell'ID dell'utente `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Ora è possibile utilizzare la chiave abbreviata `zabbix.userid` per recuperare i dati.

Esempio 2: recupero dell'utilizzo della CPU con parametri predefiniti e personalizzati.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Ciò consente di utilizzare la chiave `cpu.util` per ottenere la percentuale di utilizzo della CPU con i parametri predefiniti e di utilizzare `cpu.util[all, idle, avg15]` per ottenere dati specifici sull'utilizzo della CPU.

Esempio 3: esecuzione di più regole di low-level discovery che elaborano gli stessi item di discovery.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Ora è possibile configurare diverse regole di discovery utilizzando `vfs.fs.discovery`, con parametri diversi per ciascuna regola, ad esempio `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` e così via.

AllowKey

Consente l'esecuzione di quegli item key che corrispondono a un pattern. Il pattern della key è un'espressione wildcard che supporta il carattere * per corrispondere a qualsiasi numero di caratteri.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità. Vedi anche: Restricting agent checks.

AllowKeyRegexp

Consente l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola corrispondente avrà la priorità. Vedere anche: Limitazione dei controlli dell'agent.

Esempio:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

Consente all'agent di essere eseguito come `root`. Se disabilitato e l'agent viene avviato da `root`, l'agent tenterà invece di passare all'utente `zabbix`. Non ha effetto se avviato da un utente normale.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

BufferSend

Imposta la frequenza di invio dei valori dal buffer a Zabbix server/proxy (in secondi).

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Imposta il numero massimo di valori nel buffer di memoria.
 L'agent invierà tutti i dati raccolti al server Zabbix o al proxy se il buffer è pieno.

Predefinito: 100
 Intervallo: 2-65535

DebugLevel

Imposta il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (genera molte informazioni); - *5* - debug esteso (genera ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

DenyKey

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere * per trovare una corrispondenza con un numero qualsiasi di caratteri.
 È possibile definire più regole di

corrispondenza delle chiavi in combinazione con AllowKey, AllowKeyRegexp e DenyKeyRegexp. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità. Vedere anche: Limitazione dei controlli dell'agent.

Predefinito: DenyKey=system.run[*]

DenyKeyRegexp

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con AllowKeyRegexp, AllowKey e DenyKey. I parametri vengono elaborati uno alla volta in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità.
 Tutti gli item system.run[] sono negati per impostazione predefinita (DenyKey=system.run[*]), a meno che non venga specificata un'altra regola system.run[*].

Vedere anche: Limitazione dei controlli dell'agent.

EnableRemoteCommands

Consente i comandi remoti dal server Zabbix.
 Questo parametro è **deprecato**; utilizzare invece AllowKey=system.run[*], AllowKeyRegexp=^system\.run\[.*\]$, DenyKey=system.run[*] o DenyKeyRegexp=^system\.run\[.*\]$.
 È un alias interno per i parametri AllowKey, AllowKeyRegexp, DenyKey e DenyKeyRegexp, a seconda del valore: - 0 - DenyKey=system.run[*] (DenyKeyRegexp=^system\.run\[.*\]$) - 1 - AllowKey=system.run[*] (AllowKeyRegexp=^system\.

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

HeartbeatFrequency

Imposta la frequenza dei messaggi heartbeat in secondi. Usato per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Specifica l'interfaccia dell'host (indirizzo IP o nome DNS) utilizzata durante l'autoregistrazione dell'host. Questo valore verrà usato per popolare l'interfaccia nel nuovo host creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per ulteriori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da HostInterfaceItem.

L'agent segnalerà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Specificare un item per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante la registrazione automatica del host. Questo valore viene utilizzato solo se HostInterface non è definito. Per ulteriori informazioni, vedere Utilizzo del DNS come interfaccia predefinita.

Durante una richiesta di registrazione automatica, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 255 caratteri.

L'item system.run[] è supportato indipendentemente dalle impostazioni AllowKey, AllowKeyRegexp, DenyKey o DenyKeyRegexp.

HostMetadata

Specifica i metadati per identificare o distinguere l'host durante l'autoregistrazione dell'host (agent attivo).
HostMetadata consente di distinguere gli host oltre il nome host.

Se non definito, il valore verrà acquisito da HostMetadataItem.

L'agent genererà un errore e non verrà avviato se il valore specificato supera il limite di 2034 byte o se è una stringa non UTF-8.
Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori con codifica UTF-8 valida ma che non sono IP o nomi DNS validi verranno rifiutati e segnalati come non validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Intervallo: 0-2034 byte

HostMetadataItem

Specificare un item dell'agent Zabbix per ottenere i metadati dell'host. Questa opzione viene utilizzata solo quando HostMetadata non è definito.

Il valore di HostMetadataItem viene recuperato a ogni tentativo di autoregistrazione e viene utilizzato solo durante il processo di autoregistrazione dell'host (agent attivo). HostMetadataItem consente di distinguere gli host oltre il nome host.

Sono supportati i parametri utente e gli alias. L'item system.run[] è supportato indipendentemente dalle impostazioni AllowKey, AllowKeyRegexp, DenyKey o DenyKeyRegexp.

Durante una richiesta di autoregistrazione, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 65535 punti di codice UTF-8. Il valore restituito dall'item deve essere una stringa UTF-8; in caso contrario, verrà ignorato. Quando un parametro prevede un indirizzo IP o un nome DNS, verranno rifiutati e segnalati come non validi anche i valori UTF-8 validi che non sono indirizzi IP o nomi DNS validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Nome host

Specifica hostname univoci e sensibili alle maiuscole/minuscole. Un elenco delimitato da virgole.
 Obbligatorio per i controlli attivi e deve corrispondere agli hostname configurati sul server.
 Il valore viene acquisito da HostnameItem se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per hostname, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Specificare un item dell'agent Zabbix per ottenere il nome host.
 Questa opzione viene utilizzata solo quando Hostname non è definito.
 I parametri utente o gli alias non sono supportati, ma l'item system.run[] è supportato indipendentemente dai valori di AllowKey, AllowKeyRegexp, DenyKey o DenyKeyRegexp.

Predefinito: system.hostname

Include

Specifica una directory o singoli file da includere nel file di configurazione.
 Durante l'installazione di Zabbix, questo parametro viene impostato su /usr/local/etc, a meno che non venga modificato durante la compilazione.

Regole generali per il parametro Include:

- Il percorso può essere relativo alla posizione del file zabbix_agentd.conf.
- È possibile specificare più parametri Include in un unico file di configurazione.
- Zabbix elabora il parametro Include non appena lo legge, quindi è consigliabile posizionarlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non si avvierà.

Regole per l'inclusione di directory o pattern di file:

- Non assumere alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup che alcuni editor di testo creano automaticamente (ad esempio, la modifica di include/my_specific.conf può creare include/my_specific.conf.BAK), poiché entrambi i file verranno inclusi.

Per includere solo i file rilevanti da una directory, usare il carattere jolly (*) per la corrispondenza dei pattern.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Imposta il numero massimo di connessioni in sospeso nella coda TCP.
 Il valore predefinito è una costante hard-coded, che dipende dal sistema.
 Anche il valore massimo supportato dipende dal sistema; valori troppo elevati possono essere troncati silenziosamente al "massimo specificato dall'implementazione".

Predefinito: SOMAXCONN
 Intervallo: 0 - INT_MAX

ListenIP

Specifica l'indirizzo IP di ascolto dell'agent per le connessioni dal server/proxy. Un elenco delimitato da virgole.

Predefinito: 0.0.0.0

ListenPort

Specifica la porta di ascolto dell'agent per le connessioni dal server/proxy.

Predefinito: 10050
 Intervallo: 1024-32767

LoadModule

Specifica il modulo da caricare all'avvio dell'agent.
 I moduli vengono utilizzati per estendere le funzionalità dell'agent.
 Il modulo deve trovarsi nella directory specificata da `LoadModulePath` oppure il percorso deve precedere il nome del modulo. Se il percorso precedente è assoluto (inizia con /), `LoadModulePath` viene ignorato.
 Formati: - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

È consentito includere più parametri `LoadModule`.

LoadModulePath

Specifica il percorso completo della posizione dei moduli dell'agent.
 Il valore predefinito dipende dalle opzioni di compilazione.

LogFile

Specifica il file di log dell'agent.

Obbligatorio: Sì, se `LogType` è impostato su `file`; altrimenti no

LogFileSize

Imposta la dimensione massima del file di log (in MB).
0 - disabilita la rotazione automatica dei log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file fallisce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024

LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi.
 I comandi verranno registrati solo se eseguiti in remoto.
 Le voci di log non verranno create se `system.run[]` viene avviato localmente dai parametri `HostMetadataItem`, `HostInterfaceItem` o `HostnameItem`.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

LogType

Specifica il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro `LogFile`; - *system* - scrive il log in syslog; - *console* - scrive il log nell'output standard.

Predefinito: `file`

MaxLinesPerSecond

Imposta il numero massimo di nuove righe al secondo da inviare a Zabbix server/proxy durante l'elaborazione dei controlli attivi `log` o `logrt`.
 Il valore fornito verrà sovrascritto dal parametro `maxlines`, specificato nella chiave dell'item `log` o `logrt`.
Nota: Zabbix elaborerà un numero di nuove righe 10 volte superiore a quello impostato in `MaxLinesPerSecond` per cercare la stringa richiesta negli item di log.

Predefinito: 20
 Intervallo: 1-1000

PidFile

Specifica il file PID.

Predefinito: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi, in secondi.
 Nota che, dopo un tentativo fallito di aggiornamento dei controlli attivi, il successivo aggiornamento verrà tentato dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Specifica gli indirizzi IP, opzionalmente in notazione CIDR, oppure i nomi DNS dei server/proxy Zabbix. Un elenco delimitato da virgole.
 Le connessioni in ingresso saranno accettate solo dagli host elencati qui. Se il supporto IPv6 è abilitato, allora `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4. Si noti che gli "indirizzi IPv6 compatibili con IPv4" (`prefisso 0000::/96`) sono supportati ma deprecati da RFC4291. Sono consentiti gli spazi.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obbligatorio: sì, se StartAgents non è impostato esplicitamente a 0

ServerActive

Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
 L'indirizzo del server/proxy è un indirizzo IP o un nome DNS, con una porta opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo di server o proxy separati da punto e virgola. È possibile specificare più server/cluster Zabbix e proxy Zabbix, separati da virgola. A meno che non si usino gruppi di proxy, non dovrebbe essere specificato più di un proxy Zabbix per ciascun server/cluster Zabbix. Se è specificato un proxy Zabbix, allora non dovrebbe essere specificato il server/cluster Zabbix per quel proxy.
È possibile fornire più indirizzi delimitati da virgole per usare in parallelo più server Zabbix indipendenti. Sono consentiti gli spazi.
Se la porta non è specificata, viene usata la porta predefinita.
Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre se è specificata la porta per quell'host. Se la porta non è specificata, le parentesi quadre per gli indirizzi IPv6 sono facoltative.
Se questo parametro non è specificato, i controlli attivi sono disabilitati.

Esempio per proxy Zabbix:

```
ServerActive=127.0.0.1:10051
```

Esempio per gruppo di proxy Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Esempio per più server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Esempio per alta disponibilità:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Esempio per alta disponibilità con due cluster e un server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Specifica l'indirizzo IP di origine per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy;
- la creazione di connessioni durante l'esecuzione di alcuni item (`web.page.get`, `net.tcp.port`, ecc.).

StartAgents

Imposta il numero di istanze pre-forked di `zabbix_agentd` per l'elaborazione dei controlli passivi.
 Se impostato a 0, i controlli passivi sono disabilitati e l'agent non ascolterà su alcuna porta TCP.

Predefinito: 10
 Intervallo: 0-100

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix proxy o server.

Questo parametro definisce la durata di varie operazioni di comunicazione, tra cui:

- attesa di una risposta dal server Zabbix;
- invio di richieste al server Zabbix, incluse le richieste di configurazione degli item e i dati degli item nelle active checks;
- recupero dei dati di log tramite logfile;
- invio di messaggi heartbeat;
- durata massima per i controlli `vfs.*`;
- utilizzo da parte dei moduli dell'agent Zabbix;
- funzione di fallback negli scenari in cui un server o un proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli dell'agent che dispongono di impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o per singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Specifica il livello di crittografia richiesto per le connessioni in ingresso.
 Usato per i controlli passivi.
 È possibile specificare più valori, separati da virgola: - *unencrypted* - accetta connessioni senza crittografia (predefinito); - *psk* - accetta connessioni con TLS e una chiave precondivisa (PSK); - *cert* - accetta connessioni con TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCAFile

Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Specificare il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate con i componenti di Zabbix.

TLSCipherAll

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificati e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Specifica la stringa dei cipher per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della ciphersuite per la crittografia basata su certificati e PSK.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Esempio per OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Specifica la stringa di priorità GnuTLS o la stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificati.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Esempio per OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Specifica la stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificato.

TLSCipherPSK

Specifica la stringa di priorità GnuTLS o la stringa dei cipher OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della ciphersuite per la crittografia basata su PSK.

Esempio per GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Esempio per OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Specifica la stringa dei cipher per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sostituisce i criteri di selezione predefiniti della ciphersuite per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Specifica il livello di crittografia richiesto per le connessioni dell'agent a Zabbix server o proxy.
 Usato per i controlli attivi.
 È possibile specificare un solo valore: - *unencrypted* - connessione senza crittografia (predefinito); - *psk* - connessione tramite TLS e una chiave precondivisa (PSK); - *cert* - connessione tramite TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCRLFile

Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSPSKFile

Specifica il percorso completo del file contenente la chiave pre-condivisa dell'agent, utilizzata per le comunicazioni crittografate con il server Zabbix o il proxy.

TLSPSKIdentity

Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.

TLSServerCertIssuer

Specifica l'emittente del certificato del server (proxy) consentito.

TLSServerCertSubject

Specifica il subject del certificato del server (proxy) consentito.

UnsafeUserParameters

Consente di passare tutti i caratteri negli argomenti dei parametri definiti dall'utente. I seguenti caratteri non sono consentiti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Inoltre, i caratteri di nuova riga non sono consentiti.

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

Utente

Riduce i privilegi a un utente specifico già esistente nel sistema.
Ha effetto solo se eseguito come `root` e `AllowRoot` è disabilitato.

Predefinito: `zabbix`

UserParameter

Specifica un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: `UserParameter=<key>,<shell command>`
Nota che il comando shell non deve restituire una stringa vuota o solo EOL. I comandi shell possono avere percorsi relativi, se è specificato il parametro `UserParameterDir`.

Esempio:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Specifica il percorso di ricerca predefinito per i comandi `UserParameter`. Se utilizzato, l'agent cambierà la propria directory di lavoro in quella specificata qui prima di eseguire un comando. In questo modo, i comandi `UserParameter` possono avere un prefisso relativo `./` invece di un percorso completo.
È consentita una sola voce.

Esempio:

```
UserParameterDir=/opt/myscripts
```

Vedi anche

1. Differenze nella configurazione dell'agent Zabbix per i controlli attivi e passivi a partire dalla versione 2.0.0

Parametri di configurazione: Zabbix agent (Windows)

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento dell'agent Zabbix su Windows.

Specificare i valori dei parametri nel file di configurazione dell'agent di Zabbix (`zabbix_agentd.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome).
- Dettagli dei parametri.

Parametro	Descrizione
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un pattern.
AllowKeyRegexp	Consente l'esecuzione delle chiavi item che corrispondono a un pattern basato su un'espressione regolare.
BufferSend	Imposta la frequenza di invio dei valori dal buffer al server/proxy Zabbix.
BufferSize	Imposta il numero massimo di valori nel buffer di memoria.
DebugLevel	Imposta il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un pattern.
DenyKeyRegexp	Nega l'esecuzione delle chiavi item che corrispondono a un pattern basato su un'espressione regolare.
EnableRemoteCommands	Consente i comandi remoti dal server Zabbix. Questo parametro è deprecato; utilizzare invece `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.
HeartbeatFrequency	Imposta la frequenza dei messaggi heartbeat in secondi.
HostInterface	Specifica l'interfaccia dell'host.
HostInterfaceItem	Specifica un item per ottenere l'interfaccia dell'host.
HostMetadata	Specifica i metadati dell'host.
HostMetadataItem	Specifica un item dell'agent Zabbix per ottenere i metadati dell'host.
Hostname	Specifica il nome host.
HostnameItem	Specifica un item dell'agent Zabbix per ottenere il nome host.
Include	Specifica una directory o singoli file da includere nel file di configurazione.
ListenBacklog	Imposta il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Specifica l'indirizzo IP di ascolto dell'agent per le connessioni dal server/proxy.
ListenPort	Specifica la porta di ascolto dell'agent.
LogFile	Specifica il file di log dell'agent.
LogFileSize	Imposta la dimensione massima del file di log.
LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
LogType	Specifica il tipo di output del log.
MaxLinesPerSecond	Imposta il numero massimo di nuove righe al secondo da inviare al server/proxy Zabbix durante l'elaborazione dei controlli attivi `log`, `logrt` o `eventlog`.
PerfCounter	Definisce un nuovo parametro <parameter_name>, che rappresenta il valore medio del contatore delle prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi).
PerfCounterEn	Definisce un nuovo parametro <parameter_name>, che rappresenta il valore medio del contatore delle prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi). Rispetto a PerfCounter, i percorsi dei contatori delle prestazioni devono essere in inglese.
RefreshActiveChecks	Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi.
Server	Specifica gli indirizzi IP o i nomi DNS dei server/proxy Zabbix.
ServerActive	Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	Specifica l'indirizzo IP di origine.
StartAgents	Imposta il numero di istanze pre-fork di `zabbix_agentd` per l'elaborazione dei controlli passivi.
Timeout	Specifica per quanto tempo attendere (in secondi) la stabilizzazione della connessione e lo scambio di dati con il server o il proxy Zabbix.
TLSAccept	Specifica il livello di crittografia richiesto per le connessioni in ingresso.
TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCertFile	Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSConnect	Specifica il livello di crittografia richiesto per le connessioni al server o al proxy Zabbix.
TLSCRLFile	Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti Zabbix.
TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti Zabbix.
TLSPSKFile	Specifica il percorso completo del file contenente la chiave precondivisa dell'agent, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSPSKIdentity	Specifica la stringa identificativa della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.

Parametro	Descrizione
TLSServerCertIssuer	Specifica l'emittente del certificato del server (proxy) consentito.
TLSServerCertSubject	Specifica il soggetto del certificato del server (proxy) consentito.
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti dei parametri definiti dall'utente.
UserParameter	Specifica un parametro definito dall'utente da monitorare.
UserParameterDir	Specifica il percorso di ricerca predefinito per i comandi UserParameter.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del daemon, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta solo file di configurazione con codifica UTF-8 senza BOM.
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli del parametro

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più breve e semplice.
 Possono essere presenti più parametri `Alias`. Non sono consentiti più parametri con la stessa chiave `Alias`.
 Chiavi `Alias` diverse possono fare riferimento alla stessa chiave item.
 Gli alias possono essere utilizzati in `HostMetadataItem`, ma non nei parametri `HostnameItem` o `PerfCounter`.

Esempio 1: recupero dell'utilizzo del file di paging in percentuale dal server.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Ora è possibile utilizzare la chiave abbreviata `pg_usage` per recuperare i dati.

Esempio 2: recupero del carico della CPU con parametri predefiniti e personalizzati.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Ciò consente di utilizzare la chiave `cpu.load` per ottenere il carico della CPU con i parametri predefiniti e di utilizzare `cpu.load[percpu,avg15]` per ottenere dati specifici sul carico della CPU.

Esempio 3: esecuzione di più regole di low-level discovery che elaborano gli stessi item di discovery.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Ora è possibile configurare diverse regole di discovery utilizzando `vfs.fs.discovery`, con parametri diversi per ciascuna regola, ad esempio `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` e così via.

AllowKey

Consente l'esecuzione di quegli item key che corrispondono a un pattern. Il pattern della key è un'espressione wildcard che supporta il carattere `*` per corrispondere a qualsiasi numero di caratteri.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà priorità. Vedi anche: Restricting agent checks.

AllowKeyRegexp

Consente l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità. Vedere anche: Limitazione dei controlli dell'agent.

Esempio:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Imposta la frequenza di invio dei valori dal buffer a Zabbix server/proxy (in secondi).

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Imposta il numero massimo di valori nel buffer di memoria.
 L'agent invierà tutti i dati raccolti al server Zabbix o al proxy se il buffer è pieno.

Predefinito: 100
 Intervallo: 2-65535

DebugLevel

Imposta il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (produce molte informazioni); - *5* - debug esteso (produce ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

DenyKey

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere * per trovare una corrispondenza con un numero qualsiasi di caratteri.
 È possibile definire più regole di corrispondenza delle chiavi in combinazione con `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola corrispondente avrà la priorità. Vedere anche: Limitazione dei controlli dell'agent.

Predefinito: `DenyKey=system.run[*]`

DenyKeyRegexp

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `AllowKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità.
 Tutti gli item `system.run[]` sono negati per impostazione predefinita (`DenyKey=system.run[*]`), a meno che non venga specificata un'altra regola `system.run[*]`.

Vedere anche: Limitazione dei controlli dell'agent.

EnableRemoteCommands

Consente i comandi remoti dal server Zabbix.
 Questo parametro è **deprecato**; utilizzare invece `AllowKey=system.run[*]`, `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` o `DenyKeyRegexp=^system\.run\[.*\]$`.
 È un alias interno per i parametri `AllowKey`, `AllowKeyRegexp`, `DenyKey` e `DenyKeyRegexp`, a seconda del valore: - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.`

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

HeartbeatFrequency

Imposta la frequenza dei messaggi heartbeat in secondi. Usato per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Specificare l'interfaccia dell'host (indirizzo IP o nome DNS) utilizzata durante l'autoregistrazione dell'host. Questo valore verrà usato per popolare l'interfaccia nel nuovo host creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per ulteriori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da HostInterfaceItem.

L'agent segnalerà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Specifica un item per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante la registrazione automatica del host.
Questo valore viene utilizzato solo se `HostInterface` non è definito.
Per ulteriori informazioni, vedere Utilizzo del DNS come interfaccia predefinita.

Durante una richiesta di registrazione automatica, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 255 caratteri.

L'item `system.run[]` è supportato indipendentemente dalle impostazioni `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

HostMetadata

Specifica i metadati utilizzati per identificare o distinguere l'host durante l'autoregistrazione dell'host (agent attivo).
`HostMetadata` consente di distinguere gli host oltre il nome host.

Se non definito, il valore verrà acquisito da `HostMetadataItem`.

L'agent genererà un errore e non verrà avviato se il valore specificato supera il limite di 2034 byte o se è una stringa non UTF-8. Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori UTF-8 validi ma non corrispondenti a IP o nomi DNS validi verranno rifiutati e segnalati come non validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Intervallo: 0-2034 byte

HostMetadataItem

Specificare un item dell'agent Zabbix per ottenere i metadati dell'host. Questa opzione viene utilizzata solo quando `HostMetadata` non è definito.

Il valore di `HostMetadataItem` viene recuperato a ogni tentativo di autoregistrazione e viene utilizzato solo durante il processo di autoregistrazione dell'host (agent attivo). `HostMetadataItem` consente di distinguere gli host oltre il nome host.

Sono supportati i parametri utente, i contatori delle prestazioni e gli alias. L'item `system.run[]` è supportato indipendentemente dalle impostazioni `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

Durante una richiesta di autoregistrazione, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 65535 punti di codice UTF-8. Il valore restituito dall'item deve essere una stringa UTF-8; in caso contrario, verrà ignorato. Quando un parametro richiede un indirizzo IP o un nome DNS, verranno rifiutati e segnalati come non validi anche i valori che sono UTF-8 validi ma non sono IP o nomi DNS validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Nome host

Specifica hostname univoci e sensibili alle maiuscole/minuscole. Un elenco delimitato da virgole.
 Obbligatorio per i controlli attivi e deve corrispondere agli hostname configurati sul server. Il valore viene acquisito da `HostnameItem` se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per hostname, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Specificare un item dell'agent Zabbix per ottenere il nome host.
 Questa opzione viene utilizzata solo quando `Hostname` non è definito.
 I parametri utente, i contatori delle prestazioni e gli alias non sono supportati, ma l'item `system.run[]` è supportato indipendentemente dai valori di `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.
Vedere anche una descrizione più dettagliata.

Predefinito: `system.hostname`

Include

Specificare una directory o singoli file da includere nel file di configurazione (situato in `C:\Program Files\Zabbix Agent` per impostazione predefinita se Zabbix agent è installato utilizzando i pacchetti Windows MSI; situato nella cartella specificata durante l'installazione se Zabbix agent è installato come archivio zip).

Regole generali per il parametro `Include`:

- Il percorso può essere relativo alla posizione del file `zabbix_agentd.conf`.
- È possibile specificare più parametri `Include` in un unico file di configurazione.
- Zabbix elabora il parametro `Include` non appena lo legge; si consiglia quindi di inserirlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non verrà avviato.

Regole per l'inclusione di directory o pattern di file:

- Non presumere alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup che alcuni editor di testo creano automaticamente (ad esempio, modificando `include/my_specific.conf` potrebbe essere creato `include/my_specific.conf.BAK`), poiché entrambi i file verranno inclusi.

Per includere solo i file pertinenti da una directory, utilizzare il carattere jolly (*) per la corrispondenza dei pattern.

Esempio:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Imposta il numero massimo di connessioni in sospeso nella coda TCP.
 Il valore predefinito è una costante codificata, che dipende dal sistema.
 Anche il valore massimo supportato dipende dal sistema; valori troppo elevati possono essere troncati silenziosamente al "massimo specificato dall'implementazione".

Predefinito: SOMAXCONN
 Intervallo: 0 - INT_MAX

ListenIP

Specifica l'indirizzo IP di ascolto dell'agent per le connessioni dal server/proxy. Elenco delimitato da virgole.

Predefinito: 0.0.0.0

ListenPort

Specifica la porta di ascolto dell'agent per le connessioni dal server/proxy.

Predefinito: 10050
 Intervallo: 1024-32767

LogFile

Specifica il file di log dell'agent.

Predefinito: c:\zabbix_agentd.log
 Obbligatorio: Sì, se LogType è impostato su file; altrimenti no

LogFileSize

Imposta la dimensione massima del file di log (in MB).
0 - disabilita la rotazione automatica del log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file non riesce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024

LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi.
 I comandi verranno registrati solo se eseguiti in remoto.
 Le voci di log non verranno create se system.run[] viene avviato localmente dai parametri HostMetadataItem, HostInterfaceItem o HostnameItem.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

LogType

Specifica il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro LogFile; - *system* - scrive il log nel registro eventi di Windows; - *console* - scrive il log nell'output standard.

Predefinito: file

MaxLinesPerSecond

Imposta il numero massimo di nuove righe al secondo da inviare a Zabbix server/proxy durante l'elaborazione di controlli attivi log, logrt o eventlog.
 Il valore fornito verrà sovrascritto dal parametro maxlines, specificato nella chiave dell'item log, logrt o eventlog.
Nota: Zabbix elaborerà un numero di nuove righe 10 volte superiore a quello impostato in MaxLinesPerSecond per cercare la stringa richiesta negli item di log.

Predefinito: 20
 Intervallo: 1-1000

PerfCounter

Definisci un nuovo parametro <parameter_name> che rappresenta il valore medio del contatore di prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi).
Sintassi: <parameter_name>,"<perf_counter_path>",<p

Ad esempio, se desideri ricevere il numero medio di interrupt del processore al secondo per l'ultimo minuto, puoi definire un nuovo parametro "interrupts" come segue:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Nota le virgolette doppie attorno al percorso del contatore di prestazioni. Il nome del parametro (interrupts) deve essere usato come chiave dell'item durante la creazione di un item. I campioni per il calcolo del valore medio verranno raccolti ogni secondo.
Puoi eseguire typeperf -qx per ottenere l'elenco di tutti i contatori di prestazioni disponibili in Windows.

PerfCounterEn

Definisce un nuovo parametro <parameter_name> che è il valore medio del contatore delle prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi). Rispetto a PerfCounter, i percorsi dei perfcounter devono essere in inglese. Supportato solo su **Windows Server 2008/Vista** e versioni successive.
Sintassi: <parameter_name>,"<perf_counter_path>",<period>

Ad esempio, se si desidera ricevere il numero medio di interrupt del processore al secondo per l'ultimo minuto, è possibile definire un nuovo parametro "interrupts" come segue:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Si noti l'uso delle virgolette doppie attorno al percorso del contatore delle prestazioni. Il nome del parametro (interrupts) deve essere usato come chiave dell'item quando si crea un item. I campioni utilizzati per calcolare il valore medio verranno raccolti ogni secondo.
È possibile trovare l'elenco delle stringhe in inglese visualizzando la seguente chiave del registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi, in secondi.
 Nota che, dopo un tentativo fallito di aggiornamento dei controlli attivi, il successivo aggiornamento verrà tentato dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Specifica gli indirizzi IP, opzionalmente in notazione CIDR, oppure i nomi DNS dei server/proxy Zabbix. Un elenco delimitato da virgole.
 Le connessioni in ingresso saranno accettate solo dagli host elencati qui. Se il supporto IPv6 è abilitato, allora `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4. Nota che gli "indirizzi IPv6 compatibili con IPv4" (`0000::/96 prefix`) sono supportati ma deprecati da RFC4291. Sono consentiti gli spazi.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obbligatorio: sì, se StartAgents non è impostato esplicitamente a 0

ServerActive

Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
 L'indirizzo del server/proxy è un indirizzo IP o un nome DNS, con una porta opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo di server o proxy separati da punto e virgola. È possibile specificare più server/cluster Zabbix e proxy Zabbix, separati da virgola. A meno che non si utilizzino gruppi di proxy, non è necessario specificare più di un proxy Zabbix per ciascun server/cluster Zabbix. Se è specificato un proxy Zabbix, allora il server/cluster Zabbix per quel proxy non deve essere specificato.
È possibile fornire più indirizzi separati da virgole per usare più server Zabbix indipendenti in parallelo. Sono consentiti gli spazi.
Se la porta non è specificata, viene usata la porta predefinita.
Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre se è specificata la porta per quell'host. Se la porta non è specificata, le parentesi quadre per gli indirizzi IPv6 sono facoltative.
Se questo parametro non è specificato, i controlli attivi sono disabilitati.

Esempio per proxy Zabbix:

```
ServerActive=127.0.0.1:10051
```

Esempio per gruppo di proxy Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Esempio per più server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Esempio per alta disponibilità:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Esempio per alta disponibilità con due cluster e un server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

Intervallo: Il numero di server attivi elencati in `ServerActive` più il numero di istanze pre-fork per i controlli passivi specificate in `StartAgents` deve essere inferiore a 64.

SourceIP

Specificare l'indirizzo IP sorgente per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy;

- la creazione di connessioni durante l'esecuzione di alcuni item (`web.page.get`, `net.tcp.port`, ecc.).

StartAgents

Imposta il numero di istanze pre-fork di `zabbix_agentd` per l'elaborazione dei controlli passivi.
 Se impostato a 0, i controlli passivi sono disabilitati e l'agent non ascolterà su alcuna porta TCP.

Predefinito: 10
 Intervallo: 0-100 (*)

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con il proxy o il server di Zabbix.

Questo parametro definisce la durata di varie operazioni di comunicazione, tra cui:

- attesa di una risposta dal server Zabbix;
- invio di richieste al server Zabbix, incluse le richieste di configurazione degli item e i dati degli item nelle active checks;
- recupero dei dati di log tramite il monitoraggio dei logfile o del registro eventi di Windows;
- invio di messaggi heartbeat;
- durata massima per i controlli `vfs.*`;
- utilizzo da parte dei moduli dell'agent Zabbix;
- funzione di fallback negli scenari in cui un server o un proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli dell'agent che dispongono di impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o per singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Specifica il livello di crittografia richiesto per le connessioni in ingresso.
 Usato per i controlli passivi.
 È possibile specificare più valori, separati da virgola: - *unencrypted* - accetta connessioni senza crittografia (predefinito); - *psk* - accetta connessioni con TLS e una chiave precondivisa (PSK); - *cert* - accetta connessioni con TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCAFile

Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Specificare il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate con i componenti di Zabbix.

TLSConnect

Specifica il livello di crittografia richiesto per le connessioni dell'agent a Zabbix server o proxy.
 Usato per i controlli attivi.
 Può essere specificato un solo valore: - *unencrypted* - connessione senza crittografia (predefinito); - *psk* - connessione tramite TLS e una chiave precondivisa (PSK); - *cert* - connessione tramite TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCRLFile

Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSPSKFile

Specifica il percorso completo del file contenente la chiave pre-condivisa dell'agent, utilizzata per le comunicazioni crittografate con il server o il proxy di Zabbix.

TLSPSKIdentity

Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.

TLSServerCertIssuer

Specifica l'emittente del certificato del server (proxy) consentito.

TLSServerCertSubject

Specifica il subject del certificato del server (proxy) consentito.

UnsafeUserParameters

Consente di passare tutti i caratteri negli argomenti dei parametri definiti dall'utente. I seguenti caratteri non sono consentiti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Inoltre, i caratteri di nuova riga non sono consentiti.

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

UserParameter

Specifica un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: `UserParameter=<key>,<shell command>`
Nota che il comando shell non deve restituire una stringa vuota o solo EOL. I comandi shell possono usare percorsi relativi, se è specificato il parametro `UserParameterDir`.

Esempio:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Specifica il percorso di ricerca predefinito per i comandi `UserParameter`. Se utilizzato, l'agent cambierà la propria directory di lavoro in quella specificata qui prima di eseguire un comando. In questo modo, i comandi `UserParameter` possono avere un prefisso relativo `./` invece di un percorso completo. È consentita una sola voce.

Esempio:

```
UserParameterDir=C:\opt\myscripts
```

Zabbix agent su Microsoft Windows

Configurazione dell'agent

Entrambe le generazioni degli agent Zabbix vengono eseguite come servizio Windows. Per Zabbix agent 2, sostituire `agentd` con `agent2` nelle istruzioni riportate di seguito.

È possibile eseguire una singola istanza dell'agent Zabbix o più istanze dell'agent su un host Microsoft Windows. Una singola istanza può utilizzare:

- il file di configurazione predefinito, situato nella stessa directory del binario dell'agent;
- un file di configurazione specificato nella riga di comando.

In caso di più istanze, ogni istanza dell'agent deve avere il proprio file di configurazione (una delle istanze può utilizzare il file di configurazione predefinito).

Un file di configurazione di esempio è disponibile nell'archivio dei sorgenti di Zabbix come:

- `conf/zabbix_agentd.conf` per Zabbix agent;
- `conf/zabbix_agent2.conf` per Zabbix agent2.

Se si desidera installare Zabbix agent/agent 2 per Windows come servizio da un archivio senza specificare esplicitamente il file di configurazione, prima di installare l'agent:

- `conf/zabbix_agentd.conf` deve essere copiato manualmente nella directory in cui verrà installato `zabbix_agentd.exe`;
- `conf/zabbix_agent2.conf` e la directory `conf/zabbix_agent2.d` devono essere copiati manualmente nella directory in cui verrà installato `zabbix_agent2.exe`.

Per informazioni dettagliate sulla configurazione dell'agent Zabbix per Windows, consultare le opzioni del file di configurazione dell'agent Zabbix o del file di configurazione dell'agent 2 Zabbix.

Parametro Hostname

Per eseguire i controlli attivi su un host, l'agent Zabbix deve avere definito il nome host. Inoltre, il valore del nome host impostato sul lato agent deve corrispondere esattamente al *Nome host* configurato per l'host nel frontend.

Il valore del nome host sul lato agent può essere definito tramite il parametro `Hostname` o `HostnameItem` nel file di configurazione dell'agent; in alternativa, se nessuno di questi parametri è specificato, vengono utilizzati i valori predefiniti.

Il valore predefinito del parametro `HostnameItem` è il valore restituito dalla chiave dell'agent `system.hostname`. In Windows, questa chiave restituisce il risultato della funzione `gethostname()`, che interroga i provider dello spazio dei nomi per determinare il nome host locale. Se nessun provider dello spazio dei nomi risponde, viene restituito il nome NetBIOS.

Il valore predefinito di `Hostname` è il valore restituito dal parametro `HostnameItem`. Pertanto, se entrambi i parametri non sono specificati, il nome host effettivo sarà il nome NetBIOS dell'host; l'agent Zabbix utilizzerà il nome host NetBIOS per recuperare l'elenco dei controlli attivi dal server Zabbix e inviargli i risultati.

La chiave `system.hostname` supporta due parametri opzionali: *type* e *transform*.

Type determina il tipo di nome che l'item deve restituire:

- *netbios* (predefinito): restituisce il nome host NetBIOS, limitato a 15 caratteri e composto esclusivamente da lettere MAIUSCOLE;
- *host*: fa distinzione tra maiuscole e minuscole e restituisce il nome host Windows completo e reale (senza dominio);
- *shorthost*: restituisce la parte del nome host precedente al primo punto. Restituisce l'intera stringa se il nome non contiene punti.
- *fqdn*: restituisce il Fully Qualified Domain Name (senza il punto finale).

Transform consente di specificare una regola di trasformazione aggiuntiva per il nome host:

- *none* (predefinito): utilizza la combinazione originale di maiuscole e minuscole;
- *lower*: converte il testo in minuscolo.

Per semplificare la configurazione del file `zabbix_agentd.conf` e renderla uniforme, è possibile utilizzare diversi approcci:

1. Lasciare non definiti i parametri `Hostname` o `HostnameItem`: l'agent Zabbix utilizzerà il nome host NetBIOS come nome host.
2. Lasciare non definito il parametro `Hostname` e definire `HostnameItem` come segue:
   - `HostnameItem=system.hostname[host]`: l'agent Zabbix utilizzerà il nome host Windows completo e reale (con distinzione tra maiuscole e minuscole) come nome host.
   - `HostnameItem=system.hostname[shorthost,lower]`: l'agent Zabbix utilizzerà solo la parte del nome host precedente al primo punto, convertita in minuscolo.
   - `HostnameItem=system.hostname[fqdn]`: l'agent Zabbix utilizzerà il Fully Qualified Domain Name come nome host.

Il nome host viene utilizzato anche come parte del nome del servizio Windows, impiegato per installare, avviare, arrestare e disinstallare il servizio Windows. Ad esempio, se il file di configurazione dell'agent Zabbix specifica `Hostname=Windows_db_server`, l'agent verrà installato come servizio Windows `Zabbix Agent [Windows_db_server]`. Pertanto, per avere un nome di servizio Windows diverso per ogni istanza dell'agent Zabbix, ogni istanza deve utilizzare un nome host diverso.

Installazione di agent come servizio Windows

Prima di installare l'agent, copiare manualmente `conf/zabbix_agentd.conf` nella directory in cui verrà installato zabbix_agentd.exe.

Per installare una singola istanza di Zabbix agent con il file di configurazione predefinito:

```
zabbix_agentd.exe --install
```

> **Attention:**
> Su un sistema a 64 bit, per far funzionare correttamente tutti i controlli relativi all'esecuzione di processi a 64 bit è necessaria una versione a 64 bit di Zabbix agent.

Se si desidera usare un file di configurazione diverso da quello predefinito, è necessario usare il seguente comando per l'installazione del servizio:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

È necessario specificare il percorso completo del file di configurazione.

È possibile installare più istanze di Zabbix agent come servizi in questo modo:

```
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
```

Il servizio installato dovrebbe ora essere visibile nel Pannello di controllo.

Avvio di agent

Per avviare il servizio agent, puoi usare il Pannello di controllo oppure farlo da riga di comando.

Per avviare una singola istanza di Zabbix agent con il file di configurazione predefinito:

```
zabbix_agentd.exe --start
```

Per avviare una singola istanza di Zabbix agent con un altro file di configurazione:

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Per avviare una delle più istanze di Zabbix agent:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

Arresto dell'agent

Per arrestare il servizio dell'agent, è possibile usare il Pannello di controllo oppure eseguire l'operazione da riga di comando.

Per arrestare una singola istanza di Zabbix agent avviata con il file di configurazione predefinito:

```
zabbix_agentd.exe --stop
```

Per arrestare una singola istanza di Zabbix agent avviata con un altro file di configurazione:

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Per arrestare una delle più istanze di Zabbix agent:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Disinstallazione del servizio Windows di agent

Per disinstallare una singola istanza di Zabbix agent usando il file di configurazione predefinito:

```
zabbix_agentd.exe --uninstall
```

Per disinstallare una singola istanza di Zabbix agent usando un file di configurazione non predefinito:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Per disinstallare più istanze di Zabbix agent dai servizi Windows:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

Limitazioni

L'agent Zabbix per Windows non supporta configurazioni Windows non standard in cui le CPU sono distribuite in modo non uniforme tra i nodi NUMA. Se le CPU logiche sono distribuite in modo non uniforme, le metriche delle prestazioni della CPU potrebbero non essere disponibili per alcune CPU. Ad esempio, se ci sono 72 CPU logiche con 2 nodi NUMA, entrambi i nodi devono avere 36 CPU ciascuno.

Zabbix agent su IBM AIX

Panoramica

I file binari precompilati dell'agent Zabbix per IBM AIX sono disponibili per il download come archivio `.tar.gz`. Questa pagina descrive come installare l'agent dall'archivio, eseguirlo manualmente o con il System Resource Controller (SRC), avviarlo all'avvio del sistema e rimuoverlo da SRC.

Installazione dell'agent

Il comando `tar` di AIX non supporta l'opzione `-z`, quindi decomprimere l'archivio con `gunzip`. Utilizzare l'opzione `-o` di `tar` in modo che i file estratti siano di proprietà dell'utente che li estrae, anziché degli ID numerici dell'utente e del gruppo memorizzati nell'archivio. Ad esempio, per installare l'agent in `/opt/zabbix` (eseguire come root; `/opt/zabbix` non deve ancora esistere):

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

Nell'archivio AIX, il binario dell'agent si trova in `usr/sbin/` e il file di configurazione in `etc/zabbix/`. Prima di avviare l'agent, modificare secondo necessità `/opt/zabbix/etc/zabbix/zabbix_agentd.conf` (almeno i parametri `Server`, `ServerActive` e `Hostname`).

Se viene avviato come root, l'agent passa all'utente `zabbix` (o all'utente specificato dal parametro `User`), a meno che nel file di configurazione non sia impostato `AllowRoot=1`. Questo utente deve esistere nel sistema; in caso contrario, l'agent non viene avviato. Se l'utente `zabbix` non esiste ancora, crearlo insieme a un gruppo con lo stesso nome:

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

Gli attributi `login=false` e `rlogin=false` impediscono di utilizzare l'account per l'accesso; l'agent può comunque essere eseguito con questo utente.

Avvio manuale dell'agent

L'agent può essere avviato manualmente:

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Esecuzione dell'agent con SRC

Registrazione dell'agent

L'agent può essere registrato anche con il System Resource Controller (SRC). Quando viene avviato tramite SRC, l'agent deve essere eseguito in modalità foreground (l'opzione `-f`). In caso contrario, SRC perde il controllo del processo dell'agent e non può arrestarlo. Ad esempio (eseguire come root):

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

In questo esempio:

- `-s` imposta il nome del sottosistema utilizzato con i comandi SRC;
- `-p` imposta il percorso completo al file binario dell'agent;
- `-a` imposta gli argomenti della riga di comando dell'agent come un'unica stringa tra virgolette; SRC divide questa stringa in argomenti separati quando avvia l'agent (è accettata una sola opzione `-a`);
- `-u` imposta l'utente con cui viene eseguito l'agent; il nome utente viene convertito in un ID utente numerico quando viene creato il sottosistema, quindi l'utente deve già esistere;
- `-G` aggiunge l'agent al gruppo di sottosistemi `tcpip`, in modo che i comandi del gruppo vengano applicati anche all'agent (ad esempio, `stopsrc -g tcpip` arresta anche l'agent);
- `-S -n 15 -f 9` configura SRC per controllare l'agent tramite segnali: `stopsrc` invia `SIGTERM` (15) per un arresto normale e `SIGKILL` (9) per un arresto forzato (`stopsrc -f`);
- `-R` configura SRC per riavviare l'agent se termina in modo anomalo (ad esempio, se uno dei suoi processi si arresta in modo imprevisto); l'agent non viene riavviato dopo `stopsrc`.

Poiché l'ID utente viene memorizzato nella definizione del sottosistema, aggiornare la definizione se l'utente `zabbix` viene ricreato con un ID utente diverso:

```
chssys -s zabbix_agentd -u zabbix
```

Avvio e arresto dell'agent

L'agent può essere avviato, interrogato e arrestato con `startsrc`, `lssrc` e `stopsrc`:

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

Evitare l'arresto forzato (`stopsrc -f`). Termina solo il processo principale dell'agent; gli altri processi dell'agent continuano a essere eseguiti e mantengono aperta la porta di ascolto, pertanto non è possibile avviare nuovamente l'agent finché non vengono terminati manualmente, ad esempio con `kill -9`.

Avvio dell'agent all'avvio del sistema

La registrazione dell'agent con SRC non ne comporta l'avvio all'avvio del sistema. Per avviare automaticamente l'agent all'avvio, aggiungere una voce a `/etc/inittab`:

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

Rimozione dell'agent da SRC

Per rimuovere l'agent dall'avvio del sistema e da SRC:

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

Panoramica Zabbix agent 2 è una nuova generazione di Zabbix agent, scritta in Go (con parte del codice C riutilizzato da Zabbix agent). È progettato per:

- Ridurre il numero di connessioni TCP;
- Fornire una migliore concorrenza dei controlli;
- Essere facilmente estendibile con plugin che forniscono controlli semplici con codice minimo e supportano controlli complessi costituiti da script a lunga esecuzione e raccolta autonoma dei dati con reporting periodico;
- Funzionare come sostituto di Zabbix agent, supportando tutte le funzionalità precedenti.

Tutte le metriche in Zabbix agent 2 vengono raccolte dai plugin.

Queste funzionalità sono specifiche dei plugin:

- Intervalli pianificati e flessibili sia per i controlli passivi sia per quelli attivi
- Gestione della coda delle attività in relazione alla pianificazione e alla concorrenza delle attività
- Timeout a livello di plugin
- Verifica di compatibilità di Zabbix agent 2 e dei relativi plugin all'avvio

Esistono due tipi di plugin:

- Plugin integrati
- Plugin caricabili

Puoi anche creare i tuoi plugin.

I plugin creati personalemente non devono essere incorporati in agent 2. Puoi aggiungerli come plugin caricabili, semplificando così la creazione di plugin per nuove metriche.

I plugin caricabili possono essere aggiunti in fase di esecuzione senza ricompilare Zabbix agent 2.

Vedi anche:

- Confronto tra agent e agent 2
- Parametri di configurazione: Zabbix agent 2
- Parametri di configurazione: Zabbix agent 2 (Windows)
- Parametri di configurazione: Plugin
- Protocollo dei plugin di Zabbix agent 2

Controlli passivi e attivi

Zabbix agent 2 supporta i controlli passivi e attivi, analogamente a Zabbix agent. Inoltre, i controlli attivi di Zabbix agent 2 supportano intervalli flessibili/pianificati e la concorrenza dei controlli all'interno di un unico server attivo.

> **Note:**
> Per impostazione predefinita, dopo un riavvio, Zabbix agent 2 pianifica la prima raccolta dei dati per i controlli attivi in un momento condizionatamente casuale all'interno dell'intervallo di aggiornamento dell'item, per evitare picchi nell'utilizzo delle risorse. Per eseguire immediatamente dopo il riavvio dell'agent i controlli attivi che non hanno un intervallo di aggiornamento *Scheduling*, impostare nel file di configurazione il parametro `ForceActiveChecksOnStart` (a livello globale) oppure il parametro del plugin `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (influisce solo sui controlli del plugin specificato). Se impostato, il parametro a livello di plugin sostituirà il parametro globale.

Verifica della concorrenza

Le verifiche provenienti da plugin diversi possono essere eseguite in modo concorrente. Il numero di verifiche concorrenti all'interno di un singolo plugin è limitato dall'impostazione della capacità del plugin. Ogni plugin può avere un'impostazione di capacità hard-coded (1000 è il valore predefinito) che può essere ridotta usando l'impostazione `Plugins.<PluginName>.System.Capacity=N` nel parametro di configurazione `Plugins`.

Piattaforme supportate Per le piattaforme supportate, fare riferimento alla pagina Requisiti.

Agent 2 su sistemi Unix-like Zabbix agent 2 su sistemi Unix-like viene eseguito sull'host monitorato.

Installazione

Zabbix agent 2 può essere installato su sistemi basati su Linux utilizzando uno dei seguenti metodi:

- Pacchetti Zabbix - selezionare il componente Agent 2 (dopo aver scelto la versione di Zabbix, la distribuzione OS e la versione OS) e seguire le istruzioni;
- Sorgenti Zabbix - scaricare i file sorgente e compilare l'agent configurando con l'opzione `--enable-agent2`.

Note:
Le funzionalità di monitoraggio di Zabbix agent 2 possono essere estese con plugin caricabili, disponibili separatamente. Per i dettagli, vedere Plugin caricabili.

Se installato come pacchetto

Zabbix agent 2 viene eseguito come processo in primo piano e si affida a un gestore di servizi esterno (ad esempio, systemd) per l'esecuzione in background. Zabbix agent 2 non dispone del supporto integrato per la daemonizzazione su Linux.

L'agent può essere avviato eseguendo:

```
systemctl start zabbix-agent2
```

Per arrestare, riavviare o verificare lo stato di Zabbix agent 2, usare i seguenti comandi:

```
systemctl stop zabbix-agent2
systemctl restart zabbix-agent2
systemctl status zabbix-agent2
```

Avvio manuale

È possibile avviare Zabbix agent individuando il binario `zabbix_agent2` ed eseguendolo direttamente; ad esempio:

```
zabbix_agent2
```

Agent 2 sui sistemi Windows Zabbix agent 2 viene eseguito come processo autonomo; tuttavia, può essere eseguito anche come servizio Windows.

Installazione

Zabbix agent 2 può essere installato su Windows utilizzando uno dei seguenti metodi:

- Binarî precompilati di Zabbix agent - scaricare il pacchetto di installazione MSI dell'agent e seguire le istruzioni nella pagina Installazione dell'agent Windows da MSI;
- Sorgenti di Zabbix - scaricare i file sorgente e seguire le istruzioni nella pagina Compilazione di Zabbix agent 2 su Windows.

Note:
Le funzionalità di monitoraggio di Zabbix agent 2 possono essere estese con plugin caricabili, disponibili separatamente. Per i dettagli, vedere Plugin caricabili.

Vedere la pagina Zabbix agent su Microsoft Windows per ulteriori dettagli sull'installazione di Zabbix agent 2 (da un archivio ZIP) come servizio Windows.

Opzioni I seguenti parametri della riga di comando possono essere utilizzati con Zabbix agent 2:

Parametro	Descrizione
agent UNIX e Windows	
`-c --config <config-file>`	Percorso del file di configurazione. È possibile utilizzare questa opzione per specificare un file di configurazione diverso da quello predefinito.
`-f --foreground`	Esegue Zabbix agent in primo piano (predefinito: true).
`-p --print`	Stampa gli item conosciuti ed esce. Per restituire anche i risultati dei parametri utente, è necessario specificare il file di configurazione (se non si trova nella posizione predefinita).
`-t --test <item key>`	Verifica l'item specificato ed esce. Per restituire anche i risultati dei parametri utente, è necessario specificare il file di configurazione (se non si trova nella posizione predefinita).
`-T --test-config`	Convalida il file di configurazione ed esce.
`-h --help`	Stampa le informazioni della guida ed esce.
`-v --verbose`	Stampa le informazioni di debug. Utilizzare questa opzione con le opzioni `-p` e `-t`.
`-V --version`	Stampa la versione dell'agent e le informazioni sulla licenza.

Parametro	Descrizione
-R --runtime-control <option>	Esegue funzioni amministrative. Vedere controllo in fase di esecuzione.
Solo agent Windows	
-m --multiple-agents	Utilizza più istanze dell'agent (con le opzioni -i, -d, -s, -x). Per distinguere i nomi dei servizi delle istanze, ogni nome del servizio includerà il valore Hostname del file di configurazione specificato.
-S --startup-type <value>	Imposta il tipo di avvio del servizio Zabbix agent per Windows. Valori consentiti: automatic - *(predefinito)* avvia automaticamente il servizio all'avvio di Windows; delayed - ritarda l'avvio del servizio finché i servizi avviati automaticamente non hanno completato l'avvio; manual - avvia manualmente il servizio (da parte di un utente o di un'applicazione); disabled - disabilita il servizio, impedendone l'avvio da parte di un utente o di un'applicazione. È possibile utilizzare questa opzione insieme all'opzione -i oppure separatamente per modificare il tipo di avvio di un servizio già installato.
-i --install	Installa Zabbix agent per Windows come servizio.
-d --uninstall	Disinstalla il servizio Zabbix agent per Windows.
-s --start	Avvia il servizio Zabbix agent per Windows.
-x --stop	Arresta il servizio Zabbix agent per Windows.

Esempi specifici di utilizzo dei parametri della riga di comando:

- Stampa tutti gli item integrati dell'agent con i relativi valori.
- Verifica un parametro utente con la chiave mysql.ping definita nel file di configurazione specificato.
- Installazione di un servizio Zabbix Agent per Windows utilizzando il percorso predefinito del file di configurazione.
- Modifica del tipo di avvio di un servizio Zabbix Agent per Windows installato utilizzando il file di configurazione che si trova nella stessa cartella dell'eseguibile dell'agent.

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

Controllo runtime

Il controllo runtime fornisce alcune opzioni per il controllo remoto.

Option	Description
log_level_increase	Aumenta il livello di log.
log_level_decrease	Riduce il livello di log.
metrics	Elenca le metriche disponibili.
periodic_prof_enable	Abilita la profilazione periodica. Tutte le metriche raccolgono i dati al momento e li scaricano, ad eccezione dei dati della CPU. I dati periodici della CPU vengono raccolti nel tempo in un file temporaneo nascosto. Al termine dell'intervallo, il file viene pubblicato come profilo della CPU e viene conservato il numero configurato di file completati. Il file temporaneo non viene conteggiato in ProfilerMaxFilesPerProfile.
periodic_prof_disable	Disabilita la profilazione periodica.
periodic_prof_execute	Scrive i profili dopo un campionamento di 5 secondi. Quando la profilazione periodica è abilitata, questo comando completa l'intervallo corrente della CPU, scrive un profilo della CPU separato di 5 secondi e avvia un nuovo file periodico della CPU. I profili richiesti esplicitamente vengono ruotati in base a ProfilerMaxFilesPerProfile al completamento del comando. Durante la raccolta dei profili da parte di questo comando, tutti i comandi runtime del profiler vengono rifiutati con "profiler is busy".
periodic_prof_set_interval	Imposta l'intervallo di profilazione.
version	Visualizza la versione dell'agent.
userparameter_reload	Ricarica i valori delle opzioni UserParameter e Include dal file di configurazione corrente.
help	Visualizza le informazioni della guida sul controllo runtime.

Esempi:

- Aumento del livello di log per l'agent 2.
- Stampa delle opzioni di controllo runtime.

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

File di configurazione Il percorso predefinito del file di configurazione dipende dal sistema operativo e dal metodo di installazione di Zabbix:

- Su UNIX, il valore predefinito è:
  - Per l'installazione dai sorgenti: `/usr/local/etc/zabbix_agent2.conf`, oppure quello impostato dalle variabili di compilazione `--sysconfdir` o `--prefix`.
  - Per l'installazione dai pacchetti: `/etc/zabbix/zabbix_agent2.conf`.
- Su Windows, il valore predefinito è `C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf`.

Se su un host è in esecuzione una singola istanza dell'agent, questa può utilizzare il file di configurazione predefinito oppure un file di configurazione specificato nella riga di comando. In caso di più istanze, ogni istanza dell'agent deve avere il proprio file di configurazione (una delle istanze può utilizzare il file di configurazione predefinito).

I parametri di configurazione di Zabbix agent 2 corrispondono per lo più a quelli di Zabbix agent, con alcune eccezioni:

Parameters		Description
Parametri di configurazione specifici di Zabbix agent 2		
	`ControlSocket`	Percorso del socket di controllo runtime. Zabbix agent 2 utilizza un socket di controllo per i comandi runtime.
	`EnablePersistentBuffer`, `PersistentBufferFile`, `PersistentBufferPeriod`	Questi parametri vengono utilizzati per configurare l'archiviazione persistente su Zabbix agent 2 per gli item attivi.
	`ForceActiveChecksOnStart`	Determina se l'agent deve eseguire immediatamente i controlli attivi dopo il riavvio oppure distribuirli uniformemente nel tempo.
	`Plugins`	I plugin possono avere parametri propri, nel formato `Plugins.<Plugin name>.<Parameter>=<value>`. Un parametro comune dei plugin è `System.Capacity`, che imposta il limite dei controlli che possono essere eseguiti contemporaneamente.
	`StatusPort`	La porta sulla quale Zabbix agent 2 rimarrà in ascolto per le richieste HTTP di stato e per visualizzare un elenco dei plugin configurati e alcuni parametri interni.

Parameters		Description
Parametri di configurazione specifici di Zabbix agent non supportati da Zabbix agent 2		
	`AllowRoot`, `User`	Non supportati da Zabbix agent 2 perché non supporta la daemonizzazione.
	`EnableRemoteCommands`	Deprecato; in entrambi gli agent vengono utilizzati invece i parametri `AllowKey`, `AllowKeyRegexp`, `DenyKey` e `DenyKeyRegexp`.
	`ListenBacklog`	Utilizzato in Zabbix agent per definire il numero massimo di connessioni in attesa nella coda TCP. Non supportato da Zabbix agent 2.
	`LoadModule`, `LoadModulePath`	I moduli caricabili non sono supportati da Zabbix agent 2.
	`MaxLinesPerSecond`, `LogRemoteCommands`	In Zabbix agent 2 vengono utilizzati invece `Plugins.Log.MaxLinesPerSecond` e `Plugins.SystemRun.LogRemoteCommands`.
	`StartAgents`	Questo parametro viene utilizzato in Zabbix agent per aumentare la concorrenza dei controlli passivi o disabilitarli. In Zabbix agent 2, la concorrenza viene configurata a livello di plugin e può essere limitata tramite un'impostazione di capacità. I controlli passivi sono disabilitati in Zabbix Agent 2 se il parametro `Server` non è specificato.

Per ulteriori dettagli, vedere le opzioni del file di configurazione per Zabbix agent2.

Codici di uscita Zabbix agent 2 può essere compilato anche con versioni precedenti di OpenSSL (1.0.1, 1.0.2).

In questo caso Zabbix fornisce mutex per il locking in OpenSSL. Se il lock o l'unlock di un mutex fallisce, viene stampato un messaggio di errore sul flusso di errore standard (STDERR) e Agent 2 termina con codice di ritorno 2 o 3, rispettivamente.

Confronto tra agent e agent 2

Questa sezione descrive le differenze tra Zabbix agent e Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Linguaggio di programmazione	C	Go con alcune parti in C
Daemonization	yes	solo tramite systemd (yes su Windows)
Estensioni supportate	Moduli caricabili personalizzati loadable modules in C.	plugin personalizzati in Go.
Requirements		
Piattaforme supportate	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: tutte le versioni desktop e server da XP in poi	Linux, Windows: tutte le versioni desktop e server, sulle quali è possibile installare una supported Go version.

Parameter	Zabbix agent	Zabbix agent 2
Librerie crittografiche supportate	GnuTLS 3.1.18 e versioni successive. OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. LibreSSL - testato con le versioni 2.7.4, 2.8.2 (si applicano alcune limitazioni, vedere la pagina Encryption per i dettagli).	Linux: OpenSSL 1.0.1 e versioni successive. MS Windows: OpenSSL 1.1.1 o versioni successive. La libreria OpenSSL deve avere il supporto PSK abilitato. LibreSSL non è supportato.
Monitoring processes		
Processi	Un processo separato di active check per ogni record server/proxy.	Un singolo processo con thread creati automaticamente. Il numero massimo di thread è determinato dalla variabile di ambiente `GOMAXPROCS`.
Metriche	**UNIX**: vedere l'elenco degli item supportati. **Windows**: vedere l'elenco degli item aggiuntivi specifici per Windows.	**UNIX**: Tutte le metriche supportate da Zabbix agent. Inoltre, agent 2 fornisce una soluzione di monitoraggio nativa Zabbix per: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd e altri target di monitoraggio - vedere l'elenco completo degli item specifici di agent 2. **Windows**: Tutte le metriche supportate da Zabbix agent, oltre ai controlli `net.tcp.service*` di HTTPS, LDAP. Inoltre, agent 2 fornisce una soluzione di monitoraggio nativa Zabbix per: PostgreSQL, Redis.
Concorrenza	Gli active check per un singolo server vengono eseguiti in sequenza.	I controlli provenienti da plugin diversi o più controlli all'interno dello stesso plugin possono essere eseguiti in parallelo.
Trap di terze parti	no	yes
Additional features		
Archiviazione persistente	no	yes
File persistenti per metriche `log*[]`	yes (solo su Unix)	no
Caricamento dei dati di log	Può essere eseguito durante la raccolta dei log per liberare il buffer.	La raccolta dei log si interrompe quando il buffer è pieno, pertanto il parametro `BufferSize` deve essere almeno `MaxLinesPerSecond` x 2.
Cambio dell'utente a runtime	yes (solo sistemi Unix-like)	no (controllato da systemd)
Ciphersuite configurabili dall'utente	yes	no

Vedi anche:

- *Descrizione dei processi Zabbix*: Zabbix agent, Zabbix agent 2
- *Parametri di configurazione*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Plugin integrati

Questi plugin sono integrati in Zabbix agent 2 e sono disponibili subito, senza configurazioni aggiuntive. Fare clic sul nome del plugin per aprire il repository del plugin con ulteriori informazioni.

Nome plugin	Descrizione	Chiavi item supportate	Commenti
Agent	Metriche dell'agent Zabbix in uso.	agent.hostname, agent.ping, agent.version	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
CPU	Monitoraggio della CPU di sistema (numero di CPU/core CPU, CPU rilevate, percentuale di utilizzo).	system.cpu.discovery, system.cpu.num, system.cpu.util	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
Docker	Monitoraggio dei container Docker.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	Vedi anche: Parametri di configurazione
File	Raccolta delle metriche dei file.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
Kernel	Monitoraggio del kernel.	kernel.maxfiles, kernel.maxproc	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
Log	Monitoraggio dei file di log.	log, log.count, logrt, logrt.count	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent. Vedi anche: Parametri di configurazione del plugin (Unix/Windows)
Memcached	Monitoraggio del server Memcached.	memcached.ping, memcached.stats	
Modbus	Legge i dati Modbus.	modbus.get	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
MQTT	Riceve i valori pubblicati dei topic MQTT.	mqtt.get	Per configurare una connessione crittografata al broker MQTT, specificare i parametri TLS nel file di configurazione dell'agent come parametri di una sessione denominata o predefiniti. Al momento i parametri TLS non possono essere passati come parametri della chiave item.
MySQL	Monitoraggio di MySQL e dei suoi fork.	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	Per configurare una connessione crittografata al database, specificare i parametri TLS nel file di configurazione dell'agent come parametri di una sessione denominata o predefiniti. Al momento i parametri TLS non possono essere passati come parametri della chiave item.
NetIf	Monitoraggio delle interfacce di rete.	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.

Nome plugin	Descrizione	Chiavi item supportate	Commenti
Oracle	Monitoraggio di Oracle Database.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Installare Oracle Instant Client prima di usare il plugin.
Proc	Percentuale di utilizzo della CPU del processo.	proc.cpu.util	La chiave supportata ha gli stessi parametri della chiave di Zabbix agent.
Redis	Monitoraggio del server Redis.	redis.config, redis.info, redis.ping, redis.slowlog.count	Per configurare una connessione crittografata a Redis, specificare i parametri TLS nel file di configurazione dell'agent come parametri di una sessione denominata o predefiniti. I parametri TLS non possono essere passati come parametri della chiave item. Si noti che una configurazione TLS errata o comunque non valida può impedire l'avvio di Zabbix agent 2, quindi verificare i file dei certificati, le autorizzazioni e i percorsi prima di abilitare TLS.

Smart	Monitoraggio S.M.A.R.T.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	La versione minima richiesta di smartctl è 7.1.
			Sono necessari i privilegi sudo/root su smartctl per l'utente che esegue Zabbix agent 2. Il plugin usa solo i seguenti comandi:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Le chiavi supportate possono essere usate con Zabbix agent 2 solo su Linux/Windows, sia come controllo passivo sia attivo.
			Vedi anche:
			Parametri di configurazione

SW	Elenco dei pacchetti installati.	system.sw.packages, system.sw.packages.get	Le chiavi supportate hanno gli stessi parametri della chiave di Zabbix agent.
Swap	Dimensione dello spazio di swap in byte/percentuale.	system.swap.size	La chiave supportata ha gli stessi parametri della chiave di Zabbix agent.

Nome plugin	Descrizione	Chiavi item supportate	Commenti
SystemRun	Esegue il comando specificato.	system.run	La chiave supportata ha gli stessi parametri della chiave di Zabbix agent. Vedi anche: Parametri di configurazione del plugin (Unix/Windows)
Systemd	Monitoraggio dei servizi systemd.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	Controllo della disponibilità delle connessioni TCP.	net.tcp.port	La chiave supportata ha gli stessi parametri della chiave di Zabbix agent.
UDP	Monitoraggio della disponibilità e delle prestazioni dei servizi UDP.	net.udp.service, net.udp.service.perf	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
Uname	Recupero delle informazioni sul sistema.	system.hostname, system.sw.arch, system.uname	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
Uptime	Raccolta delle metriche di uptime del sistema.	system.uptime	La chiave supportata ha gli stessi parametri della chiave di Zabbix agent.
VFSDev	Raccolta delle metriche VFS.	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
WebCertificate	Monitoraggio dei certificati dei siti web TLS/SSL.	web.certificate.get	
WebPage	Monitoraggio delle pagine web.	web.page.get, web.page.perf, web.page.regexp	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
ZabbixAsync	Raccolta asincrona delle metriche.	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	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.
ZabbixStats	Metriche interne di server/proxy Zabbix o numero di item ritardati in una coda.	zabbix.stats	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.

Nome plugin	Descrizione	Chiavi item supportate	Commenti
ZabbixSync	Raccolta sincrona delle metriche.	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.	Le chiavi supportate hanno gli stessi parametri delle chiavi di Zabbix agent.

Plugin caricabili

I plugin caricabili non sono disponibili immediatamente per Zabbix agent 2 e devono essere installati separatamente:

- Su Linux, è possibile utilizzare i pacchetti (ad esempio, installare Ember+ su Ubuntu con `apt install zabbix-agent2-plugin-emb`
  oppure compilare i plugin
- Su Windows, è possibile installare da MSI oppure compilare i plugin

Attention:
Prima di installare un plugin, consultare il relativo file README. Potrebbe contenere requisiti specifici e istruzioni per l'installazione.

Fare clic sul nome del plugin per accedere al repository del plugin, che contiene il file README del plugin con ulteriori informazioni.

Nome del plugin	Descrizione	Chiavi item supportate	Commenti
Ceph	Monitoraggio di Ceph.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Vedere anche Parametri di configurazione del plugin Ceph.
Ember+	Monitoraggio di Ember+.	ember.get	Vedere anche Parametri di configurazione del plugin Ember+.

Nome del plugin	Descrizione	Chiavi item supportate	Commenti
MongoDB	Monitoraggio dei server e dei cluster MongoDB (database distribuito basato su documenti).	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	Per configurare connessioni crittografate al database, specificare i parametri TLS nel file di configurazione dell'agent come parametri di una sessione denominata. Attualmente, i parametri TLS non possono essere passati come parametri della chiave item. Vedere anche Parametri di configurazione del plugin MongoDB.
MSSQL	Monitoraggio del database MSSQL.	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	Per configurare una connessione crittografata al database, specificare i parametri TLS nel file di configurazione dell'agent come parametri di una sessione denominata o predefiniti. Attualmente, i parametri TLS non possono essere passati come parametri della chiave item. Vedere anche Parametri di configurazione del plugin MSSQL.
NVIDIA GPU	Monitoraggio delle GPU NVIDIA.	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	Vedere anche Parametri di configurazione del plugin NVIDIA GPU.

Nome del plugin	Descrizione	Chiavi item supportate	Commenti
PostgreSQL	Monitoraggio di PostgreSQL e dei suoi fork.	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	Per configurare connessioni crittografate al database, specificare i parametri TLS nel file di configurazione dell'agent come parametri di una sessione denominata o predefiniti. Attualmente, i parametri TLS non possono essere passati come parametri della chiave item. Vedere anche Parametri di configurazione del plugin PostgreSQL.

Note:
I plugin caricabili, quando vengono avviati con:
 - `-V --version` - visualizzano la versione del plugin e le informazioni sulla licenza;
 - `-h --help` - visualizzano le informazioni della guida.
 - `-t, --test <item key>` - avviano il plugin per i test (la configurazione del plugin viene ignorata).

Parametri di configurazione: Zabbix agent 2

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento di Zabbix agent 2.

Zabbix agent 2 è una nuova generazione di Zabbix agent e può essere utilizzato al posto di Zabbix agent.

Specificare i valori dei parametri nel file di configurazione di agent 2 di Zabbix (`zabbix_agent2.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un pattern.
AllowKeyRegexp	Consente l'esecuzione delle chiavi item che corrispondono a un pattern basato su un'espressione regolare.
BufferSend	Imposta la frequenza di invio dei valori dal buffer al server/proxy Zabbix.
BufferSize	Imposta il numero massimo di valori nel buffer di memoria.
ControlSocket	Specifica il socket di controllo per l'invio di comandi runtime con l'opzione '-R'.
DebugLevel	Imposta il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un pattern.
DenyKeyRegexp	Nega l'esecuzione delle chiavi item che corrispondono a un pattern basato su un'espressione regolare.
EnablePersistentBuffer	Abilita l'utilizzo dell'archiviazione persistente locale per gli item attivi.

Parametro	Descrizione
EnableProfiler	Abilita la profilazione periodica di Go all'avvio di agent.
ForceActiveChecksOnStart	Esegue immediatamente i controlli attivi dopo il riavvio per la prima configurazione ricevuta.
HeartbeatFrequency	Imposta la frequenza dei messaggi heartbeat in secondi.
HostInterface	Specifica l'interfaccia del host.
HostInterfaceItem	Specifica un item per ottenere l'interfaccia del host.
HostMetadata	Specifica i metadati del host.
HostMetadataItem	Specifica un item di Zabbix agent per ottenere i metadati del host.
Hostname	Specifica il nome del host.
HostnameItem	Specifica un item di Zabbix agent per ottenere il nome del host.
Include	Specifica una directory o singoli file da includere nel file di configurazione.
ListenIP	Specifica l'indirizzo IP di ascolto di agent per le connessioni dal server/proxy.
ListenPort	Specifica la porta di ascolto di agent.
LogFile	Specifica il file di log di agent.
LogFileSize	Imposta la dimensione massima del file di log.
LogType	Specifica il tipo di output del log.
PersistentBufferFile	Specifica il file in cui mantenere il database SQLite.
PersistentBufferPeriod	Imposta il periodo di tempo per l'archiviazione dei dati quando non è disponibile una connessione al server o al proxy Zabbix.
PidFile	Specifica il file PID.
Plugins.<PluginName>.System.Capacity	Imposta il limite dei controlli per plugin che possono essere eseguiti contemporaneamente.
Plugins.Log.MaxLinesPerSecond	Imposta il numero massimo di nuove righe al secondo da inviare al server/proxy Zabbix durante l'elaborazione dei controlli attivi `log` o `logrt`.
Plugins.SystemRun.LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
PluginSocket	Specifica il percorso del socket UNIX per le comunicazioni con i plugin caricabili.
PluginTimeout	Imposta il timeout per le connessioni ai plugin caricabili.
ProfilerDir	La directory in cui verranno scritti i dati di profilazione.
ProfilerMaxFilesPerProfile	Il numero massimo di file da conservare per ogni tipo di profilo.
ProfilerInterval	L'intervallo di scrittura dei dati di profilazione.
RefreshActiveChecks	Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi.
Server	Specifica gli indirizzi IP o i nomi DNS dei server/proxy Zabbix.
ServerActive	Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	Specifica l'indirizzo IP di origine.
StatusPort	Specifica la porta per l'ascolto delle richieste di stato HTTP (`http://localhost:<port>/status`).
Timeout	Specifica per quanto tempo attendere (in secondi) la stabilizzazione della connessione e lo scambio di dati con il server o il proxy Zabbix.
TLSAccept	Specifica il livello di crittografia richiesto per le connessioni in entrata.
TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCertFile	Specifica il percorso completo del file contenente il certificato di agent o la catena di certificati, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCipherAll	Specifica la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherAll13	Specifica la stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert	Specifica la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherCert13	Specifica la stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK	Specifica la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSCipherPSK13	Specifica la stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSConnect	Specifica il livello di crittografia richiesto per le connessioni al server o al proxy Zabbix.
TLSCRLFile	Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti Zabbix.
TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata di agent, utilizzata per le comunicazioni crittografate tra i componenti Zabbix.

Parametro	Descrizione
TLSPSKFile	Specifica il percorso completo del file contenente la chiave precondivisa di agent, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSPSKIdentity	Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSServerCertIssuer	Specifica l'emittente del certificato del server (proxy) consentita.
TLSServerCertSubject	Specifica il subject del certificato del server (proxy) consentito.
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti dei parametri definiti dall'utente.
UserParameter	Specifica un parametro definito dall'utente da monitorare.
UserParameterDir	Specifica il percorso di ricerca predefinito per i comandi UserParameter.

Nota:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta i file di configurazione solo con codifica UTF-8 senza BOM.
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli del parametro

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più breve e semplice.
 Possono essere presenti più parametri `Alias`. Non sono consentiti più parametri con la stessa chiave `Alias`.
 Chiavi `Alias` diverse possono fare riferimento alla stessa chiave item.
 Gli alias possono essere utilizzati in `HostMetadataItem`, ma non nel parametro `HostnameItem`.

Esempio 1: recupero dell'ID dell'utente `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Ora è possibile utilizzare la chiave abbreviata `zabbix.userid` per recuperare i dati.

Esempio 2: recupero dell'utilizzo della CPU con parametri predefiniti e personalizzati.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Questo consente di utilizzare la chiave `cpu.util` per ottenere la percentuale di utilizzo della CPU con i parametri predefiniti e di utilizzare `cpu.util[all, idle, avg15]` per ottenere dati specifici sull'utilizzo della CPU.

Esempio 3: esecuzione di più regole di low-level discovery che elaborano gli stessi item di discovery.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Ora è possibile configurare diverse regole di discovery utilizzando `vfs.fs.discovery`, con parametri diversi per ciascuna regola, ad esempio `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` e così via.

AllowKey

Consenti l'esecuzione di quelle chiavi di item che corrispondono a un pattern. Il pattern della chiave è un'espressione wildcard che supporta il carattere * per corrispondere a qualsiasi numero di caratteri.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con `DenyKey`. È possibile definire più regole di corrispondenza delle chiavi in combinazione con `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà priorità. Vedi anche: Restricting agent checks.

AllowKeyRegexp

Consente l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità. Vedere anche: Limitazione dei controlli dell'agent.

Esempio:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Imposta la frequenza di invio dei valori dal buffer a Zabbix server/proxy (in secondi).
 Nota che, se il buffer è pieno, i dati verranno inviati prima.

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Imposta il numero massimo di valori nel buffer di memoria.
 L'agent invierà tutti i dati raccolti al server Zabbix o al proxy se il buffer è pieno. Questo parametro deve essere usato solo se il buffer persistente è disabilitato (`EnablePersistentBuffer=0`).

Predefinito: 1000
 Intervallo: 2-65535

ControlSocket

Specifica il socket di controllo per inviare comandi a runtime con l'opzione `-R`.

Predefinito: `/tmp/agent.sock`

DebugLevel

Imposta il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (produce molte informazioni); - *5* - debug esteso (produce ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

DenyKey

Nega l'esecuzione degli item key che corrispondono a un pattern.
 Il pattern della key è un'espressione con caratteri jolly che supporta il carattere `*` per trovare un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle key in combinazione con `AllowKey`. È possibile definire più regole di corrispondenza delle key in combinazione con `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità. Vedere anche: Restrizione dei controlli dell'agent.

Predefinito: `DenyKey=system.run[*]`

DenyKeyRegexp

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `AllowKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità.
 Tutti gli item `system.run[]` sono negati per impostazione predefinita (`DenyKey=system.run[*]`), a meno che non venga specificata un'altra regola `system.run[*]`.

Vedere anche: Limitazione dei controlli dell'agent.

Esempio:

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

EnablePersistentBuffer

Abilita l'uso dell'archiviazione persistente locale per gli item attivi.
 Se l'archiviazione persistente è disabilitata, verrà utilizzato il buffer in memoria.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

EnableProfiler

Abilita il profiling periodico di Go all'avvio dell'agent. Il profiling può essere abilitato o disabilitato anche durante l'esecuzione utilizzando l'opzione `-R`. Mentre `periodic_prof_execute` raccoglie i profili, tutti i comandi runtime del profiler vengono rifiutati perché il profiler è occupato.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

ForceActiveChecksOnStart

Esegue i controlli attivi immediatamente dopo il riavvio, per la prima configurazione ricevuta.
 Disponibile anche come parametro di configurazione per singolo plugin, ad esempio: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

HeartbeatFrequency

Imposta la frequenza dei messaggi heartbeat in secondi.
 Usato per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Specifica l'interfaccia dell'host (indirizzo IP o nome DNS) utilizzata durante l'autoregistrazione dell'host.
 Questo valore verrà usato per popolare l'interfaccia nel nuovo host creato e consente di configurare esplicitamente un indirizzo IP o DNS.
 Per ulteriori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da `HostInterfaceItem`.

L'agent segnalerà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Specificare un item per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante la registrazione automatica del host.
 Questo valore viene utilizzato solo se `HostInterface` non è definito.
 Per ulteriori informazioni, vedere Utilizzo del DNS come interfaccia predefinita.

Durante una richiesta di registrazione automatica, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 255 caratteri.

L'item `system.run[]` è supportato indipendentemente dalle impostazioni `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

HostMetadata

Specificare i metadati utilizzati per identificare o distinguere l'host durante l'autoregistrazione dell'host.
 `HostMetadata` consente di distinguere gli host oltre il nome host.

Se non definito, il valore verrà acquisito da `HostMetadataItem`.

L'agent genererà un errore e non verrà avviato se il valore specificato supera il limite di 2034 byte o se è una stringa non UTF-8. Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori con codifica UTF-8 valida ma che non sono IP o nomi DNS validi verranno rifiutati e segnalati come non validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Intervallo: 0-2034 byte

HostMetadataItem

Specificare un item per ottenere i metadati dell'host.
 Questa opzione viene utilizzata solo quando `HostMetadata` non è definito.

Il valore di `HostMetadataItem` viene recuperato a ogni tentativo di autoregistrazione e viene utilizzato solo durante il processo di autoregistrazione dell'host. `HostMetadataItem` consente di distinguere gli host oltre il nome host.

Sono supportati i parametri utente e gli alias. L'item `system.run[]` è supportato indipendentemente dalle impostazioni `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

Durante una richiesta di autoregistrazione, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 65535 punti di codice UTF-8. Il valore restituito dall'item deve essere una stringa UTF-8; in caso contrario, verrà ignorato. Quando un parametro richiede un indirizzo IP o un nome DNS, verranno rifiutati e segnalati come non validi anche i valori che sono UTF-8 validi ma non sono IP o nomi DNS validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Nome host

Specifica hostname univoci, con distinzione tra maiuscole e minuscole. Un elenco delimitato da virgole.
 Obbligatorio per i controlli attivi e deve corrispondere agli hostname configurati sul server.
 Il valore viene acquisito da `HostnameItem` se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per ogni hostname, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Specifica un item per ottenere il nome host.
 Questa opzione viene utilizzata solo quando `Hostname` non è definito.
 I parametri utente o gli alias non sono supportati, ma l'item `system.run[]` è supportato indipendentemente dai valori di `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

Predefinito: `system.hostname`

Include

Specifica una directory o singoli file da includere nel file di configurazione.
 Durante l'installazione di Zabbix, questo parametro viene impostato su `/usr/local/etc`, a meno che non venga modificato durante la compilazione.

Regole generali per il parametro Include:

- Il percorso può essere relativo alla posizione del file zabbix_agent2.conf.
- È possibile specificare più parametri Include in un unico file di configurazione.
- Zabbix elabora il parametro Include non appena lo legge, quindi è consigliabile posizionarlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non si avvierà.

Regole per l'inclusione di directory o pattern di file:

- Non dare per scontato alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup che alcuni editor di testo creano automaticamente (ad esempio, la modifica di include/my_specific.conf può creare include/my_specific.conf.BAK), poiché entrambi i file verranno inclusi.

Per includere solo i file pertinenti da una directory, usa il carattere jolly (*) per la corrispondenza dei pattern.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Specifica l'indirizzo IP di ascolto dell'agent per le connessioni dal server/proxy. Un elenco delimitato da virgole.
 Il primo indirizzo IP viene inviato al server Zabbix, se ci si connette a esso, per recuperare l'elenco dei controlli attivi.

Predefinito: 0.0.0.0

ListenPort

Specifica la porta di ascolto dell'agent per le connessioni dal server/proxy.

Predefinito: 10050
 Intervallo: 1024-32767

LogFile

Specifica il file di log dell'agent.

Predefinito: /tmp/zabbix_agent2.log
 Obbligatorio: Sì, se LogType è impostato su file; altrimenti no

LogFileSize

Imposta la dimensione massima del file di log (in MB).
0 - disabilita la rotazione automatica del log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file fallisce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024

LogType

Specifica il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro LogFile; - *system* - scrive il log in syslog; - *console* - scrive il log nell'output standard.

Predefinito: file

PersistentBufferFile

Specifica il file per la conservazione del database SQLite. Deve essere un nome file completo.
 Questo parametro viene utilizzato solo se il buffer persistente è abilitato (EnablePersistentBuffer=1).

PersistentBufferPeriod

Imposta il periodo di tempo per l'archiviazione dei dati quando non c'è connessione al server Zabbix o al proxy. I dati più vecchi andranno persi.
 I dati di log verranno conservati. Questo parametro viene utilizzato solo se il buffer persistente è abilitato (EnablePersistentBuffer=1).

Predefinito: 1h
 Intervallo: 1m-365d

PidFile

Specifica il file PID.

Predefinito: /tmp/zabbix_agent2.pid

Plugin.<PluginName>.System.Capacity

Imposta il limite di controlli per il plugin <PluginName> che possono essere eseguiti contemporaneamente.

Predefinito: 1000 Intervallo: 1-1000

Plugins.Log.MaxLinesPerSecond

Imposta il numero massimo di nuove righe al secondo da inviare a Zabbix server/proxy durante l'elaborazione dei controlli attivi log o logrt.
 Il valore fornito verrà sovrascritto dal parametro maxlines, specificato nella chiave dell'item log e logrt.
Nota: Zabbix elaborerà 10 volte più nuove righe rispetto a quelle impostate in MaxLinesPerSecond per cercare la stringa richiesta negli item di log.

Predefinito: 20
 Intervallo: 1-1000

Plugins.SystemRun.LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi.
 I comandi verranno registrati solo se eseguiti in remoto.
 Le voci di log non verranno create se system.run[] viene avviato localmente dai parametri HostMetadataItem, HostInterfaceItem o HostnameItem.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

PluginSocket

Specificare il percorso del socket UNIX per le comunicazioni dei plugin caricabili.

Predefinito: /tmp/agent.plugin.sock

PluginTimeout

Specifica il timeout per le connessioni ai plugin caricabili, in secondi.

Predefinito: Timeout
 Intervallo: 1-30

ProfilerDir

La directory in cui verranno scritti i dati di profilazione.

Predefinito: /tmp/zabbix_agent2_profiles

ProfilerMaxFilesPerProfile

Il numero massimo di file da conservare per ogni tipo di profilo. Il limite viene applicato ai file completati dopo ogni dump del profilo. Il profilo CPU attivo viene raccolto in un file temporaneo nascosto e non viene conteggiato ai fini del limite. Anche i file creati da periodic_prof_execute sono soggetti al limite.

Predefinito: 10
 Intervallo: 1-100

ProfilerInterval

L'intervallo di scrittura dei dati di profilazione, in secondi.

Predefinito: 3600
 Intervallo: 1-86400

RefreshActiveChecks

Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi, in secondi.
 Nota che, dopo un tentativo fallito di aggiornamento dei controlli attivi, il successivo aggiornamento verrà eseguito dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Specifica gli indirizzi IP, opzionalmente in notazione CIDR, oppure i nomi DNS dei server/proxy Zabbix. Un elenco delimitato da virgole.
 Le connessioni in ingresso saranno accettate solo dagli host elencati qui. Se il supporto IPv6 è abilitato, allora 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 sono trattati allo stesso modo e ::/0 consentirà qualsiasi indirizzo IPv4 o IPv6. 0.0.0.0/0 può essere usato per consentire qualsiasi indirizzo IPv4. Sono consentiti gli spazi. Se questo parametro non è specificato, disabilita i controlli passivi e l'agent non ascolterà su alcuna porta TCP.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
 L'indirizzo del server/proxy è un indirizzo IP o un nome DNS, con una porta opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo di server o proxy separati da punto e virgola. È possibile specificare

più server/cluster Zabbix e proxy Zabbix, separati da virgola. A meno che non si usino gruppi di proxy, non dovrebbe essere specificato più di un proxy Zabbix per ciascun server/cluster Zabbix. Se viene specificato un proxy Zabbix, allora non dovrebbe essere specificato il server/cluster Zabbix per quel proxy.
È possibile fornire più indirizzi separati da virgole per usare in parallelo più server Zabbix indipendenti. Sono consentiti gli spazi.
Se la porta non è specificata, viene usata la porta predefinita.
Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre se per quell'host è specificata una porta. Se la porta non è specificata, le parentesi quadre per gli indirizzi IPv6 sono facoltative.
Se questo parametro non è specificato, i controlli attivi sono disabilitati.

Esempio per proxy Zabbix:

```
ServerActive=127.0.0.1:10051
```

Esempio per gruppo di proxy Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Esempio per più server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Esempio per alta disponibilità:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Esempio per alta disponibilità con due cluster e un server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Specifica l'indirizzo IP di origine per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy.
- la creazione di connessioni durante l'esecuzione di alcuni item (`web.page.get`, `net.tcp.port`, ecc.).

StatusPort

Specifica la porta per l'ascolto delle richieste di stato HTTP (`http://localhost:<port>/status`).

Intervallo: 1024-32767

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con il proxy o il server di Zabbix.

Questo parametro definisce la durata di varie operazioni di comunicazione, tra cui:

- l'attesa di una risposta dal server Zabbix;
- l'invio di richieste al server Zabbix, incluse le richieste di configurazione degli item e i dati degli item nelle active checks;
- il recupero dei dati di log tramite logfile;
- l'invio di messaggi heartbeat;
- la durata massima per i controlli `vfs.*`;
- l'uso come fallback negli scenari in cui un server o un proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli dell'agent che hanno impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o per singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Specifica il livello di crittografia richiesto per le connessioni in ingresso.
 Usato per i controlli passivi.
 È possibile specificare più valori, separati da virgola: - *unencrypted* - accetta connessioni senza crittografia (predefinito); - *psk* - accetta connessioni con TLS e una chiave precondivisa (PSK); - *cert* - accetta connessioni con TLS e un certificato.

Obbligatorio: sì, se sono definiti i parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCAFile

Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate con i componenti di Zabbix.

TLSCipherAll

Specifica la stringa di cifratura OpenSSL (TLS 1.2). Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificati e PSK.

Esempio:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Specifica la stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificati e su PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Specifica la stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificato.

Esempio:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Nota che questo parametro non può essere usato insieme a `TLSAccept=cert,psk`; per una connessione con certificato (`TLSConnect=cert`) usa invece `TLSCipherAll`.

TLSCipherCert13

Specifica la stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su certificato.

Nota che questo parametro non può essere usato insieme a `TLSAccept=cert,psk`; per una connessione con certificato (`TLSConnect=cert`) usa invece `TLSCipherAll13`.

TLSCipherPSK

Specifica la stringa di cifrari OpenSSL (TLS 1.2). Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su PSK.

Esempio:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Specifica la stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Specifica il livello di crittografia richiesto per le connessioni dell'agent a Zabbix server o proxy.
 Usato per i controlli attivi.
 Può essere specificato un solo valore: - *unencrypted* - connessione senza crittografia (predefinito); - *psk* - connessione tramite TLS e una chiave precondivisa (PSK); - *cert* - connessione tramite TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCRLFile

Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSPSKFile

Specifica il percorso completo del file contenente la chiave pre-condivisa dell'agent, utilizzata per le comunicazioni crittografate con il server o il proxy di Zabbix.

TLSPSKIdentity

Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con Zabbix server o proxy.

TLSServerCertIssuer

Specifica l'emittente del certificato del server (proxy) consentito.

TLSServerCertSubject

Specifica il subject del certificato del server consentito (proxy).

UnsafeUserParameters

Consente di passare tutti i caratteri negli argomenti dei parametri definiti dall'utente. I seguenti caratteri non sono consentiti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Inoltre, i caratteri di nuova riga non sono consentiti.

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

UserParameter

Specifica un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: `UserParameter=<key>,<shell command>`
Nota che il comando shell non deve restituire una stringa vuota o solo EOL. I comandi shell possono avere percorsi relativi, se è specificato il parametro `UserParameterDir`.

Esempio:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Specifica il percorso di ricerca predefinito per i comandi `UserParameter`. Se utilizzato, l'agent cambierà la propria directory di lavoro in quella specificata qui prima di eseguire un comando. In questo modo, i comandi UserParameter possono avere un prefisso relativo `./` invece di un percorso completo.
È consentita una sola voce.

Esempio:

```
UserParameterDir=/opt/myscripts
```

Parametri di configurazione: Zabbix agent 2 (Windows)

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento di Zabbix agent 2 su Windows.

Specificare i valori dei parametri nel file di configurazione di agent 2 di Zabbix (`zabbix_agent2.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un pattern.
AllowKeyRegexp	Consente l'esecuzione delle chiavi item che corrispondono a un pattern basato su un'espressione regolare.
BufferSend	Imposta la frequenza di invio dei valori dal buffer al server Zabbix.
BufferSize	Imposta il numero massimo di valori nel buffer di memoria.
ControlSocket	Specifica il socket di controllo utilizzato per inviare comandi runtime con l'opzione -R.
DebugLevel	Imposta il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un pattern.
DenyKeyRegexp	Nega l'esecuzione delle chiavi item che corrispondono a un pattern basato su un'espressione regolare.
EnablePersistentBuffer	Abilita l'utilizzo dell'archiviazione persistente locale per gli item attivi.
EnableProfiler	Abilita la profilazione periodica di Go all'avvio di agent.
ForceActiveChecksOnStart	Esegue immediatamente i controlli attivi dopo il riavvio per la prima configurazione ricevuta.
HeartbeatFrequency	Imposta la frequenza dei messaggi heartbeat in secondi.
HostInterface	Specifica l'interfaccia del host.

Parametro	Descrizione
HostInterfaceItem	Specifica un item per ottenere l'interfaccia del host.
HostMetadata	Specifica i metadati del host.
HostMetadataItem	Specifica un item di Zabbix agent utilizzato per ottenere i metadati del host.
Hostname	Specifica il nome del host.
HostnameItem	Specifica un item di Zabbix agent per ottenere il nome del host.
Include	Specifica una directory o singoli file da includere nel file di configurazione.
ListenIP	Specifica l'indirizzo IP di ascolto di agent per le connessioni dal server/proxy.
ListenPort	Specifica la porta di ascolto di agent.
LogFile	Specifica il file di log di agent.
LogFileSize	Imposta la dimensione massima del file di log.
LogType	Specifica il tipo di output del log.
PerfCounter	Specifica un nuovo parametro <parameter_name>, che rappresenta il valore medio del contatore delle prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi).
PerfCounterEn	Specifica un nuovo parametro <parameter_name>, che rappresenta il valore medio del contatore delle prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi). Rispetto a PerfCounter, i percorsi dei contatori delle prestazioni devono essere in inglese.
PersistentBufferFile	Specifica il file in cui Zabbix agent 2 deve conservare il database SQLite.
PersistentBufferPeriod	Imposta la durata di conservazione dei dati quando non è disponibile una connessione al server o al proxy.
Plugins.<PluginName>.System.Capacity	Imposta il limite dei controlli per plugin che possono essere eseguiti contemporaneamente.
Plugins.Log.MaxLinesPerSecond	Imposta il numero massimo di nuove righe che agent invierà al server o al proxy Zabbix ogni secondo durante l'elaborazione dei controlli attivi `log`, `logrt` ed `eventlog`.
Plugins.SystemRun.LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
PluginSocket	Specifica il percorso della named pipe di Windows per le comunicazioni con i plugin caricabili.
PluginTimeout	Imposta il timeout per le connessioni con i plugin caricabili, in secondi.
ProfilerDir	La directory in cui verranno scritti i dati di profilazione.
ProfilerMaxFilesPerProfile	Il numero massimo di file da conservare per ogni tipo di profilo.
ProfilerInterval	L'intervallo di scrittura dei dati di profilazione.
RefreshActiveChecks	Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi.
Server	Specifica gli indirizzi IP o i nomi DNS dei server/proxy Zabbix.
ServerActive	Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	Specifica l'indirizzo IP di origine.
StatusPort	Imposta la porta su cui agent ascolta le richieste HTTP di stato (`http://localhost:<port>/status`).
Timeout	Specifica per quanto tempo attendere (in secondi) la stabilizzazione della connessione e lo scambio di dati con il server o il proxy Zabbix.
TLSAccept	Specifica il livello di crittografia richiesto per le connessioni in entrata.
TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCertFile	Specifica il percorso completo del file contenente il certificato o la catena di certificati di agent, utilizzati per le comunicazioni crittografate tra i componenti Zabbix.
TLSCipherAll13	Specifica la stringa di cifratura OpenSSL (TLS 1.3). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert13	Specifica la stringa di cifratura OpenSSL (TLS 1.3). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK13	Specifica la stringa di cifratura OpenSSL (TLS 1.3). Sostituisce i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSConnect	Specifica il livello di crittografia richiesto per le connessioni al server o al proxy Zabbix.
TLSCRLFile	Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti Zabbix.
TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata di agent, utilizzata per le comunicazioni crittografate tra i componenti Zabbix.
TLSPSKFile	Specifica il percorso completo del file contenente la chiave precondivisa di agent, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSPSKIdentity	Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.
TLSServerCertIssuer	Specifica l'emittente del certificato del server (proxy) consentito.
TLSServerCertSubject	Specifica il subject del certificato del server (proxy) consentito.

Parametro	Descrizione
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti dei parametri definiti dall'utente.
UserParameter	Specifica un parametro definito dall'utente da monitorare.
UserParameterDir	Specifica il percorso di ricerca predefinito per i comandi `UserParameter`.

Tenere presente che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti;
- I valori supportano le variabili d'ambiente;
- Zabbix supporta i file di configurazione solo con codifica UTF-8 senza BOM;
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli del parametro

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più breve e semplice.
 Possono essere presenti più parametri `Alias`. Non sono consentiti più parametri con la stessa chiave `Alias`.
 Chiavi `Alias` diverse possono fare riferimento alla stessa chiave item.
 Gli alias possono essere utilizzati in `HostMetadataItem`, ma non nel parametro `HostnameItem`.

Esempio 1: recupero dell'utilizzo del file di paging in percentuale dal server.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Ora è possibile utilizzare la chiave abbreviata `pg_usage` per recuperare i dati.

Esempio 2: recupero del carico della CPU con parametri predefiniti e personalizzati.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Ciò consente di utilizzare la chiave `cpu.load` per ottenere il carico della CPU con i parametri predefiniti e di utilizzare `cpu.load[percpu,avg15]` per ottenere dati specifici sul carico della CPU.

Esempio 3: esecuzione di più regole di low-level discovery che elaborano gli stessi item di discovery.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Ora è possibile configurare diverse regole di discovery utilizzando `vfs.fs.discovery`, con parametri diversi per ciascuna regola, ad esempio `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` e così via.

AllowKey

Consente l'esecuzione di quegli item key che corrispondono a un pattern. Il pattern della key è un'espressione wildcard che supporta il carattere * per corrispondere a un numero qualsiasi di caratteri.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. L'ordine determina quale regola corrispondente avrà la priorità. Vedi anche: Restricting agent checks.

AllowKeyRegexp

Consente l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `DenyKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità. Vedere anche: Limitazione dei controlli dell'agent.

Esempio:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Imposta la frequenza di invio dei valori dal buffer a Zabbix server/proxy (in secondi).
Nota che, se il buffer è pieno, i dati verranno inviati prima.

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Imposta il numero massimo di valori nel buffer di memoria.
 L'agent invierà tutti i dati raccolti al server Zabbix o al proxy se il buffer è pieno.
Questo parametro dovrebbe essere usato solo se il buffer persistente è disabilitato (`EnablePersistentBuffer=0`).

Predefinito: 1000
 Intervallo: 2-65535

ControlSocket

Specificare il socket di controllo, utilizzato per inviare comandi a runtime con l'opzione `-R`.

Predefinito: `\\.\pipe\agent.sock`

DebugLevel

Imposta il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (produce molte informazioni); - *5* - debug esteso (produce ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

DenyKey

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione con caratteri jolly che supporta il carattere `*` per trovare un numero qualsiasi di caratteri.
 È possibile definire più regole di corrispondenza delle key in combinazione con `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola corrispondente avrà la priorità.

Vedere anche: Restricting agent checks.

Predefinito: `DenyKey=system.run[*]`

DenyKeyRegexp

Nega l'esecuzione degli item key che corrispondono a un pattern. Il pattern della key è un'espressione regolare.
 È possibile definire più regole di corrispondenza delle key in combinazione con `AllowKeyRegexp`, `AllowKey` e `DenyKey`. I parametri vengono elaborati uno alla volta, in base all'ordine in cui compaiono. L'ordine determina quale regola di corrispondenza avrà la priorità.
 Tutti gli item `system.run[]` sono negati per impostazione predefinita (`DenyKey=system.run[*]`), a meno che non venga specificata un'altra regola `system.run[*]`.

Vedere anche: Limitazione dei controlli dell'agent.

EnablePersistentBuffer

Abilita l'uso dell'archiviazione persistente locale per gli item attivi. Se l'archiviazione persistente è disabilitata, verrà utilizzato il buffer in memoria.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

EnableProfiler

Abilita il profiling periodico di Go all'avvio dell'agent. Il profiling può essere abilitato o disabilitato anche durante l'esecuzione utilizzando l'opzione `-R`. Mentre `periodic_prof_execute` raccoglie i profili, tutti i comandi runtime del profiler vengono rifiutati perché il profiler è occupato.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

ForceActiveChecksOnStart

Esegue i controlli attivi immediatamente dopo il riavvio per la prima configurazione ricevuta. Disponibile anche come parametro di configurazione per singolo plugin, ad esempio: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

HeartbeatFrequency

Imposta la frequenza dei messaggi heartbeat in secondi. Usato per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Specifica l'interfaccia dell'host (indirizzo IP o nome DNS) utilizzata durante l'autoregistrazione dell'host. Questo valore verrà usato per popolare l'interfaccia nel nuovo host creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per ulteriori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da HostInterfaceItem.

L'agent segnalerà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Specificare un item per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante la registrazione automatica del host. Questo valore viene utilizzato solo se `HostInterface` non è definito. Per ulteriori dettagli, vedere Utilizzo del DNS come interfaccia predefinita.

Durante una richiesta di registrazione automatica, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 255 caratteri.

L'item `system.run[]` è supportato indipendentemente dalle impostazioni `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

HostMetadata

Specifica i metadati utilizzati per identificare o distinguere l'host durante l'autoregistrazione dell'host (agent attivo). `HostMetadata` consente di distinguere gli host oltre il nome host.

Se non definito, il valore verrà acquisito da `HostMetadataItem`.

L'agent genererà un errore e non si avvierà se il valore specificato supera il limite di 2034 byte o se è una stringa non UTF-8. Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori UTF-8 validi ma non validi come IP o nomi DNS verranno rifiutati e segnalati come non validi.

I metadati su più righe non sono supportati: l'output verrà troncato al primo ritorno a capo.

Intervallo: 0-2034 byte

HostMetadataItem

Specificare un item dell'agent Zabbix per ottenere i metadati dell'host. Questa opzione viene utilizzata solo quando `HostMetadata` non è definito.

Il valore di `HostMetadataItem` viene recuperato a ogni tentativo di autoregistrazione e viene utilizzato solo durante il processo di autoregistrazione dell'host (agent attivo). `HostMetadataItem` consente di distinguere gli host oltre il nome host.

Sono supportati i parametri utente e gli alias. L'item `system.run[]` è supportato indipendentemente dalle impostazioni `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

Durante una richiesta di autoregistrazione, l'agent registrerà un messaggio di avviso se il valore restituito dall'item specificato supera il limite di 65535 punti di codice UTF-8. Il valore restituito dall'item deve essere una stringa UTF-8; in caso contrario, verrà ignorato. Quando un parametro prevede un indirizzo IP o un nome DNS, verranno rifiutati e segnalati come non validi anche i valori con codifica UTF-8 valida ma che non sono indirizzi IP o nomi DNS validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Nome host

Specifica hostname univoci e sensibili alle maiuscole/minuscole. Un elenco delimitato da virgole.
 Obbligatorio per i controlli attivi e deve corrispondere agli hostname configurati sul server. Il valore viene acquisito da HostnameItem se non definito.
 Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per hostname, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Specificare un item per ottenere il nome host. Questa opzione viene utilizzata solo quando `Hostname` non è definito. I parametri utente o gli alias non sono supportati, ma l'item `system.run[]` è supportato indipendentemente dai valori di `AllowKey`, `AllowKeyRegexp`, `DenyKey` o `DenyKeyRegexp`.

Predefinito: `system.hostname`

Include

Specificare una directory o singoli file da includere nel file di configurazione (situato per impostazione predefinita in `C:\Program Files\Zabbix Agent 2` se Zabbix agent è installato utilizzando i pacchetti Windows MSI; situato nella cartella specificata durante l'installazione se Zabbix agent è installato come archivio zip). Durante l'installazione di Zabbix, questo parametro viene impostato su `/usr/local/etc`, a meno che non venga modificato durante la compilazione.

Regole generali per il parametro `Include`:

- Il percorso può essere relativo alla posizione del file `zabbix_agent2.conf` (ad esempio, `Include=.\zabbix_agent2.d\plugins.d`
- È possibile specificare più parametri `Include` in un singolo file di configurazione.
- Zabbix elabora il parametro `Include` non appena lo legge; pertanto, è consigliabile inserirlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non verrà avviato.

Regole per l'inclusione di directory o pattern di file:

- Non presumere alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup create automaticamente da alcuni editor di testo (ad esempio, la modifica di `include/my_specific.conf` può creare `include/my_specific.conf.BAK`), poiché entrambi i file verranno inclusi.

Per includere solo i file pertinenti da una directory, utilizzare il carattere jolly (`*`) per la corrispondenza dei pattern.

Esempio:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Specifica l'indirizzo IP di ascolto dell'agent per le connessioni dal server/proxy. Un elenco delimitato da virgole.
 Il primo indirizzo IP viene inviato al server Zabbix, se ci si connette ad esso, per recuperare l'elenco dei controlli attivi.

Predefinito: `0.0.0.0`

ListenPort

Specifica la porta di ascolto dell'agent per le connessioni dal server/proxy.

Predefinito: 10050
 Intervallo: 1024-32767

LogFile

Specificare il file di log dell'agent.

Predefinito: `c:\zabbix_agent2.log`
 Obbligatorio: Sì, se `LogType` è impostato su `file`; altrimenti no

LogFileSize

Imposta la dimensione massima del file di log (in MB).
0 - disabilita la rotazione automatica del log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file non riesce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024

LogType

Specifica il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro `LogFile`; - *console* - scrive il log sull'output standard.

Predefinito: `file`

PerfCounter

Definisci un nuovo parametro <parameter_name> che rappresenta il valore medio del contatore di prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi).
Sintassi: <parameter_name>,"<perf_counter_path>",<p

Ad esempio, se desideri ricevere il numero medio di interrupt del processore al secondo nell'ultimo minuto, puoi definire un nuovo parametro "interrupts" come segue:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Tieni presente le virgolette doppie attorno al percorso del contatore di prestazioni. Il nome del parametro (interrupts) deve essere usato come chiave dell'item quando si crea un item. I campioni per il calcolo del valore medio verranno raccolti ogni secondo.
Puoi eseguire `typeperf -qx` per ottenere l'elenco di tutti i contatori di prestazioni disponibili in Windows.

PerfCounterEn

Definisce un nuovo parametro <parameter_name> che è il valore medio del contatore delle prestazioni di sistema <perf_counter_path> per il periodo di tempo specificato <period> (in secondi). Rispetto a PerfCounter, i percorsi dei perfcounter devono essere in inglese. Supportato solo su **Windows Server 2008/Vista** e versioni successive.
Sintassi: <parameter_name>,"<perf_counter_path>",<period>

Ad esempio, se si desidera ricevere il numero medio di interrupt del processore al secondo per l'ultimo minuto, è possibile definire un nuovo parametro "interrupts" come segue:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Si noti l'uso delle virgolette doppie attorno al percorso del contatore delle prestazioni. Il nome del parametro (interrupts) deve essere usato come chiave dell'item quando si crea un item. I campioni per il calcolo del valore medio verranno raccolti ogni secondo.
È possibile trovare l'elenco delle stringhe in inglese visualizzando la seguente chiave del registro: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009.

PersistentBufferFile

Specifica il file in cui Zabbix agent 2 deve conservare il database SQLite. Deve essere un nome file completo.
 Questo parametro viene utilizzato solo se il buffer persistente è abilitato (EnablePersistentBuffer=1).

PersistentBufferPeriod

Imposta la durata di conservazione dei dati quando non c'è connessione al server o al proxy. I dati più vecchi andranno persi. I dati di log verranno conservati.
 Questo parametro viene usato solo se il buffer persistente è abilitato (EnablePersistentBuffer=1).

Predefinito: 1h
 Intervallo: 1m-365d

Plugin.<PluginName>.System.Capacity

Imposta il limite di controlli per il plugin <PluginName> che possono essere eseguiti contemporaneamente.

Predefinito: 1000 Intervallo: 1-1000

Plugins.Log.MaxLinesPerSecond

Imposta il numero massimo di nuove righe che l'agent invierà al server o al proxy di Zabbix al secondo durante l'elaborazione dei controlli attivi log, logrt ed eventlog. Il valore fornito verrà sovrascritto dal parametro maxlines, specificato nella chiave dell'item log, logrt o eventlog.
Nota: Zabbix elaborerà un numero di nuove righe 10 volte superiore a quello impostato in MaxLinesPerSecond per cercare la stringa richiesta negli item di log.

Predefinito: 20
 Intervallo: 1-1000

Plugins.SystemRun.LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi. I comandi verranno registrati solo se eseguiti in remoto. Le voci di log non verranno create se system.run[] viene avviato localmente dai parametri HostMetadataItem, HostInterfaceItem o HostnameItem.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

PluginSocket

Specifica il percorso della named pipe di Windows per le comunicazioni dei plugin caricabili.

Predefinito: \\.\pipe\agent.plugin.sock

PluginTimeout

Imposta il timeout per le connessioni con i plugin caricabili, in secondi.

Predefinito: Timeout
 Intervallo: 1-30

ProfilerDir

La directory in cui verranno scritti i dati di profilazione.

Predefinito: c:\zabbix_pprof

ProfilerMaxFilesPerProfile

Il numero massimo di file da conservare per ogni tipo di profilo. Il limite viene applicato ai file completati dopo ogni dump del profilo. Il profilo CPU attivo viene raccolto in un file temporaneo nascosto e non viene conteggiato ai fini del limite. Anche i file creati da periodic_prof_execute sono soggetti al limite.

Predefinito: 10
 Intervallo: 1-100

ProfilerInterval

L'intervallo di scrittura dei dati di profilazione, in secondi.

Predefinito: 3600
 Intervallo: 1-86400

RefreshActiveChecks

Specifica la frequenza di aggiornamento dell'elenco dei controlli attivi, in secondi.
 Nota che, dopo un tentativo fallito di aggiornamento dei controlli attivi, il successivo aggiornamento verrà eseguito dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Specificare gli indirizzi IP, opzionalmente in notazione CIDR, oppure i nomi DNS dei server/proxy Zabbix. Un elenco delimitato da virgole.
 Se il parametro non è specificato, i controlli passivi sono disabilitati e l'agent non ascolterà su alcuna porta TCP.
 Le connessioni in ingresso saranno accettate solo dagli host elencati qui. Se il supporto IPv6 è abilitato, allora `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4. Sono consentiti gli spazi.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Specifica l'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
 Se il parametro non è specificato, i controlli attivi sono disabilitati.

L'indirizzo del server/proxy è un indirizzo IP o un nome DNS con una porta opzionale separata da due punti. Se la porta non è specificata, viene usata la porta predefinita.
 Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre se per quell'host è specificata una porta. Se la porta non è specificata, le parentesi quadre per gli indirizzi IPv6 sono opzionali.

La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo di server o proxy separati da punto e virgola.

È possibile specificare più server/cluster Zabbix e proxy Zabbix, separati da virgola.

A meno che non si usino gruppi di proxy, non dovrebbe essere specificato più di un proxy Zabbix per ciascun server/cluster Zabbix. Se viene specificato un proxy Zabbix, allora non deve essere specificato il server/cluster Zabbix per quel proxy.

È possibile fornire più indirizzi separati da virgole per usare più server Zabbix indipendenti in parallelo. Sono consentiti gli spazi.

Esempio per proxy Zabbix:

```
ServerActive=127.0.0.1:10051
```

Esempio per gruppo di proxy Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Esempio per più server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

Esempio per alta disponibilità:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Esempio per alta disponibilità con due cluster e un server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Specifica l'indirizzo IP di origine per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy;
- la creazione di connessioni durante l'esecuzione di alcuni item (`web.page.get`, `net.tcp.port`, ecc.).

StatusPort

Imposta la porta su cui l'agent è in ascolto per le richieste di stato HTTP (`http://localhost:<port>/status`).

Intervallo: 1024-32767

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix proxy o server.

Questo parametro definisce la durata di varie operazioni di comunicazione, tra cui:

- attesa di una risposta dal server Zabbix;
- invio di richieste al server Zabbix, incluse le richieste di configurazione degli item e i dati degli item nelle active checks;
- recupero dei dati dei log tramite il monitoraggio dei file di log o del registro eventi di Windows;
- invio di messaggi heartbeat;
- durata massima per i controlli `vfs.*`;
- utilizzo come fallback negli scenari in cui un server o un proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli dell'agent che dispongono di impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o per singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Specifica il livello di crittografia richiesto per le connessioni in ingresso.
 Usato per i controlli passivi.

È possibile specificare più valori, separati da virgole:

- *unencrypted* - accetta connessioni senza crittografia (predefinito);
- *psk* - accetta connessioni con TLS e una chiave precondivisa (PSK);
- *cert* - accetta connessioni con TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCAFile

Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate con i componenti di Zabbix.

TLSCipherAll13

Specifica la stringa di cifratura OpenSSL (TLS 1.3). Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificati e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Specifica la stringa di cifratura OpenSSL (TLS 1.3). Sostituisce i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificato.

Nota che questo parametro non può essere usato insieme a `TLSAccept=cert,psk`; per una connessione con certificato (`TLSConnect=cert`) usa invece `TLSCipherAll13`.

TLSCipherPSK13

Specifica la stringa di cifratura OpenSSL (TLS 1.3). Sostituisce i criteri di selezione della suite di cifratura predefinita per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Specifica il livello di crittografia richiesto per le connessioni dell'agent a Zabbix server o proxy.
 Usato per i controlli attivi.
 È possibile specificare un solo valore: - *unencrypted* - connessione senza crittografia (predefinito); - *psk* - connessione tramite TLS e una chiave precondivisa (PSK); - *cert* - connessione tramite TLS e un certificato.

Obbligatorio: sì, se sono definiti parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCRLFile

Specifica il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSPSKFile

Specifica il percorso completo del file contenente la chiave pre-condivisa dell'agent, utilizzata per le comunicazioni crittografate con il server Zabbix o il proxy.

TLSPSKIdentity

Specifica la stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con il server o il proxy Zabbix.

TLSServerCertIssuer

Specifica l'emittente del certificato del server (proxy) consentito.

TLSServerCertSubject

Specifica il subject del certificato del server consentito (proxy).

UnsafeUserParameters

Consente di passare tutti i caratteri negli argomenti dei parametri definiti dall'utente. I seguenti caratteri non sono consentiti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Inoltre, i caratteri di nuova riga non sono consentiti.

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

UserParameter

Specifica un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: `UserParameter=<key>,<shell command>`
Nota che il comando shell non deve restituire una stringa vuota o solo EOL. I comandi shell possono usare percorsi relativi, se è specificato il parametro `UserParameterDir`.

Esempio:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Specifica il percorso di ricerca predefinito per i comandi `UserParameter`. Se utilizzato, l'agent cambierà la propria directory di lavoro in quella specificata qui prima di eseguire un comando. In questo modo, i comandi UserParameter possono avere un prefisso relativo `.\` invece di un percorso completo.
È consentita una sola voce.

Esempio:

```
UserParameterDir=C:\opt\myscripts
```

Parametri di configurazione: Plugin

Vedi i parametri di configurazione per i plugin di Zabbix agent 2:

- plugin Ceph
- plugin Docker
- plugin Ember+
- plugin Memcached
- plugin Modbus
- plugin MongoDB
- plugin MQTT
- plugin MSSQL
- plugin MySQL
- plugin NVIDIA GPU
- plugin Oracle
- plugin PostgreSQL
- plugin Redis
- plugin S.M.A.R.T.

Principi di configurazione dei plugin

Questa pagina descrive i principi di configurazione dei plugin e le procedure consigliate.

Tutti i plugin vengono configurati utilizzando il parametro `Plugins.*`, che può essere incluso nel file di configurazione dell'agent 2 di Zabbix oppure nel file di configurazione del plugin. Se un plugin utilizza un file di configurazione separato, il percorso di questo file deve essere specificato nel parametro `Include` del file di configurazione dell'agent 2 di Zabbix.

Un parametro tipico del plugin ha la seguente struttura:

`Plugins.<PluginName>.<Parameter>=<Value>`

Inoltre, esistono due gruppi specifici di parametri:

- `Plugins.<PluginName>.Default.<Parameter>=<Value>` utilizzato per definire i valori predefiniti dei parametri.
- `Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>` utilizzato per definire set separati di parametri per diversi target di monitoraggio tramite sessioni denominate.

Tutti i nomi dei parametri devono rispettare i seguenti requisiti:

- È consigliabile utilizzare le maiuscole nei nomi dei plugin.
- Il parametro deve iniziare con una lettera maiuscola.
- I caratteri speciali non sono consentiti.
- Il livello di annidamento non è soggetto a un limite massimo.
- Il numero di parametri non è limitato.

Ad esempio, per eseguire i controlli attivi che non hanno un *intervallo di aggiornamento* di Scheduling subito dopo il riavvio dell'agent, solo per il plugin Uptime, impostare `Plugins.Uptime.System.ForceActiveChecksOnStart=1` nel file di configurazione. Analogamente, per impostare un limite personalizzato per i controlli simultanei del plugin CPU, impostare `Plugins.CPU.System.Capacity=N` nel file di configurazione.

Valori predefiniti

È possibile impostare i valori predefiniti per i parametri relativi alla connessione (URI, nome utente, password, ecc.) nel file di configurazione nel formato:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

Ad esempio, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, ecc.

Se il valore di tale parametro non viene fornito in una chiave item o nei parametri della sessione denominata, il plugin utilizzerà il valore predefinito. Se anche un parametro predefinito non è definito, verranno utilizzati i valori predefiniti codificati.

> **Note:**
> Se una chiave item non contiene alcun parametro, Zabbix agent 2 tenterà di raccogliere la metrica utilizzando i valori definiti nella sezione dei parametri predefiniti.

Sessioni denominate

Le sessioni denominate rappresentano un ulteriore livello di parametri del plugin e possono essere utilizzate per specificare insiemi separati di parametri di autenticazione per ciascuna delle istanze monitorate. Ogni parametro di sessione denominata deve avere la seguente struttura:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Un nome di sessione può essere usato come parametro chiave dell'elemento connString invece di specificare separatamente URI, nome utente e/o password.

Nelle chiavi degli item, il primo parametro può essere sia un connString sia un URI. Se il primo parametro della chiave non corrisponde a nessun nome di sessione, verrà trattato come un URI. Si noti che non è supportato passare credenziali URI incorporate nella chiave dell'item; usare invece i parametri di sessione denominati.

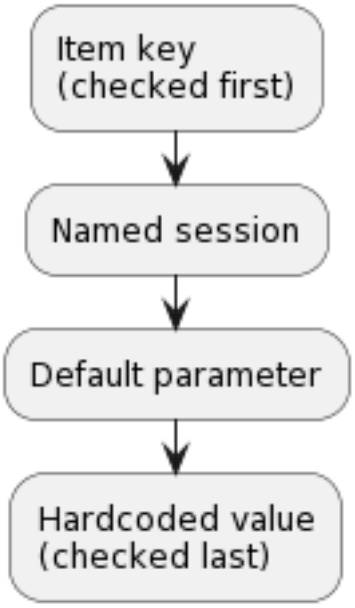
L'elenco dei parametri di sessione denominati disponibili dipende dal plugin.

È possibile sovrascrivere i parametri di sessione specificando nuovi valori nei parametri della chiave dell'item (vedere esempio).

Se un parametro non è definito per la sessione denominata, Zabbix agent 2 utilizzerà il valore definito nel parametro predefinito del plugin.

Priorità dei parametri

I plugin di Zabbix agent 2 cercano i valori dei parametri relativi alla connessione nel seguente ordine:



1. Il primo parametro della chiave dell'item viene confrontato con i nomi delle sessioni. Se non viene trovato alcun corrispondenza, viene trattato come un valore effettivo; in questo caso, il passaggio 3 verrà saltato. Se viene trovata una corrispondenza, il valore del parametro (di solito un URI) deve essere definito nella sessione denominata.
2. Gli altri parametri verranno presi dalla chiave dell'item, se definiti.
3. Se un parametro della chiave dell'item (ad esempio, password) è vuoto, il plugin cercherà il parametro corrispondente della sessione denominata.
4. Se anche il parametro della sessione non è specificato, verrà utilizzato il valore definito nel corrispondente parametro predefinito.
5. Se tutto il resto fallisce, il plugin utilizzerà il valore predefinito codificato.

Esempio 1

Monitoraggio di due istanze "MySQL1" e "MySQL2".

Parametri di configurazione:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.1:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

Come risultato di questa configurazione, ogni nome di sessione può essere utilizzato come connString in una chiave item, ad esempio `mysql.ping[MySQL1]` oppure `mysql.ping[MySQL2]`.

Esempio 2

Fornire alcuni dei parametri nella chiave dell'item.

Parametri di configurazione:

```
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
```

Chiave dell'item: `pgsql.ping[session1,new_username,,postgres]`

Come risultato di questa configurazione, l'agent si connetterà a PostgreSQL utilizzando i seguenti parametri:

- URI dal parametro della sessione: `192.0.2.234:5432`
- Nome utente dalla chiave dell'item: `new_username`
- Password dal parametro della sessione (poiché è omessa nella chiave dell'item): `session_password`
- Nome del database dalla chiave dell'item: `postgres`

Esempio 3

Raccolta di una metrica usando i parametri di configurazione predefiniti.

Parametri di configurazione:

```
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]`

Come risultato di questa configurazione, l'agent si connetterà a PostgreSQL usando i parametri:

- URI predefinito: `192.0.2.234:5432`
- Nome utente predefinito: `zabbix`
- Password predefinita: `password`
- Nome del database dall'item key: `postgres`

Connessioni

Alcuni plugin supportano la raccolta di metriche da più istanze contemporaneamente. È possibile monitorare sia istanze locali sia remote. Sono supportate connessioni TCP e Unix-socket.

Si consiglia di configurare i plugin in modo che mantengano le connessioni alle istanze in stato aperto. I vantaggi sono la riduzione della congestione di rete, della latenza e dell'uso di CPU e memoria grazie al minor numero di connessioni. La libreria client si occupa di questo.

> **Note:**
> Il periodo di tempo per cui le connessioni inutilizzate devono rimanere aperte può essere determinato dal parametro `Plugins.<PluginName>.KeepAlive`. Esempio: `Plugins.Memcached.KeepAlive`

Plugin Ceph

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Ceph.

Si consiglia di specificarli in un file di configurazione dedicato (ad esempio, `ceph.conf`), anziché nel file di configurazione di agent 2. Utilizzare quindi la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Ceph.Default.ApiKey	Specifica la chiave API predefinita per la connessione a Ceph.
Plugins.Ceph.Default.Mode	Specifica la modalità predefinita per la connessione a Ceph.
Plugins.Ceph.Default.URI	Specifica l'URI predefinito per la connessione a Ceph.
Plugins.Ceph.Default.User	Specifica il nome utente predefinito per la connessione a Ceph.
Plugins.Ceph.InsecureSkipVerify	Specifica se un client HTTP deve verificare la catena del certificato del server e il nome host.
Plugins.Ceph.KeepAlive	Imposta il tempo massimo di attesa prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.Ceph.Sessions.<Session1>.ApiKey	Specifica la chiave API della sessione denominata per la connessione a Ceph.
Plugins.Ceph.Sessions.<Session1>.Mode	Specifica la modalità della sessione denominata per la connessione a Ceph.
Plugins.Ceph.Sessions.<Session1>.URI	Specifica l'URI della sessione denominata per la connessione a Ceph.
Plugins.Ceph.Sessions.<Session1>.User	Specifica il nome utente della sessione denominata per la connessione a Ceph.
Plugins.Ceph.Timeout	Imposta il timeout di esecuzione della richiesta.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione forniti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli dei parametri

Plugins.Ceph.Default.ApiKey

Specifica la chiave API predefinita per la connessione a Ceph.
 Utilizzata se non è specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Ceph.Default.Mode

Specifica la modalità predefinita per la connessione a Ceph.
 La modalità nativa è supportata solo su Linux e a partire da Ceph 16. La modalità restful non funzionerà con la versione 20 di Ceph o successive.
 Tieni presente che il set di credenziali utente differisce per ciascuna modalità e che non sono compatibili tra loro.

Predefinito: `restful` (deprecato)
 Intervallo: native/restful

Plugins.Ceph.Default.URI

Specifica l'URI predefinito per la connessione a Ceph.
 Usato se non è specificato alcun valore in una item key o in una sessione denominata.

Non deve includere credenziali incorporate (verranno ignorate).
Deve corrispondere al formato URI.
È supportato solo lo schema `https`; lo schema può essere omesso.
La porta può essere omessa (predefinita=8003).

Predefinito: `https://localhost:8003`

Esempi:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Specifica il nome utente predefinito per la connessione a Ceph.
 Usato se non è specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Ceph.InsecureSkipVerify

Specifica se un client HTTP deve verificare la catena del certificato del server e il nome host.
 Se *true*, TLS accetta qualsiasi certificato presentato dal server e qualsiasi nome host presente in tale certificato. In questa modalità, TLS è vulnerabile ad attacchi man-in-the-middle (usare solo per test).

Predefinito: `false`
 Intervallo: false/true

Plugins.Ceph.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Specifica la chiave API della sessione denominata per la connessione a Ceph.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Ceph.Sessions.<SessionName>.Mode

Specifica la modalità della sessione denominata per la connessione a Ceph.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 La modalità native è supportata solo su Linux e a partire da Ceph 16. La modalità restful non funzionerà con la versione 20 di Ceph o successive.
 Si noti che il set di credenziali utente differisce per ciascuna modalità e che non sono compatibili tra loro.

Predefinito: `restful` (deprecato)
 Intervallo: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Specifica l'URI della sessione denominata per la connessione a Ceph.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
Non deve includere credenziali incorporate (verranno ignorate).
Deve corrispondere al formato URI.
 È supportato solo lo schema `https`; lo schema può essere omesso.
La porta può essere omessa (predefinita=8003).

Predefinito: `https://localhost:8003`

Esempi:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per la connessione a Ceph.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Ceph.Timeout

Imposta il timeout di esecuzione della richiesta (durata di attesa affinché una richiesta venga completata prima di interromperla) in secondi.

Predefinito: `global timeout`
 Intervallo: 1-30

Vedi anche

- Istruzioni per configurare i plugin
- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)

Plugin Docker

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Docker.

Si consiglia di specificarli in un file di configurazione dedicato (ad esempio, `docker.conf`), anziché nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Docker.Endpoint	Specifica la posizione del socket UNIX del daemon Docker..
Plugins.Docker.Timeout	Specifica il timeout di esecuzione della richiesta.

Si noti che:

- I valori predefiniti riflettono i valori di default del processo, non quelli presenti nei file di configurazione forniti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli dei parametri

Plugins.Docker.Endpoint

Specifica la posizione del socket UNIX del daemon Docker.
 Deve contenere uno schema (è supportato solo `unix://`).

Predefinito: `unix:///var/run/docker.sock`

Plugins.Docker.Timeout

Specifica il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Predefinito: `global timeout`
 Intervallo: 1–30

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin Ember+

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Ember+.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `emberplus.conf`), anziché nel file di configurazione di agent 2. Quindi utilizzare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Il plugin Ember+ è un plugin caricabile ed è disponibile e descritto in modo completo nel repository del plugin Ember+.

Questo plugin è attualmente disponibile solo per la compilazione dai sorgenti (sia per Unix che per Windows).

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.EmberPlus.Default.Uri	Specifica l'URI predefinito per la connessione a Ember+.
Plugins.EmberPlus.KeepAlive	Imposta il tempo massimo di attesa prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Specifica l'URI della sessione denominata per la connessione a Ember+.
Plugins.EmberPlus.System.Path	Specifica il percorso dell'eseguibile del plugin Ember+.
Plugins.EmberPlus.Timeout	Imposta il timeout di esecuzione della richiesta.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione distribuiti.
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM.
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.EmberPlus.Default.Uri

Specifica l'URI predefinito per la connessione a Ember+.
 L'unico schema supportato è `tcp://`. Lo schema può essere omesso. Le credenziali incorporate verranno ignorate.

Predefinito: `tcp://localhost:9998`

Plugins.EmberPlus.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60-900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione a Ember+.
 L'unico schema supportato è `tcp://`. Lo schema può essere omesso. Le credenziali incorporate verranno ignorate.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Predefinito: `tcp://localhost:9998`

Plugins.EmberPlus.System.Path

Specifica il percorso dell'eseguibile del plugin Ember+.

Esempio: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

Imposta la durata, in secondi, da attendere affinché un server risponda durante la connessione iniziale e nelle operazioni successive della sessione.

Predefinito: `global timeout`
 Intervallo: 1-30

Opzioni

Parameter	Description
-V --version	Stampa la versione del plugin e le informazioni sulla licenza.
-h --help	Stampa le informazioni di aiuto (scorciatoia).
-t, --test <item key>	Avvia il plugin per il test (la configurazione del plugin viene ignorata).

Vedi anche:

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin GPU NVIDIA

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin NVIDIA GPU.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `nvidiagpu.conf`), anziché nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Il plugin NVIDIA GPU è un plugin caricabile ed è disponibile e descritto in modo completo nel repository del plugin NVIDIA GPU.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.NVIDIA.System.Path	Specificare il percorso dell'eseguibile del plugin NVIDIA GPU.
Plugins.NVIDIA.Timeout	Specificare il tempo massimo (in secondi) da attendere per una risposta del server durante i tentativi di connessione e le operazioni successive nella sessione.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione distribuiti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.NVIDIA.System.Path

Specifica il percorso dell'eseguibile del plugin NVIDIA GPU.

Esempio:

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

Specifica il tempo massimo (in secondi) da attendere per una risposta del server durante i tentativi di connessione e le operazioni successive nella sessione.
 Il timeout globale del tipo item o il timeout del singolo item sostituirà questo valore se superiore.

Predefinito: `global timeout`
 Intervallo: 1–30

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin Memcached

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Memcached.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `memcached.conf`), anziché nel file di configurazione di agent 2. Utilizzare quindi la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Memcached.Default.Password	Specifica la password predefinita per la connessione a Memcached.

Parameter	Description
Plugins.Memcached.Default.Uri	Specifica l'URI predefinito per la connessione a Memcached.
Plugins.Memcached.Default.User	Specifica il nome utente predefinito per la connessione a Memcached.
Plugins.Memcached.KeepAlive	Imposta il tempo massimo di attesa prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.Memcached.Sessions.<SessionName>.Password	Specifica la password della sessione denominata per la connessione a Memcached.
Plugins.Memcached.Sessions.<SessionName>.Uri	Specifica l'URI della sessione denominata per la connessione a Memcached.
Plugins.Memcached.Sessions.<SessionName>.User	Specifica il nome utente della sessione denominata per la connessione a Memcached.
Plugins.Memcached.Timeout	Imposta il timeout di esecuzione della richiesta.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione forniti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.Memcached.Default.Password

Specifica la password predefinita per la connessione a Memcached.
 Utilizzata se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Memcached.Default.Uri

Specifica l'URI predefinito per la connessione a Memcached.
 Usato se non è specificato alcun valore in una chiave item o in una sessione denominata.
 Le credenziali incorporate verranno ignorate.
 Deve corrispondere al formato URI. Schemi supportati: `tcp`, `unix`.
 Uno schema può essere omesso.
 Una porta può essere omessa (default=11211).

Predefinito: `tcp://localhost:11211`

Esempi:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Specificare il nome utente predefinito per la connessione a Memcached.
 Usato se non è specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Memcached.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60-900

Plugins.Memcached.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per la connessione a Memcached.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Memcached.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione a Memcached.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Le credenziali incorporate verranno ignorate.
 Deve corrispondere al formato URI. Schemi supportati: `tcp`, `unix`.
 Uno schema può essere omesso.
 Una porta può essere omessa (predefinita=11211).

Esempi:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per la connessione a Memcached.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Memcached.Timeout

Imposta il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Predefinito: `global timeout`
 Intervallo: 1-30

Vedi anche:

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin Modbus

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Modbus.

Si consiglia di specificarli in un file di configurazione dedicato (ad esempio, `modbus.conf`), anziché nel file di configurazione di agent 2. Utilizzare quindi la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Specifica l'endpoint della sessione denominata per la connessione a Modbus.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Specifica l'ID slave della sessione denominata per la connessione a Modbus.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Imposta il timeout della sessione denominata in secondi.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione distribuiti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Specifica l'endpoint della sessione denominata per la connessione a Modbus.
 La stringa di connessione è composta da uno schema di protocollo, un indirizzo host e una porta oppure il nome di una porta seriale e i relativi attributi.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Specifica l'ID slave della sessione denominata per la connessione a Modbus.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Esempio: `Plugins.Modbus.Sessions.MB1.SlaveID=20`

Nota che questo parametro della sessione denominata viene controllato solo se il valore fornito nel parametro slave ID della chiave dell'item è vuoto.

Plugins.Modbus.Sessions.<SessionName>.Timeout

Imposta il timeout della sessione specificata in secondi.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Esempio: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Se è necessario impostare il timeout di esecuzione della richiesta (quanto tempo attendere il completamento di una richiesta prima di interromperla), utilizzare il modulo di configurazione dell'item.

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin MongoDB

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin MongoDB.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `mongodb.conf`), invece che nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Il plugin MongoDB è un plugin caricabile ed è disponibile e descritto in modo completo nel repository del plugin MongoDB.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Quando si connette a MongoDB, il plugin verifica la propria configurazione in un ordine specifico, che determina quale valore utilizzare. L'ordine, dal più alto al più basso, è il seguente:

1. Parametri specificati nell'URI di connessione (ad esempio, i parametri di query definiti dopo ? in `Plugins.MongoDB.Sessions.<sessi` hanno la priorità più alta).
2. Parametri nella chiave item di Zabbix agent 2 (ad esempio, mongodb.collection.stats).
3. Parametri definiti in una sessione denominata (ad esempio, `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions.<SessionName>.Password`).
4. Valori predefiniti (ad esempio, `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Ad esempio:

- Se si specifica il database per l'autenticazione di una connessione al server MongoDB nel parametro `*.<SessionName>.Uri` (ad esempio, `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSo` il plugin considererà il parametro `*.<SessionName>.AuthSource` impostato su `admin`, anche se il parametro `*.<SessionName>.AuthSource` è impostato su un valore diverso.
- Se si specifica l'URI nel parametro `*.<SessionName>.Uri` (ad esempio, `Plugins.MongoDB.Sessions.<session_name>.Uri=mon` ma si forniscono nome utente e password nella chiave item di Zabbix agent 2, il plugin userà il nome utente e la password dalla chiave item, anche se `Plugins.MongoDB.Sessions.<SessionName>.User` e `Plugins.MongoDB.Sessions.<SessionNam` sono configurati.

Attention:
Se viene usato lo schema `mongodb+srv://`, l'URI può recuperare parametri da un record DNS TXT. I parametri ottenuti da questo record TXT sostituiscono tutte le altre configurazioni, inclusi i parametri URI, i parametri della chiave item di Zabbix Agent 2 e i parametri della sessione.
 Può esistere un solo record TXT per l'hostname usato nella stringa di connessione.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.MongoDB.Default.AuthMechanism	Specify the default authentication mechanism for connecting to MongoDB server.
Plugins.MongoDB.Default.AuthSource	Specify the default database for authenticating connection to MongoDB server.
Plugins.MongoDB.Default.Disa	Disable automatic discovery of MongoDB replica set nodes.
Plugins.MongoDB.Default.Passwo	Specify the default password.
Plugins.MongoDB.Default.Replica	Specify the default replica set name, used for automatic discovery of MongoDB replica set nodes.
Plugins.MongoDB.Default.ReadPre	Specify the default order for connecting to MongoDB replica set nodes or fallback.
Plugins.MongoDB.Default.TLSCon	Specify the level of encryption required for communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSCAF	Specify full pathname of the file containing the top-level CA(s) certificates for peer certificate verification.
Plugins.MongoDB.Default.TLSCer	Specify full pathname of the file containing the agent certificate or certificate chain.
Plugins.MongoDB.Default.TLSKey	Specify full pathname of the file containing the database private key.
Plugins.MongoDB.Default.Uri	Specify the default URI.
Plugins.MongoDB.Default.User	Specify the default username.
Plugins.MongoDB.KeepAlive	Set the maximum time of waiting before unused plugin connections are closed.
Plugins.MongoDB.Sessions.<SessionName>.AuthMech	Specify the named session authentication mechanism for connecting to MongoDB server.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	Specify the named session database for authenticating connection to MongoDB server.
Plugins.MongoDB.Sessions.<SessionName>.Disable	Disable automatic discovery of MongoDB replica set nodes for named session.
Plugins.MongoDB.Sessions.<SessionName>.Passwd	Specify the named session password.

Parameter	Description
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	Specify the name of the replica set name, used for automatic discovery of MongoDB replica set nodes.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	Specify the name of session order for connecting to MongoDB replica set nodes or fallback.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Specify the level of encryption required for communications between Zabbix agent 2 and monitored databases for named session.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Specify full path name of the file containing the top-level CA(s) certificates for peer certificate verification for named session.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Specify full path name of the file containing the agent certificate or certificate chain for named session.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Specify full path name of the file containing the database private key for named session.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Specify the named session URI.
Plugins.MongoDB.Sessions.<SessionName>.User	Specify the named session username.
Plugins.MongoDB.System.Path	Specify path to the plugin executable.
Plugins.MongoDB.Timeout	Set the request execution timeout.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili di ambiente.
- Zabbix supporta file di configurazione solo in codifica UTF-8 senza BOM.
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.MongoDB.Default.AuthMechanism

Specifica il meccanismo di autenticazione predefinito per la connessione al server MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo: `SCRAM-SHA-1`, `SCRAM-SHA-256` o `MONGODB-X509`

Plugins.MongoDB.Default.AuthSource

Specifica il database predefinito per l'autenticazione della connessione al server MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Predefinito: `admin`

Se viene utilizzato il meccanismo di autenticazione `MONGODB-X509`, questo parametro viene impostato automaticamente su `$external`; modificarlo in qualsiasi altro valore causerà un errore all'avvio.

Plugins.MongoDB.Default.Direct

Disabilita il rilevamento automatico dei nodi del replica set di MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Predefinito: `false`
 Intervallo: `true` - si connette solo al server specificato, ignorando gli altri membri del replica set, `false` - rileva automaticamente tutti i nodi nel replica set.

Se viene usato lo schema `tcp://` (deprecato), il parametro viene impostato su `true` per impostazione predefinita.

Plugins.MongoDB.Default.Password

Specificare la password predefinita.
 <SessionName> - definire il nome di una sessione da usare nelle chiavi degli item.

Plugins.MongoDB.Default.ReplicaSet

Specifica il nome predefinito del replica set, utilizzato per il rilevamento automatico dei nodi del replica set MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Se questo parametro viene usato con lo schema `mongodb+srv://`, si comporterà come se fossero stati forniti più URL; un server DNS può avere un record TXT che può sovrascrivere questo parametro (o qualsiasi altro parametro).
 Non disponibile se il parametro `Plugins.MongoDB.Sessions.<SessionName>.Direct` è impostato su `true`; l'agent non riuscirà ad avviarsi.

Plugins.MongoDB.Default.ReadPreference

Specifica l'ordine predefinito per la connessione ai nodi del replica set MongoDB o il fallback.
 <SessionName> - definisce il nome di una sessione da utilizzare nelle chiavi degli item.
 Non disponibile se il parametro `Plugins.MongoDB.Sessions.<SessionNam` è impostato su `true`; l'agent non verrà avviato.

Intervallo: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` o `NearestMode`

Sono preferibili i valori `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` e `nearest`. I valori che terminano con `Mode` sono supportati come alias legacy per la compatibilità.

Plugins.MongoDB.Default.TLSConnect

Specifica il livello di crittografia richiesto per le comunicazioni tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Supportato a partire dalla versione 1.2.1 del plugin.

Intervallo: `required` - richiede una connessione TLS, `verify_ca` - verifica i certificati, `verify_full` - verifica i certificati e l'indirizzo IP

Plugins.MongoDB.Default.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer, utilizzati per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.MongoDB.Default.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile`

Plugins.MongoDB.Default.TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata del database, utilizzata per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile`

Plugins.MongoDB.Default.Uri

Specifica l'URI predefinito.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Deve corrispondere al formato URI.
 Schemi supportati: `mongodb://` e `mongodb+srv://`; lo schema `tcp://` è deprecato e mantenuto per compatibilità con le configurazioni esistenti.
 La porta può essere omessa (default=27017).
Esempi: "

Predefinito: `tcp://127.0.0.1:27017`

Esempi:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

Specificare il nome utente predefinito.
 <SessionName> - definire il nome di una sessione da usare nelle chiavi degli item.

Plugins.MongoDB.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60-900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

Specifica il meccanismo di autenticazione della sessione denominata per la connessione al server MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo: `SCRAM-SHA-1`, `SCRAM-SHA-256` o `MONGODB-X509`

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

Specifica il database della sessione denominata per l'autenticazione della connessione al server MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Predefinito: `admin`

Se viene utilizzato il meccanismo di autenticazione `MONGODB-X509`, questo parametro viene impostato automaticamente su `$external`; modificarlo in qualsiasi altro valore causerà un errore all'avvio.

Plugins.MongoDB.Sessions.<SessionName>.Direct

Disabilita il rilevamento automatico dei nodi del replica set di MongoDB per la sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Predefinito: `false`
 Intervallo: `true` - si connette solo al server specificato, ignorando gli altri membri del replica set, `false` - rileva automaticamente tutti i nodi nel replica set.

Se viene usato lo schema `tcp://` (deprecato), il parametro viene impostato su `true` per impostazione predefinita.

Plugins.MongoDB.Sessions.<SessionName>.Password

Specifica la password della sessione denominata.
 <SessionName> - definisce il nome di una sessione da utilizzare nelle chiavi degli item.

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

Specifica il nome del replica set della sessione denominata, utilizzato per il rilevamento automatico dei nodi del replica set di MongoDB.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Se questo parametro viene usato con lo schema `mongodb+srv://`, si comporterà come se fossero stati forniti più URL; un server DNS può avere un record TXT che può sovrascrivere questo parametro (o qualsiasi altro parametro).
 Non disponibile se il parametro `Plugins.MongoDB.Sessions.<SessionName>.Direct` è impostato su `true`; l'agent non riuscirà ad avviarsi.

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

Specifica l'ordine della sessione denominata per la connessione ai nodi del replica set MongoDB o il fallback.
 <SessionName> - definisce il nome di una sessione da utilizzare nelle chiavi degli item.
 Non disponibile se il parametro `Plugins.MongoDB.Sessions.<SessionName>.Direct` è impostato su `true`; l'agent non verrà avviato.

Intervallo: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` o `NearestMode`

Sono preferibili i valori `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` e `nearest`. I valori che terminano con `Mode` sono supportati come alias legacy per la compatibilità.

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Specifica il livello di crittografia richiesto per le comunicazioni tra Zabbix agent 2 e i database monitorati per la sessione specificata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Supportato a partire dalla versione 1.2.1 del plugin.

Intervallo: `required` - richiede una connessione TLS, `verify_ca` - verifica i certificati, `verify_full` - verifica i certificati e l'indirizzo IP

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati per la sessione specificata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati per la sessione specificata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile`

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata del database, utilizzata per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati per la sessione specificata.
 <SessionName> - definire il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile`

Plugins.MongoDB.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Deve corrispondere al formato URI.
 Schemi supportati: `mongodb://` e `mongodb+srv://`; lo schema `tcp://` è deprecato e mantenuto per compatibilità con le configurazioni esistenti.
 La porta può essere omessa (default=27017).
Esempi: "

Predefinito: `tcp://127.0.0.1:27017`

Esempi:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MongoDB.System.Path

Specifica il percorso dell'eseguibile del plugin MongoDB.

Esempio di utilizzo: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb`

Plugins.MongoDB.Timeout

Imposta il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Predefinito: `global timeout`
 Intervallo: 1-30

Opzioni

Parameter	Description
-V --version	Stampa la versione del plugin e le informazioni sulla licenza.
-h --help	Stampa le informazioni di aiuto (scorciatoia).
-t, --test <item key>	Avvia il plugin per il test (la configurazione del plugin viene ignorata).

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin MQTT

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin MQTT.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `mqtt.conf`), anziché nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.MQTT.Default.Password	Specificare la password predefinita per la connessione a MQTT.
Plugins.MQTT.Default.TLSCAFile	Specificare il percorso completo del file contenente i certificati delle CA di livello superiore per la verifica del certificato del peer.
Plugins.MQTT.Default.TLSCertFile	Specificare il percorso completo del file contenente il certificato di agent o la catena di certificati.
Plugins.MQTT.Default.TLSKeyFile	Specificare il percorso completo del file contenente la chiave privata MQTT.
Plugins.MQTT.Default.Topic	Specificare il topic predefinito per la sottoscrizione MQTT.
Plugins.MQTT.Default.Url	Specificare la stringa di connessione predefinita al broker MQTT.
Plugins.MQTT.Default.User	Specificare il nome utente predefinito per la connessione a MQTT.
Plugins.MQTT.Sessions.<SessionName>.Password	Specificare la password della sessione denominata per la connessione a MQTT.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	Specificare il percorso completo del file contenente i certificati delle CA di livello superiore per la verifica del certificato del peer per la sessione denominata.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	Specificare il percorso completo del file contenente il certificato di agent o la catena di certificati per la sessione denominata.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	Specificare il percorso completo del file contenente la chiave privata MQTT per la sessione denominata.

Parameter	Description
Plugins.MQTT.Sessions.<Session Name>.Topic	Specificare il topic della sessione denominata per la sottoscrizione MQTT.
Plugins.MQTT.Sessions.<Session Name>.Url	Specificare la stringa di connessione al broker MQTT della sessione denominata.
Plugins.MQTT.Sessions.<Session Name>.User	Specificare il nome utente della sessione denominata per la connessione a MQTT.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione distribuiti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con ”#” sono supportati solo all’inizio della riga.

Dettagli del parametro

Plugins.MQTT.Default.Password

Specifica la password predefinita per la connessione a MQTT.
 Utilizzata se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.MQTT.Default.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer nelle comunicazioni crittografate tra Zabbix agent 2 e il broker MQTT.
 Utilizzato se non è specificato alcun valore in una sessione denominata.

Plugins.MQTT.Default.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell’agent o la catena di certificati per le comunicazioni crittografate tra Zabbix agent 2 e il broker MQTT.
 Utilizzato se non è specificato alcun valore in una sessione denominata.

Plugins.MQTT.Default.TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata MQTT per le comunicazioni crittografate tra Zabbix agent 2 e il broker MQTT.
 Utilizzato se non viene specificato alcun valore in una sessione denominata.

Plugins.MQTT.Default.Topic

Specifica il topic predefinito per la sottoscrizione MQTT.
 Utilizzato se non viene specificato alcun valore in una chiave item o in una sessione denominata.
 Il topic può contenere caratteri jolly (”+”,”#”).

Esempi:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

Specifica la stringa di connessione predefinita del broker MQTT.
 Usata se non è specificato alcun valore in una chiave item o in una sessione denominata.
 Non deve includere parametri di query.
 Deve corrispondere al formato URL.
 Schemi supportati: `tcp` (predefinito), `ws`, `tls`
 Uno schema può essere omesso.
 Una porta può essere omessa (predefinita=1883)

Predefinito: `tcp://localhost:1883`

Esempi:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

Specifica il nome utente predefinito per la connessione a MQTT.
 Utilizzato se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.MQTT.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per la connessione a MQTT.
 <SessionName> - definisce il nome di una sessione da utilizzare nelle chiavi degli item.

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer per la sessione specificata, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e il broker MQTT.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati per la sessione specificata, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e il broker MQTT.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata MQTT per la sessione nominata, utilizzata per le comunicazioni crittografate tra Zabbix agent 2 e il broker MQTT.
 <SessionName> - definire il nome di una sessione da usare nelle chiavi degli item.

Plugins.MQTT.Sessions.<SessionName>.Topic

Specifica il topic della sessione denominata per la sottoscrizione MQTT.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Il topic può contenere caratteri jolly ("+","#").

Esempi:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Sessions.<SessionName>.Url

Specifica la stringa di connessione al broker MQTT della sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Non deve includere parametri di query.
 Deve corrispondere al formato URL.
 Schemi supportati: `tcp` (predefinito), `ws`, `tls`
 Uno schema può essere omesso.
 Una porta può essere omessa (predefinita=1883)

Predefinito: `tcp://localhost:1883`

Esempi:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per la connessione a MQTT.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Se è necessario impostare il timeout di esecuzione della richiesta (per quanto tempo attendere il completamento di una richiesta prima di interromperla), utilizzare il modulo configurazione dell'item.

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin MSSQL

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin MSSQL.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `mssql.conf`), anziché nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Il plugin MSSQL è un plugin caricabile ed è disponibile e descritto in modo completo nel repository del plugin MSSQL.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.MSSQL.CustomQueriesDir	Specifica il percorso della directory contenente i file `.sql` definiti dall'utente con query personalizzate che il plugin può eseguire.
Plugins.MSSQL.CustomQueriesEnabled	Abilita l'esecuzione della chiave item `mssql.custom.query`.
Plugins.MSSQL.Default.CACertPath	Specifica il percorso file predefinito del certificato a chiave pubblica dell'autorità di certificazione (CA) che ha emesso il certificato del server MSSQL.
Plugins.MSSQL.Default.Database	Specifica il nome del database predefinito.
Plugins.MSSQL.Default.Encrypt	Specifica il livello di crittografia predefinito.
Plugins.MSSQL.Default.HostnameInCertificate	Specifica per impostazione predefinita il nome comune (CN) del certificato del server MSSQL.
Plugins.MSSQL.Default.Password	Specifica la password predefinita.
Plugins.MSSQL.Default.TLSMinVersion	Specifica per impostazione predefinita la versione minima di TLS.
Plugins.MSSQL.Default.TrustServerCertificate	Considera attendibile il certificato del server senza verificarlo, per impostazione predefinita.
Plugins.MSSQL.Default.Uri	Specifica l'URI predefinito.
Plugins.MSSQL.Default.User	Specifica il nome utente predefinito.
Plugins.MSSQL.KeepAlive	Specifica il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath	Specifica il percorso file della sessione denominata del certificato a chiave pubblica dell'autorità di certificazione (CA) che ha emesso il certificato del server MSSQL.
Plugins.MSSQL.Sessions.<SessionName>.Database	Specifica il nome del database della sessione denominata.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Specifica il livello di crittografia della sessione denominata.
Plugins.MSSQL.Sessions.<SessionName>.HostnameInCertificate	Specifica il nome comune (CN) del certificato del server MSSQL per la sessione denominata.
Plugins.MSSQL.Sessions.<SessionName>.Password	Specifica la password della sessione denominata.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Specifica la versione minima di TLS della sessione denominata.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Considera attendibile il certificato del server della sessione denominata senza verificarlo.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Specifica l'URI della sessione denominata.
Plugins.MSSQL.Sessions.<SessionName>.User	Specifica il nome utente della sessione denominata.
Plugins.MSSQL.System.Path	Specifica il percorso dell'eseguibile del plugin MSSQL.
Plugins.MSSQL.Timeout	Imposta la durata, in secondi, dell'attesa di una risposta da un server alla prima connessione e nelle operazioni successive nella sessione.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione distribuiti.
- I valori supportano variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM.
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.MSSQL.CustomQueriesDir

Specifica il percorso della directory contenente i file `.sql` definiti dall'utente con query personalizzate che il plugin può eseguire.
 Il plugin carica tutti i file .sql disponibili nella directory configurata all'avvio. Ciò significa che eventuali modifiche ai file di query personalizzate non verranno applicate finché il plugin non viene riavviato. Il plugin viene avviato e arrestato insieme a Zabbix agent 2.

Valori predefiniti:
`/usr/local/share/zabbix/custom-queries/mssql` - per sistemi Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` - per sistemi Windows (dove `*` è il nome dell'unità dalla variabile di ambiente `ProgramFiles`)

Plugins.MSSQL.CustomQueriesEnabled

Abilita l'esecuzione della chiave item `mssql.custom.query`.
 Se disabilitato, non verranno caricate query dal percorso della directory delle query personalizzate.

Predefinito: `false`
 Intervallo: `true/false`

Plugins.MSSQL.Default.CACertPath

Specifica il percorso file predefinito del certificato della chiave pubblica dell'autorità di certificazione (CA) che ha emesso il certificato del server MSSQL.
 Il certificato deve essere in formato PEM.

Plugins.MSSQL.Default.Database

Specifica il nome del database predefinito.

Plugins.MSSQL.Default.Encrypt

Specifica il livello di crittografia predefinito.

Intervallo:
true - i dati inviati tra il plugin e il server sono crittografati;
false - i dati inviati tra il plugin e il server non sono crittografati oltre il pacchetto di accesso;
strict - i dati inviati tra il plugin e il server sono crittografati end-to-end usando TDS8;
disable - i dati inviati tra il plugin e il server non sono crittografati.

Plugins.MSSQL.Default.HostNameInCertificate

Specifica per impostazione predefinita il nome comune (CN) del certificato del server MSSQL.

Plugins.MSSQL.Default.Password

Specificare la password predefinita per un server MSSQL protetto.

Plugins.MSSQL.Default.TLSMinVersion

Specifica la versione minima di TLS predefinita.

Intervallo: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Default.TrustServerCertificate

Considera attendibile il certificato del server senza convalidarlo per impostazione predefinita.

Intervallo: `true`, `false`

Plugins.MSSQL.Default.Uri

Specifica l'URI predefinito per la connessione al server MSSQL.
 Le credenziali incorporate verranno ignorate.
 L'unico schema supportato è `sqlserver://`.
 Lo schema può essere omesso.

Predefinito: `sqlserver://localhost:1433`

Plugins.MSSQL.Default.User

Specifica il nome utente predefinito per un server MSSQL protetto.

Plugins.MSSQL.KeepAlive

Specifica il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60–900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

Specifica il percorso del file della sessione denominata verso il certificato della chiave pubblica dell'autorità di certificazione (CA) che ha emesso il certificato del server MSSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Il certificato deve essere in formato PEM.

Plugins.MSSQL.Sessions.<SessionName>.Database

Specifica il nome del database della sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

Specifica il livello di crittografia della sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo:
true - i dati inviati tra il plugin e il server sono crittografati;
false - i dati inviati tra il plugin e il server non sono crittografati oltre il pacchetto di accesso;
strict - i dati inviati tra il plugin e il server sono crittografati end-to-end usando TDS8;
disable - i dati inviati tra il plugin e il server non sono crittografati.

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

Specifica il nome comune (CN) del certificato del server MSSQL per la sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MSSQL.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per un server MSSQL protetto.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

Specifica la versione minima TLS della sessione denominata.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

Fidati del certificato del server della sessione specificata senza convalidarlo.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo: `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione al server MSSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Le credenziali incorporate verranno ignorate.
 L'unico schema supportato è `sqlserver://`.
 Lo schema può essere omesso.

Predefinito: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per un server MSSQL protetto.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.MSSQL.System.Path

Specifica il percorso dell'eseguibile del plugin MSSQL.
 Impostazione globale per il plugin MSSQL. Applicata a tutte le connessioni.

Esempio: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

Imposta la durata, in secondi, da attendere affinché un server risponda durante la prima connessione e nelle operazioni successive della sessione.

Predefinito: `global timeout`
 Intervallo: 1-30

Opzioni

Parameter	Description
-V --version	Stampa la versione del plugin e le informazioni sulla licenza.
-h --help	Stampa le informazioni di aiuto (scorciatoia).
-t, --test <item key>	Avvia il plugin per il test (la configurazione del plugin viene ignorata).

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin MySQL

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin MySQL.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `mysql.conf`), anziché nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Mysql.CallTimeout	Specifica il tempo massimo, in secondi, da attendere affinché una richiesta venga completata.
Plugins.Mysql.CustomQueriesPath	Specifica il percorso completo della directory contenente i file `.sql` con query personalizzate.
Plugins.Mysql.CustomQueriesEnabled	Abilita l'esecuzione della chiave item `mysql.custom.query`.
Plugins.Mysql.Default.Password	Specifica la password predefinita.
Plugins.Mysql.Default.TLSConnect	Specifica il livello predefinito di crittografia per le comunicazioni tra Zabbix agent 2 e i database monitorati.
Plugins.Mysql.Default.TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer.

Parameter	Description
Plugins.Mysql.Default.TLSCertFile	Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati.
Plugins.Mysql.Default.TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata del database.
Plugins.Mysql.Default.Uri	Specifica l'URI predefinito.
Plugins.Mysql.Default.User	Specifica il nome utente predefinito.
Plugins.Mysql.KeepAlive	Specifica il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.Mysql.Sessions.<SessionName>.Password	Specifica la password della sessione denominata.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	Specifica il livello di crittografia della sessione denominata per le comunicazioni tra Zabbix agent 2 e i database monitorati.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer della sessione denominata.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati della sessione denominata.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata del database della sessione denominata.
Plugins.Mysql.Sessions.<SessionName>.Uri	Specifica l'URI della sessione denominata.
Plugins.Mysql.Sessions.<SessionName>.User	Specifica il nome utente della sessione denominata.
Plugins.Mysql.Timeout	Imposta il tempo massimo, in secondi, di attesa quando è necessario stabilire una connessione.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti;
- I valori supportano le variabili d'ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.Mysql.CallTimeout

Specifica il tempo massimo, in secondi, da attendere per il completamento di una richiesta.

Predefinito: `global timeout`
 Intervallo: 1-30

Plugins.Mysql.CustomQueriesPath

Specifica il percorso completo della directory contenente i file `.sql` con query personalizzate.

Valori predefiniti:
`/usr/local/share/zabbix/custom-queries/mysql` - per sistemi Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` - per sistemi Windows (dove `*` è il nome dell'unità della variabile di ambiente `ProgramFiles`)

Plugins.Mysql.CustomQueriesEnabled

Abilita l'esecuzione della chiave item `mysql.custom.query`.
 Se disabilitato, nessuna query verrà caricata dal percorso della directory delle query personalizzate.

Predefinito: `false`
 Intervallo: `true/false`

Plugins.Mysql.Default.Password

Specificare la password predefinita per un server MySQL protetto.

Plugins.Mysql.Default.TLSConnect

Specifica il livello predefinito di crittografia per le comunicazioni tra Zabbix agent 2 e i database monitorati.

Intervallo:
required - richiedi una connessione TLS;
verify_ca - verifica i certificati;
verify_full - verifica i certificati e l'indirizzo IP

Plugins.Mysql.Default.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.

Obbligatorio: `yes`, se `Plugins.Mysql.Default.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.Mysql.Default.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.

Obbligatorio: `yes`, se è specificato `Plugins.Mysql.Default.TLSKeyFile`

Plugins.Mysql.Default.TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata del database, utilizzata per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.

Obbligatorio: `yes`, se è specificato `Plugins.Mysql.Default.TLSCertFile`

Plugins.Mysql.Default.Uri

Specifica l'URI predefinito per la connessione al server MySQL.
 Le credenziali incorporate verranno ignorate.
 Schemi supportati: `tcp`, `unix`
 Lo schema può essere omesso.
 La porta può essere omessa (default=3306).

Esempi:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Default.User

Specifica il nome utente predefinito per un server MySQL protetto.

Plugins.Mysql.KeepAlive

Specifica il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60–900

Plugins.Mysql.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per un server MySQL protetto.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Specifica il livello di crittografia della sessione denominata per le comunicazioni tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo:
required - richiede una connessione TLS;
verify_ca - verifica i certificati;
verify_full - verifica i certificati e l'indirizzo IP

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la verifica del certificato del peer della sessione denominata, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent o la catena di certificati della sessione denominata, utilizzato per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile`

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata del database della sessione denominata, utilizzata per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile`

Plugins.Mysql.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione al server MySQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Le credenziali incorporate verranno ignorate.
 Schemi supportati: `tcp`, `unix`
 Lo schema può essere omesso.
 La porta può essere omessa (default=3306).

Esempi:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per un server MySQL protetto.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Mysql.Timeout

Imposta il tempo massimo, in secondi, di attesa quando è necessario stabilire una connessione.

Predefinito: `global timeout`
 Intervallo: 1-30

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin Oracle

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Oracle.

Si consiglia di specificarli in un file di configurazione dedicato (ad esempio, `oracle.conf`), anziché nel file di configurazione di agent 2. Quindi usare la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **opzionali**, a meno che non sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Oracle.CallTimeout	Il tempo massimo di attesa, in secondi, per il completamento di una richiesta. Questo parametro al momento non ha effetto; viene invece utilizzata l'impostazione item timeout.
Plugins.Oracle.ConnectTimeout	Specifica il tempo massimo di attesa, in secondi, per stabilire una connessione.
Plugins.Oracle.CustomQueriesPath	Specifica il percorso completo della directory contenente i file `.sql` con query personalizzate.
Plugins.Oracle.CustomQueriesEnabled	Abilita l'esecuzione della chiave item `oracle.custom.query`.
Plugins.Oracle.Default.Password	Specifica la password predefinita.
Plugins.Oracle.Default.Service	Specifica il nome del servizio predefinito.
Plugins.Oracle.Default.URI	Specifica l'URI predefinito.
Plugins.Oracle.Default.User	Specifica il nome utente predefinito.
Plugins.Oracle.KeepAlive	Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.Oracle.ResolveTNS	Specifica come interpretare la stringa di connessione (ConnString) per connettersi al server Oracle.
Plugins.Oracle.Sessions.<SessionName>.Password	Specifica la password della sessione denominata.
Plugins.Oracle.Sessions.<SessionName>.Service	Specifica il nome del servizio della sessione denominata.
Plugins.Oracle.Sessions.<SessionName>.URI	Specifica l'URI della sessione denominata.
Plugins.Oracle.Sessions.<SessionName>.User	Specifica il nome utente della sessione denominata.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione distribuiti.
- I valori supportano le variabili di ambiente.
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM.
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli dei parametri

Plugins.Oracle.CallTimeout

Il tempo massimo di attesa, in secondi, affinché una richiesta venga completata.
 Questo parametro al momento non ha alcun effetto. Il tempo massimo di attesa è determinato invece dal timeout dell'item.

Predefinito: `Timeout`[1]
 Intervallo: 1-30

Plugins.Oracle.ConnectTimeout

Specifica il tempo massimo di attesa, in secondi, per stabilire una connessione.

Predefinito: `Timeout`[1]
 Intervallo: 1-30

Plugins.Oracle.CustomQueriesPath

Specifica il percorso completo della directory contenente i file `.sql` con query personalizzate.

Valori predefiniti:
`/usr/local/share/zabbix/custom-queries/oracle` - per sistemi Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Oracle` - per sistemi Windows (dove `*` è il nome dell'unità ricavato dalla variabile di ambiente `ProgramFiles`)

Esempio:

```
/etc/zabbix/oracle/sql
```

Plugin.Oracle.CustomQueriesEnabled

Abilita l'esecuzione della chiave item `oracle.custom.query`.
 Se disabilitato, non verranno caricate query dal percorso della directory delle query personalizzate.

Predefinito: `false`
 Intervallo: `true/false`

Plugins.Oracle.Default.Password

Specifica la password predefinita per la connessione a Oracle.
 Utilizzata se non è specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Oracle.Default.Service

Specifica il nome del servizio predefinito per la connessione a Oracle (SID non è supportato).
 Utilizzato se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Oracle.Default.Uri

Specifica l'URI predefinito per la connessione a Oracle.
 Usato se non è specificato alcun valore in una chiave item o in una sessione denominata.
 Le credenziali incorporate verranno ignorate.
 Deve corrispondere al formato URI.
 Schemi supportati: `tcp`, `tcps`
 Lo schema può essere omesso.
 La porta può essere omessa (default=1521).
 È anche possibile specificare la chiave TNS o il valore TNS come stringa di connessione. Il valore TNS deve essere composto senza spazi.

Esempi:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (chiave TNS)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (valore
```

Plugin.Oracle.Default.User

Specifica il nome utente predefinito per la connessione a Oracle.
 Utilizzato se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Oracle.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin inutilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60-900

Plugins.Oracle.ResolveTNS

Specifica come interpretare la stringa di connessione (ConnString) per connettersi al server Oracle.
 Se è impostato su `false`, lo schema e la porta predefiniti verranno aggiunti a ConnString (se assenti).
 Se l'opzione è impostata su `true`, lo schema e la porta predefiniti verranno omessi (a meno che non siano specificati esplicitamente in ConnString) e ConnString verrà passato al client Oracle così com'è. Se il client Oracle trova ConnString nel file `tnsnames.ora`, verrà utilizzata la descrizione di connessione trovata per connettersi al server Oracle.

Predefinito: `true`
 Intervallo: `true/false`

Plugins.Oracle.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per la connessione a Oracle.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.Oracle.Sessions.<SessionName>.Service

Specificare il nome del servizio della sessione denominata per la connessione a Oracle (SID non è supportato).
 <SessionName> - definire il nome di una sessione da usare nelle chiavi degli item.

Plugins.Oracle.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione a Oracle.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Le credenziali incorporate verranno ignorate.
 Deve corrispondere al formato URI.
 Schemi supportati: `tcp`, `tcps`
 Lo schema può essere omesso.
 La porta può essere omessa (default=1521).
 È inoltre possibile specificare la chiave TNS o il valore TNS come stringa di connessione. Il valore TNS deve essere composto senza spazi.

Esempi:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per la connessione a Oracle.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Note a piè di pagina

[1] Il timeout è definito dall'opzione `Timeout` nel file di configurazione (`zabbix_agent2.conf`) di Zabbix agent 2 (UNIX) o Zabbix agent 2 (Windows).

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin PostgreSQL

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin PostgreSQL.

Si consiglia di specificarli in un file di configurazione separato (ad esempio, `postgresql.conf`), anziché nel file di configurazione di agent 2. Utilizzare quindi la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Il plugin PostgreSQL è un plugin caricabile ed è disponibile e descritto in modo completo nel repository del plugin PostgreSQL.

Tutti i parametri sono **opzionali**, salvo quando è esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.PostgreSQL.CallTimeout	Imposta il tempo massimo di attesa, in secondi, per il completamento di una richiesta.
Plugins.PostgreSQL.CustomQueriesPath	Specifica il percorso completo della directory contenente i file `.sql` con query personalizzate.
Plugins.PostgreSQL.CustomQueriesEnabled	Abilita l'esecuzione della chiave item `postgresql.custom.query`.
Plugins.PostgreSQL.Default.CacheMode	Specifica la modalità di cache predefinita per la connessione PostgreSQL.
Plugins.PostgreSQL.Default.Database	Specifica il nome del database predefinito.
Plugins.PostgreSQL.Default.Password	Specifica la password predefinita.
Plugins.PostgreSQL.Default.TLSCAFile	Specifica il percorso completo del file contenente il certificato della/e CA di livello superiore per la verifica del certificato peer.
Plugins.PostgreSQL.Default.TLSCertFile	Specifica il percorso completo del file contenente il certificato PostgreSQL o la catena di certificati.
Plugins.PostgreSQL.Default.TLSConnect	Specifica il livello predefinito di crittografia per le comunicazioni tra Zabbix agent 2 e i database monitorati.
Plugins.PostgreSQL.Default.TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata PostgreSQL.
Plugins.PostgreSQL.Default.Uri	Specifica l'URI predefinito.
Plugins.PostgreSQL.Default.User	Specifica il nome utente predefinito.
Plugins.PostgreSQL.KeepAlive	Imposta il tempo massimo di attesa, in secondi, prima che le connessioni del plugin non utilizzate vengano chiuse.
Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode	Specifica la modalità di cache della sessione denominata per la connessione PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Specifica il nome del database della sessione denominata.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Specifica la password della sessione denominata.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Specifica il percorso completo del file contenente il certificato della/e CA di livello superiore per la verifica del certificato peer.

Parameter	Description
Plugins.PostgreSQL.Sessions.<Session>.TLSCertFile	Specifica il percorso completo del file contenente il certificato PostgreSQL o la catena di certificati.
Plugins.PostgreSQL.Sessions.<Session>.TLSConnect	Specifica il livello di crittografia della sessione denominata per le comunicazioni tra Zabbix agent 2 e i database monitorati.
Plugins.PostgreSQL.Sessions.<Session>.TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata PostgreSQL.
Plugins.PostgreSQL.Sessions.<Session>.Uri	Specifica l'URI della sessione denominata.
Plugins.PostgreSQL.Sessions.<Session>.User	Specifica il nome utente della sessione denominata.

|Plugins.PostgreSQL.System.Path|Specifica il percorso dell'eseguibile del plugin PostgreSQL.| |Plugins.PostgreSQL.Timeout|Imposta il timeout di esecuzione della richiesta.|

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione distribuiti.
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM.
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.PostgreSQL.CallTimeout

Imposta il tempo massimo di attesa, in secondi, affinché una richiesta venga completata.

Predefinito: `global timeout`
 Intervallo: 1-30

Plugins.PostgreSQL.CustomQueriesPath

Specifica il percorso completo della directory contenente i file `.sql` con query personalizzate.

Valori predefiniti:
`/usr/local/share/zabbix/custom-queries/postgreSQL` - per i sistemi Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` - per i sistemi Windows (dove `*` è il nome dell'unità dalla variabile di ambiente `ProgramFiles`)

Esempio:

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

Abilita l'esecuzione della chiave item `postgresql.custom.query`.
 Se disabilitato, non verrà caricato alcun query dal percorso della directory delle query personalizzate.

Predefinito: `false`
 Intervallo: `true/false`

Plugins.PostgreSQL.Default.CacheMode

Specifica la modalità di cache predefinita per la connessione PostgreSQL.

Predefinito: `prepare`
 Intervallo:
`prepare` - crea statement preparati sul server PostgreSQL;
`describe` - usa lo statement preparato anonimo per descrivere uno statement senza creare uno statement sul server.

Nota che `describe` è utile principalmente quando l'ambiente non consente statement preparati, ad esempio quando si utilizza un pooler di connessioni come PgBouncer.

Plugins.PostgreSQL.Default.Database

Specifica il database predefinito per la connessione a PostgreSQL.
 Usato se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.PostgreSQL.Default.Password

Specifica la password predefinita per la connessione a PostgreSQL.
 Utilizzata se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.PostgreSQL.Default.TLSCAFile

Specifica il percorso completo del file contenente il certificato CA di livello superiore per la verifica del certificato del peer nelle comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 Usato se non è specificato alcun valore in una sessione denominata.

Obbligatorio: `yes`, se `Plugins.PostgreSQL.Default.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.PostgreSQL.Default.TLSCertFile

Specifica il percorso completo del file contenente il certificato PostgreSQL o la catena di certificati per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 Usato se non è specificato alcun valore in una sessione denominata.

Obbligatorio: `yes`, se è specificato `Plugins.PostgreSQL.Default.TLSKeyFile`

Plugins.PostgreSQL.Default.TLSConnect

Specifica il livello predefinito di crittografia per le comunicazioni tra Zabbix agent 2 e i database monitorati.
 Usato se non viene specificato alcun valore in una sessione denominata.

Intervallo:
`required` - connessione tramite TLS come modalità di trasporto senza controlli di identità;
`verify_ca` - connessione tramite TLS e verifica del certificato;
`verify_full` - connessione tramite TLS, verifica del certificato e verifica che l'identità del database (CN) specificata da DBHost corrisponda al relativo certificato.

Un tipo di crittografia non definito indica una connessione non crittografata.

Plugins.PostgreSQL.Default.TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata PostgreSQL per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 Usato se non è specificato alcun valore in una sessione denominata.

Obbligatorio: `yes`, se `Plugins.PostgreSQL.Default.TLSCertFile` è specificato

Plugins.PostgreSQL.Default.Uri

Specifica l'URI predefinito per la connessione a PostgreSQL.
 Usato se non è specificato alcun valore in una chiave item o in una sessione denominata.
 Le credenziali incorporate verranno ignorate.
 Deve corrispondere al formato URI.
 Schemi supportati: `tcp`, `unix`

Esempi:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Default.User

Specifica il nome utente predefinito per la connessione a PostgreSQL.
 Usato se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.PostgreSQL.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

Specifica la modalità di cache della sessione con nome per la connessione PostgreSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Predefinito: `prepare`
 Intervallo:
`prepare` - crea statement preparati sul server PostgreSQL;
`describe` - usa lo statement preparato anonimo per descrivere uno statement senza crearne uno sul server.

Nota che `describe` è utile principalmente quando l'ambiente non consente statement preparati, ad esempio quando si esegue un pooler di connessioni come PgBouncer.

Plugins.PostgreSQL.Sessions.<SessionName>.Database

Specifica il database della sessione denominata per la connessione a PostgreSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.PostgreSQL.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per la connessione a PostgreSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Deve corrispondere al formato della password.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Specifica il percorso completo del file contenente il certificato della CA di livello superiore per la verifica del certificato del peer nelle comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se `Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Specifica il percorso completo del file contenente il certificato PostgreSQL o la catena di certificati per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se `Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile` è specificato

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Specifica il livello di crittografia della sessione denominata per le comunicazioni tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Intervallo:
`required` - connessione tramite TLS come modalità di trasporto senza controlli di identità;
`verify_ca` - connessione tramite TLS e verifica del certificato;
`verify_full` - connessione tramite TLS, verifica del certificato e verifica che l'identità del database (CN) specificata da DBHost corrisponda al relativo certificato.

Un tipo di crittografia non definito indica una connessione non crittografata.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata PostgreSQL per le comunicazioni crittografate tra Zabbix agent 2 e i database monitorati.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Obbligatorio: `yes`, se è specificato `Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile`.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione a PostgreSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.
 Le credenziali incorporate verranno ignorate.
 Deve corrispondere al formato URI.
 Schemi supportati: `tcp`, `unix`

Esempi:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per la connessione a PostgreSQL.
 <SessionName> - definisce il nome di una sessione da usare nelle chiavi degli item.

Plugins.PostgreSQL.System.Path

Specificare il percorso dell'eseguibile del plugin PostgreSQL.

Obbligatorio: `yes`

Esempio:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

Imposta il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Predefinito: `global timeout`
 Intervallo: 1-30

Opzioni

Parameter	Description
-V --version	Stampa la versione del plugin e le informazioni sulla licenza.
-h --help	Stampa le informazioni di aiuto (scorciatoia).
-t, --test <item key>	Avvia il plugin per il test (la configurazione del plugin viene ignorata).

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin Redis

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin Redis.

Si consiglia di specificarli in un file di configurazione dedicato (ad esempio, `redis.conf`), anziché nel file di configurazione di agent 2. Utilizzare quindi la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **facoltativi**, salvo quando è esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Redis.Default.Password	Specifica la password predefinita per la connessione a Redis.
Plugins.Redis.Default.TLSConnect	Specifica il livello di crittografia richiesto per le comunicazioni di Zabbix agent 2.
Plugins.Redis.Default.TLSCAFile	Specifica il percorso completo del file contenente i certificati della CA di livello superiore.
Plugins.Redis.Default.TLSCertFile	Specifica il percorso completo del file contenente il certificato di agent.
Plugins.Redis.Default.TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata di agent.
Plugins.Redis.Default.Uri	Specifica l'URI predefinito per la connessione a Redis.
Plugins.Redis.Default.User	Specifica il nome utente predefinito per la connessione a Redis.
Plugins.Redis.KeepAlive	Imposta il tempo massimo di attesa (in secondi) prima della chiusura delle connessioni del plugin non utilizzate.
Plugins.Redis.Sessions.<Session name>.Password	Specifica la password della sessione denominata per la connessione a Redis.
Plugins.Redis.Sessions.<Session name>.TLSConnect	Specifica il livello di crittografia richiesto per le comunicazioni di Zabbix agent 2 per la sessione denominata.
Plugins.Redis.Sessions.<Session name>.TLSCAFile	Specifica il percorso completo del file contenente i certificati della CA di livello superiore per la sessione denominata.
Plugins.Redis.Sessions.<Session name>.TLSCertFile	Specifica il percorso completo del file contenente il certificato di agent per la sessione denominata.
Plugins.Redis.Sessions.<Session name>.TLSKeyFile	Specifica il percorso completo del file contenente la chiave privata di agent per la sessione denominata.
Plugins.Redis.Sessions.<Session name>.Uri	Specifica l'URI della sessione denominata per la connessione a Redis.
Plugins.Redis.Sessions.<Session name>.User	Specifica il nome utente della sessione denominata per la connessione a Redis.
Plugins.Redis.Timeout	Imposta il timeout di esecuzione della richiesta.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione forniti.
- I valori supportano le variabili di ambiente.
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM.
- I commenti che iniziano con "#" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.Redis.Default.Password

Specifica la password predefinita per la connessione a Redis.
 Usata se non è specificato alcun valore in una chiave item o in una sessione denominata.

Plugins.Redis.Default.TLSConnect

Specifica il livello di crittografia richiesto per le comunicazioni di Zabbix agent 2.
 Un valore vuoto o non impostato viene considerato come assenza di TLS.

Intervallo: `required` - crittografato, non verificato (solo per test), `verify_ca` - crittografato, certificato del server, `verify_full` - crittografato, certificato del server e nome del server verificati con SAN

Plugins.Redis.Default.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore.

Plugins.Redis.Default.TLSCertFile

Specificare il percorso completo del file contenente il certificato dell'agent.

Plugins.Redis.Default.TLSKeyFile

Specifica il percorso completo del file contenente la chiave privata dell'agent.

Plugins.Redis.Default.Uri

Specifica l'URI predefinito per la connessione a Redis.
 Usato se non è specificato alcun valore in una item key o in una sessione denominata.
 Le credenziali incorporate verranno ignorate. Deve corrispondere al formato URI.
 Schemi supportati: `tcp`, `unix`
 Uno schema può essere omesso.
 Una porta può essere omessa (default=6379).

Predefinito: `tcp://localhost:6379`

Esempi:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Specifica il nome utente predefinito per la connessione a Redis.
 Utilizzato se non viene specificato alcun valore in una chiave item o in una sessione denominata.

Predefinito: `default`

Plugins.Redis.KeepAlive

Imposta il tempo massimo di attesa (in secondi) prima che le connessioni del plugin non utilizzate vengano chiuse.

Predefinito: 300
 Intervallo: 60–900

Plugins.Redis.Sessions.<SessionName>.Password

Specifica la password della sessione denominata per la connessione a Redis.
 <SessionName> - definisce il nome della sessione da utilizzare nelle chiavi degli item.

Plugins.Redis.Sessions.<SessionName>.TLSConnect

Specifica il livello di crittografia richiesto per le comunicazioni di Zabbix agent 2 per la sessione denominata.
 <SessionName> - definisce il nome della sessione da usare nelle chiavi degli item.
 Un valore vuoto o non impostato viene considerato come assenza di TLS.

Intervallo: `required` - crittografato, non verificato (solo per test), `verify_ca` - crittografato, certificato del server, `verify_full` - crittografato, certificato del server e nome del server verificato con SAN

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

Specifica il percorso completo del file contenente i certificati CA di livello superiore per la sessione denominata.
 <SessionName> - definisce il nome della sessione da utilizzare nelle chiavi degli item.

Obbligatorio: `yes`, se `Plugins.Redis.Sessions.<SessionName>.TLSConnect` è impostato su `verify_ca` o `verify_full`

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

Specifica il percorso completo del file contenente il certificato dell'agent per la sessione denominata.
 <SessionName> - definisce il nome della sessione da usare nelle chiavi degli item.

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

Specificare il percorso completo del file contenente la chiave privata dell'agent per la sessione denominata.
 <SessionName> - definisce il nome della sessione da usare nelle chiavi degli item.

Plugins.Redis.Sessions.<SessionName>.Uri

Specifica l'URI della sessione denominata per la connessione a Redis.
 <SessionName> - definisce il nome della sessione da usare nelle chiavi degli item.
 Le credenziali incorporate verranno ignorate. Deve corrispondere al formato URI.
 Schemi supportati: `tcp`, `unix`
 Uno schema può essere omesso.
 Una porta può essere omessa (default=6379).

Predefinito: `tcp://localhost:6379`

Esempi:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Specifica il nome utente della sessione denominata per la connessione a Redis.
 <SessionName> - definisce il nome della sessione da usare nelle chiavi dell'item.

Plugins.Redis.Timeout

Imposta il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Predefinito: `global timeout<br>` Intervallo: 1–30

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Plugin SMART

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento del plugin SMART.

Si consiglia di specificarli in un file di configurazione dedicato (ad esempio, `smart.conf`), anziché nel file di configurazione di agent 2. Utilizzare quindi la direttiva Include per includere questo file nella configurazione di Zabbix agent 2.

Tutti i parametri sono **facoltativi**, salvo che sia esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per vedere i dettagli completi di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parameter	Description
Plugins.Smart.Path	Specifica il percorso dell'eseguibile smartctl.
Plugins.Smart.Timeout	Imposta il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti;
- I valori supportano variabili di ambiente;
- Il percorso dell'eseguibile `smartctl` deve essere impostato aggiungendo la sua directory alla variabile di ambiente `PATH` del sistema oppure configurando `Plugins.Smart.Path`; ciò vale sia per Linux sia per Windows;
- Zabbix supporta file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con "`#`" sono supportati solo all'inizio della riga.

Dettagli del parametro

Plugins.Smart.Path

Specifica il percorso dell'eseguibile smartctl.

Predefinito: `smartctl`

Plugins.Smart.Timeout

Imposta il timeout di esecuzione della richiesta (la durata, in secondi, da attendere affinché una richiesta venga completata prima di interromperla).

Predefinito: `global timeout<br>` Intervallo: 1–30

Vedi anche

- Descrizione dei parametri di configurazione generali di Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Istruzioni per la configurazione dei plugin

Protocollo del plugin di Zabbix agent 2

Il protocollo di Zabbix agent 2 si basa su codice, dimensione e modello dei dati.

Codice

Type	Size	Comments
Byte	4	Tipo di payload, attualmente è supportato solo JSON.

Dimensione

Type	Size	Comments
Byte	4	Dimensione del payload corrente in byte.

Dati del payload

Type	Size	Comments
Byte	Definito dal campo *Size*	Dati in formato JSON.

Definizione dei dati del payload

Dati comuni

Questi parametri sono presenti in tutte le richieste/risposte:

Name	Type	Comments
id	uint32	Per le richieste - l'identificatore incrementale usato per collegare le richieste alle risposte. Unico all'interno di una direzione di richiesta (cioè da agent al plugin o dal plugin all'agent). Per le risposte - l'ID della richiesta corrispondente.
type	uint32	Il tipo di richiesta.

Richiesta di log

Una richiesta inviata da un plugin per scrivere un messaggio di log nel file di log dell'agent.

direction	plugin → agent
response	no

Parametri specifici delle richieste di log:

Name	Type	Comments
severity	uint32	La gravità del messaggio (livello di log).
message	string	Il messaggio da registrare.

Esempio:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Richiesta di registrazione

Una richiesta inviata dall'agent durante la fase di avvio dell'agent per ottenere le metriche fornite e registrare un plugin.

direction	agent → plugin
response	yes

Parametri specifici delle richieste di registrazione:

Name	Type	Comments
version	string	La versione del protocollo <major>.<minor>

Esempio:

```
{"id":1,"type":2,"version":"1.0"}
```

Risposta di registrazione

Risposta del plugin alla richiesta di registrazione.

direction	plugin → agent
response	n/a

Parametri specifici delle risposte di registrazione:

Name	Type	Comments
name	string	Il nome del plugin.
metrics	array of strings (optional)	Le metriche con descrizioni, come usate nel plugin. Restituisce RegisterMetrics(). Assente se viene restituito un errore.
interfaces	uint32 (optional)	La maschera di bit delle interfacce supportate dal plugin. Assente se viene restituito un errore.
error	string (optional)	Un messaggio di errore restituito se un plugin non può essere avviato. Assente se vengono restituite le metriche.

Esempi:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

oppure

```
{"id":2,"type":3,"error":"error message"}
```

Richiesta di avvio

Una richiesta per eseguire la funzione Start dell'interfaccia Runner.

direction	agent → plugin
response	no

La richiesta non ha parametri specifici, contiene solo i parametri dei dati comuni.

Esempio:

```
{"id":3,"type":4}
```

Richiesta di terminazione

Una richiesta inviata dall'agent per arrestare un plugin.

direction	agent → plugin
response	no

La richiesta non ha parametri specifici, contiene solo i parametri dei dati comuni.

Esempio:

```
{"id":3,"type":5}
```

Richiesta di esportazione

Una richiesta per eseguire la funzione Export dell'interfaccia Exporter.

direction	agent → plugin
response	no

Parametri specifici per le richieste di esportazione:

Name	Type	Comments
key	string	La chiave del plugin.
parameters	array of strings (optional)	I parametri per la funzione Export.

Esempio:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Risposta di esportazione

Risposta della funzione Export dell'interfaccia Exporter.

direction	plugin → agent
response	n/a

Parametri specifici delle risposte di esportazione:

Name	Type	Comments
value	string (optional)	Valore di risposta della funzione Export. Assente se viene restituito un errore.
error	string (optional)	Messaggio di errore se la funzione Export non è stata eseguita correttamente. Assente se viene restituito value.

Esempi:

```
{"id":5,"type":7,"value":"response"}
```

oppure

```
{"id":5,"type":7,"error":"error message"}
```

Configurazione richiesta

Una richiesta per eseguire la funzione *Configure* dell'interfaccia *Configurator*.

direction	agent → plugin
response	n/a

Parametri specifici per le richieste *Configure*:

Name	Type	Comments
global_options	JSON object	Oggetto JSON contenente le opzioni di configurazione globali dell'agent.
private_options	JSON object (optional)	Oggetto JSON contenente le opzioni di configurazione private del plugin, se fornite.

Esempio:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Convalida richiesta

Una richiesta per eseguire la funzione *Validate* dell'interfaccia *Configurator*.

direction	agent → plugin
response	yes

Parametri specifici delle richieste *Validate*:

Name	Type	Comments
private_options	JSON object (optional)	Oggetto JSON contenente le opzioni di configurazione private del plugin, se fornite.

Esempio:

```
{"id":7,"type":9,"private_options":{...}}
```

Convalida risposta

Risposta della funzione *Validate* dell'interfaccia *Configurator*.

direction	plugin → agent
response	n/a

Parametri specifici delle risposte *Validate*:

Name	Type	Comments
error	string (optional)	Un messaggio di errore restituito se la funzione Validate non viene eseguita correttamente. Assente se eseguita correttamente.

Esempio:

```
{"id":8,"type":10}
```

oppure

```
{"id":8,"type":10,"error":"error message"}
```

Java gateway

Panoramica Zabbix Java gateway può essere installato dal codice sorgente o dai pacchetti.

Il supporto nativo per il monitoraggio delle applicazioni JMX è disponibile sotto forma di un demone Zabbix denominato "Zabbix Java gateway". Zabbix Java gateway è un demone scritto in Java. Per trovare il valore di un determinato contatore JMX su un host, Zabbix server interroga Zabbix Java gateway, che utilizza la JMX management API per interrogare da remoto l'applicazione interessata. Non è necessario installare software aggiuntivo sull'applicazione; è sufficiente avviarla con l'opzione `-Dcom.sun.management.jmxremote` nella riga di comando.

Java gateway accetta connessioni in ingresso da Zabbix server o proxy e può essere utilizzato solo come "proxy passivo". A differenza del proxy Zabbix, può essere utilizzato anche da un proxy Zabbix (i proxy Zabbix non possono essere concatenati). L'accesso a ogni Java gateway viene configurato direttamente nel file di configurazione di Zabbix server o del Zabbix proxy; pertanto, è possibile configurare un solo Java gateway per ogni Zabbix server o Zabbix proxy. Se un host contiene item di tipo *JMX agent* e item di altri tipi, solo gli item *JMX agent* verranno passati a Java gateway per il recupero.

Quando un item deve essere aggiornato tramite Java gateway, Zabbix server o proxy si connette al Java gateway e richiede il valore, che il Java gateway recupera a sua volta e restituisce al server o al proxy. Pertanto, Java gateway non memorizza nella cache alcun valore.

Zabbix server o proxy dispone di un tipo specifico di processi che si connettono al Java gateway, controllato dall'opzione `StartJavaPollers`. Internamente, Java gateway avvia più thread, controllati dall'opzione `START_POLLERS` option. Sul lato server, se una connessione richiede più di "Timeout" secondi, viene terminata, ma Java gateway potrebbe essere ancora impegnato nel recupero del valore dal contatore JMX. Per risolvere questo problema, Java gateway fornisce l'opzione `TIMEOUT`, che consente di impostare un timeout per le operazioni di rete JMX.

Vedere anche:

- Parametri di configurazione: Zabbix Java gateway
- Configurazione dai pacchetti Debian/Ubuntu
- Configurazione dai pacchetti RHEL

- Configurazione dai sorgenti

Zabbix server o proxy cercherà di raggruppare il più possibile le richieste verso un singolo target JMX (influenzato dagli intervalli degli item) e di inviarle al Java gateway in un'unica connessione per ottenere prestazioni migliori.

Si consiglia di impostare `StartJavaPollers` su un valore minore o uguale a `START_POLLERS`; in caso contrario, potrebbero verificarsi situazioni in cui non sono disponibili thread nel Java gateway per gestire le richieste in arrivo. In tali casi, il Java gateway utilizza `ThreadPoolExecutor.CallerRunsPolicy`, il che significa che il thread principale gestirà la richiesta in arrivo e temporaneamente non accetterà nuove richieste.

Se si sta cercando di monitorare applicazioni Java basate su Wildfly con Zabbix Java gateway, installare la versione più recente di `jboss-client.jar` disponibile nella pagina di download di Wildfly.

Parametri di configurazione: Zabbix Java gateway

Se utilizzi gli script `startup.sh` e `shutdown.sh` per avviare Zabbix Java gateway, puoi specificare i parametri di configurazione necessari nel file `settings.sh`. Gli script di avvio e arresto leggono il file delle impostazioni e si occupano di convertire le variabili della shell (elencate nella prima colonna) in proprietà Java (elencate nella seconda colonna).

Se avvii Zabbix Java gateway manualmente eseguendo direttamente `java`, allora specifichi le corrispondenti proprietà Java dalla riga di comando.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	Indirizzo IP su cui ascoltare.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Porta su cui ascoltare.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Nome del file PID. Se omesso, Zabbix Java Gateway viene avviato come applicazione console.
PROPERTIES_FILE	zabbix.propertiesFile	no			Nome del file delle proprietà. Può essere usato per impostare proprietà aggiuntive usando un formato chiave-valore in modo che non siano visibili sulla riga di comando oppure per sovrascrivere quelle esistenti. Ad esempio: "javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Numero di thread worker da avviare.
TIMEOUT	zabbix.timeout	no	1-30	3	Tempo di attesa per le operazioni di rete (in secondi).

Warning:
La porta 10052 non è registrata presso IANA.

Configurazione dai pacchetti Debian/Ubuntu

Panoramica

Se installato da pacchetti, le seguenti informazioni ti aiuteranno a configurare Zabbix Java gateway.

Configurazione ed esecuzione di Java gateway

La configurazione di Java gateway può essere modificata nel file:

```
/etc/zabbix/zabbix_java_gateway.conf
```

Per ulteriori dettagli, vedere i parametri di configurazione di Zabbix Java gateway.

Per avviare Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Per avviare automaticamente Zabbix Java gateway all'avvio:

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

Configurazione di server per l'uso con Java gateway

Con Java gateway in esecuzione, devi indicare a Zabbix server dove trovare Zabbix Java gateway. Questo si fa specificando i parametri `JavaGateway` e `JavaGatewayPort` nel file di configurazione del server. Se l'host su cui è in esecuzione l'applicazione JMX è monitorato da Zabbix proxy, allora devi specificare i parametri di connessione nel file di configurazione del proxy invece.

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

Per impostazione predefinita, server non avvia alcun processo relativo al monitoraggio JMX. Se desideri utilizzarlo, tuttavia, devi specificare il numero di istanze pre-fork di Java pollers. Lo fai nello stesso modo in cui specifichi i pollers e i trappers standard.

```
StartJavaPollers=5
```

Non dimenticare di riavviare server o proxy, una volta terminata la configurazione.

Debugging Java gateway

Il file di log di Zabbix Java gateway è:

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

Se desideri aumentare il livello di logging, modifica il file:

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

e cambia `info` in `level="info"` in debug o persino `trace` (per una risoluzione dei problemi approfondita):

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

</configuration>
```

Monitoraggio JMX

Vedere la pagina Monitoraggio JMX per maggiori dettagli.

Configurazione dai pacchetti RHEL

Panoramica

Se installato da pacchetti, le seguenti informazioni ti aiuteranno a configurare lo Java gateway.

Configurazione ed esecuzione di Java gateway

I parametri di configurazione di Zabbix Java gateway possono essere modificati nel file:

```
/etc/zabbix/zabbix_java_gateway.conf
```

Per ulteriori dettagli, vedere i parametri di configurazione di Zabbix Java gateway.

Per avviare Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Per avviare automaticamente Zabbix Java gateway all'avvio del sistema:

RHEL 7 e versioni successive:

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

RHEL precedenti alla 7:

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

Configurazione di server per l'uso con Java gateway

Con Java gateway attivo e in esecuzione, devi indicare a Zabbix server dove trovare Zabbix Java gateway. Questo si fa specificando i parametri `JavaGateway` e `JavaGatewayPort` nel file di configurazione del server. Se l'host su cui è in esecuzione l'applicazione JMX è monitorato da Zabbix proxy, allora devi specificare i parametri di connessione nel file di configurazione del proxy invece.

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

Per impostazione predefinita, server non avvia alcun processo relativo al monitoraggio JMX. Se desideri utilizzarlo, tuttavia, devi specificare il numero di istanze pre-forked di Java pollers. Lo fai nello stesso modo in cui specifichi i pollers e i trappers normali.

```
StartJavaPollers=5
```

Non dimenticare di riavviare server o proxy, una volta terminata la configurazione.

Debugging Java gateway

Il file di log di Zabbix Java gateway è:

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

Se desideri aumentare il livello di logging, modifica il file:

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

e cambia `info` in `level="info"` in debug o persino `trace` (per una risoluzione approfondita dei problemi):

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

</configuration>
```

Monitoraggio JMX

Vedere la pagina JMX monitoring per maggiori dettagli.

Installazione dai sorgenti

Panoramica

Se installato dai sorgenti, le seguenti informazioni ti aiuteranno a configurare il Java gateway di Zabbix.

Panoramica dei file

Se hai ottenuto Java gateway dai sorgenti, dovresti avere una raccolta di script shell, file JAR e file di configurazione in `$PREFIX/sbin/zabbix_java`. Il ruolo di questi file è riepilogato di seguito.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Il file JAR di Java gateway.

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Dipendenze di Java gateway: le librerie Logback, SLF4J e Android JSON.

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

File di configurazione per Logback.

```
shutdown.sh
startup.sh
```

Script di utilità per avviare e arrestare Java gateway.

```
settings.sh
```

File di configurazione caricato dagli script di avvio e arresto sopra indicati.

Configurazione ed esecuzione di Java gateway

Per impostazione predefinita, Java gateway ascolta sulla porta 10052. Se prevedi di eseguire Java gateway su una porta diversa, puoi specificarlo nello script `settings.sh`. Consulta la descrizione del file di configurazione di Java gateway per sapere come specificare questa e altre opzioni.

> **Warning:**
> La porta 10052 non è registrata presso IANA.

Una volta che hai familiarità con le impostazioni, puoi avviare Java gateway eseguendo lo script `startup.sh`:

```
./startup.sh
```

Allo stesso modo, quando non hai più bisogno di Java gateway, esegui lo script `shutdown.sh` per arrestarlo:

```
./shutdown.sh
```

Nota che, a differenza di server o proxy, Java gateway è leggero e non necessita di un database.

Configurazione del server per l'uso con Java gateway

Con Java gateway in esecuzione, è necessario indicare a Zabbix server dove trovare Zabbix Java gateway. Questo si fa specificando i parametri `JavaGateway` e `JavaGatewayPort` nel file di configurazione del server. Se l'host su cui è in esecuzione l'applicazione JMX è monitorato da Zabbix proxy, allora i parametri di connessione vanno specificati invece nel file di configurazione del proxy.

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

Per impostazione predefinita, server non avvia alcun processo relativo al monitoraggio JMX. Se tuttavia si desidera utilizzarlo, è necessario specificare il numero di istanze pre-forked di Java pollers. Questo si fa nello stesso modo in cui si specificano i pollers e i trappers normali.

```
StartJavaPollers=5
```

Non dimenticare di riavviare server o proxy, una volta terminata la configurazione.

Debugging Java gateway

Nel caso ci siano problemi con Java gateway oppure il messaggio di errore che vedi riguardo a un item nel frontend non sia sufficientemente descrittivo, potresti voler dare un'occhiata al file di log di Java gateway.

Per impostazione predefinita, Java gateway registra le proprie attività nel file `/tmp/zabbix_java.log` con livello di log "info". A volte queste informazioni non sono sufficienti ed è necessario avere informazioni con livello di log "debug". Per aumentare il livello di logging, modifica il file `lib/logback.xml` e cambia l'attributo level del tag `<root>` in debug:

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Nota che, a differenza di Zabbix server o Zabbix proxy, non è necessario riavviare Zabbix Java gateway dopo aver modificato il file `logback.xml` - le modifiche in `logback.xml` verranno applicate automaticamente. Quando hai terminato il debugging, puoi riportare il livello di logging a "info".

Se desideri registrare i log in un file diverso o in un supporto completamente diverso, come un database, modifica il file `logback.xml` in base alle tue esigenze. Per maggiori dettagli, consulta il Manuale di Logback.

A volte, per scopi di debugging, è utile avviare Java gateway come applicazione console invece che come daemon. Per farlo, commenta la variabile `PID_FILE` in `settings.sh`. Se `PID_FILE` viene omessa, lo script `startup.sh` avvia Java gateway come applicazione console e fa in modo che Logback utilizzi il file `lib/logback-console.xml` invece, che non solo registra i log nella console, ma ha anche abilitato il livello di logging "debug".

Infine, nota che poiché Java gateway utilizza SLF4J per il logging, puoi sostituire Logback con il framework che preferisci inserendo un file JAR appropriato nella directory lib. Per maggiori dettagli, consulta il Manuale di SLF4J.

Monitoraggio JMX

Vedere la pagina Monitoraggio JMX per maggiori dettagli.

Mittente

Panoramica Zabbix sender è un'utilità da riga di comando che può essere utilizzata per inviare dati sulle prestazioni al server o al proxy Zabbix per l'elaborazione.

L'utilità viene generalmente utilizzata negli script utente a esecuzione prolungata per l'invio periodico dei dati di disponibilità e delle prestazioni.

Per inviare i risultati direttamente al server o al proxy Zabbix, è necessario configurare un item di tipo *Zabbix trapper*.

Vedere anche:

- Protocollo Zabbix sender
- Libreria a collegamento dinamico Zabbix sender per Windows
- Libreria Python per Zabbix, che dispone di funzionalità integrate per agire come Zabbix sender

Esecuzione di Zabbix sender Un esempio di esecuzione di Zabbix sender UNIX:

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

dove:

- `z` - host del server Zabbix (è possibile utilizzare anche un indirizzo IP);
- `s` - nome tecnico dell'host monitorato (come registrato nel frontend Zabbix);
- `k` - chiave dell'item;
- `o` - valore da inviare.

Attention:
Le opzioni che contengono spazi devono essere racchiuse tra virgolette doppie.

Zabbix sender può essere utilizzato per inviare più valori da un file di input. Per ulteriori informazioni, consultare la pagina man di Zabbix sender.

Se viene specificato un file di configurazione, Zabbix sender utilizza tutti gli indirizzi definiti nel parametro di configurazione `ServerActive` dell'agent per inviare i dati. Se l'invio a un indirizzo non riesce, sender tenta di inviare i dati agli altri indirizzi. Se l'invio di dati in batch a un indirizzo non riesce, i batch successivi non vengono inviati a tale indirizzo.

Zabbix sender accetta stringhe con codifica UTF-8 (sia per i sistemi Unix-like sia per Windows) senza un byte order mark (BOM) all'inizio del file.

Zabbix sender su Windows può essere eseguito in modo analogo:

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

Gli scenari di invio in tempo reale di `zabbix_sender` raccolgono più valori passati al programma in rapida successione e li inviano al server tramite un'unica connessione. Un valore che non dista più di 0,2 secondi dal valore precedente può essere inserito nello stesso stack, ma il tempo massimo di polling rimane di 1 secondo.

Note:
Zabbix sender termina se nel file di configurazione specificato è presente una voce di parametro non valida (che non segue la notazione `parameter=value`).

Esecuzione di Zabbix sender con low-level discovery Un esempio di esecuzione di Zabbix sender per inviare un valore in formato JSON per il low-level discovery:

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

Perché questo funzioni, la regola di low-level discovery deve avere il tipo di item *Zabbix trapper* (in questo esempio, con la chiave `trapper.discovery.item`).

Protocollo Zabbix sender

Zabbix server e Zabbix proxy utilizzano un protocollo di comunicazione basato su JSON per ricevere dati da Zabbix sender. I dati possono essere ricevuti con l'aiuto di un trapper item oppure di un HTTP agent item con il trapping abilitato.

I messaggi di richiesta e risposta devono iniziare con header and data length.

Richiesta di Zabbix sender

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Risposta del server Zabbix

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Richiesta di Zabbix sender con timestamp

In alternativa, Zabbix sender può inviare una richiesta con un timestamp e nanosecondi.

```
{
    "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
}
```

Risposta del server Zabbix

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Libreria a collegamento dinamico Zabbix sender per Windows

In un ambiente Windows, le applicazioni possono inviare dati a server/proxy Zabbix utilizzando la libreria a collegamento dinamico Zabbix sender (`zabbix_sender.dll`) invece di dover avviare un processo esterno (`zabbix_sender.exe`).

Per compilare applicazioni utente con `zabbix_sender.dll` sono necessari `zabbix_sender.h` e `zabbix_sender.lib`.

Ottenimento

Esistono due modi per ottenere `zabbix_sender.dll`.

1. Scaricare i file `zabbix_sender.h`, `zabbix_sender.lib` e `zabbix_sender.dll` come archivio ZIP.

Quando si scelgono le opzioni di download, assicurarsi di selezionare "No encryption" in *Encryption* e "Archive" in *Packaging*. Quindi scaricare Zabbix agent (non Zabbix agent 2).

I file `zabbix_sender.h`, `zabbix_sender.lib` e `zabbix_sender.dll` si troveranno all'interno dell'archivio ZIP scaricato nella directory `bin\dev`. Decomprimere i file nel percorso desiderato.

2. Compilare `zabbix_sender.dll` dai sorgenti (vedere le istruzioni).

La libreria a collegamento dinamico con i file di sviluppo si troverà nella directory `bin\winXX\dev`. Per utilizzarla, includere il file di intestazione `zabbix_sender.h` ed eseguire il linking con la libreria `zabbix_sender.lib`.

Vedi anche

- esempio di una semplice utility Zabbix sender implementata con la libreria a collegamento dinamico Zabbix sender per illustrare l'uso della libreria;
- file zabbix_sender.h con le funzioni di interfaccia della libreria a collegamento dinamico Zabbix sender. Questo file contiene la documentazione che spiega lo scopo di ciascuna funzione di interfaccia, i relativi argomenti e il valore restituito.

Ottieni

Panoramica Zabbix get è un'utilità da riga di comando che può essere utilizzata per comunicare con l'agent di Zabbix e recuperare le informazioni richieste dall'agent.

L'utilità viene solitamente usata per la risoluzione dei problemi degli agent di Zabbix.

Vedi anche la libreria Python per Zabbix, che dispone di funzionalità integrate per comportarsi come Zabbix get.

Esecuzione di Zabbix get Un esempio di esecuzione di Zabbix get su UNIX per ottenere il valore del carico del processore dall'agent:

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

Un altro esempio di esecuzione di Zabbix get per acquisire una stringa da un sito web:

```
cd bin
./zabbix_get -s 192.168.1.1 -p 10050 -k "web.page.regexp[www.example.com,,,\"USA: ([a-zA-Z0-9.-]+)\",,\1]"
```

Si noti che la chiave dell'item qui contiene uno spazio, quindi vengono usate le virgolette per indicare la chiave dell'item alla shell. Le virgolette non fanno parte della chiave dell'item; verranno rimosse dalla shell e non saranno passate all'agent Zabbix.

Se una chiave dell'item non è supportata, Zabbix get restituirà il codice di uscita 1.

Zabbix get accetta i seguenti parametri da riga di comando:

```
-s --host <host name or IP>             Specifica il nome host o l'indirizzo IP di un host
-p --port <port number>                 Specifica il numero di porta dell'agent in esecuzione sull'host (p
-I --source-address <IP address>        Specifica l'indirizzo IP sorgente
-t --timeout <seconds>                  Specifica il timeout. Intervallo valido: 1-600 secondi (predefinit
-k --key <item key>                     Specifica la chiave dell'item di cui recuperare il valore
-P --protocol <value>                   Protocollo usato per comunicare con l'agent. Valori:
                                            auto - connessione tramite protocollo JSON, fallback e nuovo t
                                            json - connessione tramite protocollo JSON
                                            plaintext - connessione tramite protocollo plaintext in cui vi
-h --help                               Mostra questo messaggio di aiuto
-V --version                            Mostra il numero di versione

--tls-connect <value>                   Modalità di connessione all'agent. Valori:
                                            unencrypted - connessione senza crittografia (predefinito)
                                            psk - connessione tramite TLS e una chiave precondivisa
                                            cert - connessione tramite TLS e un certificato
--tls-ca-file <CA file>                 Percorso completo di un file contenente i certificati CA di livell
--tls-crl-file <CRL file>               Percorso completo di un file contenente i certificati revocati
--tls-agent-cert-issuer <cert issuer>   Emittente del certificato dell'agent consentito
--tls-agent-cert-subject <cert subject> Soggetto del certificato dell'agent consentito
--tls-cert-file <cert file>             Percorso completo di un file contenente il certificato o la catena
--tls-key-file <key file>               Percorso completo di un file contenente la chiave privata
--tls-psk-identity <PSK-identity>       Stringa univoca, sensibile alle maiuscole/minuscole, usata per ide
--tls-psk-file <PSK-file>               Percorso completo di un file contenente la chiave precondivisa
```

```
--tls-cipher13 <cipher-string>          Stringa di cifratura per OpenSSL 1.1.1 o versioni successive per T
--tls-cipher <cipher-string>            Stringa di priorità GnuTLS (per TLS 1.2 e versioni successive) o s
```

Per ulteriori informazioni, vedere anche la pagina man di Zabbix get.

Zabbix get su Windows può essere eseguito in modo analogo:

```
zabbix_get.exe [options]
```

JS

Panoramica zabbix_js è un'utilità da riga di comando che può essere utilizzata per testare script incorporati.

Questa utilità eseguirà uno script utente con un parametro stringa e stamperà il risultato. Gli script vengono eseguiti utilizzando il motore di scripting incorporato di Zabbix.

In caso di errori di compilazione o di esecuzione, zabbix_js stamperà l'errore su stderr e terminerà con il codice 1.

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

Utilizzo zabbix_js accetta i seguenti parametri da riga di comando:

```
-s, --script script-file          Specifica il nome del file dello script da eseguire
                                  Se come nome del file viene specificato "-", lo script verrà letto da st
-i, --input input-file            Specifica il nome del file delle informazioni di input
                                  Il contenuto verrà passato allo script come valore
                                  Mutuamente esclusivo con -p
                                  Se come nome del file viene specificato "-", lo script verrà letto da st
-p, --param input-param           Specifica il parametro di input: la variabile da passare allo script com
                                  Mutuamente esclusivo con -i
-l, --loglevel log-level          Specifica il livello di log
                                  Intervallo: 0-5
-t, --timeout timeout             Specifica il timeout in secondi
                                  Intervallo valido: 1-600 secondi (predefinito: 10 secondi)
-h, --help                        Mostra le informazioni di aiuto
-V, --version                     Mostra il numero di versione
-w <webdriver url>                Abilita il monitoraggio del browser
```

Esempi: Esempio 1: Esecuzione di uno script con una variabile contenente un valore

Lo script (script-file.js):

```
return value;
```

Utilizzo:

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

Output: example_value

Esempio 2: Esecuzione di uno script con un file contenente informazioni di input

Il file con lo script (script-file.js):

```
return value;
```

Il file con le informazioni di input (example.txt):

```
Example of input information from the file
```

Utilizzo:

```
zabbix_js -s script-file.js -i example.txt
```

Output: `Example of input information from the file`

Esempio 3: Esecuzione di uno script con input letto da stdin

Il file con lo script (`script-file.js`):

```
return value;
```

Utilizzo:

```
zabbix_js -s script-file.js -i -
```

Input (stdin):

```
Example of input from stdin
```

Output: `Example of input from stdin`

Servizio web

Panoramica Zabbix web service è un processo utilizzato per la comunicazione con servizi web esterni. Attualmente, Zabbix web service viene utilizzato per generare e inviare report pianificati, con l'obiettivo di aggiungere ulteriori funzionalità in futuro.

Zabbix server si connette al servizio web tramite HTTP(S). Zabbix web service richiede che Google Chrome sia installato sullo stesso host; su alcune distribuzioni il servizio può funzionare anche con Chromium (vedere problemi noti).

Installazione Il pacchetto ufficiale zabbix-web-service è disponibile nel repository Zabbix.

Per compilare il web service Zabbix dai sorgenti, specificare l'opzione di configurazione `--enable-webservice`.

Per configurare il web service Zabbix, aggiornare i parametri del file di configurazione del web service.

> **Attention:**
> Si consiglia vivamente di configurare la crittografia tra il server Zabbix e il web service Zabbix utilizzando certificati. Per impostazione predefinita, i dati trasmessi tra il server Zabbix e il web service Zabbix non sono crittografati, il che può consentire accessi non autorizzati.

Parametri di configurazione: Zabbix web service

Panoramica

Il servizio web di Zabbix è un processo utilizzato per la comunicazione con servizi web esterni.

I parametri supportati dal file di configurazione del servizio web di Zabbix (`zabbix_web_service.conf`) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare i dettagli completi.

Parameter	Description
AllowedIP	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di server Zabbix e proxy Zabbix.
DebugLevel	Il livello di debug.
IgnoreURLCertErrors	Specifica la gestione degli errori di convalida del certificato TLS quando si accede all'URL del frontend.
Include	È possibile includere singoli file o tutti i file in una directory nel file di configurazione.
ListenPort	Il servizio ascolterà su questa porta le connessioni dal server.
LogFile	Il nome del file di log.
LogFileSize	La dimensione massima del file di log.
LogType	Il tipo di output del log.
Timeout	Il tempo massimo (in secondi) impiegato per la formattazione del PDF report di una dashboard.
TLSAccept	Quali connessioni in ingresso accettare.

Parameter	Description
TLSCAFile	Il percorso completo di un file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato del servizio o la catena di certificati, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata del servizio, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

Tutti i parametri sono facoltativi, salvo quando sia esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non quelli presenti nei file di configurazione distribuiti;
- I valori supportano le variabili di ambiente;
- Zabbix supporta i file di configurazione solo in codifica UTF-8 senza BOM;
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli del parametro

AllowedIP

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS dei server Zabbix e dei proxy Zabbix. Le connessioni in ingresso saranno accettate solo dagli host elencati qui.
Se il supporto IPv6 è abilitato, allora `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere usato per consentire qualsiasi indirizzo IPv4.

Esempio:

`127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com`

Obbligatorio: sì

DebugLevel

Specifica il livello di debug: - *0* - informazioni di base sull'avvio e l'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (produce molte informazioni); - *5* - debug esteso (produce ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

IgnoreURLCertErrors

Specifica la gestione degli errori di convalida del certificato TLS quando si accede all'URL del frontend:
0 - non ignorare gli errori del certificato;
1 - ignorare gli errori del certificato.

Predefinito: 0
 Intervallo: 0-1

Include

Specifica una directory o singoli file da includere nel file di configurazione.
 Durante l'installazione di Zabbix, questo parametro viene impostato su `/usr/local/etc`, a meno che non venga modificato durante la compilazione.

Regole generali per il parametro `Include`:

- Il percorso può essere relativo alla posizione del file `zabbix_web_service.conf`.
- È possibile specificare più parametri `Include` in un unico file di configurazione.
- Zabbix elabora il parametro `Include` non appena lo legge, quindi è consigliabile posizionarlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non si avvierà.

Regole per l'inclusione di directory o pattern di file:

- Non dare per scontato alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup che alcuni editor di testo creano automaticamente (ad esempio, la modifica di `include/my_specific.conf` può creare `include/my_specific.conf.BAK`), poiché entrambi i file verranno inclusi.

Per includere solo i file pertinenti da una directory, utilizzare il carattere jolly (*) per la corrispondenza dei pattern.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Il servizio ascolterà su questa porta le connessioni provenienti dal server.

Predefinito: 10053
 Intervallo: 1024-32767

LogFile

Il nome del file di log.

Esempio:

`/tmp/zabbix_web_service.log`

Obbligatorio: Sì, se LogType è impostato su `file`; altrimenti no

LogFileSize

La dimensione massima di un file di log in MB.
0 - disabilita la rotazione automatica dei log.
Nota: se viene raggiunto il limite di dimensione del file di log e la rotazione del file non riesce, per qualsiasi motivo, il file di log esistente viene troncato e ricreato da zero.

Predefinito: 1
 Intervallo: 0-1024

LogType

Il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro LogFile; - *system* - scrive il log in syslog; - *console* - scrive il log nell'output standard.

Predefinito: `file`

Timeout

Il tempo massimo (in secondi) impiegato per formattare il PDF del report di una dashboard.

Predefinito: 10
 Intervallo: 1-30

TLSAccept

Quali connessioni in ingresso accettare: - *unencrypted* - accetta connessioni senza crittografia (predefinito); - *cert* - accetta connessioni con TLS e un certificato.

Predefinito: `unencrypted`

TLSCAFile

Il percorso completo del file contenente i certificati della CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCertFile

Il percorso completo del file contenente il certificato del servizio o la catena di certificati, utilizzato per le comunicazioni crittografate con i componenti di Zabbix.

TLSKeyFile

Il percorso completo del file contenente la chiave privata del servizio, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

Adattatore bridge

L'adattatore bridge di Zabbix consente al tuo server Zabbix di comunicare con l'app mobile Zabbix.

L'app mobile Zabbix non si connette direttamente al tuo server Zabbix. Al contrario, due componenti trasferiscono i dati tra loro:

- Zabbix Bridge - un servizio ospitato centralmente che scambia dati con l'app mobile Zabbix sul tuo dispositivo mobile.
- Adattatore bridge di Zabbix - un componente che installi ed esegui. Connette il tuo server Zabbix a Zabbix Bridge.

L'adattatore bridge può essere installato sullo stesso sistema del tuo server Zabbix oppure su un sistema diverso, utilizzando i pacchetti.

Per configurare l'adattatore bridge di Zabbix, aggiorna il relativo file di configurazione (percorso predefinito: `/etc/zabbix/zabbix_bridge_`

Per mantenere sicuro l'adattatore bridge di Zabbix, consulta Accesso mobile sicuro.

Flusso dei dati Quando si utilizza l'app mobile Zabbix, i dati seguono il flusso seguente:

1. Si esegue un'azione nell'app mobile Zabbix, che invia quindi una query a Zabbix Bridge.
2. Zabbix Bridge passa la query a Zabbix bridge adapter come richiesta JSON-RPC.
3. Zabbix bridge adapter verifica se il metodo API Zabbix richiesto è consentito. Se il metodo non è consentito, bridge adapter rifiuta la richiesta. Se il metodo è consentito, bridge adapter invia la richiesta all'API Zabbix.
4. L'API Zabbix elabora la richiesta e restituisce una risposta a Zabbix bridge adapter.
5. Zabbix bridge adapter restituisce la risposta a Zabbix Bridge.
6. Zabbix Bridge passa la risposta all'app mobile Zabbix, che visualizza i dati sul dispositivo mobile.

Parametri di configurazione: adattatore bridge di Zabbix

Panoramica

I parametri di configurazione consentono di personalizzare il comportamento dell'adattatore bridge di Zabbix.

Specificare i valori dei parametri nel file di configurazione (`zabbix_bridge_adapter.conf`). Tutti i parametri sono **facoltativi**, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
AllowedAPIMethods	Specificare quali metodi API di Zabbix possono essere chiamati tramite l'adattatore bridge di Zabbix.
DataDir	Specificare la directory in cui vengono creati e archiviati il file di stato dell'adattatore bridge e le chiavi di crittografia.
DebugLevel	Impostare il livello di debug.
Include	Specificare una directory o singoli file da includere nel file di configurazione.
ListenIP	Specificare l'indirizzo IP di ascolto dell'adattatore bridge per le connessioni HTTP(S) in ingresso dal server Zabbix.
ListenHealthPort	Specificare la porta di ascolto dell'adattatore bridge per le connessioni HTTP in ingresso per i controlli di integrità.
ListenPort	Specificare la porta di ascolto dell'adattatore bridge per le connessioni HTTP(S) in ingresso dal server Zabbix.
LogFile	Specificare il percorso completo del file di log dell'adattatore bridge.
LogFileSize	Impostare la dimensione massima del file di log (in MB).
LogType	Specificare il tipo di output del log.
PidFile	Specificare il file PID.
TLSConfig.AllowedCN	Specificare un elenco separato da virgole dei Common Name client consentiti (solo `mTLS`).
TLSConfig.CAFile	Specificare il percorso completo di un file contenente il certificato o i certificati CA di livello superiore utilizzati per verificare i certificati client in modalità `mTLS`.
TLSConfig.CertFile	Specificare il percorso completo di un file contenente il certificato o la catena di certificati.
TLSConfig.KeyFile	Specificare il percorso completo di un file contenente la chiave privata.
TLSConfig.Mode	Impostare la modalità TLS per le connessioni in ingresso su `ListenPort`.
ServerId	Specificare l'identificatore univoco del server Zabbix. **Obbligatorio**.
ZabbixAPITLSCAFile	Specificare il percorso completo di un file contenente il certificato o i certificati CA di livello superiore utilizzati dall'adattatore bridge per verificare il certificato dell'API Zabbix.
ZabbixAPITLSConnect	Impostare la modalità TLS per le connessioni in uscita all'API Zabbix.
ZabbixAPIURL	Specificare l'URL dell'endpoint dell'API Zabbix. **Obbligatorio**.
ZabbixBridgeURL	Specificare l'URL di Zabbix Bridge per il recupero della configurazione di Zabbix Bridge (recuperata all'avvio dell'adattatore bridge).

Tenere presente quanto segue:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta solo file di configurazione con codifica UTF-8 senza BOM;
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli dei parametri

AllowedAPIMethods

Specifica quali metodi dell'API Zabbix possono essere chiamati tramite Zabbix bridge adapter.
 Per più metodi, ripeti il parametro, specificando un metodo per riga.

Ogni valore del parametro deve essere un nome di metodo esatto (ad esempio, `problem.get`) oppure un modello con caratteri jolly che sostituisce solo l'intero nome dell'API (ad esempio, `*.get`) o l'intero nome del metodo (ad esempio, `problem.*`).

Qualsiasi altro valore non è valido e impedisce l'avvio di Zabbix bridge adapter. Sono inclusi un carattere jolly isolato (`*`), un modello con carattere jolly parziale (`host*` o `host*.*`) oppure un valore vuoto (`AllowedAPIMethods=`).

Esempio:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

L'insieme minimo di metodi richiesti dall'app mobile Zabbix:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

Predefinito: non impostato (tutti i metodi dell'API Zabbix sono negati)

DataDir

Specificare la directory in cui vengono creati e archiviati il file di stato del bridge adapter e le chiavi di crittografia.
 La directory deve avere i permessi 0700 e tutti i file archiviati al suo interno devono avere i permessi 0600.

Predefinito: `/var/lib/zabbix-bridge-adapter`

DebugLevel

Imposta il livello di debug:

- 0 - informazioni di base sull'avvio e l'arresto dei processi Zabbix
- 1 - informazioni critiche
- 2 - informazioni sugli errori
- 3 - avvisi
- 4 - per il debug (genera molte informazioni)
- 5 - debug esteso (genera ancora più informazioni)

Predefinito: 3
 Intervallo: 0-5

Include

Specifica una directory o singoli file da includere nel file di configurazione.
 Durante l'installazione di Zabbix, questo parametro viene impostato su `/usr/local/etc`, a meno che non venga modificato durante la compilazione.

Regole generali per il parametro `Include`:

- Il percorso può essere relativo alla posizione del file `zabbix_bridge_adapter.conf`.
- È possibile specificare più parametri `Include` in un unico file di configurazione.
- Zabbix elabora il parametro `Include` non appena lo legge; pertanto, è consigliabile inserirlo alla fine del file di configurazione.

Regole per i file inclusi:

- Ogni file deve essere leggibile.
- Ogni file deve avere una sintassi corretta. In caso contrario, il componente non verrà avviato.

Regole per l'inclusione di directory o pattern di file:

- Non presumere alcun ordine di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico).
- Non definire lo stesso parametro in più di un file incluso (ad esempio, per sovrascrivere un'impostazione generale con una specifica), poiché non è possibile prevedere quale valore verrà utilizzato.
- Prestare attenzione alle copie di backup create automaticamente da alcuni editor di testo (ad esempio, modificando `include/my_specific.conf` potrebbe essere creato `include/my_specific.conf.BAK`), poiché entrambi i file verranno inclusi.

Per includere solo i file pertinenti da una directory, utilizzare il carattere jolly (*) per la corrispondenza dei pattern.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Specificare l'indirizzo IP di ascolto dell'adattatore bridge per le connessioni HTTP(S) in ingresso dal server Zabbix.
 Se non specificato, l'adattatore bridge ascolta su tutti gli indirizzi IP.

ListenHealthPort

Specificare la porta di ascolto dell'adattatore bridge per le connessioni HTTP in ingresso dei controlli di integrità.
 Fornisce gli endpoint `/health` e `/ready`.
 Se non specificata, il servizio di integrità non viene avviato.

Predefinito: 8081
 Intervallo: 1024-32767

ListenPort

Specificare la porta di ascolto dell'adattatore bridge per le connessioni HTTP(S) in entrata dal server Zabbix.

Predefinito: 10054
 Intervallo: 1024-32767

LogFile

Specificare il percorso completo del file di log dell'adattatore bridge.

Obbligatorio: **Sì** se `LogType` è impostato su `file`

Predefinito: `/tmp/zabbix_bridge_adapter.log`

LogFileSize

Imposta la dimensione massima del file di log (in MB).
 0 - disabilita la rotazione automatica dei log.

Predefinito: 1
 Intervallo: 0-1024

LogType

Specificare il tipo di output del log:

- `file` - scrive il log nel file specificato dal parametro `LogFile`.
- `system` - scrive il log nel syslog.
- `console` - scrive il log sull'output standard.

Predefinito: `file`

PidFile

Specificare il file PID.

Predefinito: `/tmp/zabbix_bridge_adapter.pid`

ServerId

Specificare l'identificatore univoco del server Zabbix.
 Disponibile in *Report* > *Informazioni di sistema* nell'interfaccia utente di Zabbix.

Obbligatorio: **Sì**

TLSConfig.AllowedCN

Specificare un elenco separato da virgole dei Common Name dei client consentiti (solo `mTLS`).
 Se non specificato, l'adattatore bridge non verifica il Common Name del client.

TLSConfig.CAFile

Specificare il percorso completo di un file contenente i certificati CA di livello superiore utilizzati per verificare i certificati client in modalità `mTLS`.

TLSConfig.CertFile

Specificare il percorso completo di un file contenente il certificato o la catena di certificati.

TLSConfig.KeyFile

Specificare il percorso completo del file contenente la chiave privata.

TLSConfig.Mode

Imposta la modalità TLS per le connessioni in entrata su `ListenPort`:

- `none` - accetta connessioni senza crittografia.
- `tls` - TLS standard con v1.2 come versione minima.
- `mtls` - estende TLS con autenticazione reciproca.

Predefinito: `none`

ZabbixAPITLSCAFile

Specificare il percorso completo di un file contenente il/i certificato/i CA di livello superiore utilizzato/i dall'adattatore bridge per verificare il certificato dell'API Zabbix.

Obbligatorio: **Sì** se `ZabbixAPITLSConnect` è impostato su `verify_ca` o `verify_full`.

ZabbixAPITLSConnect

Imposta la modalità TLS per le connessioni in uscita all'API di Zabbix:

- `disabled` (o vuoto) - usa HTTP semplice; `ZabbixAPIURL` deve usare `http://`.
- `required` - usa HTTPS senza verifica del certificato (solo per i test).
- `verify_ca` - usa HTTPS e verifica il certificato dell'API di Zabbix.
- `verify_full` - usa HTTPS e verifica il certificato e il nome host dell'API di Zabbix.

ZabbixAPIURL

Specificare l'URL dell'endpoint dell'API di Zabbix.

Obbligatorio: **Sì**

Esempio:

`ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php`

ZabbixBridgeURL

Specificare l'URL del bridge Zabbix per il recupero della configurazione del bridge Zabbix (recuperata all'avvio dell'adattatore bridge).

Predefinito: `ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration`

server MCP

Panoramica Il server MCP di Zabbix è un processo che consente agli agenti AI di lavorare con Zabbix utilizzando il Model Context Protocol (MCP). Il server MCP comunica con i client MCP e con l'API di Zabbix.

Utilizzando il server MCP, un agent AI può:

- Recuperare gruppi di host, gruppi di template, host, template, item, regole LLD, interfacce, macro, trigger, problemi, periodi di manutenzione e dati storici degli item (ad eccezione degli item binari);
- Eseguire aggiornamenti dei problemi: aggiungere commenti, riconoscere o annullare il riconoscimento, modificare la gravità, sopprimere o annullare la soppressione oppure chiudere i problemi;
- Creare, aggiornare ed eliminare periodi di manutenzione.

I token API di Zabbix vengono utilizzati per l'autorizzazione; i token vengono inviati con ogni richiesta. L'accesso è limitato dalle autorizzazioni del proprietario del token: sono visibili solo i dati dei gruppi di host disponibili per tale utente e le azioni che l'utente non è autorizzato a eseguire vengono rifiutate.

Il server MCP è stateless: tra le richieste non viene mantenuta alcuna sessione.

Note:
Sebbene il server MCP di Zabbix possa funzionare con versioni precedenti o successive di Zabbix, la compatibilità è ufficialmente supportata solo con la stessa versione di Zabbix del server MCP.

Vedere anche:

- Parametri di configurazione: server MCP
- Strumenti e risorse

Installazione Il pacchetto ufficiale zabbix-mcp-server è disponibile nel repository Zabbix.

Per installare Zabbix MCP server dai sorgenti, compilarlo con `make`.

Per configurare Zabbix MCP server, aggiornare i parametri del file di configurazione del server MCP. È necessario impostare almeno i parametri `FrontendURL` e `AllowedIP`.

Il file di configurazione predefinito (`zabbix_mcp_server.conf`) si trova in `/etc/zabbix/`; nega gli strumenti che più spesso espongono dati sensibili. Per abilitare uno strumento, rimuovere la riga `DenyTool` corrispondente.

Vedere anche: Procedure consigliate per la sicurezza

Esecuzione Per avviare il server MCP, eseguire:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Per verificare il file di configurazione e uscire senza avviare il server, utilizzare l'opzione `-T`:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
```

Sono supportate le seguenti opzioni della riga di comando:

Opzione	Descrizione
`-c <file>`	Il percorso del file di configurazione. Il valore predefinito è `/etc/zabbix/zabbix_mcp_server.conf`.
`-T`	Convalida il file di configurazione ed esce.
`-h`	Visualizza le informazioni della guida.
`-V`	Visualizza il numero di versione.

Quando il server MCP viene installato dai sorgenti, è necessario creare manualmente un'unità systemd, ad esempio:

```
[Unit]
Description=Zabbix MCP server
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Connessione di un client MCP Qualsiasi client MCP che supporti il trasporto HTTP Streamable può essere connesso al server MCP. Sono necessarie le seguenti impostazioni:

Impostazione	Valore
Trasporto	HTTP Streamable
URL	`https://<host>:<ListenPort>/mcp`
Header	`Authorization: Bearer <Zabbix API token>`, inviato con ogni richiesta

Ogni utente deve configurare il proprio token API.

Per verificare la configurazione, inviare una singola richiesta al server MCP:

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

Il codice di stato HTTP 200 e un elenco degli strumenti disponibili nella risposta confermano che il server MCP è raggiungibile e accetta il token; l'elenco mostra quali strumenti sono esposti. Gli altri codici di stato indicano i seguenti problemi:

- 400 - l'header `Accept` non include entrambi i tipi di contenuto richiesti dal trasporto;
- 401 - manca l'header `Authorization` con il token;
- 403 - l'indirizzo del client non è elencato nel parametro `AllowedIP`.

Approvazione e accesso agli strumenti Se il client MCP supporta la richiesta di compilazione di moduli, ogni modifica deve essere approvata dall'utente prima di essere inviata a Zabbix. Il client MCP mostra all'utente cosa verrà modificato e la modifica viene applicata solo se l'utente la conferma; se l'utente rifiuta o ignora la richiesta, in Zabbix non viene modificato nulla. Un'approvazione è valida per cinque minuti e solo per la modifica esatta mostrata; durante questo periodo, l'approvazione può essere riutilizzata.

Se il client MCP non supporta la richiesta di compilazione di moduli (inclusi i client che supportano solo la richiesta tramite URL), le modifiche vengono applicate immediatamente, senza approvazione. In questo caso, limitare gli strumenti di scrittura e le autorizzazioni dell'API Zabbix.

L'approvazione è particolarmente importante perché l'output degli strumenti può contenere testo non attendibile proveniente dai sistemi monitorati, che potrebbe influenzare un agent.

Utilizzare AllowTool, DenyTool, AllowToolRegexp e DenyToolRegexp per controllare quali strumenti espone il server MCP. Gli strumenti negati non possono essere chiamati né elencati e non è possibile accedere alle risorse a essi associate.

Warning:
Le regole AllowTool, DenyTool, AllowToolRegexp e DenyToolRegexp limitano l'accesso MCP, ma non modificano le autorizzazioni del token Zabbix. Se l'indirizzo del frontend è esposto a un agent AI, l'agent potrebbe tentare di inviare chiamate API dirette. Mantenere la posizione del frontend Zabbix e la configurazione del server MCP inaccessibili all'agent, in modo che le richieste debbano passare attraverso il server MCP e i relativi controlli di sicurezza.

Procedure consigliate per la sicurezza Token API

L'autenticazione e l'autorizzazione sono gestite interamente da Zabbix. Il client MCP invia un token API Zabbix con ogni richiesta; il server MCP lo inoltra senza memorizzarlo né registrarlo nei log, mentre Zabbix convalida il token, applica le autorizzazioni e registra le azioni nel registro di audit.

Utilizzare un token separato, con privilegi minimi e in scadenza, per ogni agent e utente. Conservare i token in un luogo sicuro e ruotare regolarmente i token API.

Crittografia TLS

Per impedire l'invio dei token in testo non crittografato, impostare `TLSAccept=cert` e specificare il certificato e la chiave (`TLSCertFile` e `TLSKeyFile`), oppure eseguire il server MCP dietro un reverse proxy affidabile con terminazione TLS e impostare `TLSAccept=unencrypted`. HTTP non crittografato è accettabile solo in una rete affidabile.

Restrizioni di accesso

Limitare il parametro `AllowedIP` agli indirizzi dei client MCP attendibili e utilizzare le autorizzazioni di Zabbix per limitare le operazioni che i token API possono eseguire.

Il server MCP non fornisce la limitazione della frequenza delle richieste. Ogni richiesta MCP normalmente comporta una chiamata all'API di Zabbix, sebbene un agent in esecuzione in un ciclo possa comunque generare un carico significativo.

Protezione dei dati sensibili

Le chiavi degli item possono contenere credenziali (ad esempio negli item SSH, JMX, ODBC e HTTP agent), i valori delle macro possono contenere password e stringhe di community, le interfacce rivelano indirizzi interni e i valori degli item di tipo log o testo possono contenere qualsiasi dato. Per questo motivo, gli strumenti `item_get`, `lld_get`, `interface_get`, `macro_get` e `history_get` sono negati per impostazione predefinita nel file di configurazione fornito.

Per evitare l'esposizione dei valori sensibili delle macro al provider LLM dell'agent connesso, utilizzare le macro segrete; i relativi valori non vengono mai restituiti dall'API.

Parametri di configurazione: Zabbix MCP server

Panoramica

Il server MCP di Zabbix è un processo che consente agli agent AI di lavorare con Zabbix utilizzando il Model Context Protocol (MCP).

In questa sezione sono elencati i parametri supportati dal file di configurazione del server MCP di Zabbix (`zabbix_mcp_server.conf`).

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare tutti i dettagli.

Parametro	Descrizione
AllowedIP	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, o di nomi DNS dei client MCP.
AllowTool	Una regola con caratteri jolly che consente uno strumento.
AllowToolRegexp	Una regola con espressione regolare che consente gli strumenti.
DebugLevel	Il livello di debug.
DenyTool	Una regola con caratteri jolly che nega uno strumento.
DenyToolRegexp	Una regola con espressione regolare che nega gli strumenti.
FrontendURL	L'endpoint dell'API di Zabbix.
IgnoreURLCertErrors	Specifica la gestione degli errori di convalida dei certificati TLS durante l'accesso all'API di Zabbix.
ListenIP	L'indirizzo IP su cui rimanere in ascolto.
ListenPort	La porta su cui rimanere in ascolto.
LogFile	Il nome del file di log.
LogFileSize	La dimensione massima del file di log.
LogType	Il tipo di output del log.
MaxResponseSize	La dimensione massima di una risposta restituita da uno strumento di lettura.
PidFile	Il nome del file PID.
Timeout	Il tempo massimo (in secondi) di attesa di una risposta dall'API di Zabbix.
TLSAccept	Quali connessioni in ingresso accettare.
TLSCertFile	Il percorso completo di un file contenente il certificato del server o la catena di certificati.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata del server.
ToolPrefix	Un prefisso aggiunto ai nomi di tutti gli strumenti.

Tutti i parametri sono facoltativi, salvo quando viene esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori presenti nei file di configurazione forniti;
- I valori supportano le variabili d'ambiente nel formato `${NAME}`;
- Zabbix supporta i file di configurazione solo con codifica UTF-8 senza BOM;
- I commenti che iniziano con # sono supportati solo all'inizio della riga.

Dettagli dei parametri

AllowedIP

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, o di nomi DNS dei client MCP. Le connessioni in entrata saranno accettate solo dagli host elencati qui.

Esempio:

```
AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com
```

Obbligatorio: sì

AllowTool

Una regola con caratteri jolly che consente uno strumento. Il parametro può essere specificato più volte.
 I modelli con caratteri jolly utilizzano la sintassi delle chiavi degli item, dove * corrisponde a qualsiasi carattere. La regola verifica il nome semplice dello strumento e ignora ToolPrefix.
 Tutte le regole `AllowTool`, `DenyTool`, `AllowToolRegexp` e `DenyToolRegexp` formano un unico elenco nell'ordine in cui sono specificate; decide la prima regola corrispondente. Uno strumento che non corrisponde ad alcuna regola è consentito.

Esempio:

```
AllowTool=host_get
```

AllowToolRegexp

Una regola di espressione regolare che consente l'utilizzo degli strumenti. Il parametro può essere specificato più volte.
 Le espressioni regolari utilizzano la sintassi RE2 e corrispondono a qualsiasi parte del nome semplice dello strumento, a meno che non siano ancorate.

Esempio:

```
AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$
```

DebugLevel

Specificare il livello di debug: - *0* - informazioni di base sull'avvio e sull'arresto dei processi Zabbix; - *1* - informazioni critiche; - *2* - informazioni sugli errori; - *3* - avvisi; - *4* - per il debug (genera molte informazioni); - *5* - debug esteso (genera ancora più informazioni).

Predefinito: 3
 Intervallo: 0-5

DenyTool

Una regola con caratteri jolly che nega uno strumento. Il parametro può essere specificato più volte.
 Uno strumento negato non viene incluso nell'elenco degli strumenti e non può essere chiamato. Per i dettagli, vedere AllowTool.

Esempio:

```
DenyTool=maintenance_delete
```

DenyToolRegexp

Una regola di espressione regolare che nega gli strumenti. Il parametro può essere specificato più volte.
 Per i dettagli, vedere AllowToolRegexp.

Esempio:

```
DenyToolRegexp=^problem_(close|suppress)$
```

FrontendURL

L'endpoint dell'API di Zabbix: l'URL del frontend di Zabbix che termina con `api_jsonrpc.php`.

Esempio:

```
FrontendURL=https://zabbix.example.com/api_jsonrpc.php
```

Obbligatorio: sì

IgnoreURLCertErrors

Specifica la gestione degli errori di convalida dei certificati TLS durante l'accesso all'API di Zabbix:
 false - non ignorare gli errori dei certificati;
 true - ignorare gli errori dei certificati (utilizzare solo in un ambiente di sviluppo).

Predefinito: `false`

ListenIP

L'indirizzo IP su cui rimanere in ascolto.

Predefinito: `0.0.0.0`

ListenPort

La porta su cui rimanere in ascolto. Il percorso dell'endpoint MCP è `/mcp`.

Predefinito: 8443
 Intervallo: 1024-32767

LogFile

Il nome del file di log.

Predefinito: `/tmp/zabbix_mcp_server.log`

Obbligatorio: No

LogFileSize

La dimensione massima di un file di log in MB.
0 - disabilita la rotazione automatica dei log.

Predefinito: 1
 Intervallo: 0-1024

LogType

Il tipo di output del log: - *file* - scrive il log nel file specificato dal parametro LogFile; - *system* - scrive il log nel syslog; - *console* - scrive il log nell'output standard.

Predefinito: `file`

MaxResponseSize

La dimensione massima di una risposta restituita da uno strumento di lettura. La risposta si interrompe al raggiungimento del primo tra questo limite e l'argomento `limit` dello strumento.
 Supporta suffissi delle unità.

Predefinito: 64K
 Intervallo: 16K-256K

PidFile

Il nome del file PID.

Predefinito: `/tmp/zabbix_mcp_server.pid`

Timeout

Il tempo massimo (in secondi) di attesa per una risposta dall'API di Zabbix.

Predefinito: 10
 Intervallo: 3-30

TLSAccept

Quali connessioni in ingresso accettare: - *unencrypted* - accetta connessioni senza crittografia (impostazione predefinita); - *cert* - accetta connessioni con TLS e un certificato.

Predefinito: `unencrypted`

TLSCertFile

Il percorso completo del file contenente il certificato del server o la catena di certificati.

Obbligatorio: Sì, se `TLSAccept` è impostato su `cert`; altrimenti, no

TLSKeyFile

Il percorso completo del file contenente la chiave privata del server.

Obbligatorio: Sì, se `TLSAccept` è impostato su `cert`; altrimenti, no

ToolPrefix

Un prefisso aggiunto ai nomi di tutti gli strumenti, ad esempio per evitare conflitti di nomi con gli strumenti di altri server MCP. Ad esempio, con `ToolPrefix=zbx_`, lo strumento `host_get` viene esposto come `zbx_host_get`.
 Caratteri consentiti: lettere, cifre, carattere di sottolineatura (`_`), trattino (`-`) e punto (`.`).
 Le regole di accesso agli strumenti corrispondono ai nomi degli strumenti senza il prefisso.

Esempio:

`ToolPrefix=zbx_`

Strumenti e risorse

Panoramica

Il server MCP di Zabbix espone le funzionalità di Zabbix ai client MCP come *strumenti* e *risorse*.

I nomi degli strumenti descrivono l'attività. Se il parametro ToolPrefix è impostato, il prefisso viene aggiunto a ogni nome dello strumento; le regole di accesso agli strumenti corrispondono ai nomi semplici elencati di seguito.

Strumenti di lettura

Strumento	Metodo API Zabbix	Argomenti	Ordinato per	Risorsa
`hostgroup_get`	hostgroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://hostgroup/<i`
`templategroup_get`	templategroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://templategrou`
`host_get`	host.get	`hostids`, `groupids`, `search`, `limit`	name	`zabbix://host/<id>`
`template_get`	template.get	`templateids`, `groupids`, `search`, `limit`	name	`zabbix://template/<id`
`item_get`	item.get	`hostids`, `itemids`, `search`, `limit`	name	`zabbix://item/<id>`
`lld_get`	discoveryrule.get	`hostids`, `ruleids`, `limit`	name	-
`interface_get`	hostinterface.get	`hostids`, `interfaceids`, `limit`	ID dell'interfaccia	-

Strumento	Metodo API Zabbix	Argomenti	Ordinato per	Risorsa
macro_get	usermacro.get	hostids o global, macro_names, limit	macro	-
problem_get	problem.get	eventids, groupids, hostids, severities, acknowledged, suppressed, recent, limit	ID dell'evento, decrescente	-
maintenance_get	maintenance.get	maintenanceids, groupids, hostids, limit	name	-
history_get	history.get	itemids, time_from, time_till, type, limit	clock, decrescente	-
trigger_get	trigger.get	triggerids, hostids, search, limit	description	zabbix://trigger/<id>

Nota:

- host_get e template_get restituiscono anche i trigger di ciascun oggetto (un insieme limitato di campi), i tag e i template collegati. Gli item, le regole di discovery, le interfacce e le macro vengono restituiti dagli strumenti corrispondenti.
- history_get richiede un intervallo di tempo non superiore a 7 giorni e con inizio non antecedente a 7 giorni fa. L'argomento type deve essere uno tra float, unsigned, character, text, log o json e deve corrispondere al tipo di valore degli item; se il tipo è errato, viene restituito un risultato vuoto, che non può essere distinto da un item privo di dati. La cronologia degli item binari (tipo di valore 5) non può essere recuperata tramite il server MCP.
- Gli strumenti di lettura richiedono un elenco fisso di campi; tutti i campi (output: extend) non vengono mai richiesti.
- I risultati degli strumenti che dispongono di una risorsa contengono il campo uri, in modo che un agent possa passare da un elenco a una singola entità.

Formato della risposta

Tutti gli strumenti di lettura restituiscono la risposta nello stesso formato:

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

Una risposta è limitata dall'argomento limit (predefinito: 100, massimo: 1000) e dalla dimensione serializzata della risposta (MaxResponseSize, predefinita: 64K). La risposta viene interrotta quando viene raggiunto uno dei due limiti. La risposta viene troncata solo tra i risultati, mai all'interno di un risultato, e una risposta troncata indica quale limite è stato raggiunto. Ogni strumento ha un ordine di ordinamento fisso, quindi una risposta troncata contiene sempre l'inizio dell'intero set di risultati. La paginazione non è supportata; per ottenere altri risultati, restringere la query.

Strumenti per i problemi

Tutti gli strumenti per i problemi chiamano event.acknowledge con i flag di azione corrispondenti.
L'argomento eventids è obbligatorio; gli ID degli eventi possono essere recuperati con problem_get.
È possibile aggiungere un messaggio (commento) con qualsiasi strumento per i problemi.
L'argomento message è facoltativo, tranne che per problem_annotate, per il quale è obbligatorio; se specificato, il messaggio viene registrato nella stessa chiamata all'API di Zabbix.

Strumento	Effetto	Argomenti aggiuntivi
problem_acknowledge	Conferma il problema.	-
problem_unacknowledge	Rimuove la conferma.	-
problem_close	Chiude manualmente il problema (solo per i trigger che consentono la chiusura manuale).	-
problem_annotate	Registra un messaggio.	message (obbligatorio)
problem_change_severity	Modifica la gravità del problema.	severity (0-5)
problem_suppress	Sopprime il problema fino a un'ora specificata o a tempo indeterminato.	suppress_until
problem_unsuppress	Rimuove la soppressione.	-

Non è possibile creare o eliminare problemi; tutti gli strumenti per i problemi aggiornano i problemi esistenti.

Strumenti di manutenzione

Strumento	Metodo API di Zabbix	Argomenti
`maintenance_create`	maintenance.create	`name`, `groupids` e/o `hostids`, `active_since`, `active_till`, `timeperiods`, `description`, `tags`, `tags_evaltype`
`maintenance_update`	maintenance.update	`maintenanceid` e uno qualsiasi degli argomenti precedenti. Gli argomenti elenco sostituiscono l'intero insieme di valori; un elenco vuoto cancella l'insieme.
`maintenance_delete`	maintenance.delete	`maintenanceids`

Quando viene aggiornato un periodo di manutenzione, gli argomenti elenco (come `hostids`, `groupids` e `timeperiods`) sostituiscono i valori esistenti invece di aggiungervi nuovi valori. Ad esempio, se un periodo di manutenzione include gli host A e B e `maintenance_update` viene chiamato con `hostids` contenente solo l'host C, il periodo di manutenzione includerà solo l'host C.

`maintenance_create` è l'unica operazione non idempotente e `maintenance_delete` è l'unica operazione che non può essere annullata. Per impedirne l'utilizzo, negare questi strumenti con il parametro DenyTool.

Comportamento degli strumenti di scrittura

Gli strumenti per i problemi e la manutenzione sono *strumenti di scrittura*. Funzionano come segue:

- Gli argomenti vengono convalidati prima di qualsiasi chiamata all'API di Zabbix; se la convalida non riesce, l'errore indica il vincolo non rispettato.
- Se il client lo supporta, la modifica viene applicata solo dopo essere stata approvata dall'utente.
- La modifica viene eseguita da Zabbix utilizzando il token del chiamante. Zabbix verifica le autorizzazioni e registra la modifica nel registro di audit associandola al proprietario del token.
- Gli errori di Zabbix, inclusi gli errori di autorizzazione, vengono restituiti invariati come oggetto di errore con i campi `code`, `message` e `data`.
- Gli strumenti di scrittura restituiscono direttamente il risultato di Zabbix; i limiti di risposta degli strumenti di lettura non si applicano.
- Le chiamate di scrittura non vengono mai ritentate dal server MCP.

Risorse

Le risorse utilizzano lo schema URI `zabbix://<kind>/<id>`. Solo `zabbix://about` è incluso nella risposta `resources/list`; gli ID delle entità non possono essere enumerati e vengono ottenuti dai risultati degli strumenti. La risposta `resources/templates/list` elenca i tipi di entità.

URI	Contenuto
`zabbix://about`	L'URL dell'API, l'URL del frontend, la versione di Zabbix, l'identità a cui corrisponde il token del chiamante e l'elenco degli strumenti esposti.
`zabbix://host/<hostid>`	Il record del host senza item, trigger, interfacce e macro.
`zabbix://hostgroup/<groupid>`	Il record del gruppo di host senza l'elenco dei membri.
`zabbix://template/<templateid>`	Il record del template senza item, trigger e regole di discovery.
`zabbix://templategroup/<groupid>`	Il record del gruppo di template senza l'elenco dei membri.
`zabbix://item/<itemid>`	La definizione dell'item senza il valore corrente.
`zabbix://trigger/<triggerid>`	La definizione del trigger senza host, dipendenze, funzioni e tag.

Una risorsa il cui strumento corrispondente è negato non esiste. Un'entità inesistente e un'entità a cui il proprietario del token non ha accesso vengono entrambe segnalate come *resource not found*; questi casi sono deliberatamente indistinguibili. La risorsa `zabbix://about` non è interessata dalle regole di accesso agli strumenti.

Variabili di ambiente

Panoramica Le variabili d'ambiente consentono di configurare i componenti di Zabbix senza hardcodare i valori nei file di configurazione. Questo semplifica la gestione delle configurazioni in ambienti dinamici, come Docker, dove le variabili possono essere passate in fase di esecuzione per adattarsi a diverse configurazioni.

Nel caso più semplice, impostando il valore del parametro di configurazione DebugLevel del server Zabbix su una variabile d'ambiente, è possibile usarla per configurare il server all'avvio:

```
### File di configurazione del server Zabbix:
DebugLevel=${NEW_DEBUG_LEVEL}

### Avvio del server Zabbix:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

Le variabili d'ambiente sono supportate dai seguenti componenti di Zabbix:

- Server
- Proxy
- Agent (UNIX o Windows)
- Agent 2 (UNIX o Windows), inclusi i plugin
- Web service
- Zabbix sender (quando si utilizza l'opzione -c, --config)

Note importanti

- Quando un parametro di configurazione è impostato su una variabile di ambiente, che non viene specificata all'avvio del componente, viene utilizzato il valore predefinito del parametro.
- Quando si usano comandi runtime (ad esempio, per aumentare il livello di log dell'agent), è necessario specificare tutte le variabili di ambiente precedentemente utilizzate. Questo perché i componenti Zabbix usano il proprio file di configurazione per eseguire i comandi runtime; se le variabili di ambiente vengono omesse, verranno utilizzati i valori predefiniti dei parametri di configurazione. Vedere Esempi.
- Il comando runtime `userparameter_reload` comando runtime non supporta il ricaricamento delle variabili di ambiente. Durante il ricaricamento, le variabili vengono ignorate e vengono ricaricati solo i parametri con valori normali.
- Le variabili di ambiente correnti del processo, utilizzate nei file di configurazione, vengono cancellate dopo l'avvio del componente Zabbix. Questo garantisce che i processi figlio (ad esempio, script remoti eseguiti da Zabbix) non possano accedere a queste variabili. Tuttavia, si noti che le variabili iniziali del processo possono ancora essere recuperate (ad esempio, tramite il file `/proc/<PID>/environ`).

Sintassi Le variabili di ambiente devono usare la seguente sintassi: `${alphanumerics/underscores}`.

Il nome della variabile può includere solo lettere (a-z, A-Z), underscore (_) e cifre (0-9), e non deve iniziare con una cifra.

Le variabili che non corrispondono alla sintassi richiesta o che sono combinate con un valore normale saranno trattate come valori normali, il che può generare errori.

Sintassi corretta delle variabili:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Sintassi errata delle variabili:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

> **Note:**
> In Windows, i nomi delle variabili di ambiente non fanno distinzione tra maiuscole e minuscole.

Esempi I seguenti esempi mostrano come configurare e utilizzare le variabili di ambiente con i componenti di Zabbix.

Esempio 1: Configurazione e test dell'agent Zabbix

1. Impostare le variabili d'ambiente nel file di configurazione dell'agent:

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Testare il file di configurazione:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. Avviare l'agent con le variabili d'ambiente:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Quando si utilizzano i comandi runtime (ad esempio, per aumentare il livello di log dell'agent), è necessario specificare tutte le variabili d'ambiente utilizzate in precedenza:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Questo perché l'agent utilizza il proprio file di configurazione per eseguire i comandi runtime; se le variabili d'ambiente vengono omesse, verranno utilizzati i valori predefiniti dei parametri di configurazione.

In alternativa, dopo aver impostato le variabili d'ambiente nel file di configurazione dell'agent, è possibile renderle disponibili ai processi (ad esempio, utilizzando il comando `export`). In questo modo si riduce il rischio di comportamenti imprevisti dovuti a variabili mancanti o impostate in modo errato.

```
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
```

Esempio 2: Configurazione di Zabbix agent per un container

Se si crea e configura una propria immagine personalizzata per i componenti Zabbix (ad esempio, Zabbix agent), è possibile definire i parametri di configurazione utilizzando variabili d'ambiente e quindi avviare il container con tali variabili.

1. Durante la preparazione dell'immagine, impostare le variabili d'ambiente nel file di configurazione dell'agent:

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. Dopo aver creato l'immagine del container, avviare il container dell'agent (ad esempio, Docker) con le variabili d'ambiente:

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. Quando si utilizzano i comandi runtime (ad esempio, per aumentare il livello di log dell'agent), accedere alla shell del container ed eseguire il comando runtime:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

> **Attention:**
> Il comando runtime `userparameter_reload` non supporta il ricaricamento delle variabili d'ambiente. Durante il ricaricamento, le variabili vengono ignorate e vengono ricaricati solo i parametri con valori regolari.

Configurazione

Panoramica La configurazione in Zabbix consiste nel specificare quali host e sistemi monitorare, definire quali dati raccogliere e stabilire come vengono inviate le notifiche quando si verificano problemi.

Usa la barra laterale per navigare tra gli aspetti di configurazione pertinenti.

Attività di configurazione chiave

- Configurare un nuovo host o gruppo di host → Host e gruppi di host
- Definire quali dati raccogliere dagli host → Item
- Creare regole per rilevare i problemi → Rilevamento dei problemi con i trigger
- Configurare come vengono recapitate le notifiche → Notifiche sugli eventi
- Visualizzare i dati di monitoraggio tramite dashboard, grafici e mappe → Visualizzazione
- Riutilizzare le configurazioni su più host → Template e gruppi di template, Template pronti all'uso
- Controllare l'accesso degli utenti e le autorizzazioni → Utenti e gruppi di utenti

Best practice di configurazione Le scelte di configurazione influenzano le prestazioni e la manutenibilità. Sebbene ogni installazione di Zabbix sia diversa, queste best practice possono essere usate come base da adattare secondo necessità.

Host separato per ogni istanza di monitoraggio

Un host in Zabbix è un'entità logica piuttosto che una macchina o un dispositivo fisico. Crea un host separato per ogni istanza indipendente, come una macchina virtuale, un database, un container o uno switch di rete.

Questo approccio offre:

1. Dati di monitoraggio più puliti, con item, trigger e notifiche di allarme separati per ogni host.
2. Controllo granulare dell'accesso degli utenti tramite user-roles in linea con il principio del privilegio minimo.

Usare i template e l'aggiornamento di massa per host simili

Per host simili come *Network switch 1* e *Network switch 2*, puoi creare un template con le metriche richieste (*Network switch template*) e raggruppare gli host in un gruppo host (*Network switches*).

Quando aggiungi nuovi switch, puoi usare l'aggiornamento di massa in *Data Collection* > *Hosts* per collegare il template invece di clonare manualmente gli host.

Configurare gli item master e dipendenti per ridurre al minimo le richieste

Per ridurre al minimo le richieste all'entità di destinazione, configura un item master per raccogliere i dati in un'unica richiesta, quindi usa item dipendenti con preprocessing per estrarre valori specifici.

Ad esempio, l'item master potrebbe raccogliere una risposta JSON o XML con più metriche oppure eseguire una query al database che restituisce più colonne (ad esempio, numero di connessioni aperte, connessioni interrotte, numero massimo di connessioni simultanee consentite e totale cumulativo delle connessioni dall'avvio), mentre gli item dipendenti analizzeranno e memorizzeranno ogni valore separatamente.

Elimina la cronologia dell'item master dopo la raccolta e conserva solo i dati degli item dipendenti.

Distribuire un proxy Zabbix quando necessario

Se tutti i tuoi host si trovano sulla stessa rete locale del server Zabbix e non ci sono problemi di scalabilità o prestazioni, potresti non aver bisogno di un proxy. Negli ambienti più grandi o complessi, un proxy consente una distribuzione più uniforme del carico.

Aggiungi un proxy Zabbix quando:

1. Monitori più host utilizzando diversi metodi di raccolta delle metriche dietro un firewall. Il proxy raccoglie i dati dagli host e li inoltra al server Zabbix, riducendo il numero di porte del firewall da aprire.
2. Monitori sedi, filiali o reti remote. I proxy Zabbix nelle sedi remote continuano a raccogliere dati durante le interruzioni di rete e li inviano quando la connessione viene ripristinata.
3. Gestisci distribuzioni su larga scala. Se disponi di molti host e/o raccogli un numero elevato di valori al secondo, distribuisci un proxy per ridurre il carico del server e migliorare le prestazioni.

Configurare il recupero indipendente delle macro segrete

Potresti voler usare le macro utente secret come testo segreto oppure come macro secret vault.

Per impostazione predefinita, i valori delle macro segrete vengono recuperati da Zabbix server e propagati ai proxy. Per una maggiore sicurezza con le macro secret vault, configura i valori delle macro in modo che vengano recuperati in modo indipendente da Zabbix server e dai proxy.

Configurazione di un template

Panoramica

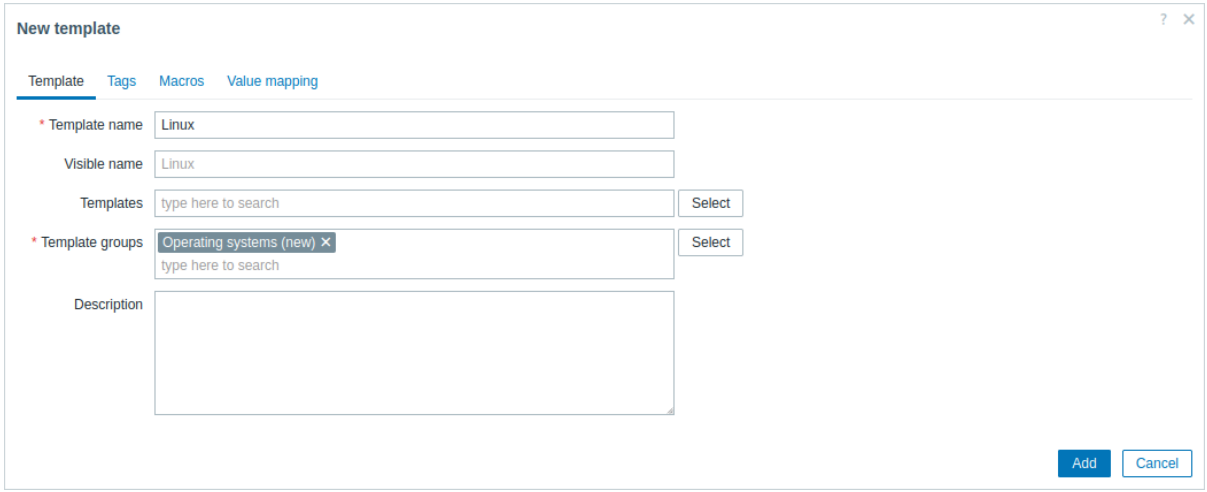
La configurazione di un template richiede innanzitutto la creazione del template definendone i parametri generali, quindi l'aggiunta di entità (item, trigger, grafici, ecc.).

Creazione di un template

Per creare un template, procedere come segue:

1. Andare a *Raccolta dati* > *Templates*.
2. Fare clic su *Crea template*.
3. Modificare gli attributi del template.

La scheda **Template** contiene gli attributi generali del template.

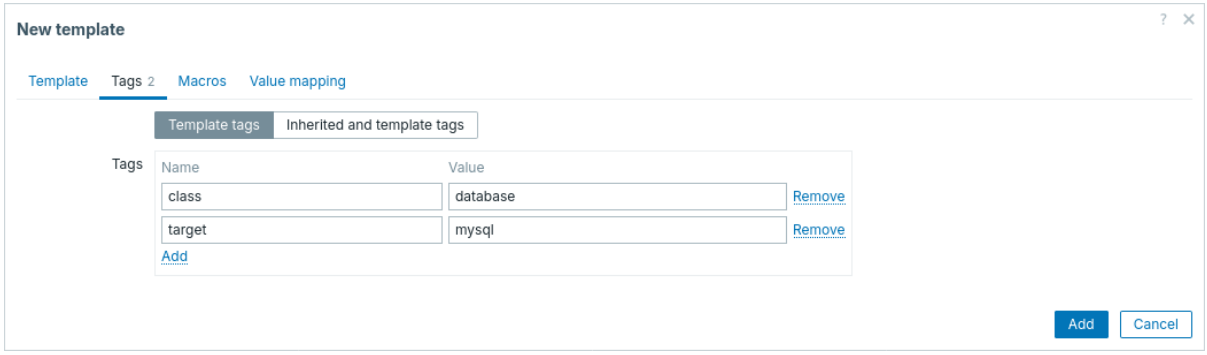


Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

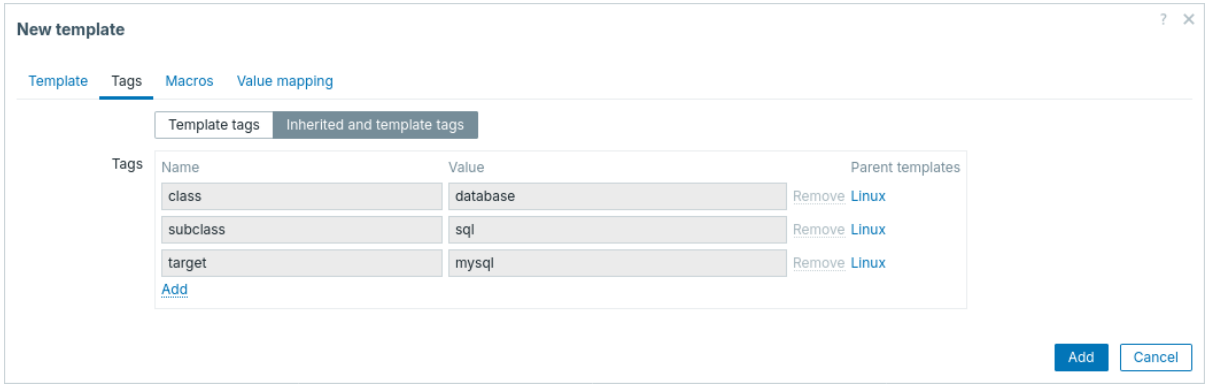
Attributi del template:

Parametro	Descrizione
Nome del template	Nome univoco del template. Sono consentiti caratteri alfanumerici, spazi, punti, trattini e caratteri di sottolineatura. Non sono consentiti spazi iniziali o finali.
Nome visibile	Se si imposta questo nome, sarà quello visualizzato negli elenchi, nelle mappe e così via.
Templates	Collega uno o più template a questo template. Tutte le entità (item, trigger e così via) verranno ereditate dai template collegati. Per collegare un nuovo template, digitare il relativo nome nel campo *Templates*: verrà visualizzato un elenco a discesa dei template corrispondenti. In alternativa, fare clic su *Seleziona* per aprire la finestra popup *Templates*, che inizialmente non mostra alcun template. Per compilare l'elenco, digitare il nome di un gruppo di template nel campo *Gruppo di template* e selezionarne uno dall'elenco suggerito, oppure fare clic sul pulsante *Seleziona* accanto al campo *Gruppo di template* per aprire la finestra popup *Gruppi di template*. Dopo aver scelto un gruppo di template, la finestra popup *Templates* viene aggiornata e mostra i template appartenenti a quel gruppo. I template selezionati verranno collegati al salvataggio o all'aggiornamento della configurazione. Per scollegare un template, utilizzare una delle due opzioni nel blocco *Templates*: *Scollega* - scollega il template, ma ne conserva le entità (item, trigger e così via); *Scollega e cancella* - scollega il template e rimuove tutte le relative entità (item, trigger e così via).
Gruppi di template	Gruppi di template a cui appartiene il template.
Descrizione	Descrizione del template.
Vendor e versione	Vendor e versione del template; visualizzati solo durante l'aggiornamento di template esistenti (template predefiniti forniti da Zabbix, template importati o template modificati tramite la Template API) se la configurazione del template contiene tali informazioni. Non possono essere modificati nel frontend di Zabbix. Per i template predefiniti, la versione viene visualizzata come segue: versione principale di Zabbix, delimitatore ("-"), numero di revisione (incrementato a ogni nuova versione del template e reimpostato a ogni versione principale di Zabbix). Ad esempio, 7.0-0, 7.0-5, 8.0-0, 8.0-3.

La scheda **Tag** consente di definire i tag a livello di template. Tutti i problemi degli host collegati a questo template verranno contrassegnati con i valori inseriti qui.



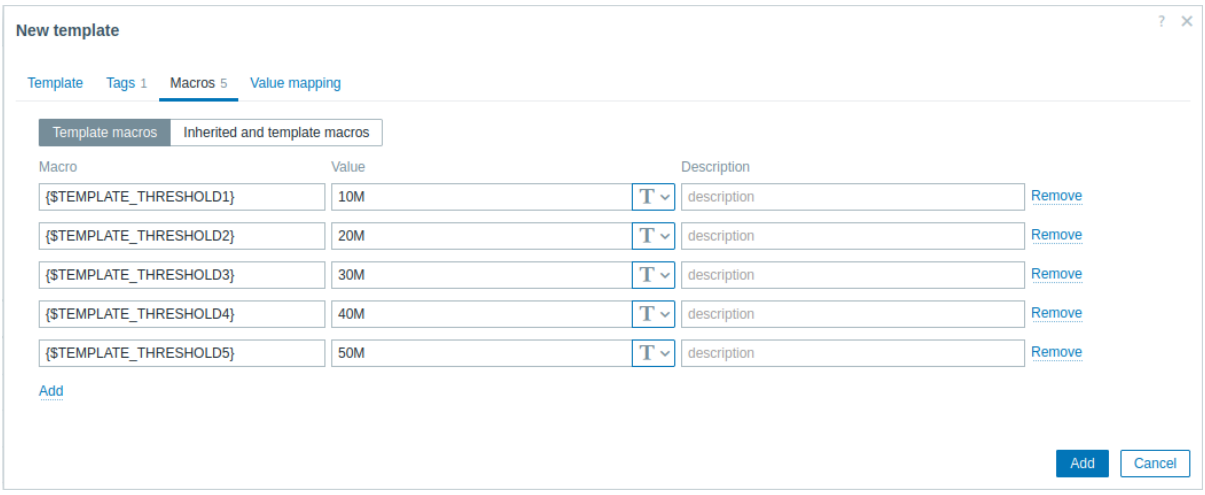
Se si seleziona l'opzione *Tag ereditati e del template*, è inoltre possibile visualizzare i tag dei template collegati che verranno ereditati dal template.



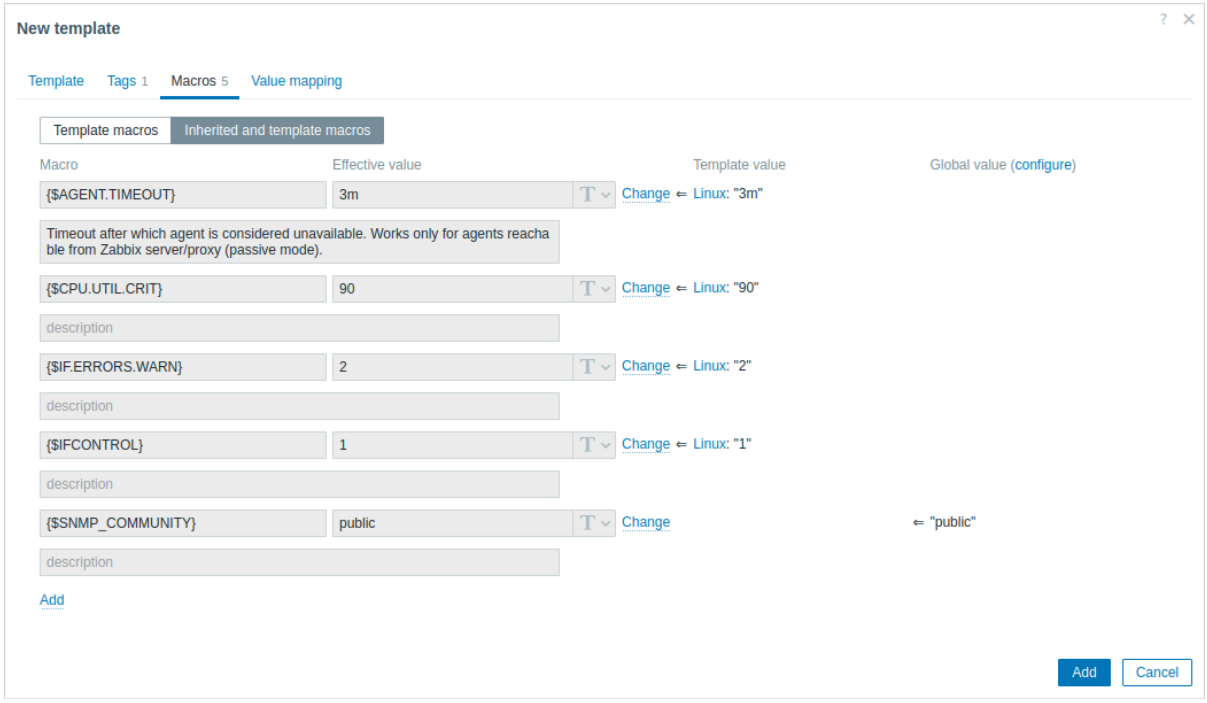
Per comodità, vengono forniti i collegamenti ai template corrispondenti.

Nei tag sono supportati le macro utente, le macro {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID}.

La scheda **Macro** consente di definire le macro utente a livello di template come coppie nome-valore. I valori delle macro possono essere memorizzati come testo normale, testo segreto o segreto Vault. È inoltre possibile aggiungere una descrizione.



Se si seleziona l'opzione *Macro ereditate e del template*, è inoltre possibile visualizzare le macro dei template collegati e le macro globali che il template erediterà, nonché i valori a cui verranno risolte le macro.



Per comodità, vengono forniti i collegamenti ai template corrispondenti e un collegamento alla configurazione delle macro globali. È inoltre possibile modificare una macro del template collegato o una macro globale a livello di template, creando di fatto una copia della macro nel template.

La scheda **Mappatura dei valori** consente di configurare una rappresentazione intuitiva dei dati degli item nelle mappature dei valori.

Pulsanti:

Add	Aggiunge il template. Il template aggiunto dovrebbe essere visualizzato nell'elenco.
Update	Aggiorna le proprietà di un template esistente.
Clone	Crea un altro template basato sulle proprietà del template corrente. Questo **include** le entità (item, trigger e così via) ereditate dai template collegati e direttamente associate al template corrente, ma **esclude** il vendor e la versione del template corrente, in modo che il template clonato sia distinguibile dall'originale.
Delete	Elimina il template; le entità del template (item, trigger e così via) rimangono associate agli host collegati.
Delete and clear	Elimina il template e tutte le relative entità dagli host collegati.
Cancel	Annulla la modifica delle proprietà del template.

Aggiunta di item, trigger, grafici

> **Attention:**
> Gli item devono essere aggiunti prima a un template. Trigger e grafici non possono essere aggiunti senza il corrispondente item.

È possibile aggiungere item esistenti a un template oppure creare nuovi item.

Per aggiungere item al template:

1. Vai a *Raccolta dati* > *Host* (o *Template*).
2. Fai clic su *Item* nella riga dell'host/template richiesto.
3. Seleziona le caselle di controllo degli item che desideri aggiungere al template.
4. Fai clic su *Copia* sotto l'elenco degli item.
5. Seleziona il template (o il gruppo di template) in cui gli item devono essere copiati e fai clic su *Copia*.
 Tutti gli item selezionati devono essere copiati nel template.

L'aggiunta di trigger e grafici avviene in modo simile (rispettivamente dall'elenco dei trigger e dei grafici), ricordando ancora una volta che possono essere aggiunti solo se gli item richiesti sono stati aggiunti per primi.

Aggiunta di dashboard

Per aggiungere dashboard a un template in *Data collection* > *Templates*, procedi come segue:

1. Fai clic su *Dashboards* nella riga del template.
2. Configura una dashboard seguendo le linee guida per la configurazione delle dashboard.

Attention:
Quando configuri i widget su una dashboard del template (anziché su una dashboard globale), i parametri relativi all'host non sono disponibili e alcuni parametri hanno un'etichetta diversa. Questo perché le dashboard del template visualizzano i dati solo dall'host a cui il template è collegato. Ad esempio, i parametri *Host groups*, *Exclude host groups* e *Hosts* nel widget *Problems* non sono disponibili, il parametro *Host groups* nel widget *Host availability* non è disponibile e il parametro *Show hosts in maintenance* viene rinominato in *Show data in maintenance*, ecc. Per ulteriori informazioni sulla disponibilità dei parametri nei widget delle dashboard del template, consulta i parametri specifici di ciascun widget della dashboard.

Note:
Per i dettagli sull'accesso alle dashboard dell'host create a partire dalle dashboard del template, consulta la sezione dashboard dell'host.

Configurazione delle regole di low-level discovery

Vedere la sezione low-level discovery del manuale.

Aggiunta di web scenario

Per aggiungere web scenario a un template in *Data collection* > *Templates*, procedere come segue:

1. Fare clic su *Web* nella riga del template.
2. Configurare un web scenario seguendo il metodo abituale di configurazione dei web scenario.

Configurazione di un template group

Panoramica

I template group vengono utilizzati per il raggruppamento logico dei template e per l'assegnazione dei permessi utente su di essi.

Ogni template deve avere assegnato almeno un template group. Un template può appartenere a più template group e ogni template group può contenere più template.

Si noti che in Zabbix tutti i permessi si basano sui gruppi: user groups, host groups e template group. Quindi, anche se un singolo utente necessita dell'accesso a un singolo template, questo viene concesso aggiungendo l'utente a un user group e assegnando a quel user group i permessi sul template group che contiene quel template.

Configurazione

Attention:
Solo gli utenti Super admin possono creare gruppi di template.

Ci sono due opzioni per creare un gruppo di template nel frontend di Zabbix.

Opzione uno:

1. Vai a *Data collection* > *Template groups*.
2. Fai clic su *Create template group* nell'angolo in alto a destra della schermata.
3. Inserisci il nome del gruppo nel modulo.



Opzione due: quando configuri un template, inserisci un nome di gruppo inesistente nel campo di input *Template groups*.

Una volta creato il gruppo di template, puoi fare clic sul nome del template nell'elenco sotto *Data collection* > *Template groups* per modificare il nome del gruppo, clonare il gruppo o eliminarlo.

L'eliminazione di un gruppo di template rimuove solo il gruppo logico, non i template nel gruppo. Non è possibile eliminare un gruppo di template che sia l'unico gruppo per un qualsiasi template esistente.

Creazione di sottogruppi di template

Un sottogruppo di template (o gruppo di template nidificato) è un elemento figlio del gruppo di template padre che lo contiene.

Un sottogruppo viene creato utilizzando la barra obliqua '/' nel campo di input del nome del gruppo per indicare la sua relazione con il gruppo o i gruppi padre. Ad esempio:

- inserendo `Linux servers/Databases` si crea il sottogruppo `Linux servers/Databases` del gruppo padre `Linux servers`.
- inserendo `Linux servers/Databases/MySQL/Tokyo` si crea il rispettivo sottogruppo all'interno dei gruppi padre nidificati `Linux servers`, `Linux servers/Databases`, `Linux servers/Databases/MySQL`.

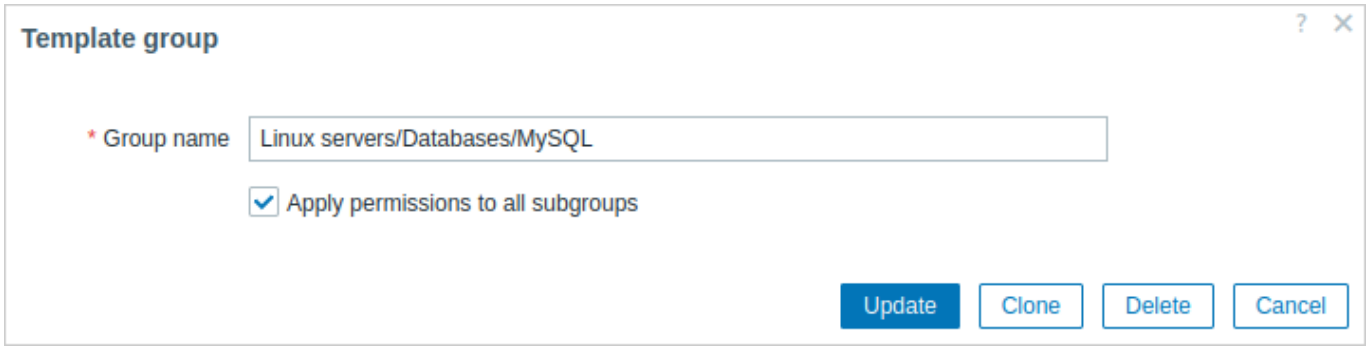
Quando si crea un sottogruppo, non è consentito utilizzare barre oblique iniziali o finali, né più barre oblique consecutive. L'escape di '/' non è supportato.

Non è necessario creare alcun gruppo di template padre prima di creare un sottogruppo. È possibile scegliere se iniziare creando un sottogruppo (ad esempio, `Linux servers/Databases`) oppure uno qualsiasi dei gruppi di template padre (nel nostro esempio, `Linux servers`). Se si inizia creando un sottogruppo, i gruppi di template padre **non** verranno creati automaticamente.

Permessi per i template group

- Quando si crea un sottogruppo di un template group padre esistente (ad esempio, creando `Linux servers/Databases` quando `Linux servers` esiste già), i permessi del user group per il sottogruppo vengono ereditati dal padre.
- Quando si crea un template group padre per un sottogruppo esistente (ad esempio, creando `Linux servers` quando `Linux servers/Databases` esiste già), non vengono impostati permessi per il padre.

Quando si modifica qualsiasi template group, è possibile impostare anche un'opzione aggiuntiva, *Applica i permessi a tutti i sottogruppi*.



Selezionando questa casella e facendo clic su *Update* si applicherà lo stesso livello di permessi a tutti i sottogruppi attuali e futuri del template group in modifica.

Quindi, se a uno qualsiasi dei user group sono stati assegnati permessi diversi per i sottogruppi del template group in modifica, selezionando la casella tutti i sottogruppi attuali e futuri riceveranno gli stessi permessi utente del gruppo in modifica.

Si noti che questa opzione non viene salvata nel database e sovrascriverà i permessi esistenti. Eventuali modifiche apportate tramite questa opzione possono essere annullate solo manualmente.

Collegamento/annullamento del collegamento

Panoramica

Il collegamento è un processo mediante il quale i template vengono applicati agli host, mentre lo scollegamento rimuove l'associazione con il template da un host.

Collegamento di un template

Per collegare un template al host, procedere come segue:

1. Andare a *Raccolta dati* > *Host*.
2. Fare clic sull'host richiesto.
3. Iniziare a digitare il nome del template nel campo *Template*. Verrà visualizzato un elenco di template corrispondenti; scorrere l'elenco per selezionare quello desiderato. In alternativa, fare clic su *Seleziona* accanto al campo e selezionare uno o più template dall'elenco in una finestra popup.
4. Fare clic su *Aggiungi/Aggiorna* nel modulo degli attributi dell'host.

L'host avrà ora tutte le entità del template. Sono inclusi item, trigger, grafici, regole di discovery di basso livello, scenari web e dashboard.

Attention:
Il collegamento di più template allo stesso host non riuscirà se tali template contengono item con la stessa chiave item. Inoltre, poiché trigger e grafici utilizzano gli item, non possono essere collegati a un singolo host da più template se utilizzano chiavi item identiche.

Quando le entità (item, trigger e così via) vengono aggiunte dal template:

- le entità identiche già esistenti sull'host vengono aggiornate come entità del template e **tutte le personalizzazioni esistenti a livello di host per l'entità vengono perse**;
- le entità del template vengono aggiunte;
- tutte le entità collegate direttamente che, prima del collegamento del template, esistevano solo sull'host rimangono invariate.

Negli elenchi, tutte le entità del template sono ora precedute dal nome del template, a indicare che appartengono a quel template specifico. Il nome del template stesso (visualizzato in grigio) è un collegamento che consente di accedere all'elenco di tali entità a livello di template.

Note:
Per alcuni item, come controlli esterni, controlli con agent HTTP, controlli semplici, controlli SSH e controlli Telnet, un'interfaccia host è facoltativa. Se, al momento del collegamento di un template, l'host non dispone di un'interfaccia definita, questi item verranno aggiunti senza un'interfaccia. Se si aggiunge successivamente un'interfaccia host, questa non verrà assegnata automaticamente agli item già esistenti. Per assegnare contemporaneamente la nuova interfaccia host a tutti gli item del template, scollegare il template dall'host e quindi collegarlo nuovamente. Per conservare la cronologia degli item, utilizzare l'opzione *Scollega*; non utilizzare *Scollega e cancella*.

Se un'entità non è preceduta dal nome del template, significa che esisteva sull'host prima del collegamento e non è stata aggiunta dal template.

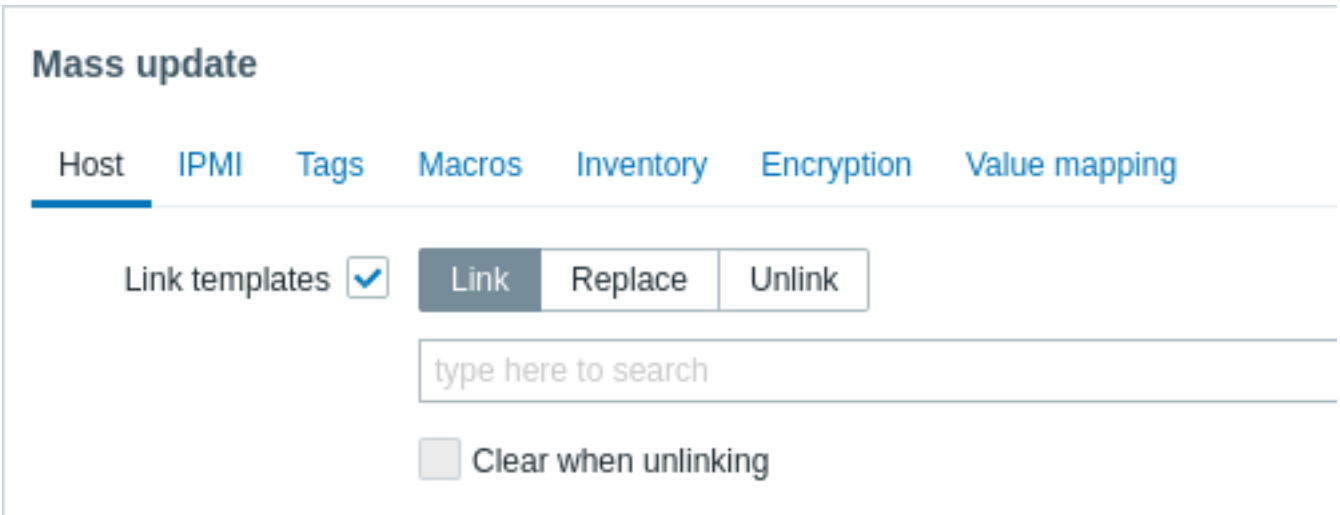
Criteri di unicità delle entità

Quando si aggiungono entità (item, trigger, ecc.) da un template è importante sapere quali di queste entità esistono già sull'host e devono essere aggiornate e quali entità sono diverse. I criteri di unicità per stabilire l'uguaglianza/la differenza sono:

- per gli item - la chiave dell'item;
- per i trigger - il nome del trigger e l'espressione;
- per i grafici personalizzati - il nome del grafico e i relativi item.

Collegamento di template a più host

Per aggiornare il collegamento dei template di molti host, in *Raccolta dati* > *Host* selezionare alcuni host contrassegnando le relative caselle di controllo, quindi fare clic su **Aggiornamento di massa** sotto l'elenco e selezionare *Collega template*:



Per collegare altri template, iniziare a digitare il nome del template nel campo di completamento automatico finché non viene visualizzato un elenco a discesa con i template corrispondenti. Scorrere quindi verso il basso per selezionare il template da collegare.

L'opzione *Sostituisci* consente di collegare un nuovo template scollegando qualsiasi template precedentemente collegato agli host. L'opzione *Scollega* consente di specificare quali template scollegare. L'opzione *Cancella durante lo scollegamento* consente non solo di scollegare i template precedentemente collegati, ma anche di rimuovere tutte le entità ereditate da essi (item, trigger, ecc.).

Note:
Zabbix offre un'ampia serie di template predefiniti. È possibile utilizzarli come riferimento, ma è consigliabile non usarli senza modifiche in produzione, poiché potrebbero contenere troppi item e raccogliere dati con una frequenza eccessiva. Se si desidera utilizzarli, adattarli alle proprie esigenze effettive.

Modifica delle entità collegate

Se provi a modificare un item o un trigger collegato dal template, potresti notare che molte opzioni chiave sono disabilitate per la modifica. Questo è logico, poiché l'idea dei template è che gli elementi vengano modificati in modo centralizzato a livello di template. Tuttavia, puoi comunque, ad esempio, abilitare/disabilitare un item su singoli host e impostare l'intervallo di aggiornamento, la durata dello storico e alcuni altri parametri.

Se vuoi modificare completamente l'entità, devi modificarla a livello di template (la scorciatoia al livello di template è visualizzata nel nome del modulo), tenendo presente che queste modifiche influenzeranno tutti gli host a cui questo template è collegato.

Attention:
Qualsiasi personalizzazione delle entità effettuata a livello di template sovrascriverà le precedenti personalizzazioni delle entità a livello di host.

Scollegare un template

Per scollegare un template da un host, procedere come segue:

1. Andare in *Data collection* > *Hosts*.
2. Fare clic sull'host richiesto e individuare il campo *Templates*.
3. Fare clic su *Unlink* o *Unlink and clear* accanto al template per scollegare il template.
4. Fare clic su *Update* nel modulo degli attributi dell'host.

La scelta dell'opzione *Unlink* rimuoverà semplicemente l'associazione con il template, lasciando però tutte le sue entità sull'host. Questo include item, trigger, grafici, regole di discovery a basso livello e scenari web, ma esclude i dashboard. Si noti che verranno rimossi anche i value map e i tag ereditati dai template collegati.

La scelta dell'opzione *Unlink and clear* rimuoverà sia l'associazione con il template sia tutte le sue entità (item, trigger, ecc.).

Annidamento

Panoramica

L'annidamento è un modo in cui un template comprende uno o più altri template.

Poiché ha senso separare le entità in singoli template per vari servizi, applicazioni, ecc., potresti ritrovarti con parecchi template, tutti potenzialmente da collegare a numerosi host. Per semplificare il quadro, è possibile collegare alcuni template tra loro in un unico template.

Il vantaggio dell'annidamento è che devi collegare un solo template all'host, e l'host erediterà automaticamente tutte le entità dai template collegati a quel template. Ad esempio, se colleghiamo *T1* e *T2* a *T3*, integriamo *T3* con tutte le entità di *T1* e *T2*, ma non viceversa. Se colleghiamo *T1* a *T2* e *T3*, integriamo *T2* e *T3* con le entità di *T1*.

Configurazione di template nidificati

Per collegare dei template, è necessario prendere un template esistente (o crearne uno nuovo), quindi:

1. Aprire il modulo di configurazione del template.
2. Trovare il campo *Templates*.
3. Fare clic su *Select* per aprire la finestra popup *Templates*.
4. Nella finestra popup, scegliere i template richiesti, quindi fare clic su *Select* per aggiungere i template all'elenco.
5. Fare clic su *Add* o *Update* nel modulo di configurazione del template.

In questo modo, tutte le entità del template configurato, così come tutte le entità dei template collegati, appariranno ora nella configurazione del template. Questo include item, trigger, grafici, regole di discovery a basso livello e scenari web, ma esclude i dashboard. Tuttavia, i dashboard dei template collegati verranno comunque ereditati dagli host.

Per scollegare uno qualsiasi dei template collegati, fare clic su *Unlink* o *Unlink and clear* nel modulo di configurazione del template, quindi fare clic su *Update*.

L'opzione *Unlink* rimuoverà semplicemente l'associazione con il template collegato, senza rimuovere nessuna delle sue entità (item, trigger, ecc.).

L'opzione *Unlink and clear* rimuoverà sia l'associazione con il template collegato, sia tutte le sue entità (item, trigger, ecc.).

Aggiornamento di massa

Panoramica

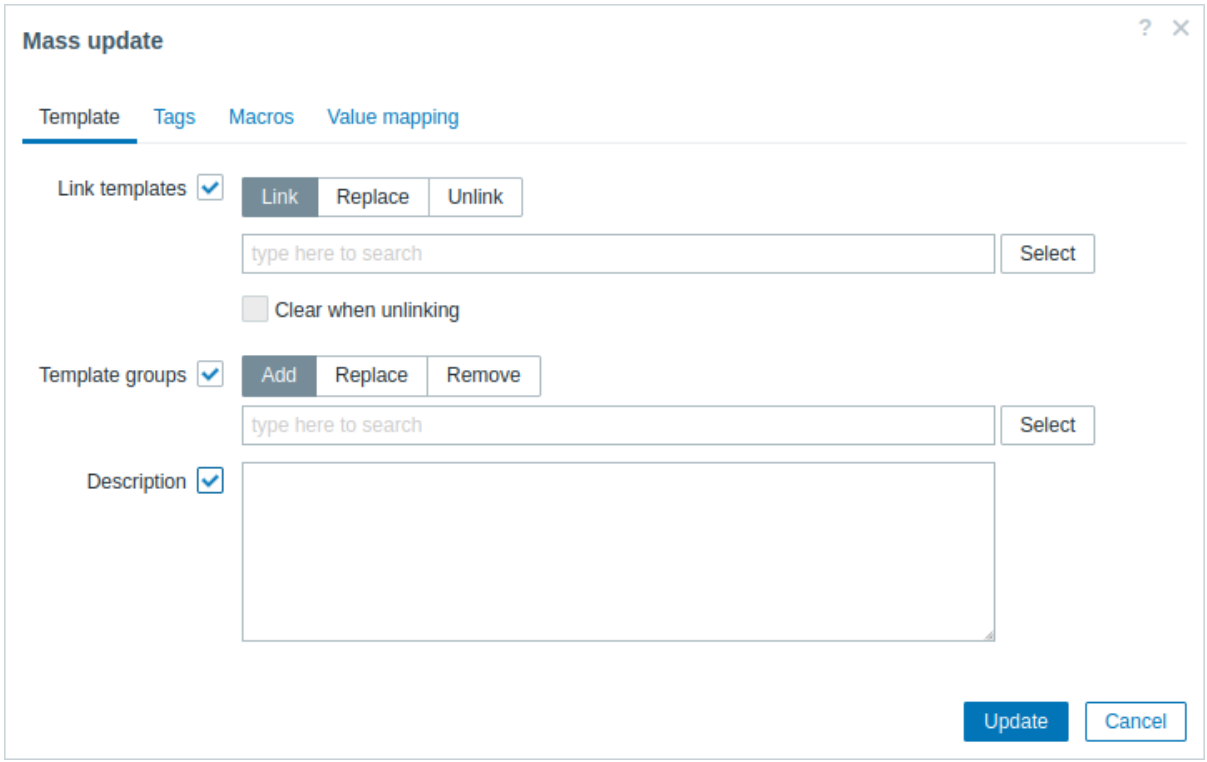
A volte può essere necessario modificare contemporaneamente un attributo per più template. Invece di aprire ogni singolo template per la modifica, è possibile utilizzare la funzione di aggiornamento di massa.

Utilizzo dell'aggiornamento di massa

Per aggiornare in massa alcuni template, procedere come segue:

1. Selezionare le caselle di controllo accanto ai template che si desidera aggiornare nell'elenco dei template.
2. Fare clic su *Aggiornamento di massa* sotto l'elenco.
3. Passare alla scheda con gli attributi richiesti (*Template*, *Tags*, *Macros* o *Value mapping*).
4. Selezionare le caselle di controllo degli attributi da aggiornare e immettere un nuovo valore per essi.

La scheda **Template** contiene le opzioni generali per l'aggiornamento di massa dei template.



Quando si seleziona il pulsante corrispondente per l'aggiornamento *Collega template*, sono disponibili le seguenti opzioni:

- *Collega* - specificare quali template aggiuntivi collegare;
- *Sostituisci* - specificare quali template collegare scollegando contemporaneamente tutti i template collegati in precedenza;
- *Scollega* - specificare quali template scollegare.

Per specificare i template da collegare/scollegare, iniziare a digitare il nome del template nel campo di completamento automatico finché non viene visualizzato un elenco a discesa con i template corrispondenti. Scorrere l'elenco verso il basso per selezionare i template richiesti.

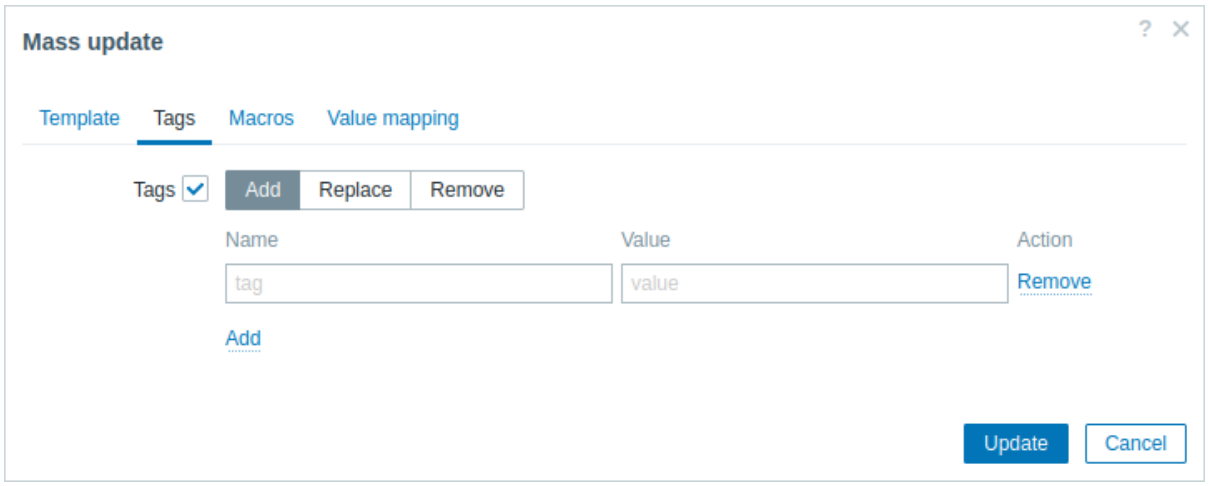
L'opzione *Cancella durante lo scollegamento* consente non solo di scollegare i template collegati in precedenza, ma anche di rimuovere tutte le entità ereditate da essi (item, trigger, grafici, ecc.).

Quando si seleziona il pulsante corrispondente per l'aggiornamento dei *Gruppi di template*, sono disponibili le seguenti opzioni:

- *Aggiungi* - specificare gruppi di template aggiuntivi tra quelli esistenti oppure inserire gruppi di template completamente nuovi per i template;
- *Sostituisci* - rimuovere il template da tutti i gruppi di template esistenti e sostituirli con quelli specificati in questo campo (gruppi di template esistenti o nuovi);
- *Rimuovi* - rimuovere gruppi di template specifici dai template.

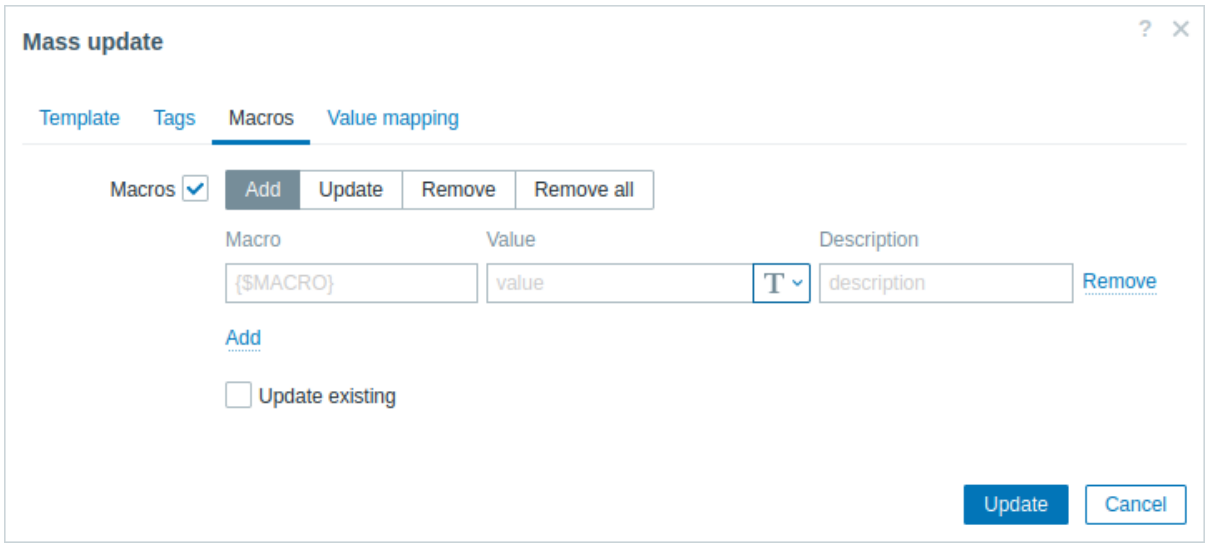
Questi campi supportano il completamento automatico: iniziando a digitare al loro interno viene visualizzato un elenco a discesa con i gruppi di template corrispondenti. Se il gruppo di template è nuovo, viene visualizzato anch'esso nell'elenco a discesa ed è indicato da *(nuovo)* dopo la stringa. Scorrere l'elenco verso il basso per selezionarlo.

La scheda **Tag** consente di aggiornare in massa i tag a livello di template.



Le macro utente, le macro {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} sono supportate nei tag. Si noti che i tag con lo stesso nome, ma valori diversi, non sono considerati "duplicati" e possono essere aggiunti allo stesso template.

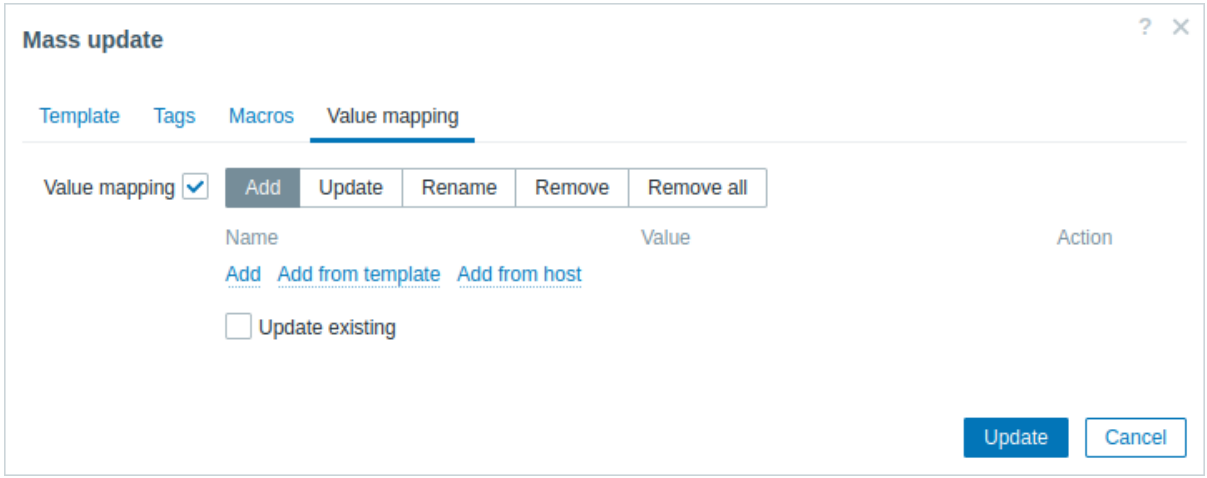
La scheda **Macro** consente di aggiornare in massa le macro a livello di template.



Quando si seleziona il pulsante corrispondente per l'aggiornamento delle macro, sono disponibili le seguenti opzioni:

- *Aggiungi* - specificare macro utente aggiuntive per i template. Se la casella di controllo *Aggiorna esistenti* è selezionata, verranno aggiornati il valore, il tipo e la descrizione per il nome della macro specificato. Se non è selezionata e nei template è già presente una macro con quel nome, questa non verrà aggiornata.
- *Aggiorna* - sostituire i valori, i tipi e le descrizioni delle macro specificate in questo elenco. Se la casella di controllo *Aggiungi mancanti* è selezionata, la macro che in precedenza non esisteva in un template verrà aggiunta come nuova macro. Se non è selezionata, verranno aggiornate solo le macro già presenti in un template.
- *Rimuovi* - rimuovere dai template le macro specificate. Se la casella di controllo *Eccetto quelle selezionate* è selezionata, verranno rimosse tutte le macro tranne quelle specificate nell'elenco. Se non è selezionata, verranno rimosse solo le macro specificate nell'elenco.
- *Rimuovi tutte* - rimuovere tutte le macro utente dai template. Se la casella di controllo *Confermo di voler rimuovere tutte le macro* non è selezionata, si aprirà una nuova finestra popup che chiederà di confermare la rimozione di tutte le macro.

La scheda **Mappatura dei valori** consente di aggiornare in massa le mappature dei valori.



Per aggiornare la mappa dei valori sono disponibili pulsanti con le seguenti opzioni:

- *Aggiungi* - aggiunge le mappe dei valori ai template. Se si seleziona *Aggiorna esistenti*, verranno aggiornate tutte le proprietà della mappa dei valori con questo nome. In caso contrario, se una mappa dei valori con questo nome esiste già, non verrà aggiornata.
- *Aggiorna* - aggiorna le mappe dei valori esistenti. Se si seleziona *Aggiungi mancanti*, una mappa dei valori che non esisteva precedentemente in un template verrà aggiunta come nuova mappa dei valori. In caso contrario, verranno aggiornate solo le mappe dei valori già esistenti in un template.
- *Rinomina* - assegna un nuovo nome a una mappa dei valori esistente.
- *Rimuovi* - rimuove le mappe dei valori specificate dai template. Se si seleziona *Tranne quelle selezionate*, verranno rimosse tutte le mappe dei valori **tranne** quelle specificate.
- *Rimuovi tutto* - rimuove tutte le mappe dei valori dai template. Se la casella di controllo *Confermo di voler rimuovere tutte le mappe dei valori* non è selezionata, si aprirà una nuova finestra popup per chiedere conferma della rimozione.

Le opzioni *Aggiungi dal template* e *Aggiungi dal host* sono disponibili per le operazioni di aggiunta/aggiornamento delle mappe dei valori.
Consentono di selezionare le mappature dei valori rispettivamente da un template o da un host.

Quando hai completato tutte le modifiche richieste, fai clic su *Update*. Gli attributi verranno aggiornati di conseguenza per tutti i template selezionati.

Host e gruppi di host

Che cos'è un "host"?

In Zabbix, un "host" si riferisce a qualsiasi dispositivo fisico o virtuale, applicazione, servizio o qualsiasi altra raccolta logicamente correlata di parametri monitorati.

La creazione degli host è una delle prime attività di monitoraggio in Zabbix. Ad esempio, se si desidera monitorare alcuni parametri su un server "x", è necessario prima creare un host chiamato, ad esempio, "Server X" e quindi sarà possibile aggiungervi item di monitoraggio.

Gli host sono organizzati in gruppi di host.

Procedere a:

- Host Wizard per un'interfaccia guidata, passo dopo passo, per creare e configurare un host nuovo o esistente.
- Configurazione di un host per un approccio classico alla creazione e configurazione di un host.

Procedura guidata host

Panoramica

Host Wizard è un'interfaccia guidata, passo dopo passo, per configurare il target di monitoraggio (dispositivo, applicazione, servizio, ecc.) in Zabbix. Ti accompagna attraverso i seguenti passaggi:

- Selezione di un template
- Creazione o selezione di un host
- Installazione di Zabbix agent o agent 2
- Aggiunta di un'interfaccia host

- Applicazione di configurazioni aggiuntive al target di monitoraggio o all'host Zabbix (se richiesto dal template)

Per accedere a Host Wizard nel frontend di Zabbix, procedi come segue:

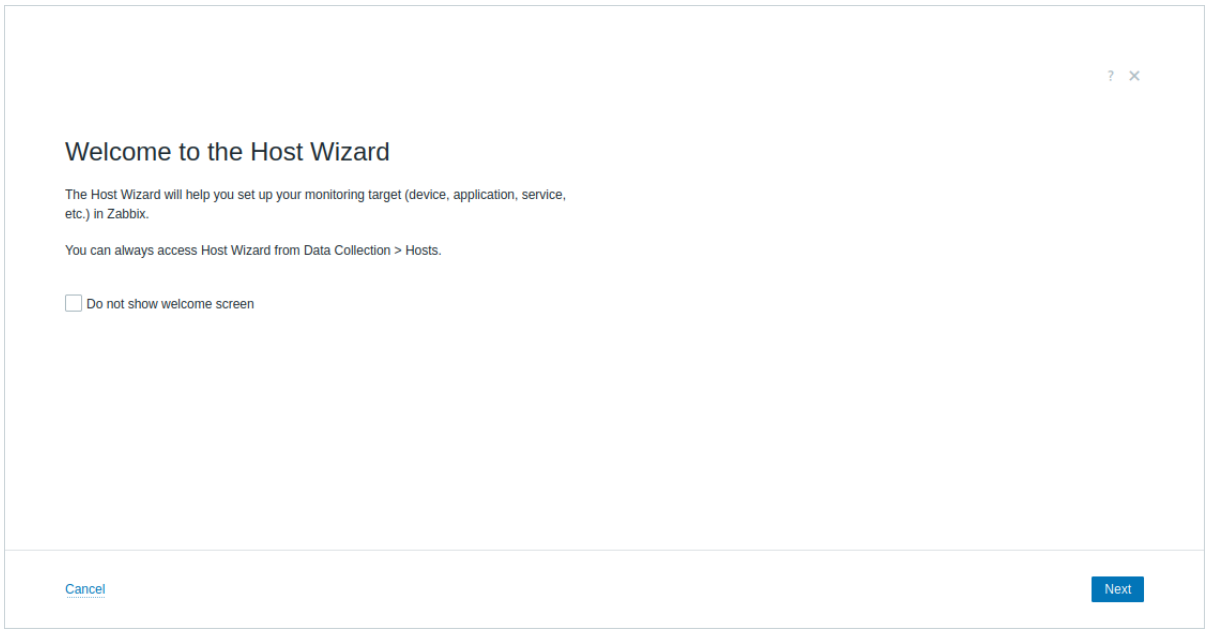
1. Vai a: *Data collection* > *Hosts*.
2. Fai clic su *Host Wizard* nell'angolo in alto a destra della schermata.
3. Segui le istruzioni nella procedura guidata.

Puoi anche modificare un host esistente facendo clic sul suo nome (o sull'icona con i tre puntini accanto ad esso) in varie sezioni del frontend per accedere a Host Wizard dal menu host. Nota che alcuni passaggi vengono saltati quando si modifica un host esistente se questo include già la configurazione richiesta dal template selezionato.

In alternativa, puoi usare l'approccio classico per creare e configurare un host.

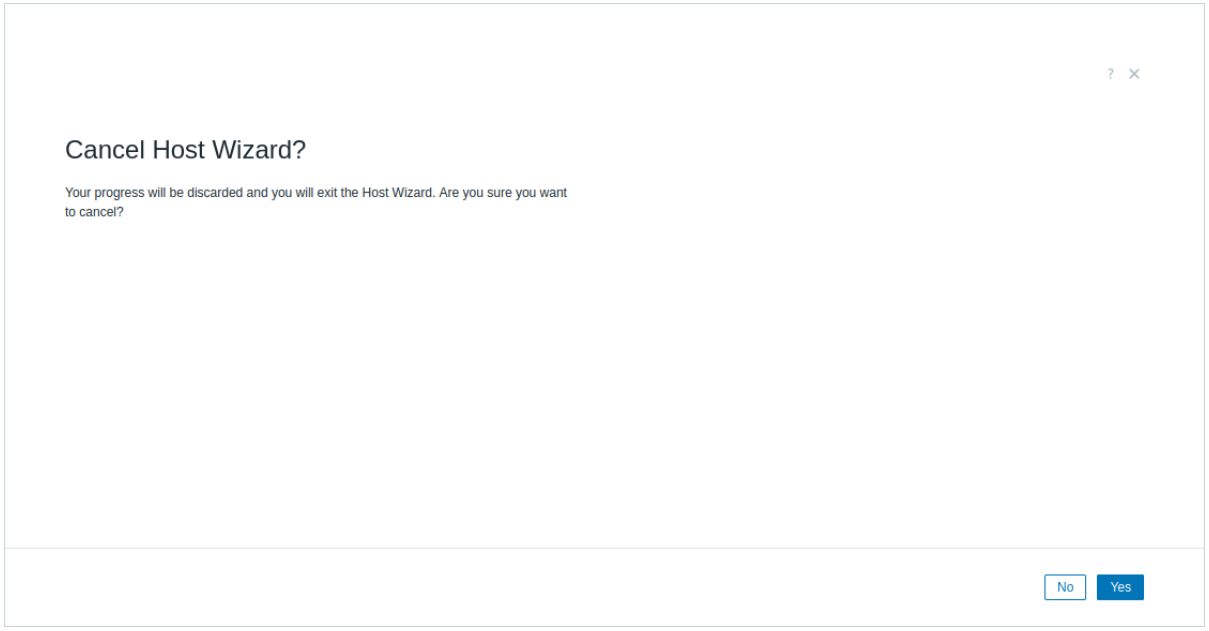
Benvenuti nella procedura guidata Host

Quando si avvia la procedura guidata Host, viene visualizzata una schermata di benvenuto:



Per saltare questa schermata nelle sessioni future, è possibile selezionare la casella di controllo *Non mostrare la schermata di benvenuto* e fare clic su *Avanti*.

Se si inizia a configurare le impostazioni nei passaggi successivi della procedura guidata Host e si tenta di chiudere la procedura guidata facendo clic sull'icona di chiusura o premendo Esc, verrà visualizzata una schermata di conferma:



Facendo clic su *Sì* si uscirà dalla procedura guidata Host senza salvare le modifiche. Facendo clic su *No*, Esc o sull'icona di chiusura si tornerà all'ultimo passaggio.

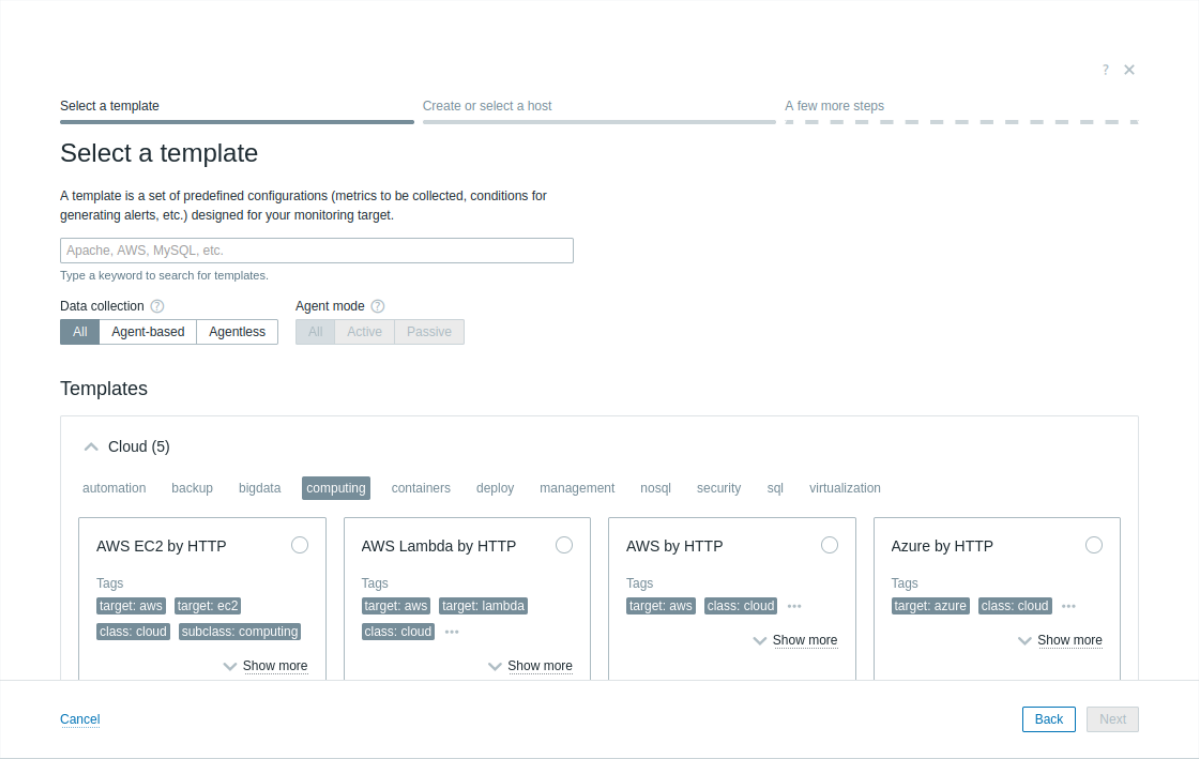
Seleziona un template

Il primo passaggio per configurare l'oggetto del monitoraggio consiste nel selezionare un template, ovvero un insieme di configurazioni predefinite (metriche da raccogliere, condizioni per la generazione degli avvisi e così via) progettate per l'oggetto del monitoraggio.

I template pronti all'uso di Zabbix offrono una varietà di configurazioni di monitoraggio predefinite per sistemi operativi, applicazioni, database, dispositivi di rete, servizi e altro.

In questo passaggio è possibile:

- Sfogliare i template per classe (Cloud, Database, Rete e così via) e sottoclasse (ad esempio, automazione, backup), basate sui tag dei template.
- Cercare i template utilizzando parole chiave presenti nel nome del template o nei nomi e nei valori dei tag del template.
- Filtrare i template in base al metodo di raccolta dei dati (basato su agent o senza agent; i template basati su agent includono almeno un item Zabbix agent).
- Filtrare i template in base alla modalità dell'agent (attiva o passiva; i template attivi includono almeno un item Zabbix agent (active), mentre i template passivi includono almeno un item passivo; per i dettagli, vedere Controlli dell'agent passivi e attivi).
- Quando si utilizza la procedura guidata Host per configurare un host esistente, è inoltre possibile filtrare i template già collegati all'host.



Attention:
Quando si seleziona un template, potrebbe essere visualizzato il seguente messaggio: *Alcuni template (n) non sono compatibili con la procedura guidata Host. Vedere come aggiornarli. I template personalizzati non sono supportati.* Questo indica che alcuni template pronti all'uso di Zabbix non sono ancora compatibili con la procedura guidata Host e devono essere aggiornati; vedere le istruzioni per l'aggiornamento dei template.

Il template selezionato determinerà i passaggi successivi nella procedura guidata Host. Ad esempio, se si seleziona il template *MySQL by Zabbix agent*, che utilizza Zabbix agent per la raccolta dei dati, la procedura guidata guiderà l'utente attraverso il processo di installazione dell'agent.

Creare o selezionare un host

Un template deve essere collegato a un host, ovvero un'entità in Zabbix che rappresenta l'oggetto del monitoraggio. Una volta collegato, l'host riceve tutte le entità del template, come item (metriche da raccogliere) e trigger (condizioni per la generazione di avvisi), necessarie per il monitoraggio.

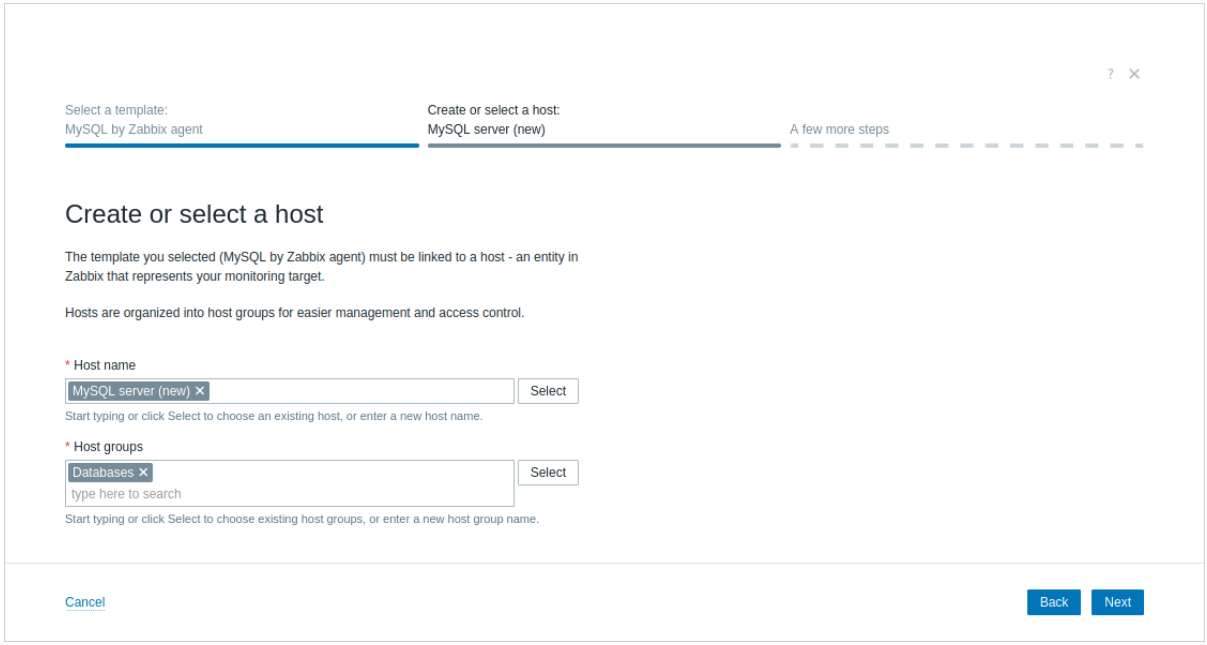
Ogni host deve inoltre appartenere ad almeno un gruppo di host, utilizzato per organizzare gli host e assegnare loro le autorizzazioni degli utenti.

In questo passaggio è possibile:

- Creare un nuovo host e un gruppo di host inserendo i nomi di entrambi.
- Creare un nuovo host inserendo il relativo nome e selezionare un gruppo di host esistente a cui assegnare il nuovo host.
- Selezionare un host esistente; è inoltre possibile assegnare questo host a gruppi di host aggiuntivi senza rimuoverlo dai gruppi attuali.

Note:
La selezione di un host esistente non rimuove né sovrascrive la configurazione attuale, salvo indicazione esplicita nei passaggi successivi. Gli host rilevati non possono essere selezionati.

Ad esempio, è possibile creare un nuovo host denominato *MySQL server* per rappresentare il server MySQL installato localmente che si desidera monitorare. Il template *MySQL by Zabbix agent* selezionato in precedenza verrà collegato a questo host. Inoltre, è possibile selezionare il gruppo di host esistente *Databases* (oppure crearne uno nuovo, ad esempio *MySQL servers*) per organizzare l'host insieme ad altri host correlati ai database e semplificare in seguito la gestione delle autorizzazioni.



Installare Zabbix agent

I template basati su agent (come *MySQL by Zabbix agent*) richiedono l'installazione di Zabbix agent o agent 2, ovvero un processo distribuito sul target di monitoraggio per monitorare attivamente le risorse e le applicazioni locali (se richiesto dal template).

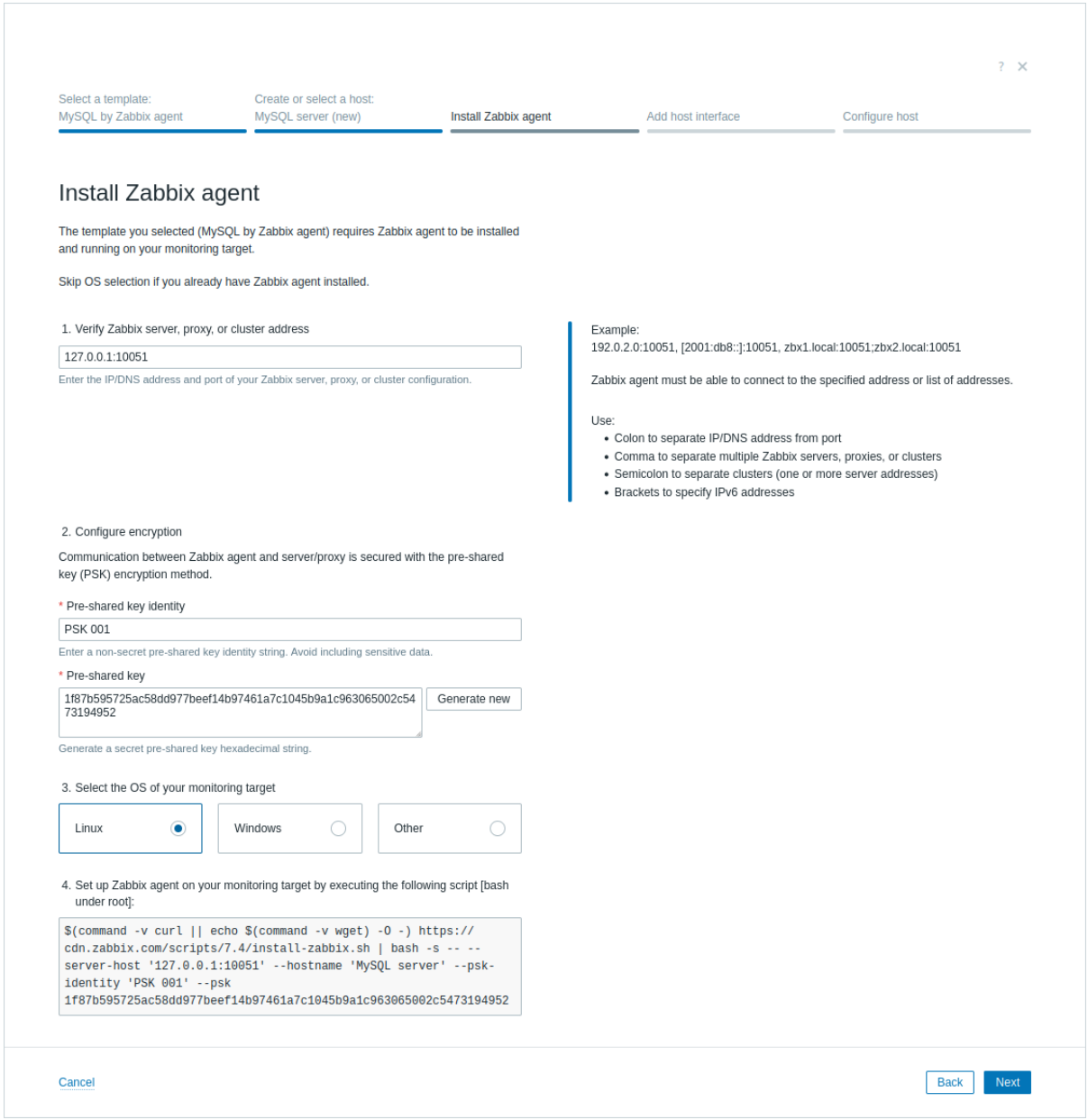
In questo passaggio:

- Verificare l'indirizzo del server Zabbix, del proxy o del cluster (ad esempio, 192.0.2.1:10051). Questo valore verrà utilizzato per configurare i parametri Server e ServerActive per Zabbix agent.
- Immettere una nuova identità univoca e non segreta della chiave precondivisa (ad esempio, *PSK 001* o *mysql-agent-psk1*) e generare la chiave precondivisa. Questi valori verranno utilizzati per configurare la crittografia con chiave precondivisa (PSK) per l'host Zabbix (ad esempio, *MySQL server*) e Zabbix agent.

Attention:
Quando si modifica un host esistente, la configurazione PSK sovrascriverà tutte le impostazioni di crittografia esistenti sull'host.

- Selezionare il sistema operativo della macchina che ospita il target di monitoraggio.
- Installare Zabbix agent su tale sistema eseguendo lo script fornito oppure seguendo le istruzioni di installazione.

Ad esempio, è possibile impostare *Identità della chiave precondivisa* su *PSK 001* e generare una nuova *Chiave precondivisa*. Quindi, è possibile selezionare *Linux* come sistema operativo ed eseguire lo script fornito su tale sistema. Dopo aver installato Zabbix agent, tornare alla procedura guidata Host.

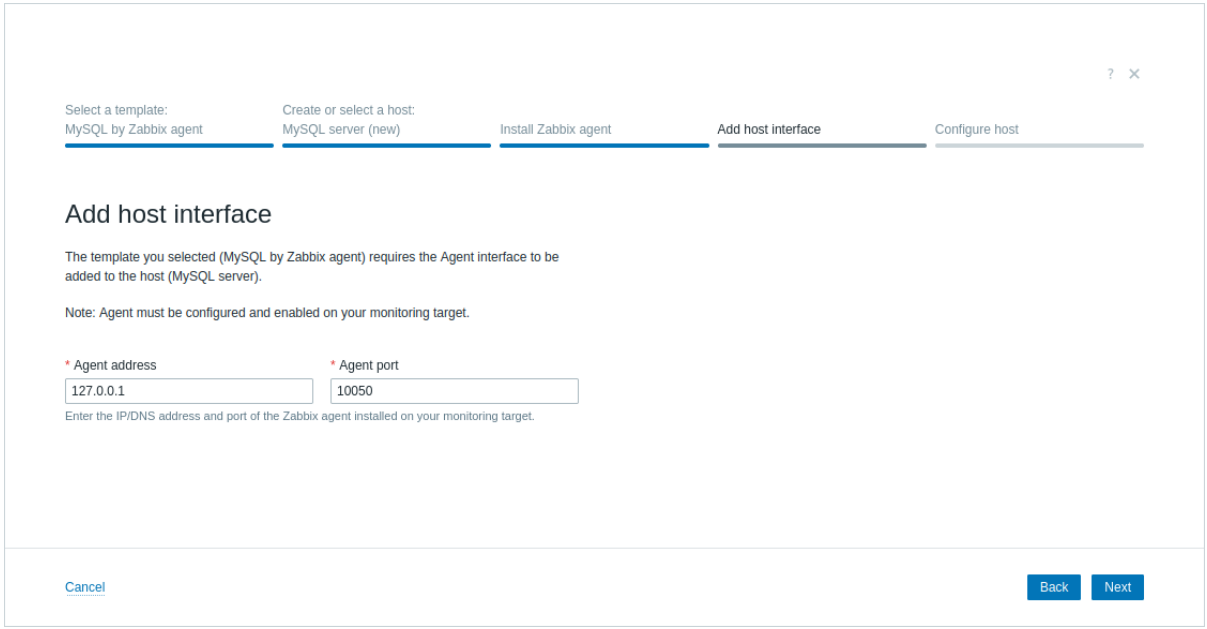


Aggiungi interfaccia host

La host interface definisce come Zabbix server si connette al target di monitoraggio tramite la rete. Definisce parametri di connessione come indirizzo IP, nome DNS, porta e tipo di interfaccia (Agent, SNMP, JMX o IPMI), a seconda del metodo di raccolta dati richiesto dal template selezionato.

In questo passaggio, inserisci i dettagli dell'host interface richiesti dal template selezionato.

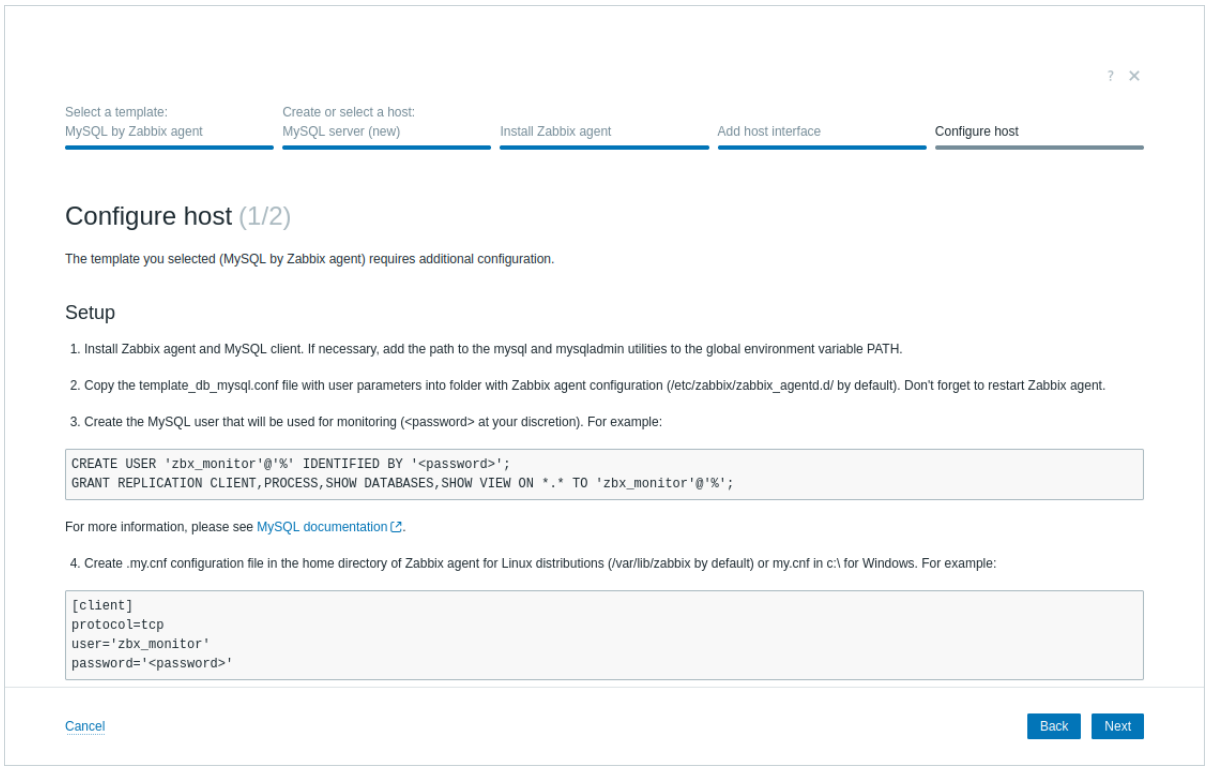
Ad esempio, puoi usare l'indirizzo agent predefinito (`127.0.0.1`) e la porta (10050) se Zabbix server, Zabbix agent e il server MySQL sono tutti in esecuzione localmente sulla stessa macchina.



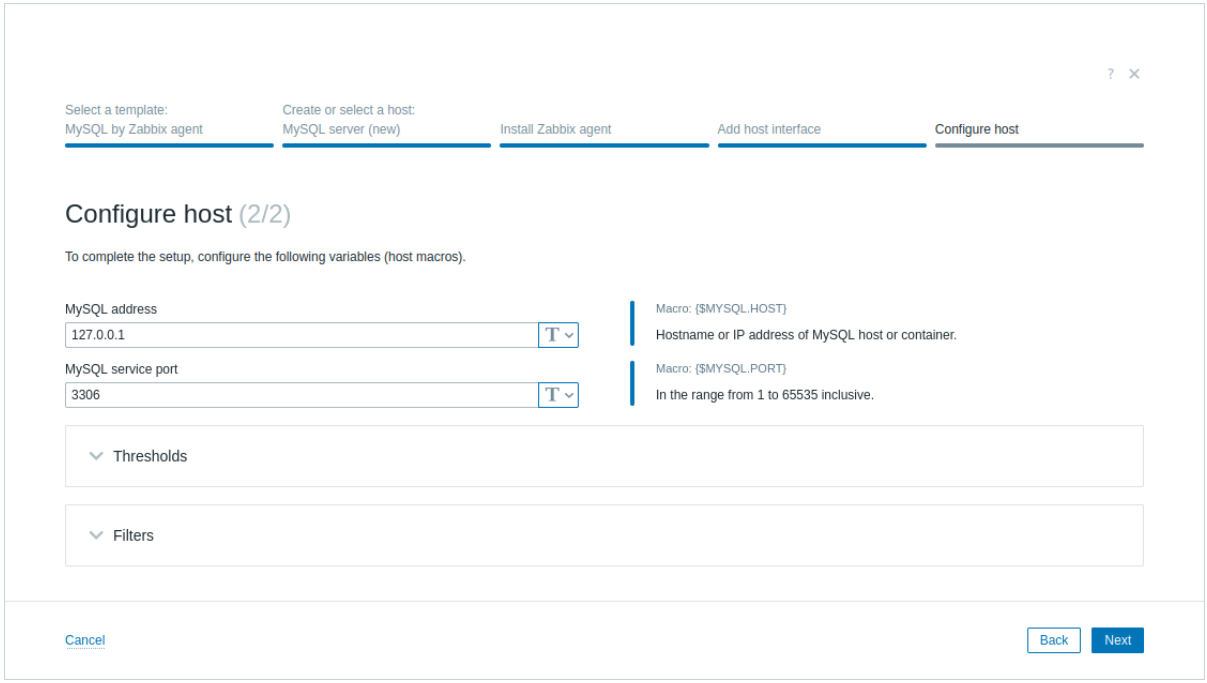
Configurazione dell'host

Alcuni template richiedono una configurazione aggiuntiva prima di poter creare l'host. Questa può includere:

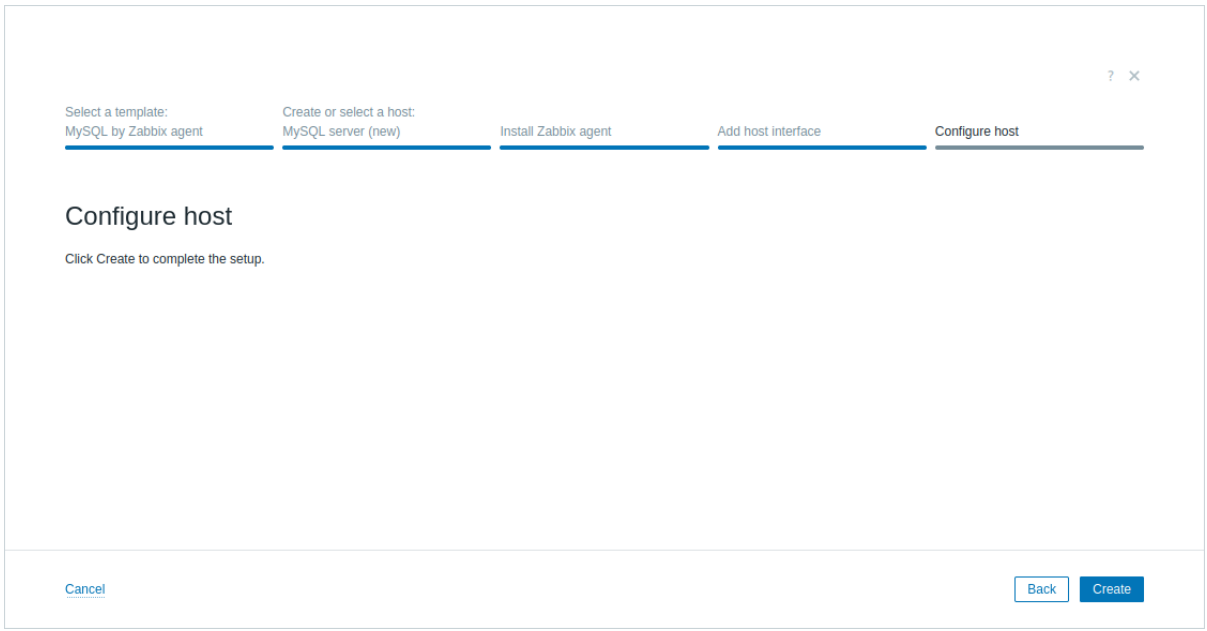
- Configurazione manuale dell'oggetto monitorato (ad esempio, abilitazione di servizi specifici, creazione di utenti di servizio o concessione di autorizzazioni):



- Definizione delle macro dell'host, ovvero variabili che controllano il comportamento degli item, le impostazioni di connessione, le credenziali di autenticazione e altri parametri:



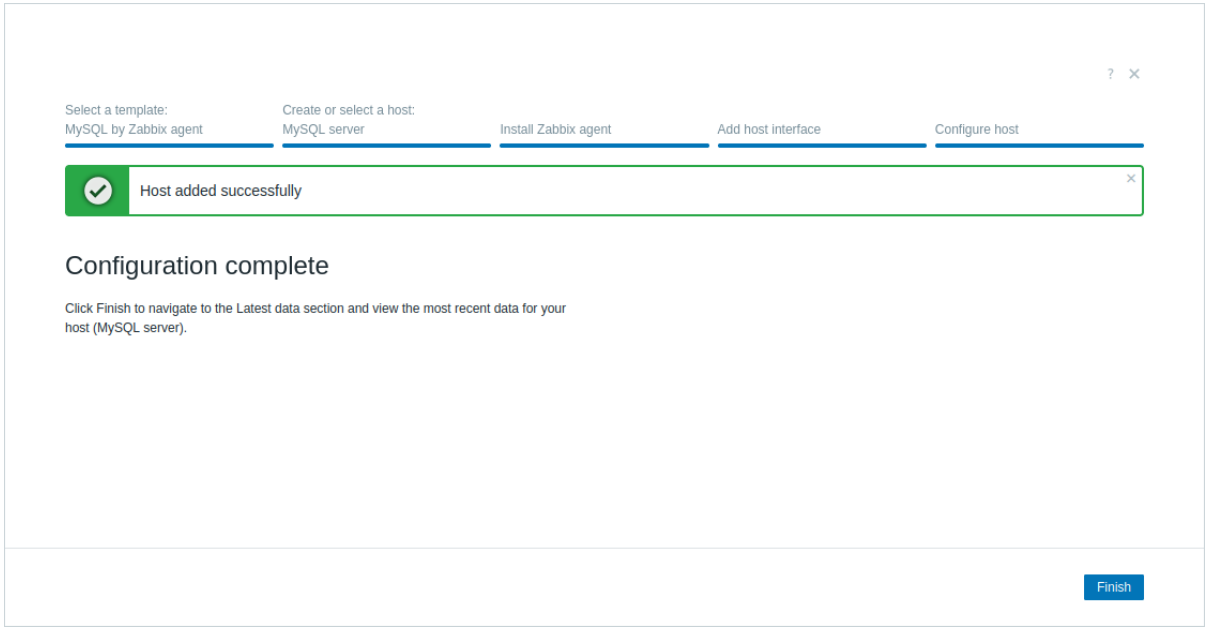
Dopo aver completato tutti i passaggi di configurazione richiesti, fare clic su *Crea* per aggiungere l'host a Zabbix (oppure su *Aggiorna* per aggiornare la configurazione di un host esistente):



Configurazione completata

A questo punto, Zabbix sta già monitorando il tuo target.

Per i nuovi host, fai clic su *Finish* per passare alla sezione Latest data e visualizzare i dati più recenti del tuo host. Per gli host esistenti, fai clic su *Finish* per chiudere la procedura guidata Host Wizard e tornare alla schermata da cui è stata aperta.



Configurazione di un host

Panoramica

Per configurare un host nel frontend di Zabbix, procedere come segue:

1. Andare su *Raccolta dati* > *Host* oppure *Monitoraggio* > *Host*.
2. Fare clic su *Crea host* nell'angolo in alto a destra della schermata (oppure sul nome dell'host per modificare un host esistente).
3. Inserire i parametri dell'host nel modulo.

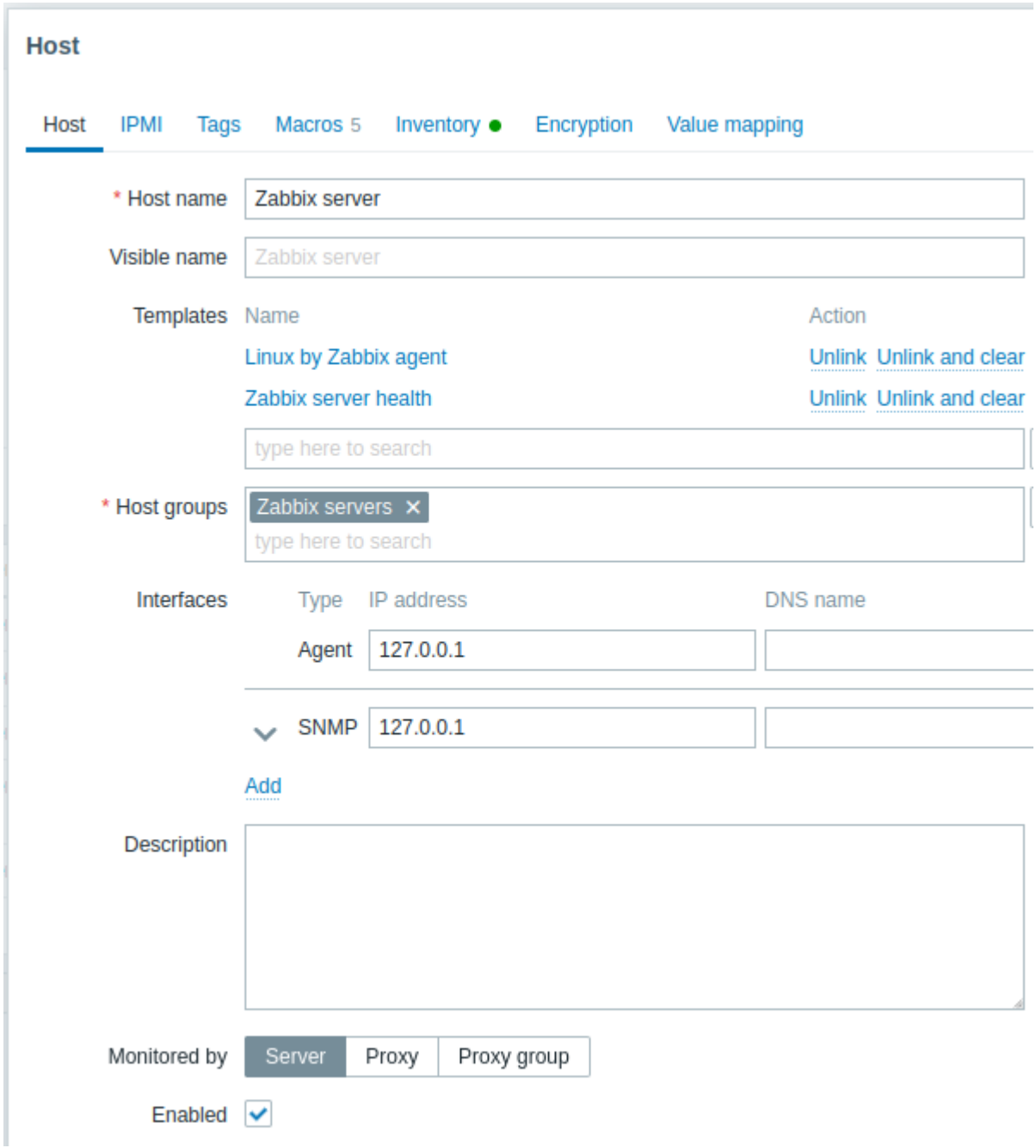
È inoltre possibile usare il pulsante *Clona* nel modulo di configurazione di un host esistente per creare un nuovo host. Questo host avrà tutte le proprietà dell'host esistente, inclusi i template collegati, le entità (item, trigger, ecc.) provenienti da tali template, nonché le entità collegate direttamente all'host esistente.

Si noti che, quando un host viene clonato, manterrà tutte le entità del template così come sono originariamente nel template. Eventuali modifiche apportate a tali entità a livello dell'host esistente (come l'intervallo di un item modificato, un'espressione regolare modificata o prototipi aggiunti alla regola di discovery a basso livello) non verranno clonate nel nuovo host; al loro posto saranno presenti come nel template.

In alternativa, è possibile usare la Host Wizard per configurare un host tramite un'interfaccia guidata, passo dopo passo.

Configurazione

La scheda **Host** contiene gli attributi generali dell'host:



Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

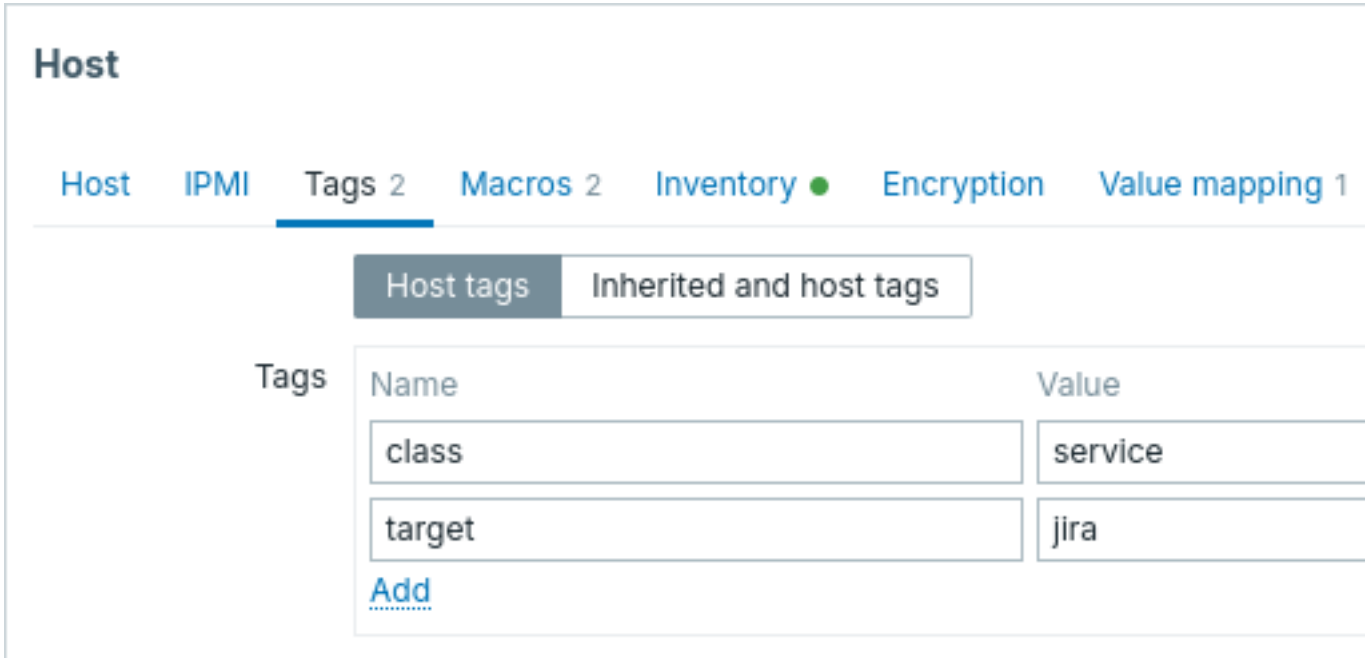
Parametro	Descrizione
Nome host	Inserire un nome host univoco. Sono consentiti caratteri alfanumerici, spazi, punti, trattini e caratteri di sottolineatura. Tuttavia, gli spazi iniziali e finali non sono consentiti. *Nota:* se Zabbix agent è in esecuzione sull'host in fase di configurazione, il parametro *Hostname* del file di configurazione dell'agent deve avere lo stesso valore del nome host inserito qui. Il nome specificato nel parametro è necessario per l'elaborazione dei controlli attivi.
Nome visibile	Inserire un nome visibile univoco per l'host. Se si imposta questo nome, sarà quello visualizzato negli elenchi, nelle mappe ecc., al posto del nome host tecnico. Questo attributo supporta UTF-8.

Parametro	Descrizione
Template	Collegare i template all'host. Tutte le entità (item, trigger ecc.) verranno ereditate dal template. Per collegare un nuovo template, iniziare a digitare il nome del template nel campo di testo. Verrà visualizzato un elenco di template corrispondenti; scorrere verso il basso per selezionarne uno. In alternativa, fare clic su *Seleziona* accanto al campo e selezionare i template dall'elenco in una finestra popup. Tutti i template selezionati verranno collegati all'host quando il modulo di configurazione dell'host viene salvato o aggiornato. Per scollegare un template, utilizzare una delle due opzioni nel blocco *Template collegati*: *Scollega* - scollega il template, ma ne conserva le entità (item, trigger ecc.); *Scollega e cancella* - scollega il template e rimuove tutte le relative entità (item, trigger ecc.). I nomi dei template elencati sono collegamenti selezionabili che conducono al modulo di configurazione del template.
Gruppi di host	Selezionare i gruppi di host a cui appartiene l'host. Un host deve appartenere ad almeno un gruppo di host. È possibile creare un nuovo gruppo e collegarlo all'host aggiungendo il nome di un gruppo inesistente.
Interfacce	Per un host sono supportati diversi tipi di interfaccia: *Agent*, *SNMP*, *JMX* e *IPMI*. Per impostazione predefinita non è definita alcuna interfaccia. Per aggiungere una nuova interfaccia, fare clic su *Aggiungi* nel blocco *Interfacce*, selezionare il tipo di interfaccia e inserire le informazioni *IP/DNS*, *Connetti a* e *Porta*. *Nota:* le interfacce utilizzate in uno o più item non possono essere rimosse e il collegamento *Rimuovi* è disabilitato per tali interfacce. L'"IP" o il "DNS" di un'interfaccia SNMP viene utilizzato anche per i trap SNMP. Durante la corrispondenza, viene utilizzato solo l'"IP" o il "DNS" selezionato nell'interfaccia dell'host. Per ulteriori dettagli sulla configurazione di un'interfaccia SNMP (v1, v2 e v3), vedere Configurazione del monitoraggio SNMP.
Indirizzo IP	Indirizzo IP dell'host (facoltativo).
Nome DNS	Nome DNS dell'host (facoltativo).
Connetti a	Facendo clic sul pulsante corrispondente, si indica al server Zabbix quale metodo utilizzare per recuperare i dati dagli agent: **IP** - connessione all'indirizzo IP dell'host (consigliato) **DNS** - connessione al nome DNS dell'host
Porta	Numero della porta TCP/UDP. I valori predefiniti sono: 10050 per Zabbix agent, 161 per l'agent SNMP, 12345 per JMX e 623 per IPMI.
Predefinita	Selezionare il pulsante di opzione per impostare l'interfaccia predefinita.
Descrizione	Inserire la descrizione dell'host.
Monitorato da	Selezionare se l'host è monitorato da: **Server** - l'host è monitorato dal server Zabbix; **Proxy** - l'host è monitorato da un proxy autonomo; **Gruppo di proxy** - l'host è monitorato da un gruppo di proxy.
Proxy	Viene visualizzato il nome del proxy assegnato (solo se il server Zabbix ne ha assegnato uno dal gruppo di proxy selezionato). Questo campo viene visualizzato solo se l'host è monitorato da un gruppo di proxy.
Abilitato	Quando la casella di controllo è selezionata, l'host è abilitato e pronto per il monitoraggio. Quando la casella di controllo non è selezionata, l'host è disabilitato e non monitorato: per le richieste di dati passive avviate dal server/proxy Zabbix (ad esempio Zabbix agent, agent SNMP, controlli semplici), il monitoraggio viene disabilitato dopo la sincronizzazione della configurazione. Anche i trigger e le azioni collegati all'host vengono disabilitati solo dopo il ricaricamento della configurazione. Per i controlli attivi di Zabbix agent, il monitoraggio si interrompe entro l'intervallo di tempo (circa 5 secondi) necessario affinché Zabbix agent riceva l'informazione che l'host è stato disabilitato. Durante questo breve intervallo, l'host continuerà a raccogliere localmente i dati per i controlli attivi e tenterà di inviarli al server/proxy; tuttavia, poiché l'host è contrassegnato come *Disabilitato*, il server/proxy rifiuterà i dati. Quando si disabilita l'host, i relativi item vengono rimossi immediatamente dalla cache della cronologia (ad eccezione dei loro ultimi valori, conservati per i log).

La scheda **IPMI** contiene gli attributi di gestione IPMI.

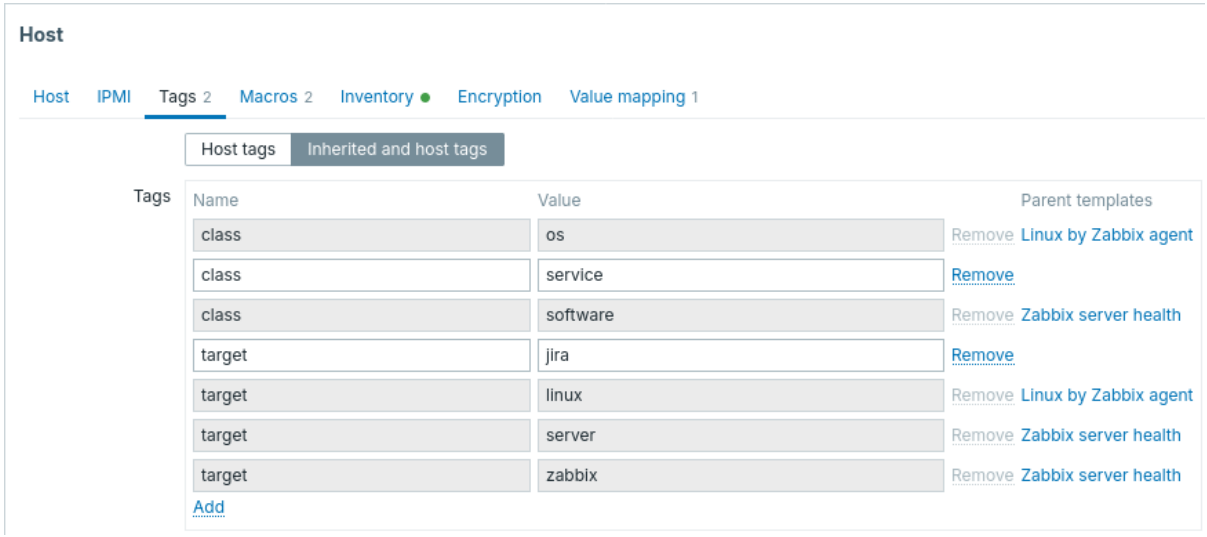
Parametro	Descrizione
Algoritmo di autenticazione	Selezionare l'algoritmo di autenticazione.
Livello di privilegio	Selezionare il livello di privilegio.
Nome utente	Nome utente per l'autenticazione. È possibile utilizzare macro utente.
Password	Password per l'autenticazione. È possibile utilizzare macro utente.

La scheda **Tag** consente di definire i tag a livello di host. Tutti i problemi di questo host verranno contrassegnati con i valori inseriti qui.



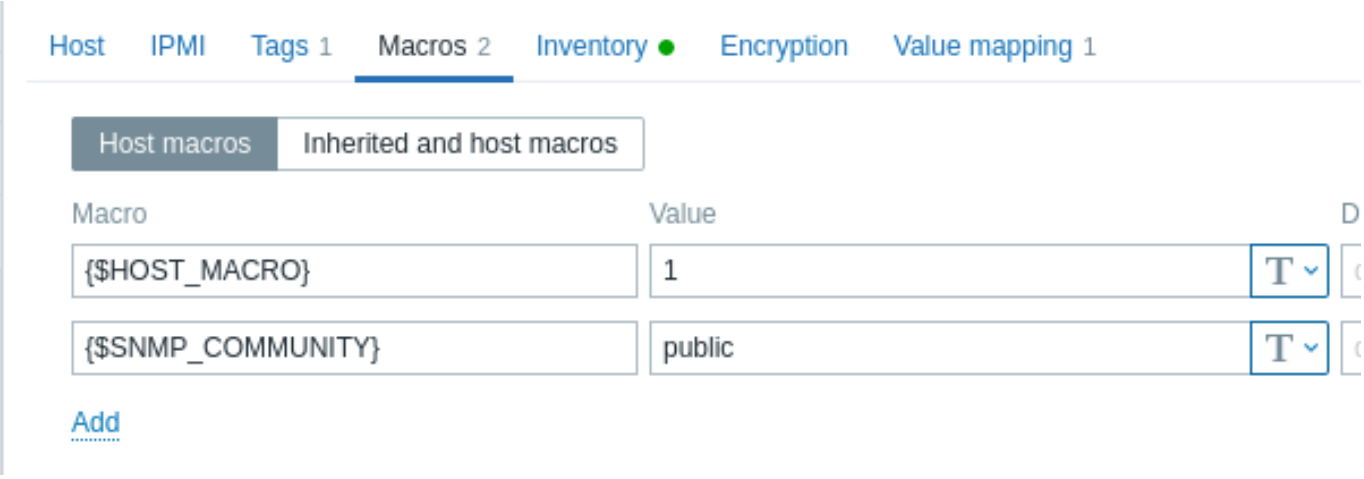
Nei tag sono supportate le macro utente, le macro `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` e `{HOST.ID}`.

È inoltre possibile visualizzare qui i tag a livello di template selezionando l'opzione *Tag ereditati e dell'host*. In questa sezione vengono visualizzati tutti i tag utente definiti per l'host e la relativa origine.

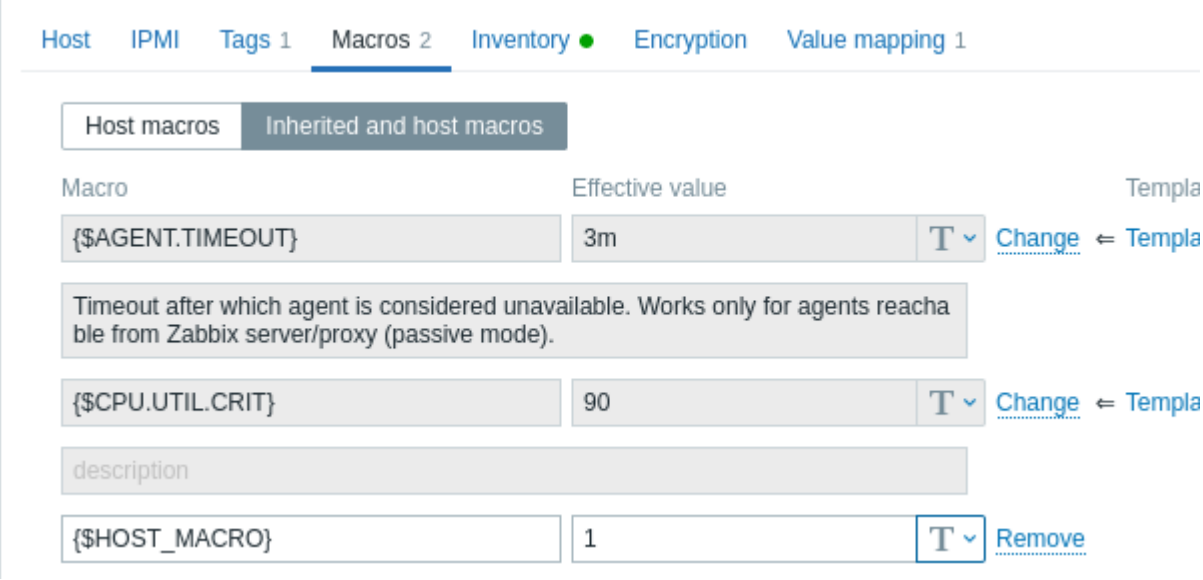


Per comodità, vengono forniti collegamenti ai template corrispondenti.

La scheda **Macro** consente di definire le macro utente a livello di host come coppie nome-valore. I valori delle macro possono essere memorizzati come testo normale, testo segreto o segreto Vault. È inoltre possibile aggiungere una descrizione.

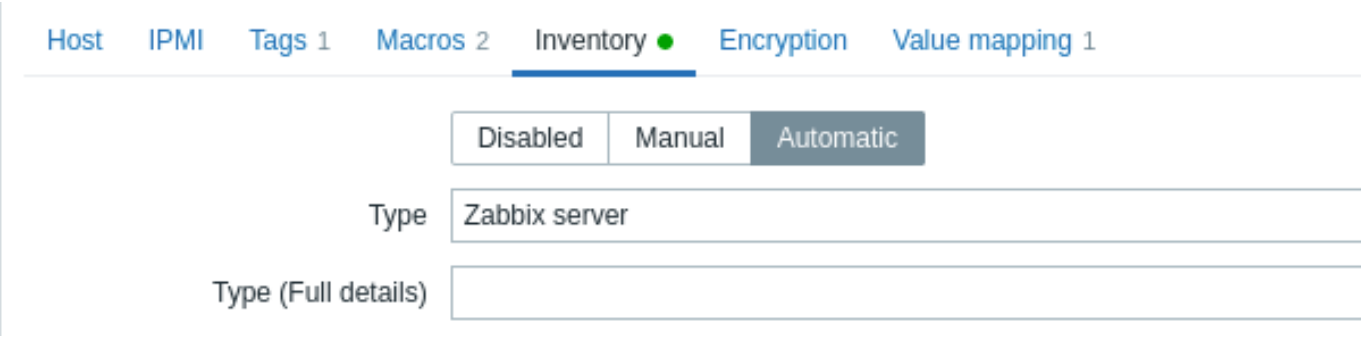


È inoltre possibile visualizzare qui le macro utente a livello di template e globali selezionando l'opzione *Macro ereditate e dell'host*. In questa sezione vengono visualizzate tutte le macro utente definite per l'host, il valore a cui vengono risolte e la relativa origine.



Per comodità, vengono forniti collegamenti ai template corrispondenti e alla configurazione delle macro globali. È inoltre possibile modificare una macro di template/globale a livello di host, creando di fatto una copia della macro sull'host.

La scheda **Inventario** consente di inserire manualmente le informazioni di inventario dell'host. È inoltre possibile scegliere di abilitare la compilazione *Automatica* dell'inventario oppure disabilitare la compilazione dell'inventario per questo host.



Se l'inventario è abilitato (manualmente o automaticamente), accanto al nome della scheda viene visualizzato un punto verde.

Crittografia

La scheda **Crittografia** consente di richiedere connessioni crittografate con l'host.

Parameter	Description
Connections to host	Come server o proxy Zabbix si connette a Zabbix agent su un host: nessuna crittografia (predefinito), tramite PSK (chiave precondivisa) o certificato.

Parameter	Description
Connections from host	Seleziona quale tipo di connessioni sono consentite dall'host (cioè da Zabbix agent e Zabbix sender). È possibile selezionare più tipi di connessione contemporaneamente (utile per i test e per passare a un altro tipo di connessione). Il valore predefinito è "No encryption".
Issuer	Issuer consentito del certificato. Il certificato viene prima convalidato con la CA (autorità di certificazione). Se è valido, firmato dalla CA, il campo *Issuer* può essere usato per limitare ulteriormente la CA consentita. Questo campo è pensato per essere usato se la tua installazione di Zabbix utilizza certificati di più CA. Se questo campo è vuoto, viene accettata qualsiasi CA.
Subject	Subject consentito del certificato. Il certificato viene prima convalidato con la CA. Se è valido, firmato dalla CA, il campo *Subject* può essere usato per consentire un solo valore della stringa *Subject*. Se questo campo è vuoto, viene accettato qualsiasi certificato valido firmato dalla CA configurata.
PSK identity	Stringa di identità della chiave precondivisa. Non inserire informazioni sensibili nella PSK identity, perché viene trasmessa non crittografata sulla rete per informare il destinatario su quale PSK usare.
PSK	Chiave precondivisa (stringa esadecimale). Lunghezza massima: 512 cifre esadecimali (PSK da 256 byte) se Zabbix usa la libreria GnuTLS o OpenSSL, 64 cifre esadecimali (PSK da 32 byte) se Zabbix usa la libreria mbed TLS (PolarSSL). Esempio: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Mappatura dei valori

La scheda **Mappatura dei valori** consente di configurare una rappresentazione intuitiva per i dati degli item nelle mappature dei valori.

Configurazione di un gruppo host

Panoramica

I gruppi host vengono utilizzati per il raggruppamento logico degli host e per assegnare loro i permessi degli utenti.

A ogni host deve essere assegnato almeno un gruppo host. Un host può appartenere a più gruppi host e ogni gruppo host può contenere più host.

Si noti che in Zabbix tutti i permessi si basano sui gruppi: gruppi utente, gruppi host e template groups. Quindi, anche se un singolo utente necessita dell'accesso a un singolo host, questo viene concesso aggiungendo l'utente a un gruppo utente e assegnando a quel gruppo utente i permessi sul gruppo host che contiene quell'host.

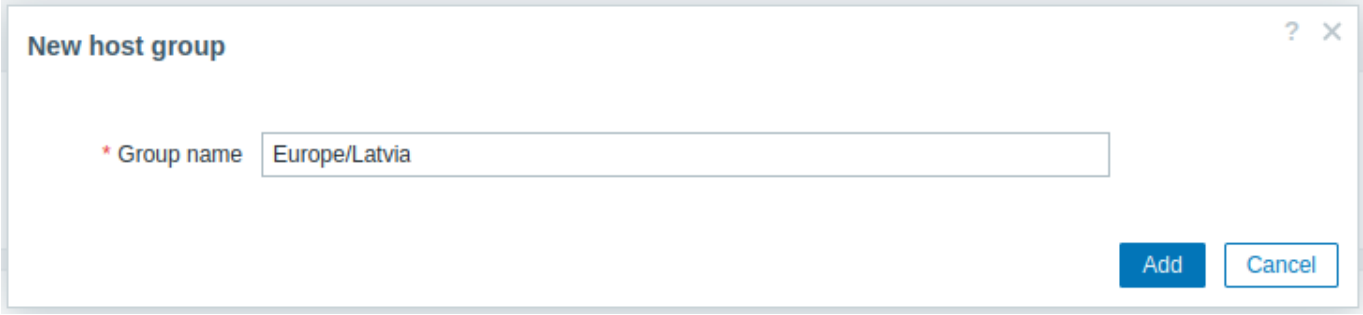
Configurazione

> **Attention:**
> Solo gli utenti Super admin possono creare gruppi di host.

Ci sono due opzioni per creare un gruppo di host nel frontend di Zabbix.

Opzione uno:

1. Vai su *Data collection* > *Host groups*.
2. Fai clic su *Create host group* nell'angolo in alto a destra della schermata.
3. Inserisci il nome del gruppo nel modulo.



Opzione due: quando configuri un host, inserisci un nome di gruppo inesistente nel campo di input *Host groups*.

Una volta creato il gruppo di host, puoi fare clic sul nome del gruppo nell'elenco sotto *Data collection* > *Host groups* per modificare il nome del gruppo, clonare il gruppo o eliminarlo.

L'eliminazione di un gruppo di host rimuove solo il gruppo logico, non gli host nel gruppo. Non è possibile eliminare un gruppo di host che sia l'unico gruppo per un host esistente.

Creazione di sottogruppi di host

Un sottogruppo di host (o gruppo di host nidificato) è un elemento figlio del gruppo di host padre che lo contiene.

Un sottogruppo viene creato utilizzando la barra obliqua '/' nel campo di input del nome del gruppo per indicarne la relazione con il gruppo o i gruppi padre. Ad esempio:

- inserendo `Europe/Latvia` si crea il sottogruppo `Europe/Latvia` del gruppo padre `Europe`.
- inserendo `Europe/Latvia/Riga/Zabbix servers` si crea il rispettivo sottogruppo all'interno dei gruppi padre nidificati `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

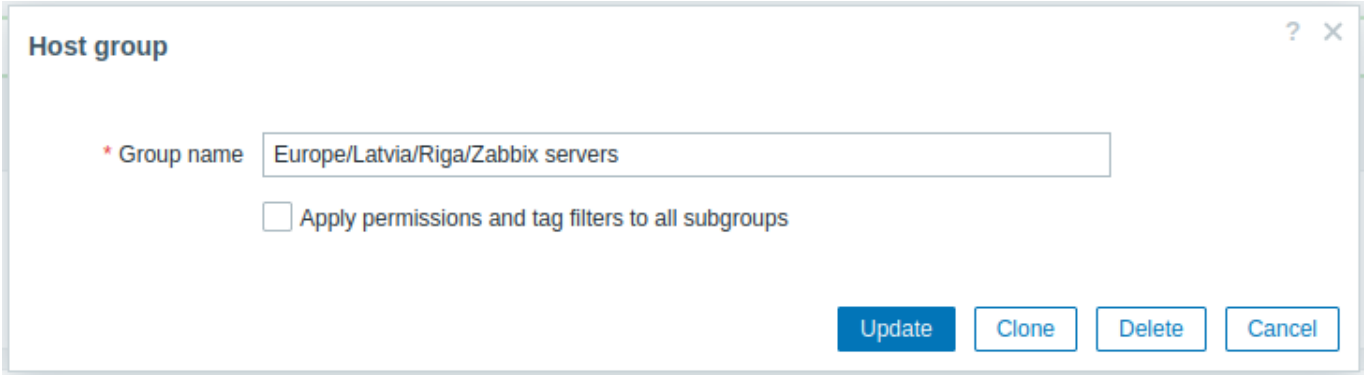
Quando si crea un sottogruppo, non è consentito utilizzare barre oblique iniziali o finali, né più barre oblique consecutive. L'escape di '/' non è supportato.

Non è necessario creare alcun gruppo di host padre prima di creare un sottogruppo. È possibile scegliere se iniziare creando un sottogruppo (ad esempio, `Europe/Latvia`) oppure uno qualsiasi dei gruppi di host padre (nel nostro esempio, `Europe`). Se si inizia creando un sottogruppo, i gruppi di host padre **non** verranno creati automaticamente.

Permessi ai gruppi host

- Quando si crea un sottogruppo di un gruppo host padre esistente (ad esempio, creando `Europe/Latvia` quando `Europe` esiste già), i permessi del gruppo utenti sul sottogruppo vengono ereditati dal padre.
- Quando si crea un gruppo host padre per un sottogruppo esistente (ad esempio, creando `Europe` quando `Europe/Latvia` esiste già), non vengono impostati permessi sul padre.

Quando si modifica qualsiasi gruppo host, è anche possibile impostare un'opzione aggiuntiva, *Applica i permessi e i filtri dei tag a tutti i sottogruppi*.



Selezionando questa casella di controllo e facendo clic su *Aggiorna* si applicherà lo stesso livello di permessi e filtri dei tag a tutti i sottogruppi attuali e futuri del gruppo host in modifica.

Quindi, se a uno qualsiasi dei gruppi utenti sono stati assegnati permessi diversi ai sottogruppi del gruppo host in modifica, selezionando la casella di controllo tutti i sottogruppi attuali e futuri riceveranno gli stessi permessi utente e permessi basati sui tag del gruppo in modifica.

Si noti che questa opzione non viene salvata nel database e sovrascriverà i permessi esistenti. Eventuali modifiche apportate tramite questa opzione possono essere annullate solo manualmente.

Inventario

Panoramica

In Zabbix è possibile mantenere l'inventario dei dispositivi di rete.

Nel frontend di Zabbix è presente uno speciale menu *Inventario*. Tuttavia, inizialmente non vi saranno dati visibili e non è lì che si inseriscono i dati. La compilazione dei dati di inventario viene eseguita manualmente durante la configurazione di un host oppure automaticamente utilizzando alcune opzioni di popolamento automatico.

Costruire l'inventario

Modalità manuale

Quando si configura un host, nella scheda *Inventario* è possibile inserire dettagli quali il tipo di dispositivo, il numero di serie, la posizione, la persona responsabile, gli URL, ecc. - dati che popoleranno le informazioni dell'inventario.

Se nelle informazioni dell'inventario dell'host è incluso un URL che inizia con 'http' o 'https', nella sezione *Inventario* verrà visualizzato come link cliccabile.

Modalità automatica

L'inventario degli host può essere compilato anche automaticamente. Per farlo, durante la configurazione di un host, la modalità di inventario nella scheda *Inventario* deve essere impostata su *Automatico*.

È quindi possibile configurare gli item dell'host per compilare qualsiasi campo dell'inventario dell'host con il relativo valore, indicando il campo di destinazione tramite l'attributo corrispondente (denominato *L'item compilerà il campo dell'inventario dell'host*) nella configurazione dell'item.

Item particolarmente utili per la raccolta automatizzata dei dati di inventario:

- system.hw.chassis[full|type|vendor|model|serial] - il valore predefinito è [full], sono necessarie le autorizzazioni root
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - il valore predefinito è [all,full]
- system.hw.devices[pci|usb] - il valore predefinito è [pci]
- system.hw.macaddr[interface,short|full] - il valore predefinito è [all,full], interface è un'espressione regolare
- system.sw.arch
- system.sw.os[name|short|full] - il valore predefinito è [name]
- system.sw.packages[regexp,manager,short|full] - il valore predefinito è [all,all,full]

Selezione della modalità di inventario

La modalità di inventario può essere selezionata nel modulo di configurazione dell'host.

Per i nuovi host, la modalità di inventario predefinita viene selezionata in base all'impostazione *Modalità di inventario predefinita dell'host* in *Amministrazione* > *Generale* > *Altro*.

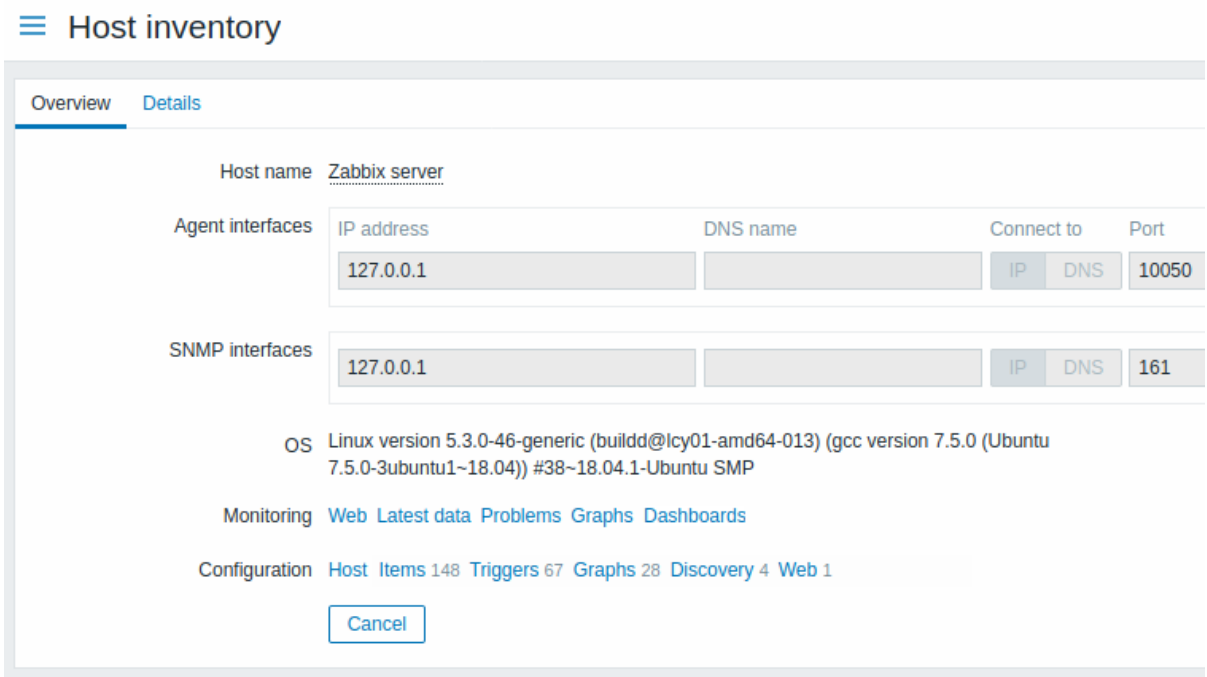
Per gli host aggiunti tramite il rilevamento di rete o le azioni di autoregistrazione, è possibile definire un'operazione *Imposta modalità di inventario dell'host* selezionando la modalità manuale o automatica. Questa operazione sostituisce l'impostazione *Modalità di inventario predefinita dell'host*.

Panoramica dell'inventario

I dettagli di tutti i dati di inventario esistenti sono disponibili nel menu *Inventory*.

In *Inventory* > *Overview* puoi ottenere il conteggio degli host in base a vari campi dell'inventario.

In *Inventory* > *Hosts* puoi vedere tutti gli host che dispongono di informazioni di inventario. Facendo clic sul nome dell'host verranno visualizzati i dettagli dell'inventario in un modulo.



La scheda **Overview** mostra:

Parameter	Description
Host name	Nome dell'host. Facendo clic sul nome si apre un menu con gli script definiti per l'host. Il nome dell'host viene visualizzato con un'icona arancione se l'host è in manutenzione.
Visible name	Nome visibile dell'host (se definito).
Host (Agent, SNMP, JMX, IPMI) interfaces	Questo blocco fornisce i dettagli delle interfacce configurate per l'host.
OS	Campo di inventario del sistema operativo dell'host (se definito).
Hardware	Campo di inventario dell'hardware dell'host (se definito).
Software	Campo di inventario del software dell'host (se definito).
Description	Descrizione dell'host.
Monitoring	Collegamenti alle sezioni di monitoraggio con i dati di questo host: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Collegamenti alle sezioni di configurazione per questo host: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. La quantità di entità configurate è elencata dopo ogni collegamento.

La scheda **Details** mostra tutti i campi di inventario compilati (non vuoti).

Macro di inventario

Sono disponibili macro dell'inventario dell'host `{INVENTORY.*}` da usare nelle notifiche, ad esempio:

"Il server in {INVENTORY.LOCATION1} ha un problema, la persona responsabile è {INVENTORY.CONTACT1}, il numero di telefono è {INVENTORY.POC.PRIMARY.PHONE.A1}."

Per ulteriori dettagli, vedere la pagina Macro supportate.

Aggiornamento di massa

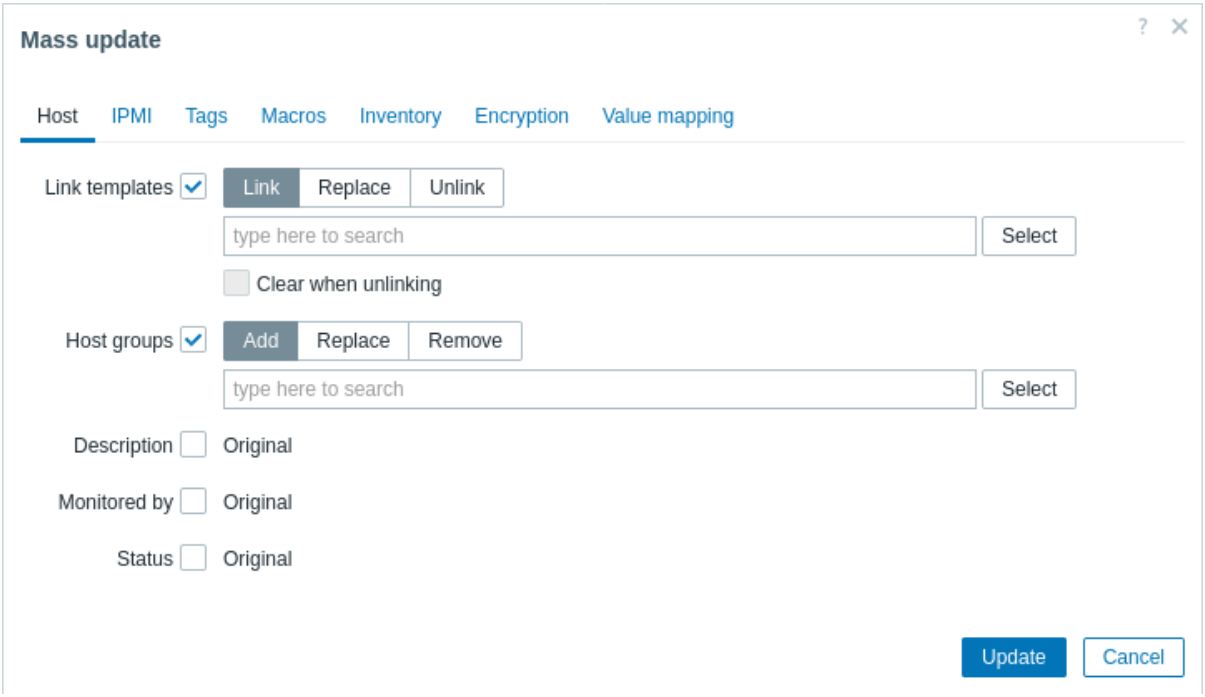
Panoramica

A volte potresti voler modificare alcuni attributi per un numero di host in una singola volta. Invece di aprire ogni singolo host per la modifica, puoi utilizzare la funzione di aggiornamento di massa.

Utilizzo dell'aggiornamento di massa

Per aggiornare più host contemporaneamente, procedere come segue:

- Selezionare le caselle di controllo accanto agli host che si desidera aggiornare nell'elenco degli host
- Fare clic su *Aggiornamento di massa* sotto l'elenco
- Accedere alla scheda con gli attributi richiesti (*Host*, *IPMI*, *Tag*, *Macro*, *Inventario*, *Crittografia* o *Mappatura dei valori*)
- Selezionare le caselle di controllo degli attributi da aggiornare e immettere un nuovo valore



Quando si seleziona il pulsante corrispondente per aggiornare il collegamento ai **template**, sono disponibili le seguenti opzioni:

- *Collega* - specificare i template aggiuntivi da collegare
- *Sostituisci* - specificare i template da collegare scollegando qualsiasi template precedentemente collegato agli host
- *Scollega* - specificare i template da scollegare

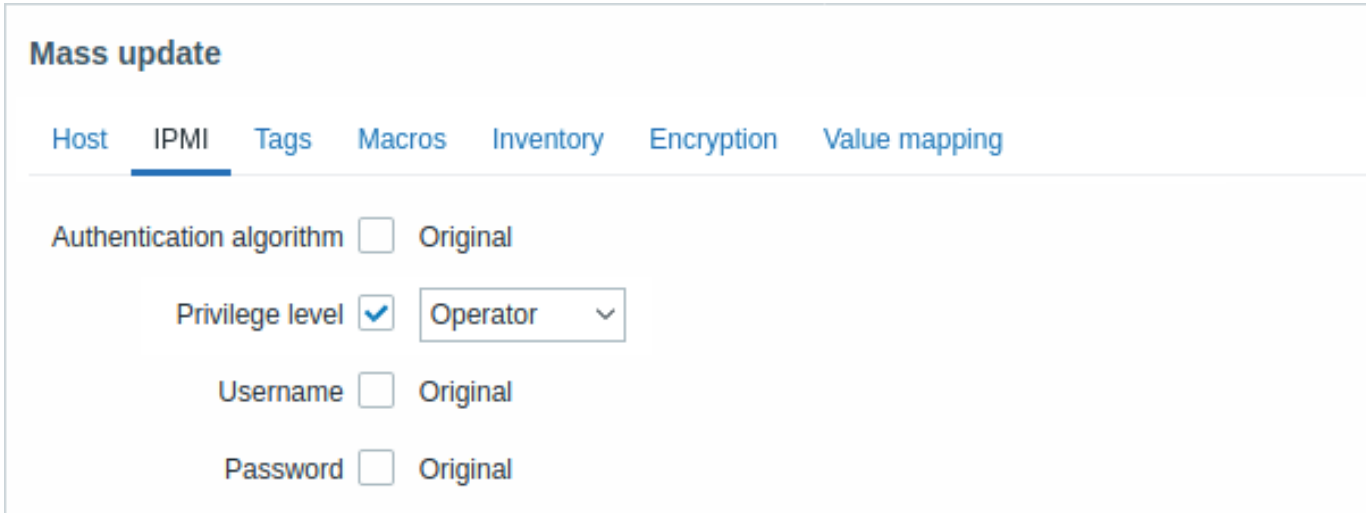
Per specificare i template da collegare/scollegare, iniziare a digitare il nome del template nel campo di completamento automatico finché non viene visualizzato un elenco a discesa con i template corrispondenti. Scorrere verso il basso per selezionare il template richiesto.

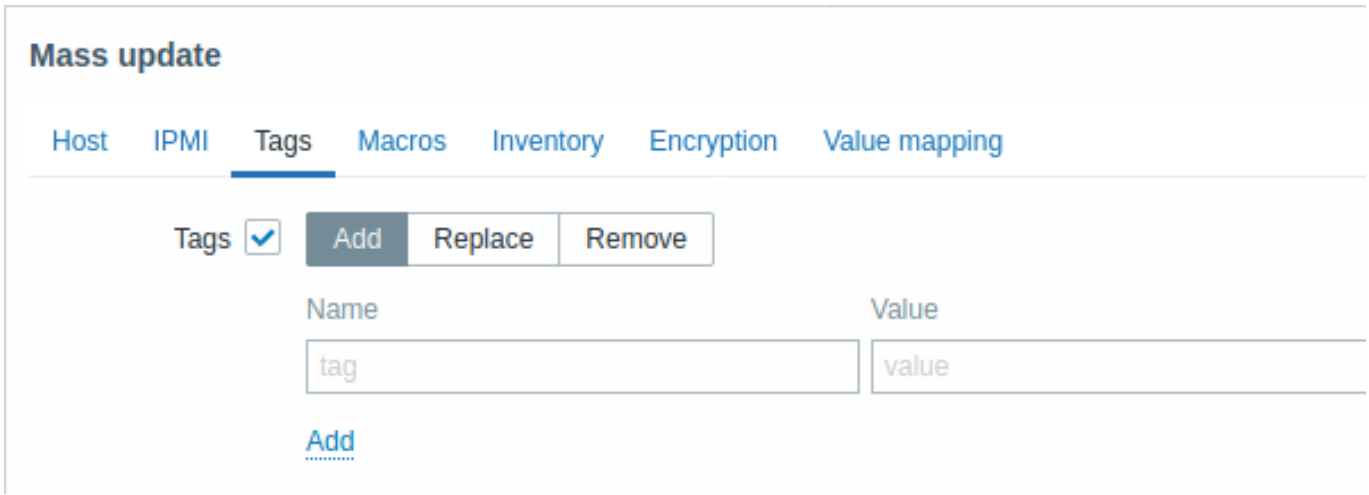
L'opzione *Cancella durante lo scollegamento* consente non solo di scollegare i template precedentemente collegati, ma anche di rimuovere tutte le entità ereditate da essi (item, trigger, grafici e così via).

Quando si seleziona il pulsante corrispondente per aggiornare il **gruppo di host**, sono disponibili le seguenti opzioni:

- *Aggiungi* - specificare gruppi di host aggiuntivi tra quelli esistenti oppure immettere gruppi di host completamente nuovi per gli host;
- *Sostituisci* - rimuovere l'host da tutti i gruppi di host esistenti e sostituirli con quelli specificati in questo campo (gruppi di host esistenti o nuovi);
- *Rimuovi* - rimuovere gruppi di host specifici dagli host.

Questi campi supportano il completamento automatico: iniziando a digitare al loro interno viene visualizzato un elenco a discesa dei gruppi di host corrispondenti. Se il gruppo di host è nuovo, viene visualizzato anch'esso nell'elenco a discesa ed è indicato da *(nuovo)* dopo la stringa. Scorrere verso il basso per selezionarlo.

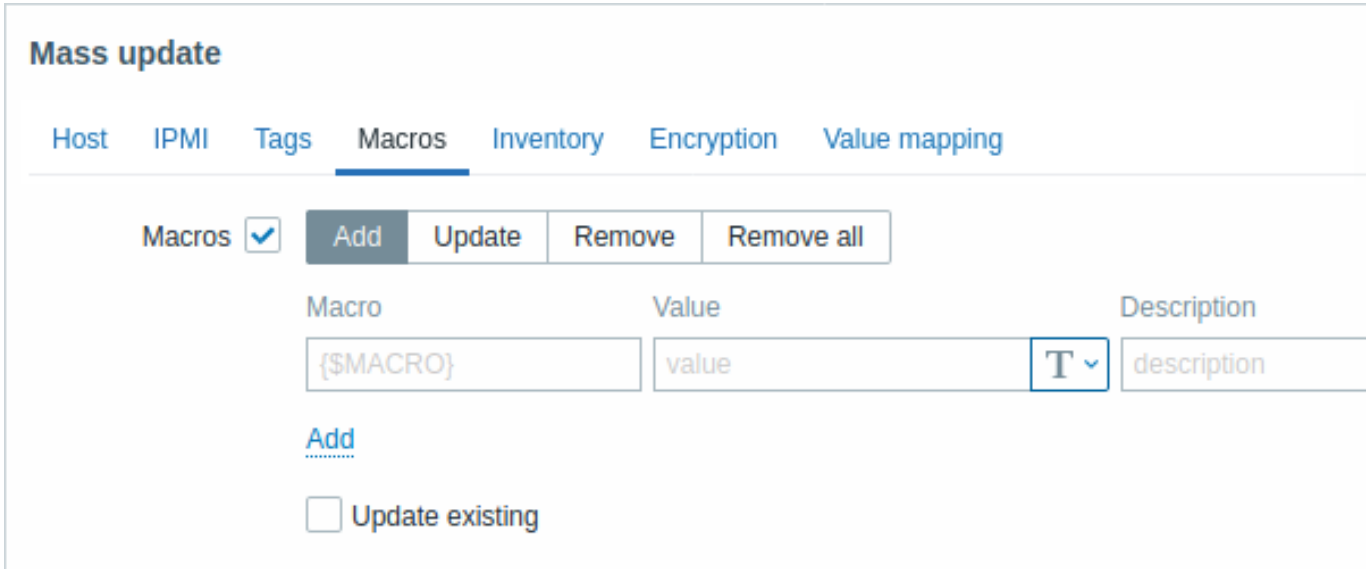




Quando si seleziona il pulsante corrispondente per aggiornare i tag, sono disponibili le seguenti opzioni:

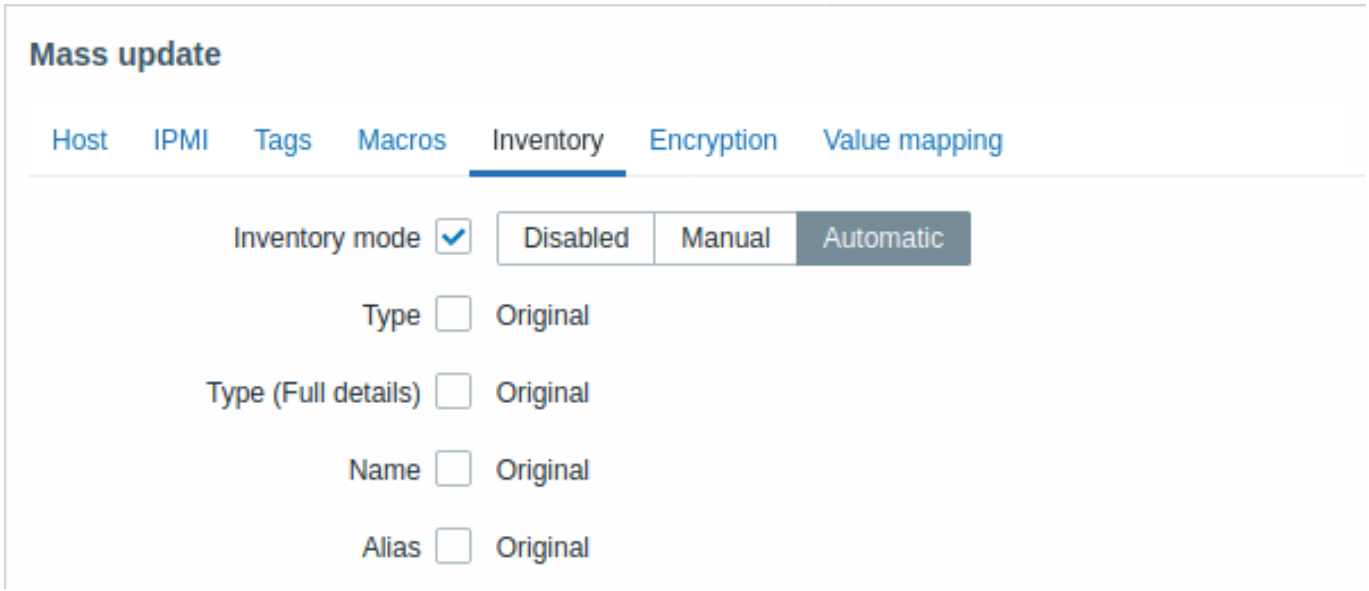
- *Aggiungi* - aggiungere nuovi tag agli host;
- *Sostituisci* - rimuovere tutti i tag esistenti dall'host e sostituirli con quelli specificati di seguito;
- *Rimuovi* - rimuovere i tag specificati dagli host.

Le macro utente, le macro `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` e `{HOST.ID}` sono supportate nei tag. I tag con lo stesso nome ma valori diversi non sono considerati "duplicati" e possono essere aggiunti allo stesso host.

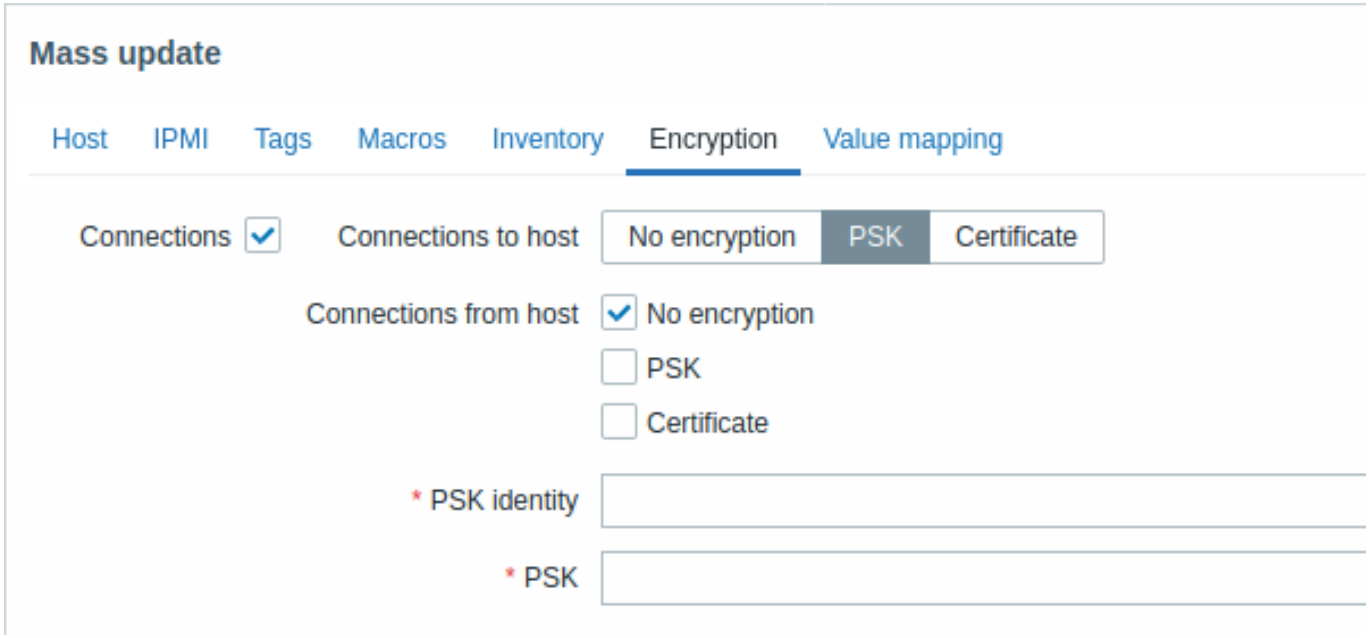


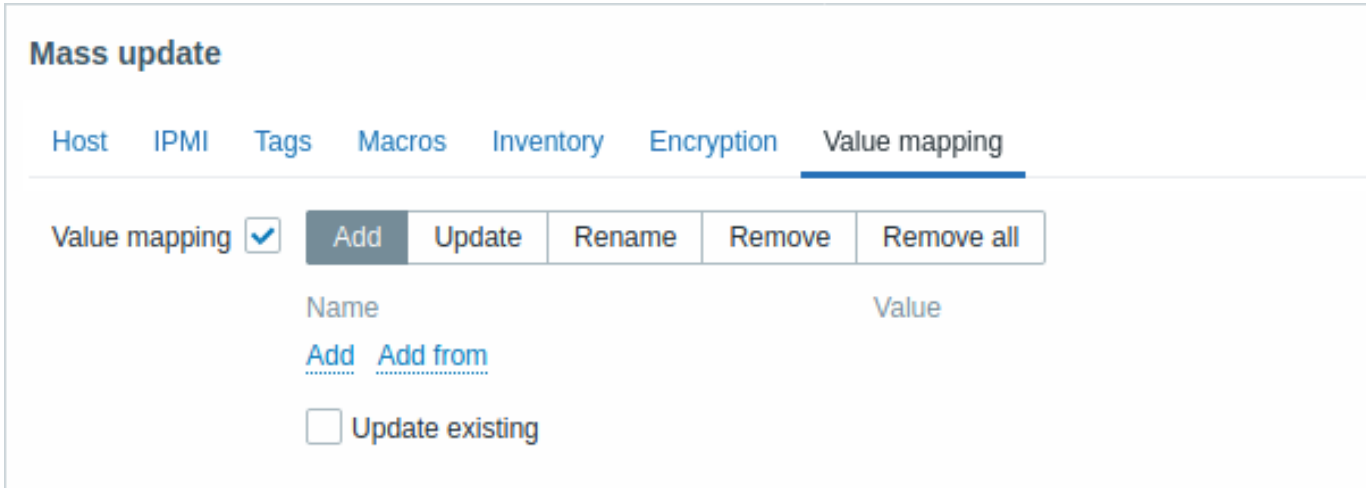
Quando si seleziona il pulsante corrispondente per aggiornare le macro, sono disponibili le seguenti opzioni:

- *Aggiungi* - specificare macro utente aggiuntive per gli host. Se la casella di controllo *Aggiorna esistenti* è selezionata, il valore, il tipo e la descrizione della macro specificata verranno aggiornati. Se non è selezionata e una macro con quel nome esiste già sull'host o sugli host, non verrà aggiornata.
- *Aggiorna* - sostituire i valori, i tipi e le descrizioni delle macro specificate in questo elenco. Se la casella di controllo *Aggiungi mancanti* è selezionata, la macro che in precedenza non esisteva su un host verrà aggiunta come nuova macro. Se non è selezionata, verranno aggiornate solo le macro già esistenti su un host.
- *Rimuovi* - rimuovere le macro specificate dagli host. Se la casella di controllo *Eccetto quelle selezionate* è selezionata, verranno rimosse tutte le macro tranne quelle specificate nell'elenco. Se non è selezionata, verranno rimosse solo le macro specificate nell'elenco.
- *Rimuovi tutto* - rimuovere tutte le macro utente dagli host. Se la casella di controllo *Confermo di voler rimuovere tutte le macro* non è selezionata, si aprirà una nuova finestra popup per chiedere conferma della rimozione di tutte le macro.



Per poter aggiornare in massa i campi dell'inventario, la *Modalità inventario* deve essere impostata su "Manuale" o "Automatica".





Per l'aggiornamento della mappatura dei valori sono disponibili pulsanti con le seguenti opzioni:

- *Aggiungi* - aggiungere mappe dei valori agli host. Se si seleziona *Aggiorna esistenti*, verranno aggiornate tutte le proprietà della mappa dei valori con questo nome. In caso contrario, se una mappa dei valori con quel nome esiste già, non verrà aggiornata.
- *Aggiorna* - aggiornare le mappe dei valori esistenti. Se si seleziona *Aggiungi mancanti*, una mappa dei valori che in precedenza non esisteva su un host verrà aggiunta come nuova mappa dei valori. In caso contrario, verranno aggiornate solo le mappe dei valori già esistenti su un host.
- *Rinomina* - assegnare un nuovo nome a una mappa dei valori esistente
- *Rimuovi* - rimuovere dagli host le mappe dei valori specificate. Se si seleziona *Eccetto quelle selezionate*, verranno rimosse tutte le mappe dei valori **tranne** quelle specificate.
- *Rimuovi tutto* - rimuovere tutte le mappe dei valori dagli host. Se la casella di controllo *Confermo di voler rimuovere tutte le mappe dei valori* non è selezionata, si aprirà una nuova finestra popup per chiedere conferma della rimozione.

Al termine di tutte le modifiche richieste, fare clic su *Aggiorna*. Gli attributi verranno aggiornati di conseguenza per tutti gli host selezionati.

Item

Overview

Un item è una singola metrica.

Gli item sono usati per raccogliere dati. Dopo aver configurato un host, devi aggiungere item per raccogliere gli effettivi dati. Un modo per aggiungere velocemente molti dati è quello di collegare un template all'host. Tuttavia, per prestazioni di sistema ottimale, potrebbe essere necessario aggiustare il template al fine di raccogliere solo i dati necessari, con la frequenza desiderata.

Per specificare quali dati raccogliere, è possibile usare la item key. Ad esempio, un item con valore della key **system.cpu.load** raccoglierà i dati relativi al carico per processore, mentre un item con key **net.if.in** collezionerà informazioni sul traffico in ingresso.

È inoltre possibile specificare parametri addizionali nelle parentesi quadre dopo il nome della key. Ad esempio, system.cpu.load**[avg5]** restituirà il carico medio del processore registrato negli ultimi 5 minuti, mentre net.if.in**[eth0]** mostrerà il traffico in ingresso per l'interfaccia "eth0".

Note:
Vedi la sezione specifica sui tipi di item per tutti i tipi di item e le item key supportate.

Prosegui su creazione e configurazione di un item.

Creazione di un item

Panoramica

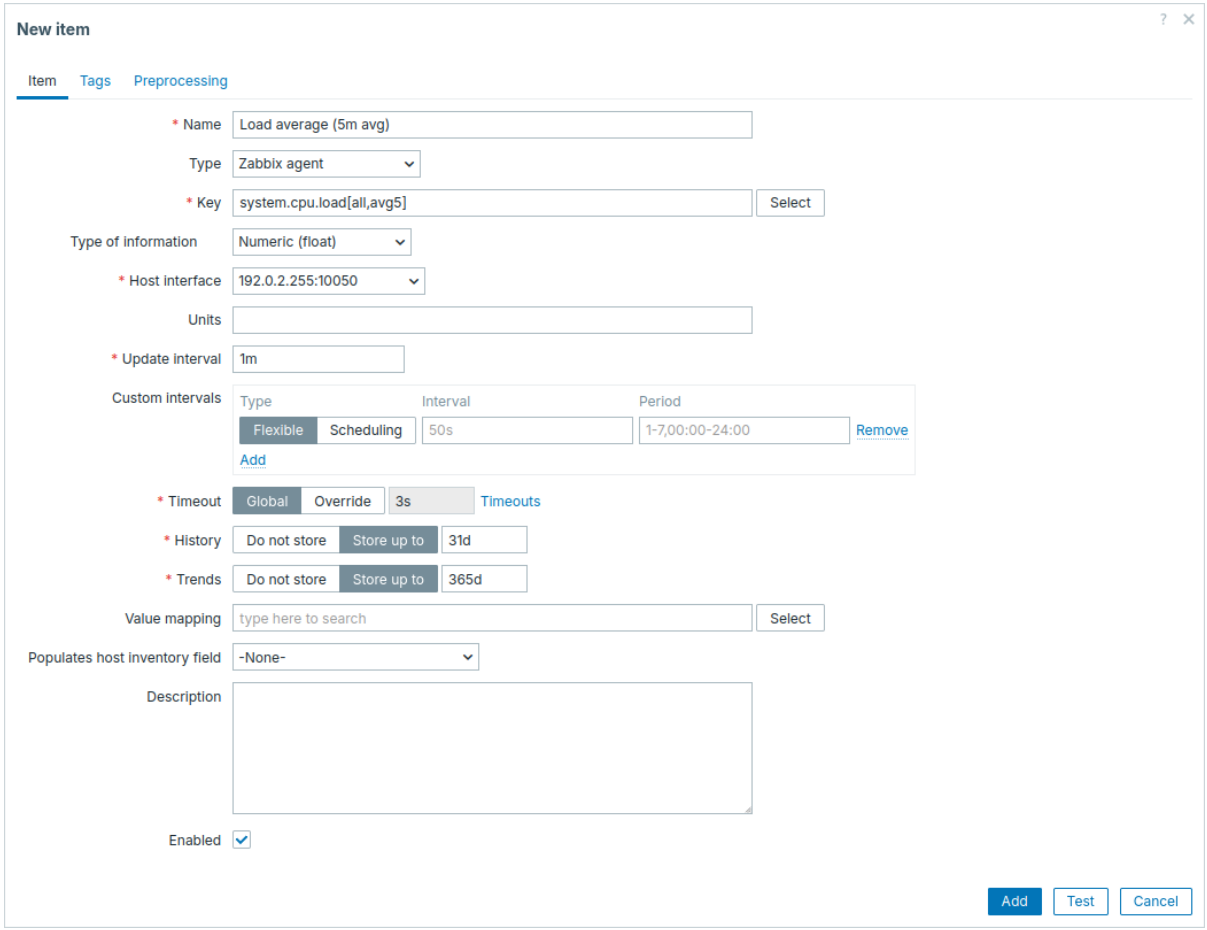
Per creare un item nel frontend di Zabbix, procedere come segue:

1. Andare a *Raccolta dati* > *Host*.
2. Fare clic su *Item* nella riga dell'host.
3. Fare clic su *Crea item* nell'angolo superiore destro della schermata.
4. Immettere i parametri dell'item nel modulo.

In alternativa, è possibile creare un item aprendo un item esistente, facendo clic sul pulsante *Clona* e salvandolo con un nome diverso.

Configurazione

La scheda **Item** contiene gli attributi generali dell'item.



Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Name	Nome dell'item. Sono supportate le macro utente.

Parametro	Descrizione
Type	Tipo di item. Consultare le sezioni relative ai singoli tipi di item.
Key	Chiave dell'item (fino a 2048 caratteri). Le chiavi degli item supportate sono disponibili nelle sezioni relative ai singoli tipi di item. La chiave deve essere univoca all'interno di un singolo host. Se il tipo di chiave è 'Zabbix agent', 'Zabbix agent (active)' o 'Simple check', il valore della chiave deve essere supportato da Zabbix agent o Zabbix server. Vedere anche il formato corretto della chiave.
Type of information	Tipo di dati utilizzato per convalidare il valore dell'item e memorizzarlo nel database, anche dopo eventuali conversioni: **Numeric (unsigned)** - intero senza segno a 64 bit; **Numeric (float)** - numero in virgola mobile a 64 bit; **Character** - testo breve; **Log** - testo lungo con o senza proprietà del log (timestamp, origine, gravità, ID dell'evento di log); **Text** - testo lungo; **Binary** - numero binario (supportato solo per gli item dipendenti); **JSON** - dati JSON strutturati, memorizzati nativamente nel database (non supportato per gli item calcolati). Gli item che restituiscono valori con tipo di dati binario o JSON non sono supportati nelle formule degli item calcolati o nelle espressioni dei trigger. Consultare Limiti dei dati degli item per i limiti di memorizzazione per tipo di dati e backend del database. Per gli item che restituiscono dati in un unico formato specifico, viene selezionato automaticamente un tipo di dati corrispondente; ad esempio, `system.cpu.load` e `system.cpu.util` restituiscono entrambi un numero in virgola mobile a 64 bit, quindi viene selezionato automaticamente *Numeric (float)*.
Host interface	Selezionare l'interfaccia dell'host. Questo campo è disponibile quando si modifica un item a livello di host.
Units	Se viene impostato un simbolo di unità, Zabbix applica una post-elaborazione al valore ricevuto dell'item e lo visualizza con il suffisso dell'unità specificata. Simboli di unità supportati con formattazione speciale (ed esempi di valore ricevuto → valore visualizzato): **B** - byte (1024 → 1 KB) **Bps** - byte al secondo (1024 → 1 KBps) **s** - secondi, visualizzati utilizzando fino a tre unità di tempo più grandi diverse da zero (881764 → 10d 4h 56m) **uptime** - tempo trascorso nel formato hh:mm:ss o N giorni, hh:mm:ss (881764 → 10 days, 04:56:04) **unixtime** - timestamp Unix, formattato come yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); per una formattazione corretta, il valore ricevuto dell'item deve essere *Numeric (unsigned)*. Per altre unità (come Hz, W e così via), se il valore ricevuto supera 1000, viene diviso per 1000 e visualizzato con il prefisso corrispondente (5000 → 5 KHz, 881764 → 881.76 KW). Se il simbolo dell'unità è preceduto da `!`, la conversione dell'unità e l'aggiunta del prefisso vengono disabilitate (1024 !B → 1024 B, 61 !s → 61 s). Per ulteriori esempi e dettagli sui simboli delle unità e sulla conversione delle unità, consultare Suffissi dei valori degli item.
Update interval	Recupera un nuovo valore per questo item ogni N secondi. L'intervallo di aggiornamento massimo consentito è di 86400 secondi (1 giorno). Sono supportati i suffissi temporali, ad esempio 30s, 1m, 2h, 1d. Sono supportate le macro utente. Una singola macro deve occupare l'intero campo. Non sono supportate più macro nello stesso campo o macro combinate con testo. *Nota*: l'intervallo di aggiornamento può essere impostato su '0' solo se esistono intervalli personalizzati con un valore diverso da zero. Se impostato su '0' e se esiste un intervallo personalizzato (flessibile o pianificato) con un valore diverso da zero, l'item verrà interrogato per la durata dell'intervallo personalizzato. *Nota* che il primo polling dell'item dopo la sua attivazione o dopo la modifica dell'intervallo di aggiornamento potrebbe avvenire prima del valore configurato. I nuovi item verranno controllati entro 60 secondi dalla loro creazione, a meno che non abbiano un intervallo di aggiornamento Scheduling o Flexible e *Update interval* sia impostato su 0. Un item passivo esistente può essere interrogato immediatamente per ottenere il valore premendo il pulsante *Execute now*.

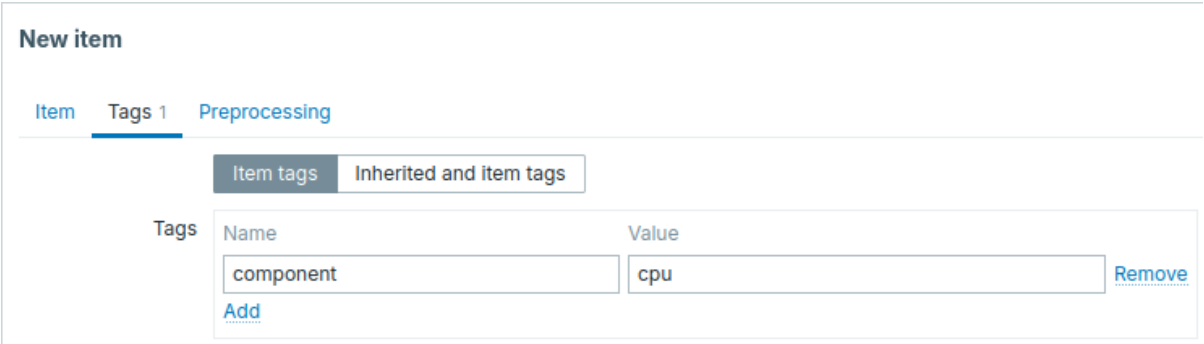
Parametro	Descrizione
Custom intervals	È possibile creare regole personalizzate per il controllo dell'item: **Flexible** - crea un'eccezione a *Update interval* (intervallo con frequenza diversa). **Scheduling** - crea una pianificazione personalizzata per il polling. Per informazioni dettagliate, consultare Intervalli personalizzati. Sono supportati i suffissi temporali nel campo *Interval*, ad esempio 30s, 1m, 2h, 1d. Sono supportate le macro utente. Una singola macro deve occupare l'intero campo. Non sono supportate più macro nello stesso campo o macro combinate con testo.
Timeout	Imposta il timeout del controllo dell'item (disponibile per i tipi di item supportati). Selezionare l'opzione di timeout: **Global** - viene utilizzato il timeout del proxy/globale (visualizzato nel campo *Timeout* disabilitato). **Override** - viene utilizzato un timeout personalizzato (impostato nel campo *Timeout*; intervallo consentito: 1 - 600s). Sono supportati i suffissi temporali, ad esempio 30s, 1m, e le macro utente. Facendo clic sul collegamento *Timeouts* è possibile configurare i timeout del proxy o i timeout globali (se non viene utilizzato un proxy). Il collegamento *Timeouts* è visibile solo agli utenti di tipo *Super admin* con autorizzazioni per le sezioni frontend *Administration* > *General* o *Administration* > *Proxies*.
History	Selezionare una delle opzioni: **Do not store** - la cronologia dell'item non viene memorizzata. Utile per gli item master se solo gli item dipendenti devono conservare la cronologia. Questa impostazione non può essere sovrascritta dalle impostazioni globali dell'housekeeper. **Store up to** - specificare la durata di conservazione della cronologia dettagliata nel database (da 1 ora a 25 anni). I dati meno recenti verranno rimossi dall'housekeeper. Memorizzata in secondi. Sono supportati i suffissi temporali, ad esempio 2h, 1d. Sono supportate le macro utente. Il valore *Store up to* può essere sovrascritto globalmente in *Administration* > *Housekeeping*. Se esiste un'impostazione globale di sovrascrittura, viene visualizzata un'icona informativa arancione. Posizionando il mouse sull'icona, viene visualizzato un messaggio di avviso, ad esempio *Overridden by global housekeeper settings (1d)*. È consigliabile conservare i valori registrati per il minor tempo possibile, per ridurre le dimensioni della cronologia dei valori nel database. Invece di memorizzare una cronologia lunga dei valori, è possibile conservare più a lungo i dati delle tendenze. Vedere anche Cronologia e tendenze.
Trends	Selezionare una delle opzioni: **Do not store** - le tendenze non vengono memorizzate. Questa impostazione non può essere sovrascritta dalle impostazioni globali dell'housekeeper. **Store up to** - specificare la durata di conservazione nel database della cronologia aggregata (minimo, massimo, media e conteggio orari) (da 1 giorno a 25 anni). I dati meno recenti verranno rimossi dall'housekeeper. Memorizzata in secondi. Sono supportati i suffissi temporali, ad esempio 24h, 1d. Sono supportate le macro utente. Il valore *Store up to* può essere sovrascritto globalmente in *Administration* > *Housekeeping*. Se esiste un'impostazione globale di sovrascrittura, viene visualizzata un'icona informativa arancione. Posizionando il mouse sull'icona, viene visualizzato un messaggio di avviso, ad esempio *Overridden by global housekeeper settings (7d)*. *Nota:* la conservazione delle tendenze non è disponibile per i dati non numerici: character, log e text. Vedere anche Cronologia e tendenze.
Value mapping	Applica la mappatura dei valori a questo item. La mappatura dei valori non modifica i valori ricevuti, ma serve solo per visualizzare i dati. Funziona con gli item *Numeric(unsigned)*, *Numeric(float)* e *Character*. Ad esempio, "Windows service states".

Parametro	Descrizione
Log time format	Disponibile solo per gli item di tipo **Log**. Segnaposto supportati: **y**: *anno (1970-2038)*. **M**: *mese (01-12)*. **d**: *giorno (01-31)*. **h**: *ora (00-23)*. **m**: *minuto (00-59)*. **s**: *secondo (00-59)*. Se lasciato vuoto, il timestamp verrà impostato su 0 nel formato Unix, che rappresenta il 1° gennaio 1970. Ad esempio, si consideri la seguente riga dal file di log di Zabbix agent: " 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)." Inizia con sei posizioni di caratteri per il PID, seguite da data, ora e dal resto del messaggio. Il formato dell'ora del log per questa riga sarebbe "pppppp:yyyyMMdd:hhmmss". Si noti che i caratteri "p" e ":" sono segnaposto e possono essere qualsiasi carattere eccetto "yMdhms".
Populates host inventory field	È possibile selezionare un campo dell'inventario dell'host che verrà popolato dal valore dell'item. Funziona se per l'host è abilitato il popolamento automatico dell'inventario e con gli item *Numeric(unsigned)*, *Numeric(float)*, *Character* e *Text*.
Description	Inserire una descrizione dell'item. Sono supportate le macro utente.
Enabled	Selezionare la casella di controllo per abilitare l'item, in modo che venga elaborato. Quando si disabilita l'item, questo viene rimosso immediatamente dalla cache della cronologia (ad eccezione del suo ultimo valore, che viene conservato per i log).
Latest data	Fare clic sul collegamento per visualizzare i dati più recenti dell'item. Questo collegamento è disponibile solo quando si modifica un item già esistente.

Note:
I campi specifici del tipo di item sono descritti nelle pagine corrispondenti.

Note:
Quando si modifica a livello di host un item a livello di template esistente, alcuni campi sono di sola lettura. È possibile utilizzare il collegamento nell'intestazione del modulo per passare al livello del template e modificarli, tenendo presente che le modifiche a livello del template modificheranno l'item per tutti gli host a cui è collegato il template.

La scheda **Tags** consente di definire i tag a livello di item.



Preprocessing del valore dell'item

La scheda **Preprocessing** consente di definire regole di trasformazione per i valori ricevuti.

Test dell'item

Attention:
Per eseguire il test di un item, assicurati che l'ora di sistema sul server e sul proxy sia sincronizzata. Nel caso in cui l'ora del server sia indietro, il test dell'item potrebbe restituire il messaggio di errore "The task has been expired." Tuttavia, l'impostazione di fusi orari diversi sul server e sul proxy non influisce sul risultato del test.

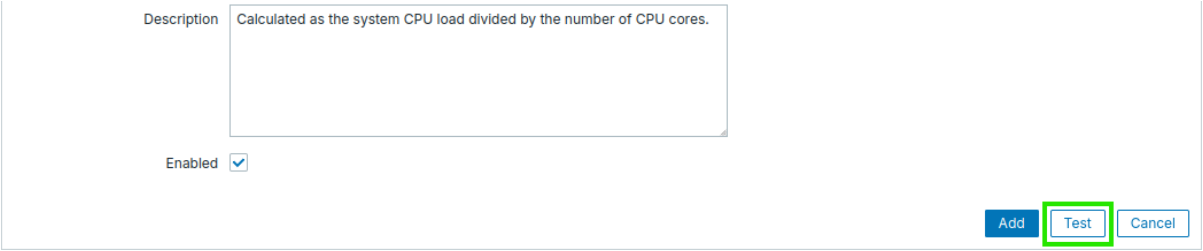
È possibile testare un item e, se configurato correttamente, ottenere in risposta un valore reale. Il test può essere eseguito anche prima che un item venga salvato.

Il test è disponibile per gli item di host e template, i prototipi di item e le regole di discovery a basso livello. Il test non è disponibile per gli item attivi.

Il test degli item è disponibile per i seguenti tipi di item passivi:

- Zabbix agent
- SNMP agent (v1, v2, v3)
- IPMI agent
- SSH checks
- Telnet checks
- JMX agent
- Simple checks (except `icmpping*`, `vmware.*` items)
- Zabbix internal
- Calculated items
- External checks
- Database monitor
- HTTP agent
- Script
- Browser

Per testare un item, fai clic sul pulsante *Test* in fondo al modulo di configurazione dell'item. Nota che il pulsante *Test* sarà disabilitato per gli item che non possono essere testati (come i controlli attivi, i simple checks esclusi).

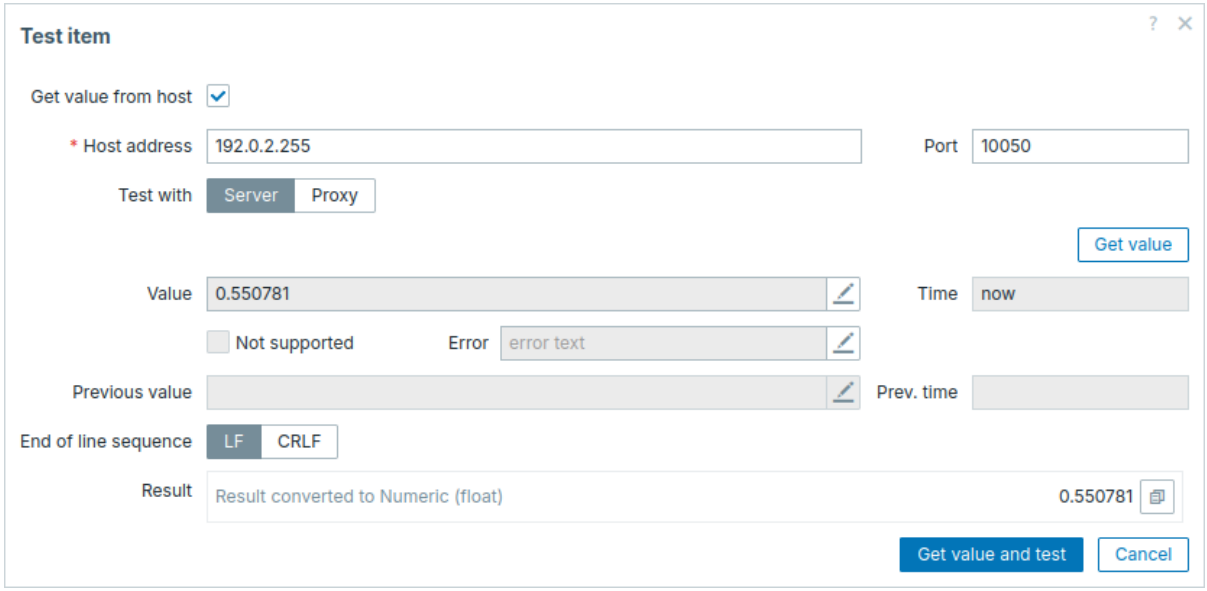


Il modulo di test dell'item contiene campi per i parametri host richiesti (indirizzo host, porta, test con server/proxy (nome del proxy)) e dettagli specifici dell'item (come la community SNMPv2 o le credenziali di sicurezza SNMPv3). Questi campi sono sensibili al contesto:

- I valori vengono precompilati quando possibile, ad esempio per gli item che richiedono un agent, prendendo le informazioni dall'interfaccia agent selezionata dell'host.
- I valori devono essere compilati manualmente per gli item dei template.
- I valori delle macro in testo normale vengono risolti.
- I campi in cui il valore (o parte del valore) è un segreto o una macro Vault sono vuoti e devono essere inseriti manualmente. Se un parametro dell'item contiene un valore di macro segreta, viene visualizzato il seguente messaggio di avviso: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- I campi sono disabilitati quando non sono necessari nel contesto del tipo di item (ad esempio, il campo indirizzo host e il campo proxy sono disabilitati per gli item calcolati)

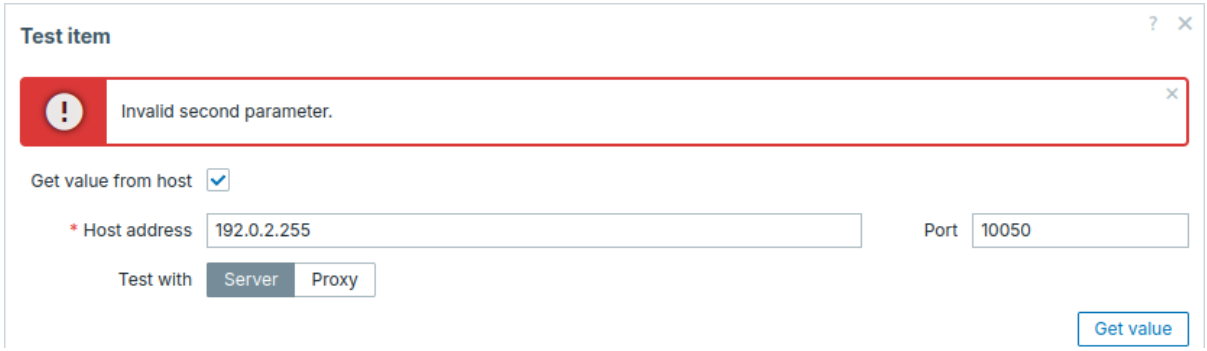
Per testare l'item, fai clic su *Get value*. Se il valore viene recuperato correttamente, verrà inserito nel campo *Value*, spostando il valore corrente (se presente) nel campo *Previous value* e calcolando anche il campo *Prev. time*, cioè la differenza di tempo tra i due valori (clic) e tentando di rilevare una sequenza EOL e passare a CRLF se viene rilevato "\n\r" nel valore recuperato.

Fai clic su *Get value and test* per testare il preprocessing.



I valori recuperati da un host e i risultati del test vengono troncati a una dimensione massima di 512KB quando vengono inviati al frontend. Se un risultato del test viene troncato, viene visualizzata un'icona di avviso; passa il mouse sopra di essa per i dettagli. Se un valore viene troncato, potrebbe non superare la convalida del tipo di dato nei test successivi che confrontano valori grandi (ad esempio, JSON); in questi casi, viene visualizzato un messaggio di errore. Nota che i dati superiori a 512KB vengono comunque elaborati completamente da Zabbix server.

Se la configurazione non è corretta, viene visualizzato un messaggio di errore che descrive la possibile causa.



Un valore recuperato con successo da un host può essere utilizzato anche per testare i passi di preprocessing.

Pulsanti del modulo

I pulsanti nella parte inferiore del modulo consentono di eseguire diverse operazioni.

Add	Aggiunge un item. Questo pulsante è disponibile solo per i nuovi item.
Update	Aggiorna le proprietà di un item.
Clone	Crea un altro item basato sulle proprietà dell'item corrente.
Execute now	Esegue immediatamente un controllo per ottenere un nuovo valore dell'item. Supportato solo per i controlli **passivi** (vedere ulteriori dettagli). *Nota*: quando si verifica immediatamente un valore, la cache di configurazione non viene aggiornata; pertanto, il valore non rifletterà le modifiche più recenti alla configurazione dell'item.
Test	Verifica se la configurazione dell'item è corretta ottenendo un valore.
Clear history and trends	Elimina la cronologia e i trend dell'item.
Delete	Elimina l'item.
Cancel	Annulla la modifica delle proprietà dell'item.

Limiti dei dati degli item

I limiti dei dati degli item dipendono dal tipo di dato dell'item e dal backend del database.

I valori numerici (unsigned) vengono memorizzati così come sono. Se viene ricevuto un valore in virgola mobile, viene memorizzata solo la parte intera (ad esempio, 1.23 → 1).

I valori numerici (float) vengono memorizzati così come sono e supportano approssimativamente da 15 a 17 cifre di precisione, con un intervallo di circa -1.79E+308 a 1.79E+308. Sono supportati anche i valori in notazione scientifica (ad esempio, 1.23E+7, 1e308, 1.1E-4).

I valori di testo vengono troncati prima di essere memorizzati per corrispondere al limite del tipo di valore del database:

Database	Type of information		
	Character	**Log**	**Text**
MySQL	255 characters	65536 bytes	65536 bytes
PostgreSQL	255 characters	65536 characters	65536 characters
SQLite (only Zabbix proxy)	255 characters	65536 characters	65536 characters

I valori binari sono limitati a 16MiB (16777216 bytes). I valori che superano questo limite vengono scartati e viene visualizzato un errore corrispondente nel frontend.

I valori con tipo di dato JSON sono limitati a 128MiB (134217728 bytes). I valori che superano questo limite vengono scartati e viene visualizzato un errore corrispondente nel frontend. Per MySQL/MariaDB, considera di configurare partizioni giornaliere per la tabella `history_json` (usata per memorizzare i valori JSON), poiché può crescere rapidamente e bloccare il housekeeper per un periodo prolungato. Se il tuo caso d'uso prevede valori JSON di grandi dimensioni (1MiB o più), consulta come configurare il sistema per il supporto di valori JSON di grandi dimensioni.

Quando Zabbix scrive qualsiasi dato di un item nel database, utilizza query INSERT e le registra (se `DebugLevel` è impostato su 4 o 5). Le voci di log di grandi dimensioni vengono troncate a 64KB.

Limite degli script personalizzati

La lunghezza disponibile degli script personalizzati dipende dal database utilizzato:

Database	Limite in caratteri	Limite in byte
MySQL	65535	65535
PostgreSQL	65535	non limitato
SQLite (solo Zabbix proxy)	65535	non limitato

Timeout dell'item

Il timeout dell'item specifica per quanto tempo Zabbix deve attendere prima di interrompere il controllo considerandolo non riuscito.

Quando viene raggiunto il timeout, il controllo viene interrotto anche se il recupero dei dati non è ancora terminato. Se i dati vengono ricevuti solo parzialmente, l'item diventa non supportato e viene registrato un messaggio di errore (ad esempio, se i dati vengono raccolti correttamente solo per uno dei più OID in un controllo SNMP).

Per molti tipi di item, è possibile impostare timeout **flessibili** per gli item:

- per singolo item
- per tipo di item (a livello di proxy)
- per tipo di item (a livello globale)

Un timeout personalizzato *per singolo item* è utile se si desidera specificare un timeout più lungo per un item specifico, mantenendo bassi i timeout degli altri item.

Precedenza dei timeout

1. Il timeout del singolo item ha la precedenza su qualsiasi altro timeout.
2. I timeout a livello di proxy hanno la precedenza sui timeout globali.

Il timeout della configurazione di Zabbix server o proxy non ha **alcun** ruolo nei controlli per i quali sono impostati timeout flessibili.

Supporto per timeout flessibili

I timeout flessibili degli item sono supportati per questi tipi di item:

- Zabbix agent (sia controlli passivi che attivi)
- SNMP agent (eccetto i controlli SNMP legacy[1])
- Simple check (eccetto `icmpping*` e gli item VMware[2])
- SSH agent
- Telnet agent
- External check
- Database monitor
- Script
- HTTP agent
- Browser

[1] Per i controlli SNMP legacy (polling di un singolo OID), si applicano le impostazioni di timeout di server o proxy.
 [2] Per gli item `icmpping*`, il valore del timeout è specificato direttamente nella chiave dell'item; si applica inoltre un timeout massimo possibile hardcoded di 600 secondi. Gli item di monitoraggio VMware usano il proprio parametro VMwareTimeout dalla configurazione di server o proxy.

Item non supportati

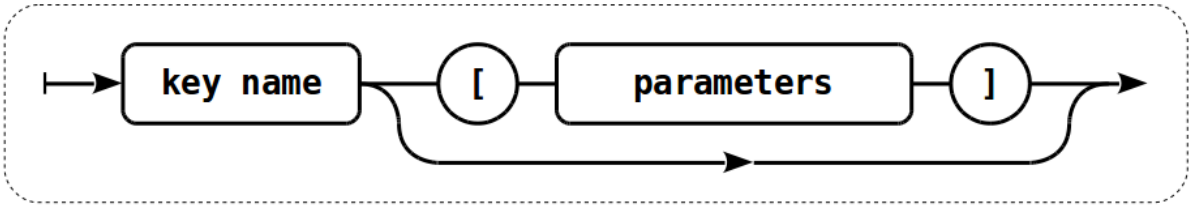
Gli item non supportati vengono segnalati con stato *Non supportato* e continuano a essere ricontrollati secondo il loro *Intervallo di aggiornamento* standard.

Un item diventa non supportato se il suo valore non può essere recuperato per qualche motivo (ad esempio, errore di connessione, nessun poller configurato per elaborare l'item). Inoltre, gli item che non ricevono dati storici rimangono nello stato *Non supportato*. Si tratta di un comportamento previsto: un item passa allo stato *Normale* solo quando riceve nuovi dati storici validi.

Come i problemi, gli item non supportati vengono rivalutati solo quando vengono ricevuti nuovi dati, anche se per quell'item non sono più disponibili dati storici. In altre parole, item e trigger cambiano stato esclusivamente alla ricezione di nuovi dati. Tuttavia, se un'espressione di trigger include una funzione date and time e/o nodata(), il trigger verrà ricalcolato ogni 30 secondi (per i dettagli, vedere tempo di calcolo dei trigger).

Formato della chiave dell'item

Il formato della chiave dell'item, inclusi i parametri della chiave, deve seguire le regole di sintassi. Le illustrazioni seguenti mostrano la sintassi supportata. Gli elementi e i caratteri consentiti in ogni punto possono essere determinati seguendo le frecce - se un blocco può essere raggiunto tramite la linea, è consentito; in caso contrario - non è consentito.



Per costruire una chiave dell'item valida, si inizia specificando il nome della chiave, poi c'è la possibilità di avere o meno dei parametri - come mostrato dalle due linee che possono essere seguite.

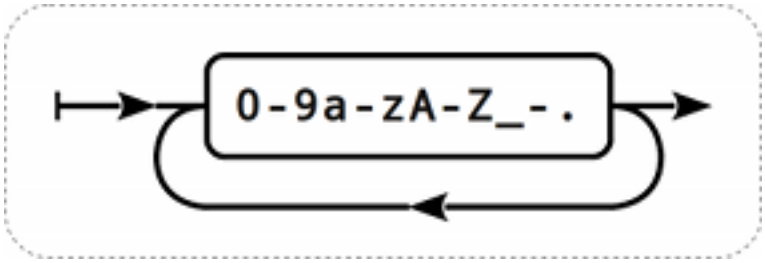
Nome della chiave

Il nome della chiave stesso ha un insieme limitato di caratteri consentiti, che si susseguono semplicemente uno dopo l'altro. I caratteri consentiti sono:

`0-9a-zA-Z_-.`

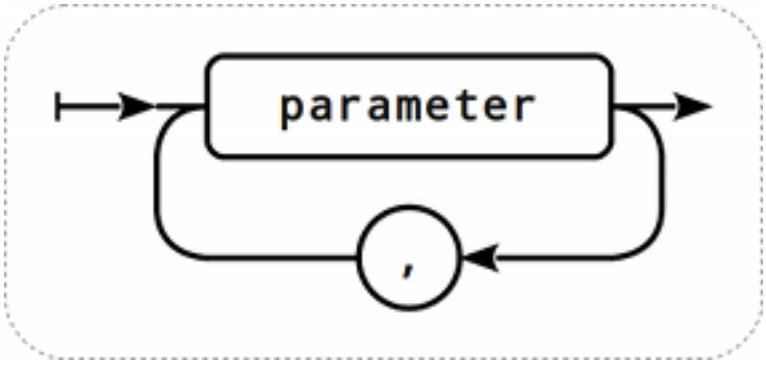
Il che significa:

- tutti i numeri;
- tutte le lettere minuscole;
- tutte le lettere maiuscole;
- trattino basso;
- trattino;
- punto.

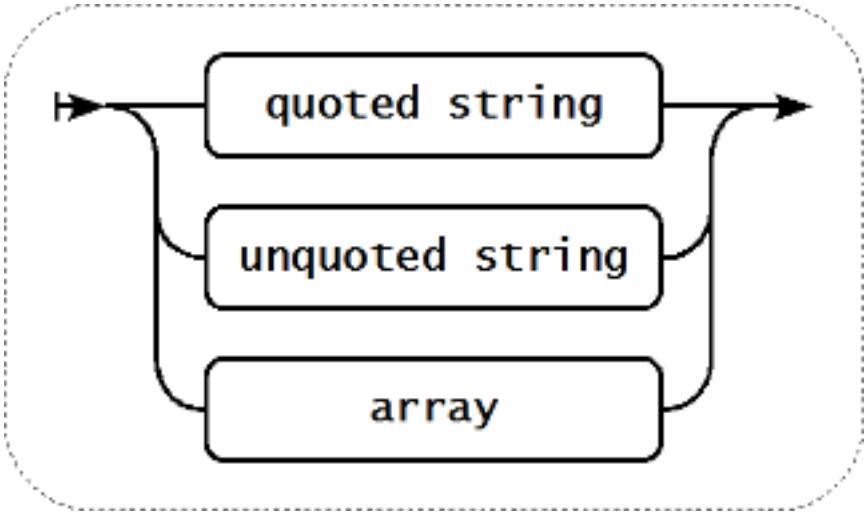


Parametri chiave

Una chiave item può avere più parametri separati da virgole.



Ogni parametro della chiave può essere una stringa tra virgolette, una stringa non tra virgolette oppure un array.

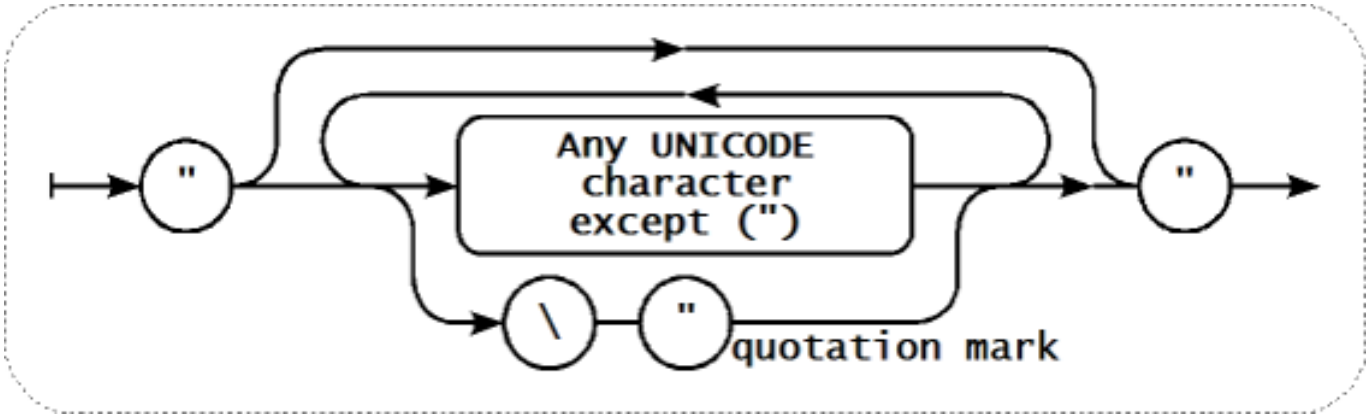


Il parametro può anche essere lasciato vuoto, utilizzando così il valore predefinito. In questo caso, deve essere aggiunto il numero appropriato di virgole se vengono specificati ulteriori parametri. Per esempio, la chiave item **icmpping[„200„500]** specifica che l'intervallo tra i singoli ping è di 200 millisecondi, il timeout è di 500 millisecondi, e tutti gli altri parametri vengono lasciati ai valori predefiniti.

È possibile includere macro nei parametri. Queste possono essere macro utente o alcune delle macro integrate. Per vedere quali macro integrate specifiche sono supportate nei parametri della chiave item, cerca nella pagina Macro supportate "item key parameters".

Parametro - stringa tra virgolette

Se il parametro della chiave è una stringa tra virgolette, è consentito qualsiasi carattere Unicode. Se la stringa del parametro della chiave contiene un segno di virgolette, questo parametro deve essere racchiuso tra virgolette e ogni segno di virgolette che fa parte della stringa del parametro deve essere preceduto da un carattere backslash (\). Se la stringa del parametro della chiave contiene una virgola, questo parametro deve essere racchiuso tra virgolette.

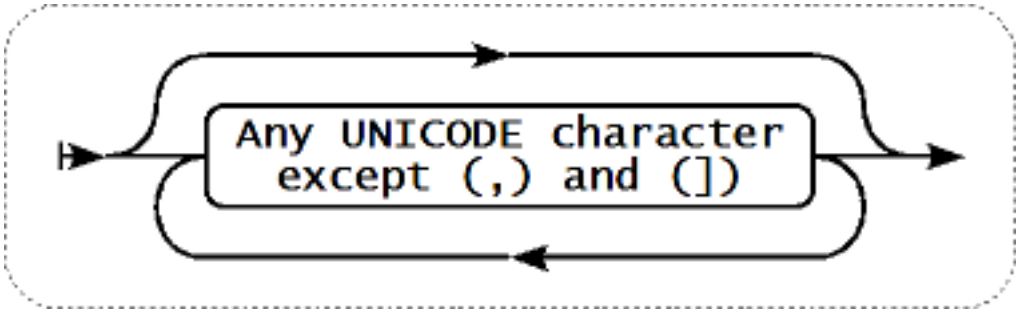


Warning:
Per racchiudere tra virgolette i parametri della chiave dell'item, utilizzare solo le virgolette doppie. Le virgolette singole non sono supportate.

Attention:
Gli array di parametri multilivello, ad esempio `[a,[b,[c,d]],e]`, non sono consentiti.

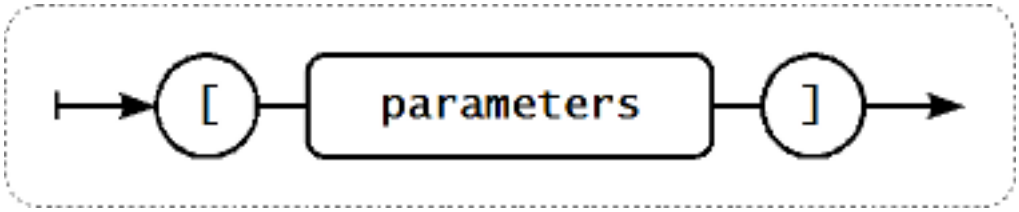
Parametro - stringa senza virgolette

Se il parametro della chiave è una stringa senza virgolette, è consentito qualsiasi carattere Unicode tranne la virgola e la parentesi quadra destra (]). Un parametro senza virgolette non può iniziare con una parentesi quadra sinistra ([).



Parametro - array

Se il parametro key è un array, viene nuovamente racchiuso tra parentesi quadre, dove i singoli parametri sono riportati in conformità con le regole e la sintassi per la specifica di parametri multipli.



Intervalli personalizzati

Panoramica

È possibile creare regole personalizzate relative agli orari in cui viene verificato un item. I due metodi disponibili sono gli *intervalli flessibili*, che consentono di ridefinire l'intervallo di aggiornamento predefinito, e la *pianificazione*, che permette di eseguire il controllo di un item a un'ora specifica o secondo una sequenza di orari.

Intervalli flessibili

Gli intervalli flessibili consentono di ridefinire l'intervallo di aggiornamento predefinito per periodi di tempo specifici. Un intervallo flessibile è definito con *Intervallo* e *Periodo*, dove:

- *Intervallo* - l'intervallo di aggiornamento per il periodo di tempo specificato. Sono supportati i suffissi temporali, ad esempio 30s, 1m, 2h, 1d.
- *Periodo* - il periodo di tempo durante il quale l'intervallo flessibile è attivo (consultare i periodi di tempo per una descrizione dettagliata del formato di *Periodo*)

Se più intervalli flessibili si sovrappongono, per il periodo di sovrapposizione viene utilizzato il valore di *Intervallo* più piccolo. Si noti che, se il valore più piccolo degli intervalli flessibili sovrapposti è '0', non verrà eseguito alcun polling. Al di fuori degli intervalli flessibili viene utilizzato l'intervallo di aggiornamento predefinito.

Si noti che, se l'intervallo flessibile è uguale alla durata del periodo, l'item verrà controllato esattamente una volta. Se l'intervallo flessibile è maggiore del periodo, l'item potrebbe essere controllato una volta oppure potrebbe non essere controllato affatto (pertanto, questa configurazione non è consigliata). Se l'intervallo flessibile è minore del periodo, l'item verrà controllato almeno una volta.

Se l'intervallo flessibile è impostato su '0', l'item non viene sottoposto a polling durante il periodo dell'intervallo flessibile e il polling riprende in base all'*Intervallo di aggiornamento* predefinito al termine del periodo. Esempi:

Intervallo	Periodo	Descrizione
10	1-5,09:00-18:00	L'item verrà controllato ogni 10 secondi durante l'orario lavorativo.
0	1-7,00:00-7:00	L'item non verrà controllato durante la notte.
0	7-7,00:00-24:00	L'item non verrà controllato la domenica.

Intervallo	Periodo	Descrizione
60	1-7,12:00-12:01	L'item verrà controllato ogni giorno alle 12:00. Si noti che questa configurazione veniva utilizzata come soluzione alternativa per i controlli pianificati; per tali controlli è consigliato utilizzare gli intervalli di pianificazione.

Intervalli di pianificazione

Gli intervalli di pianificazione vengono utilizzati per controllare gli item in momenti specifici. Mentre gli intervalli flessibili sono progettati per ridefinire l'intervallo predefinito di aggiornamento dell'item, gli intervalli di pianificazione vengono utilizzati per specificare una pianificazione di controllo indipendente, eseguita in parallelo.

Un intervallo di pianificazione è definito come: `md<filter>wd<filter>h<filter>m<filter>s<filter>` dove:

- **md** - giorni del mese
- **wd** - giorni della settimana
- **h** - ore
- **m** - minuti
- **s** - secondi

`<filter>` viene utilizzato per specificare i valori per il relativo prefisso (giorni, ore, minuti, secondi) ed è definito come: `[<from>[-<to>]][/<step>][,<filter>]` dove:

- `<from>` e `<to>` definiscono l'intervallo dei valori corrispondenti (inclusi). Se `<to>` viene omesso, il filtro corrisponde a un intervallo `<from> - <from>`. Se anche `<from>` viene omesso, il filtro corrisponde a tutti i valori possibili.
- `<step>` definisce i salti del valore numerico all'interno dell'intervallo. Per impostazione predefinita, `<step>` ha valore 1, il che significa che tutti i valori dell'intervallo definito corrispondono.

Sebbene le definizioni dei filtri siano facoltative, deve essere utilizzato almeno un filtro. Un filtro deve avere un intervallo oppure il valore di *<step>* definito.

Un filtro vuoto corrisponde a '0' se non è definito alcun filtro di livello inferiore, oppure a tutti i valori possibili negli altri casi. Ad esempio, se il filtro delle ore viene omesso, corrisponderà solo l'ora '0', a condizione che anche i filtri dei minuti e dei secondi siano omessi; altrimenti, un filtro ore vuoto corrisponderà a tutti i valori delle ore.

I valori validi di `<from>` e `<to>` per il rispettivo prefisso del filtro sono:

Prefix	Description	*<from>*	*<to>*
md	Giorni del mese	1-31	1-31
wd	Giorni della settimana	1-7	1-7
h	Ore	0-23	0-23
m	Minuti	0-59	0-59
s	Secondi	0-59	0-59

Il valore `<from>` deve essere minore o uguale al valore `<to>`. Il valore `<step>` deve essere maggiore o uguale a 1 e minore o uguale a `<to>` - `<from>`.

I valori a una sola cifra per giorni del mese, ore, minuti e secondi possono essere preceduti da 0. Ad esempio, `md01-31` e `h/02` sono intervalli validi, ma `md01-031` e `wd01-07` non lo sono.

Nel frontend di Zabbix, più intervalli di pianificazione vengono inseriti in righe separate. Nella API di Zabbix, vengono concatenati in una singola stringa con un punto e virgola `;` come separatore.

Se un orario corrisponde a più intervalli, viene eseguito una sola volta. Ad esempio, `wd1h9;h9` verrà eseguito una sola volta il lunedì alle 9:00.

Esempi:

Interval	Will be executed
m0-59	ogni minuto
h9-17/2	ogni 2 ore a partire dalle 9:00 (9:00, 11:00 ...)
m0,30 or m/30	ogni ora alle hh:00 e hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	ogni cinque minuti
wd1-5h9	ogni lunedì-venerdì alle 9:00

Interval	Will be executed
wd1-5h9-18	ogni lunedì-venerdì alle 9:00,10:00,...,18:00
h9,10,11 or h9-11	ogni giorno alle 9:00, 10:00 e 11:00
md1h9m30	ogni 1° giorno di ogni mese alle 9:30
md1wd1h9m30	ogni 1° giorno di ogni mese alle 9:30 se è lunedì
h9m/30	ogni giorno alle 9:00, 9:30
h9m0-59/30	ogni giorno alle 9:00, 9:30
h9,10m/30	ogni giorno alle 9:00, 9:30, 10:00, 10:30
h9-10m30	ogni giorno alle 9:30, 10:30
h9m10-40/30	ogni giorno alle 9:10, 9:40
h9,10m10-40/30	ogni giorno alle 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	ogni giorno alle 9:10, 9:40, 10:10, 10:40
h9m10-40	ogni giorno alle 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	ogni giorno alle 9:10, 9:11, 9:12, ... 9:40
h9-12,15	ogni giorno alle 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0	ogni giorno alle 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	ogni giorno alle 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	ogni giorno alle 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10 (*sintassi specifica dell'API*)	ogni giorno alle 9:00, 9:30, 10:00
h9m/30 h10 (*aggiungere questa come un'altra riga nel frontend*)	ogni giorno alle 9:00, 9:30, 10:00

Allineamento dei fusi orari per proxy e agent

Si noti che i proxy e gli agent di Zabbix utilizzano i rispettivi fusi orari locali durante l'elaborazione degli intervalli di pianificazione.

Per questo motivo, quando gli intervalli di pianificazione vengono applicati agli item monitorati da Zabbix proxy o agli item attivi dell'agent, si consiglia di impostare il fuso orario dei rispettivi proxy o agent uguale a quello del server Zabbix; in caso contrario, la queue potrebbe segnalare in modo errato i ritardi degli item.

Il fuso orario per Zabbix proxy o agent può essere impostato utilizzando la variabile d'ambiente `TZ` nel file unit di `systemd`:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

Pre-elaborazione del valore dell'item

Panoramica

La preelaborazione consente di applicare trasformazioni ai valori degli item ricevuti prima di salvarli nel database. Queste trasformazioni/fasi di preelaborazione vengono eseguite da Zabbix server o proxy (se gli item sono monitorati dal proxy).

Questa funzionalità supporta una varietà di casi d'uso, ad esempio:

- conversione dei byte in bit (ad esempio, moltiplicando i valori del traffico di rete per "8");
- calcolo di statistiche al secondo per valori incrementali crescenti;
- applicazione di espressioni regolari per estrarre o modificare valori;
- esecuzione di script personalizzati sui valori;
- scarto dei valori invariati per ottimizzare l'archiviazione nel database.

Per un item è possibile configurare una o più fasi di preelaborazione. Queste fasi vengono eseguite nell'ordine in cui sono configurate.

Attention:
Se una fase di preelaborazione fallisce, un item diventa unsupported. Questo può essere evitato tramite la gestione degli errori *Custom on fail* (disponibile per la maggior parte delle trasformazioni), che consente di scartare i valori o impostare valori personalizzati.

 Per gli item di log, i metadati del log (senza valore) reimposteranno sempre lo stato unsupported dell'item, rendendolo nuovamente supportato. Ciò avviene anche se l'errore iniziale si è verificato dopo aver ricevuto un valore di log da agent.

Tutti i valori passati alla preelaborazione vengono inizialmente trattati come stringhe. La conversione nel tipo di valore desiderato (come definito nella configurazione dell'item) viene eseguita alla fine della pipeline di preelaborazione. Tuttavia, specifiche fasi

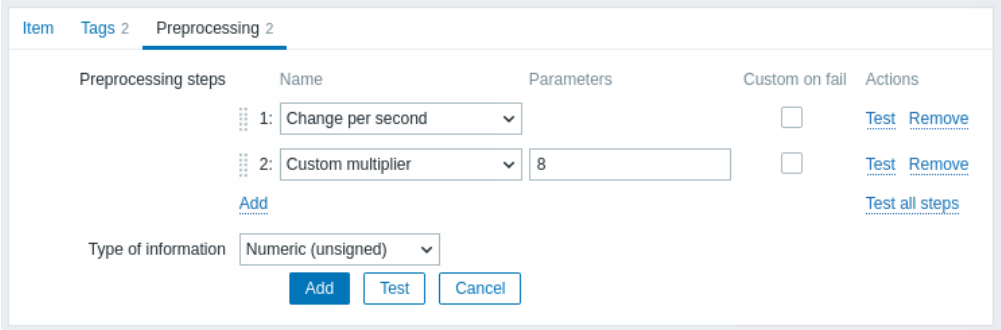
di preelaborazione possono attivare conversioni anticipate, se necessario. Per informazioni tecniche dettagliate, vedere Dettagli della preelaborazione.

Per assicurarti che la configurazione della preelaborazione funzioni come previsto, puoi testarla.

Vedi anche: Esempi di preelaborazione

Configurazione

I passaggi di pre-elaborazione sono definiti nella scheda **Preprocessing** del modulo di configurazione dell'item.



Fare clic su *Add* per selezionare una trasformazione supportata.

Il campo *Type of information* viene visualizzato nella parte inferiore della scheda quando è definito almeno un passaggio di preelaborazione. Se necessario, è possibile modificare il tipo di informazioni senza uscire dalla scheda *Preprocessing*. Per una descrizione dettagliata dei parametri, vedere Creazione di un item.

Trasformazioni supportate

Di seguito sono elencate tutte le trasformazioni supportate.
Fare clic sul nome della trasformazione per visualizzarne i dettagli.

Nome	Descrizione	Tipo
Espressione regolare	Confronta il valore con l'espressione regolare e lo sostituisce con l'output richiesto.	Testo
Sostituzione	Trova la stringa di ricerca e la sostituisce con un'altra stringa (o con nulla).	
Rimozione iniziale e finale	Rimuove i caratteri specificati dall'inizio e dalla fine del valore.	
Rimozione finale	Rimuove i caratteri specificati dalla fine del valore.	
Rimozione iniziale	Rimuove i caratteri specificati dall'inizio del valore.	
XML XPath	Estrae un valore o un frammento dai dati XML utilizzando la funzionalità XPath.	Dati strutturati
JSON Path	Estrae un valore o un frammento dai dati JSON utilizzando la funzionalità JSONPath.	
CSV in JSON	Converte i dati di un file CSV nel formato JSON.	
XML in JSON	Converte i dati in formato XML in JSON.	
Valore SNMP walk	Estrae il valore tramite il nome OID/MIB specificato e applica le opzioni di formattazione.	SNMP
SNMP walk in JSON	Converte i valori SNMP in JSON.	
Valore SNMP get	Applica le opzioni di formattazione al valore SNMP get.	
Moltiplicatore personalizzato	Moltiplica il valore per il valore intero o a virgola mobile specificato.	Aritmetica
Modifica semplice	Calcola la differenza tra il valore corrente e quello precedente.	Modifica
Modifica al secondo	Calcola la velocità di variazione del valore (differenza tra il valore corrente e quello precedente) al secondo.	
Booleano in decimale	Converte il valore dal formato booleano al formato decimale.	Sistemi numerici
Ottale in decimale	Converte il valore dal formato ottale al formato decimale.	
Esadecimale in decimale	Converte il valore dal formato esadecimale al formato decimale.	
JavaScript	Inserisce codice JavaScript.	Script personalizzati
Nell'intervallo	Definisce un intervallo in cui deve rientrare il valore.	Convalida
Corrisponde all'espressione regolare	Specifica un'espressione regolare a cui il valore deve corrispondere.	

Nome	Descrizione	Tipo
Non corrisponde all'espressione regolare	Specifica un'espressione regolare a cui il valore non deve corrispondere.	
Verifica la presenza di un errore in JSON	Verifica la presenza di un messaggio di errore a livello applicativo situato in JSONPath.	
Verifica la presenza di un errore in XML	Verifica la presenza di un messaggio di errore a livello applicativo situato in XPath.	
Verifica la presenza di un errore tramite un'espressione regolare	Verifica la presenza di un messaggio di errore a livello applicativo utilizzando un'espressione regolare.	
Verifica la presenza di un valore non supportato	Verifica se non è stato possibile recuperare alcun valore dell'item.	
Ignora se invariato	Ignora un valore se non è cambiato.	Limitazione
Ignora se invariato con heartbeat	Ignora un valore se non è cambiato entro il periodo di tempo definito.	
Pattern Prometheus	Utilizza la query seguente per estrarre i dati richiesti dalle metriche Prometheus.	Prometheus
Prometheus in JSON	Converte le metriche Prometheus richieste in JSON.	

Se la casella di controllo *Personalizzato in caso di errore* è abilitata per la trasformazione selezionata, è possibile selezionarla e specificare opzioni personalizzate per la gestione degli errori: *Ignora valore* (ignora il valore), *Imposta valore su* (imposta un valore specificato) oppure *Imposta errore su* (imposta un messaggio di errore specificato).
Se un passaggio di pre-elaborazione non riesce, l'item non diventerà non supportato quando è selezionata l'opzione *Ignora valore* o *Imposta valore su*.

Si noti che per i passaggi di pre-elaborazione *Modifica* e *Limitazione*, Zabbix deve ricordare l'ultimo valore per calcolare/confrontare il nuovo valore come richiesto.
Questi valori precedenti vengono gestiti dal gestore della pre-elaborazione.
Se il server Zabbix o il proxy viene riavviato oppure viene apportata una modifica ai passaggi di pre-elaborazione, l'ultimo valore dell'item corrispondente viene reimpostato, con i seguenti risultati:

- per i passaggi *Modifica semplice* e *Modifica al secondo*: il valore successivo verrà ignorato perché non esiste alcun valore precedente da cui calcolare la modifica;
- per i passaggi *Ignora se invariato* e *Ignora se invariato con heartbeat*: il valore successivo non verrà mai ignorato, anche se avrebbe dovuto esserlo in base alle regole di scarto.

Espressione regolare

Confronta il valore con l'espressione regolare e sostituiscilo con l'output richiesto.

Parametri:

- **pattern** - l'espressione regolare;
- **output** - il template di formattazione dell'output. Una sequenza di escape \N (dove N=1...9) viene sostituita con il gruppo corrispondente acquisito. Una sequenza di escape \0 viene sostituita con il testo corrispondente.

Commenti:

- Se non viene trovata alcuna corrispondenza con il valore di input, l'item diventerà non supportato;
- L'espressione regolare supporta l'estrazione di un massimo di 10 gruppi acquisiti con la sequenza \N;
- Per alcuni esempi esistenti, consultare la sezione espressioni regolari.

Sostituisci

Trova la stringa di ricerca e sostituiscila con un'altra stringa (o con niente).

Parametri:

- **stringa di ricerca** - la stringa da trovare e sostituire, con distinzione tra maiuscole e minuscole (obbligatorio);
- **sostituzione** - la stringa con cui sostituire la stringa di ricerca. La stringa di sostituzione può anche essere vuota, consentendo di eliminare la stringa di ricerca quando viene trovata.

Commenti:

- Tutte le occorrenze della stringa di ricerca verranno sostituite;

- È possibile utilizzare sequenze di escape per cercare o sostituire interruzioni di riga, ritorni a capo, tabulazioni e spazi "\n \r \t \s"; il carattere barra rovesciata può essere sottoposto a escape come "\\" e le sequenze di escape possono essere sottoposte a escape come "\\n";
- L'escape delle interruzioni di riga, dei ritorni a capo e delle tabulazioni viene eseguito automaticamente durante la low-level discovery.

Trim

Rimuove i caratteri specificati dall'inizio e dalla fine del valore.

Eliminazione a destra

Rimuove i caratteri specificati dalla fine del valore.

Left trim

Rimuove i caratteri specificati dall'inizio del valore.

XML XPath

Estrae un valore o un frammento dai dati XML utilizzando la funzionalità XPath.

Note:

- Affinché questa opzione funzioni, Zabbix server (o Zabbix proxy) deve essere compilato con il supporto per libxml;
- Gli spazi dei nomi non sono supportati.

Esempi:

```
number(/document/item/value) #estrae '10' da <document><item><value>10</value></item></document>
number(/document/item/@attribute) #estrae '10' da <document><item attribute="10"></item></document>
/document/item #estrae '<item><value>10</value></item>' da <document><item><value>10</value></item></docum
```

JSON Path

Estrai un valore o un frammento dai dati JSON utilizzando la funzionalità JSONPath.

Da CSV a JSON

Converte i dati di un file CSV nel formato JSON.

Per ulteriori informazioni, vedere: Pre-elaborazione da CSV a JSON.

XML in JSON

Convertire i dati dal formato XML a JSON.

Per ulteriori informazioni, vedere: Regole di serializzazione.

Valore SNMP walk

Estrae il valore tramite l'OID/nome MIB specificato e applica le opzioni di formattazione:

- **Invariato** - restituisce la stringa esadecimale come stringa esadecimale non sottoposta a escape (*nota* che i suggerimenti di visualizzazione vengono comunque applicati);
- **UTF-8 da STRINGA esadecimale** - converte la stringa esadecimale in una stringa UTF-8;
- **MAC da STRINGA esadecimale** - convalida la stringa esadecimale come indirizzo MAC e restituisce una stringa con un indirizzo MAC corretto (dove `' '` viene sostituito da `':'`);
- **Intero da BITS** - converte i primi 8 byte di una stringa di bit espressa come sequenza di caratteri esadecimali (ad esempio, "1A 2B 3C 4D") in un intero senza segno a 64 bit. Nelle stringhe di bit più lunghe di 8 byte, i byte successivi verranno ignorati.

SNMP walk in JSON

Converti i valori SNMP in JSON.

Specifica un nome di campo nel JSON e il percorso OID SNMP corrispondente. I valori dei campi verranno popolati con i valori del percorso OID SNMP specificato.

Commenti:

- Sono disponibili opzioni di formattazione dei valori simili a quelle del passaggio *Valore SNMP walk*;
- Puoi utilizzare questo passaggio di pre-elaborazione per il rilevamento degli OID SNMP.

Ottieni valore SNMP

Applica le opzioni di formattazione al valore ottenuto tramite SNMP:

- **UTF-8 da Hex-STRING** - converte la stringa esadecimale in una stringa UTF-8;
- **MAC da Hex-STRING** - convalida la stringa esadecimale come indirizzo MAC e restituisce una stringa con un indirizzo MAC valido (dove ' ' viene sostituito da ':');
- **Intero da BITS** - converte i primi 8 byte di una stringa di bit espressa come sequenza di caratteri esadecimali (ad esempio, "1A 2B 3C 4D") in un intero senza segno a 64 bit. Nelle stringhe di bit più lunghe di 8 byte, i byte successivi verranno ignorati.

Moltiplicatore personalizzato

Moltiplica il valore per il valore intero o a virgola mobile specificato.

Commenti:

- Utilizzare questa opzione per convertire i valori ricevuti in KB, MBps, ecc. in B, Bps. In caso contrario, Zabbix non può impostare correttamente i prefissi (K, M, G, ecc.).
- Notare che, se il tipo di informazione dell'item è *Numeric (unsigned)*, i valori in ingresso con una parte frazionaria verranno troncati (ovvero, '0.9' diventerà '0') prima dell'applicazione del moltiplicatore personalizzato;
- Se si utilizza un moltiplicatore personalizzato o si memorizza il valore come *Change per second* per gli item con il tipo di informazione impostato su *Numeric (unsigned)* e il valore calcolato risultante è effettivamente un numero a virgola mobile, il valore calcolato viene comunque accettato come corretto troncando la parte decimale e memorizzandolo come intero;
- Supportati: notazione scientifica, ad esempio, 1e+70; macro utente e macro LLD; stringhe che includono macro, ad esempio, `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Le macro devono risolversi in un numero intero o a virgola mobile.

Modifica semplice

Calcola la differenza tra il valore corrente e quello precedente.

Commenti:

- Questo passaggio può essere utile per misurare un valore in costante crescita;
- Valutato come **value**-**prev_value**, dove *value* è il valore corrente; *prev_value* è il valore ricevuto in precedenza;
- È consentita una sola operazione di modifica per item ("Modifica semplice" o "Modifica al secondo");
- Se il valore corrente è inferiore a quello precedente, Zabbix scarta tale differenza (non memorizza nulla) e attende un altro valore.

Variazione al secondo

Calcola la velocità di variazione del valore (differenza tra il valore corrente e quello precedente) al secondo.

Commenti:

- Questo passaggio è utile per calcolare la velocità al secondo di un valore in costante crescita;
- Poiché questo calcolo può produrre numeri in virgola mobile, si consiglia di impostare il campo "Tipo di informazioni" su *Numerico (float)*, anche se i valori grezzi in ingresso sono numeri interi. Ciò è particolarmente rilevante per i numeri piccoli, per i quali la parte decimale è importante. Se i valori in virgola mobile sono grandi e possono superare la lunghezza del campo 'float', nel qual caso l'intero valore potrebbe andare perso, si consiglia invece di utilizzare *Numerico (senza segno)*, in modo da eliminare solo la parte decimale.
- Valutato come (**value**-**prev_value**)/(**time**-**prev_time**), dove *value* è il valore corrente; *prev_value* è il valore ricevuto in precedenza; *time* è il timestamp corrente; *prev_time* è il timestamp del valore precedente;
- È consentita una sola operazione di variazione per item ("Variazione semplice" o "Variazione al secondo");
- Se il valore corrente è inferiore a quello precedente, Zabbix scarta tale differenza (non memorizza nulla) e attende un altro valore. Ciò consente di funzionare correttamente, ad esempio, con il riavvolgimento (overflow) dei contatori SNMP a 32 bit.

Boolean to decimal

Converte il valore dal formato booleano a decimale.

Commenti:

- La rappresentazione testuale viene convertita in 0 o 1. Pertanto, 'TRUE' viene memorizzato come 1 e 'FALSE' viene memorizzato come 0. Tutti i valori vengono confrontati senza distinzione tra maiuscole e minuscole. I valori attualmente riconosciuti sono, per *TRUE*: true, t, yes, y, on, up, running, enabled, available, ok, master; per *FALSE*: false, f, no, n, off, down, unused, disabled, unavailable, err, slave. Inoltre, qualsiasi valore numerico diverso da zero viene considerato TRUE, mentre zero viene considerato FALSE.

Da ottale a decimale

Converte il valore dal formato ottale a quello decimale.

Da esadecimale a decimale

Converte il valore dal formato esadecimale a decimale.

JavaScript

Inserire il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona della matita accanto ad esso.

> **Attention:**
> Non utilizzare assegnazioni non dichiarate nel JavaScript di preprocessing. Usare `var` per dichiarare le variabili locali.

Commenti:

- La lunghezza disponibile per JavaScript dipende dal database utilizzato;
- Per ulteriori informazioni, vedere: Preprocessing JavaScript.

Nell'intervallo

Definire un intervallo in cui deve rientrare un valore specificando i valori minimo/massimo (inclusi).

Commenti:

- Sono accettati valori numerici (incluso qualsiasi numero di cifre, una parte decimale facoltativa e una parte esponenziale facoltativa, valori negativi);
- Il valore minimo deve essere inferiore al valore massimo;
- Deve essere presente almeno un valore;
- È possibile utilizzare macro utente e macro di low-level discovery.

Corrisponde all'espressione regolare

Specifica un'espressione regolare a cui deve corrispondere un valore.

Non corrisponde all'espressione regolare

Specificare un'espressione regolare a cui il valore non deve corrispondere.

Verifica della presenza di un errore nel JSON

Verifica la presenza di un messaggio di errore a livello applicativo situato nel JSONPath. Interrompe l'elaborazione se l'operazione ha esito positivo e il messaggio non è vuoto; in caso contrario, continua l'elaborazione con il valore precedente a questo passaggio di pre-elaborazione.

Commenti:

- Questi errori del servizio esterno vengono segnalati all'utente così come sono, senza aggiungere informazioni sul passaggio di pre-elaborazione;
- Non viene segnalato alcun errore se non è possibile analizzare JSON non valido.

Verifica degli errori nell'XML

Verifica la presenza di un messaggio di errore a livello applicativo situato nel percorso XPath. Interrompi l'elaborazione se l'operazione ha esito positivo e il messaggio non è vuoto; in caso contrario, continua l'elaborazione con il valore precedente a questo passaggio di pre-elaborazione.

Commenti:

- Questi errori del servizio esterno vengono segnalati all'utente così come sono, senza aggiungere informazioni sul passaggio di pre-elaborazione;
- Non verrà segnalato alcun errore in caso di mancata analisi di un XML non valido.

Verifica degli errori tramite un'espressione regolare

Verifica la presenza di un messaggio di errore a livello applicativo tramite un'espressione regolare.
Interrompe l'elaborazione se l'operazione ha esito positivo e il messaggio non è vuoto; in caso contrario, continua l'elaborazione con il valore precedente a questo passaggio di pre-elaborazione.

Parametri:

- **pattern** - l'espressione regolare;
- **output** - il template di formattazione dell'output.
  La sequenza di escape \N (dove N=1…9) viene sostituita con il gruppo corrispondente individuato.
  La sequenza di escape \0 viene sostituita con il testo individuato.

Commenti:

- Questi errori del servizio esterno vengono segnalati all'utente così come sono, senza aggiungere informazioni sul passaggio di pre-elaborazione.

Verifica del valore non supportato

Verifica se non è stato possibile recuperare alcun valore dell'item. Specifica come elaborare l'errore in base all'analisi del messaggio di errore restituito.

Parametri:

- **scope** - seleziona l'ambito dell'elaborazione dell'errore:
any error - qualsiasi errore;
error matches - solo l'errore che corrisponde all'espressione regolare specificata in *pattern*;
error does not match - solo l'errore che non corrisponde all'espressione regolare specificata in *pattern*
- **pattern** - l'espressione regolare con cui confrontare l'errore. Se nel parametro scope è selezionato *any error*, questo campo non viene visualizzato. Se visualizzato, questo campo è obbligatorio.

Commenti:

- Normalmente, l'assenza di un valore o l'impossibilità di recuperarlo farebbe diventare l'item non supportato. Questo passaggio di pre-elaborazione consente di modificare questo comportamento.
- Questo passaggio di pre-elaborazione verifica solo se non è stato possibile recuperare alcun valore dell'item. Non verifica, ad esempio, se il tipo del valore recuperato (ad esempio, stringa) corrisponde al tipo di informazione dell'item (ad esempio, numerico); per i dettagli, vedere Esempi di pre-elaborazione. In caso di mancata corrispondenza dei tipi, l'item potrebbe comunque diventare non supportato dopo l'esecuzione di tutti i passaggi di pre-elaborazione. Per verificare una mancata corrispondenza dei tipi, è possibile utilizzare, ad esempio, il passaggio di pre-elaborazione *Moltiplicatore personalizzato*; vedere Esempi di pre-elaborazione.
- L'acquisizione dei gruppi delle espressioni regolari è supportata nei campi *Imposta valore su* o *Imposta errore su*. Utilizzare \N (dove N=1...9) per recuperare l'N-esimo gruppo corrispondente; utilizzare \0 per recuperare il testo corrispondente;
- Questi passaggi vengono sempre eseguiti come primi passaggi di pre-elaborazione e vengono posizionati sopra tutti gli altri dopo il salvataggio delle modifiche all'item;
- Sono supportati più passaggi *Verifica del valore non supportato*, nell'ordine specificato. Un passaggio per *any error* verrà posizionato automaticamente come ultimo passaggio di questo gruppo.

Scarta invariato

Scarta un valore se non è cambiato.

Commenti:

- Se un valore viene scartato, non viene salvato nel database e Zabbix server non ha alcuna conoscenza del fatto che questo valore sia stato ricevuto. Non verrà valutata alcuna espressione di trigger e, di conseguenza, non verranno creati/risolti problemi per i trigger correlati. Le funzioni funzioneranno solo in base ai dati effettivamente salvati nel database. Poiché i trend vengono costruiti in base ai dati nel database, se per un'ora non viene salvato alcun valore, non ci saranno dati di trend nemmeno per quell'ora.
- È possibile specificare una sola opzione di limitazione per item.

Scarta i valori invariati con heartbeat

Scarta un valore se non è cambiato entro il periodo di tempo definito (in secondi).

Commenti:

- Sono supportati valori interi positivi per specificare i secondi (minimo: 1 secondo);
- È possibile utilizzare i suffissi temporali (ad esempio, `30s`, `1m`, `2h`, `1d`);
- È possibile utilizzare macro utente e macro di low-level discovery;
- Se un valore viene scartato, non viene salvato nel database e Zabbix server non è a conoscenza del fatto che questo valore sia stato ricevuto. Non verrà valutata alcuna espressione di trigger e, di conseguenza, non verranno creati o risolti problemi per i trigger correlati. Le funzioni funzioneranno solo in base ai dati effettivamente salvati nel database. Poiché i trend vengono creati in base ai dati presenti nel database, se per un'ora non viene salvato alcun valore, non saranno disponibili nemmeno i dati dei trend per quell'ora.
- È possibile specificare una sola opzione di limitazione per item.

Pattern di Prometheus

Usa la seguente query per estrarre i dati richiesti dalle metriche di Prometheus.

Vedi Controlli Prometheus per maggiori dettagli.

Prometheus in JSON

Converti le metriche Prometheus richieste in JSON.

Vedi Prometheus checks per maggiori dettagli.

Supporto delle macro

Le macro utente e le macro utente con contesto sono supportate in:

- parametri dei passaggi di preprocessing, incluso il codice JavaScript;
- parametri personalizzati di gestione degli errori (campi *Imposta valore a* e *Imposta errore a*).

Note:
Il contesto della macro viene ignorato quando una macro viene sostituita con il suo valore. Il valore della macro viene inserito nel codice così com'è; non è possibile aggiungere un escaping aggiuntivo prima di inserire il valore nel codice JavaScript. Si noti che in alcuni casi ciò può causare errori JavaScript.

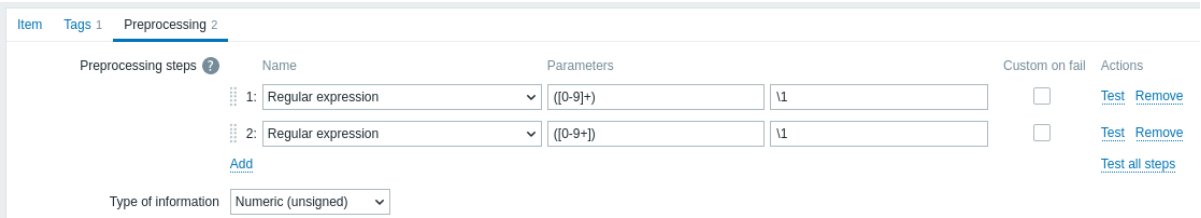
Test

Vedere test di preprocessing.

Test di preprocessing

Test

Testare i passaggi di preprocessing è utile per assicurarsi che pipeline di preprocessing complesse producano i risultati attesi, senza dover attendere che il valore dell'item venga ricevuto e sottoposto a preprocessing.

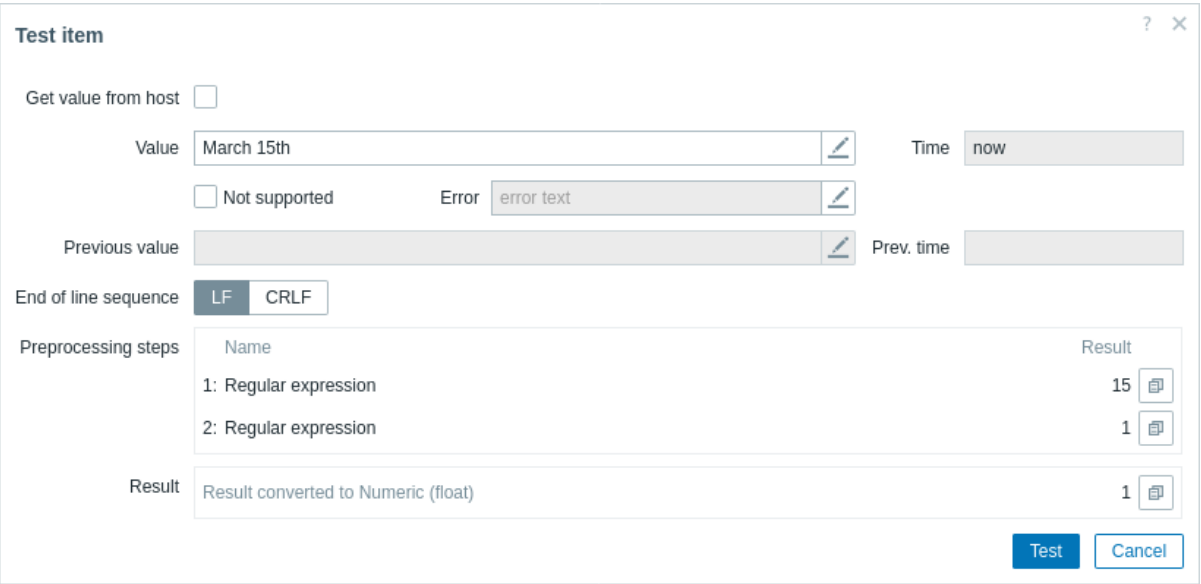


È possibile eseguire il test:

- rispetto a un valore ipotetico
- rispetto a un valore reale proveniente da un host

Ogni passaggio di preprocessing può essere testato singolarmente, così come tutti i passaggi possono essere testati insieme. Quando si fa clic sul pulsante *Test* o *Test all steps* rispettivamente nel blocco Actions, viene aperta una finestra di test.

Test del valore ipotetico



Parameter	Description
Get value from host	Se si desidera testare un valore ipotetico, lasciare questa casella di controllo non selezionata. Vedi anche: Testing real value.
Value	Immettere il valore di input da testare. Facendo clic nel campo del parametro o sul pulsante di visualizzazione/modifica si aprirà una finestra di area di testo per l'immissione del valore o del blocco di codice.
Not supported	Selezionare questa casella di controllo per testare un valore non supportato. Questa opzione è utile per testare il passaggio di preprocessing *Check for not supported value*.

Parameter	Description
Error	Immettere il testo dell'errore. Questo campo è abilitato quando *Get value from host* non è selezionato, ma *Not supported* è selezionato. Se *Get value from host* è selezionato, questo campo viene compilato con il messaggio di errore effettivo (sola lettura) proveniente dall'host.
Time	Viene visualizzato l'orario del valore di input: `now` (sola lettura).
Previous value	Immettere un valore di input precedente da confrontare. Solo per i passaggi di preprocessing *Change* e *Throttling*.
Previous time	Immettere l'orario del valore di input precedente da confrontare. Solo per i passaggi di preprocessing *Change* e *Throttling*. Il valore predefinito si basa sul valore del campo 'Update interval' dell'item (se è '1m', questo campo viene compilato con `now-1m`). Se non viene specificato nulla o l'utente non ha accesso all'host, il valore predefinito è `now-30s`.
Macros	Se vengono utilizzate macro, queste vengono elencate insieme ai rispettivi valori. I valori sono modificabili a scopo di test, ma le modifiche verranno salvate solo all'interno del contesto di test.
End of line sequence	Selezionare la sequenza di fine riga per i valori di input multilinea: **LF** - sequenza LF (line feed) **CRLF** - sequenza CRLF (carriage-return line-feed).
Preprocessing steps	I passaggi di preprocessing sono elencati; il risultato del test viene visualizzato per ciascun passaggio dopo aver fatto clic sul pulsante *Test*. I risultati del test vengono troncati a una dimensione massima di 512KB quando vengono inviati al frontend. I risultati del test possono essere copiati (non oltre i 512KB troncati). Se un risultato viene troncato, viene visualizzata un'icona di avviso. La descrizione dell'avviso viene visualizzata al passaggio del mouse. Si noti che i dati di dimensioni superiori a 512KB vengono comunque elaborati integralmente da Zabbix server. Se il passaggio non è riuscito nel test, viene visualizzata un'icona di errore. La descrizione dell'errore viene visualizzata al passaggio del mouse. Nel caso in cui per il passaggio sia specificato "Custom on fail" e tale azione venga eseguita, appare una nuova riga subito dopo la riga del passaggio di test di preprocessing, che mostra quale azione è stata eseguita e quale risultato ha prodotto (errore o valore).
Result	Il risultato finale del test dei passaggi di preprocessing viene visualizzato in tutti i casi in cui tutti i passaggi vengono testati insieme (quando si fa clic sul pulsante *Test all steps*). Viene inoltre visualizzato il tipo di conversione al tipo di valore dell'item, ad esempio `Result converted to Numeric (unsigned)`. I risultati del test vengono troncati a una dimensione massima di 512KB quando vengono inviati al frontend. Il risultato finale può essere copiato (non oltre i 512KB troncati). Se un risultato viene troncato, viene visualizzata un'icona di avviso. La descrizione dell'avviso viene visualizzata al passaggio del mouse. Si noti che i dati di dimensioni superiori a 512KB vengono comunque elaborati integralmente da Zabbix server.

Fare clic su *Test* per vedere il risultato dopo ciascun passaggio di preprocessing.

I valori di test vengono memorizzati tra le sessioni di test, sia per i singoli passaggi sia per tutti i passaggi, consentendo all'utente di modificare i passaggi di preprocessing o la configurazione dell'item e poi tornare alla finestra di test senza dover reinserire le informazioni. Tuttavia, i valori vengono persi al refresh della pagina.

Il test viene eseguito da Zabbix server. Il frontend invia una richiesta corrispondente al server e attende il risultato. La richiesta contiene il valore di input e i passaggi di preprocessing (con le macro utente espanse). Per i passaggi *Change* e *Throttling*, è possibile specificare facoltativamente un valore e un orario precedenti. Il server risponde con i risultati di ciascun passaggio di preprocessing.

Tutti gli errori tecnici o gli errori di validazione dell'input vengono visualizzati nella casella degli errori nella parte superiore della finestra di test.

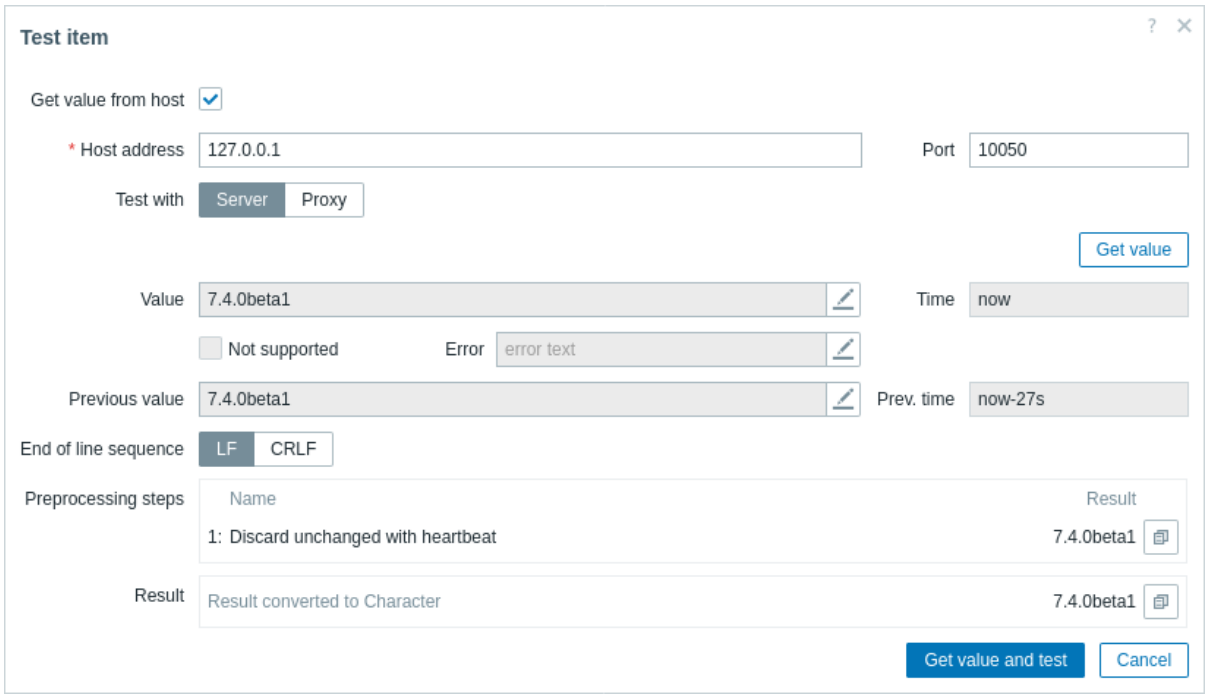
Test del valore reale

Per testare il preprocessing con un valore reale:

- Selezionare la casella *Get value from host*
- Inserire o verificare i parametri dell'host (indirizzo dell'host, porta, nome del proxy o nessun proxy) e i dettagli specifici dell'item (come la community SNMPv2 o le credenziali di sicurezza SNMPv3). Questi campi tengono conto del contesto:
  - I valori vengono precompilati quando possibile, ad esempio per gli item che richiedono un agent, utilizzando le informazioni dell'interfaccia agent selezionata dell'host
  - I valori devono essere compilati manualmente per gli item del template

- I valori delle macro in testo normale vengono risolti
- Se il valore del campo (o parte del valore) è una macro secret o Vault, il campo sarà vuoto e dovrà essere compilato manualmente. Se un parametro dell'item contiene il valore di una macro secret, viene visualizzato il seguente messaggio di avviso: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- I campi sono disabilitati quando non sono necessari nel contesto del tipo di item (ad esempio, i campi dell'indirizzo dell'host e del proxy sono disabilitati per gli item calcolati)

- Fare clic su *Get value and test* per testare il preprocessing



Se è stata specificata una mappatura dei valori nel modulo di configurazione dell'item (il campo *Show value*), la finestra di dialogo del test dell'item mostrerà un'altra riga dopo il risultato finale, denominata *Result with value map applied*.

Parametri specifici per ottenere un valore reale da un host:

Parametro	Descrizione
Get value from host	Selezionare questa casella per ottenere un valore reale dall'host.
Host address	Inserire l'indirizzo dell'host. Questo campo viene compilato automaticamente con l'indirizzo dell'interfaccia dell'host dell'item.
Port	Inserire la porta dell'host. Questo campo viene compilato automaticamente con la porta dell'interfaccia dell'host dell'item.
Additional fields for SNMP interfaces (SNMP version, SNMP community, Context name, etc.)	Per ulteriori dettagli sulla configurazione di un'interfaccia SNMP (v1, v2 e v3), vedere Configurazione del monitoraggio SNMP. Questi campi vengono compilati automaticamente dall'interfaccia dell'host dell'item.
Proxy	Specificare il proxy se l'host è monitorato da un proxy. Questo campo viene compilato automaticamente con il proxy dell'host, se presente.
Value	Valore recuperato dall'host. Facendo clic nel campo del parametro o sul pulsante di visualizzazione/modifica, si aprirà una finestra con un'area di testo contenente il valore o il blocco di codice. I valori vengono troncati a una dimensione massima di 512 KB e solo nel frontend. Se un risultato viene troncato, viene visualizzata un'icona di avviso. La descrizione dell'avviso viene visualizzata al passaggio del mouse. I dati di dimensioni superiori a 512 KB vengono comunque elaborati completamente dal server Zabbix.

Per gli altri parametri, vedere Test del valore ipotetico sopra.

Dettagli di pre-elaborazione

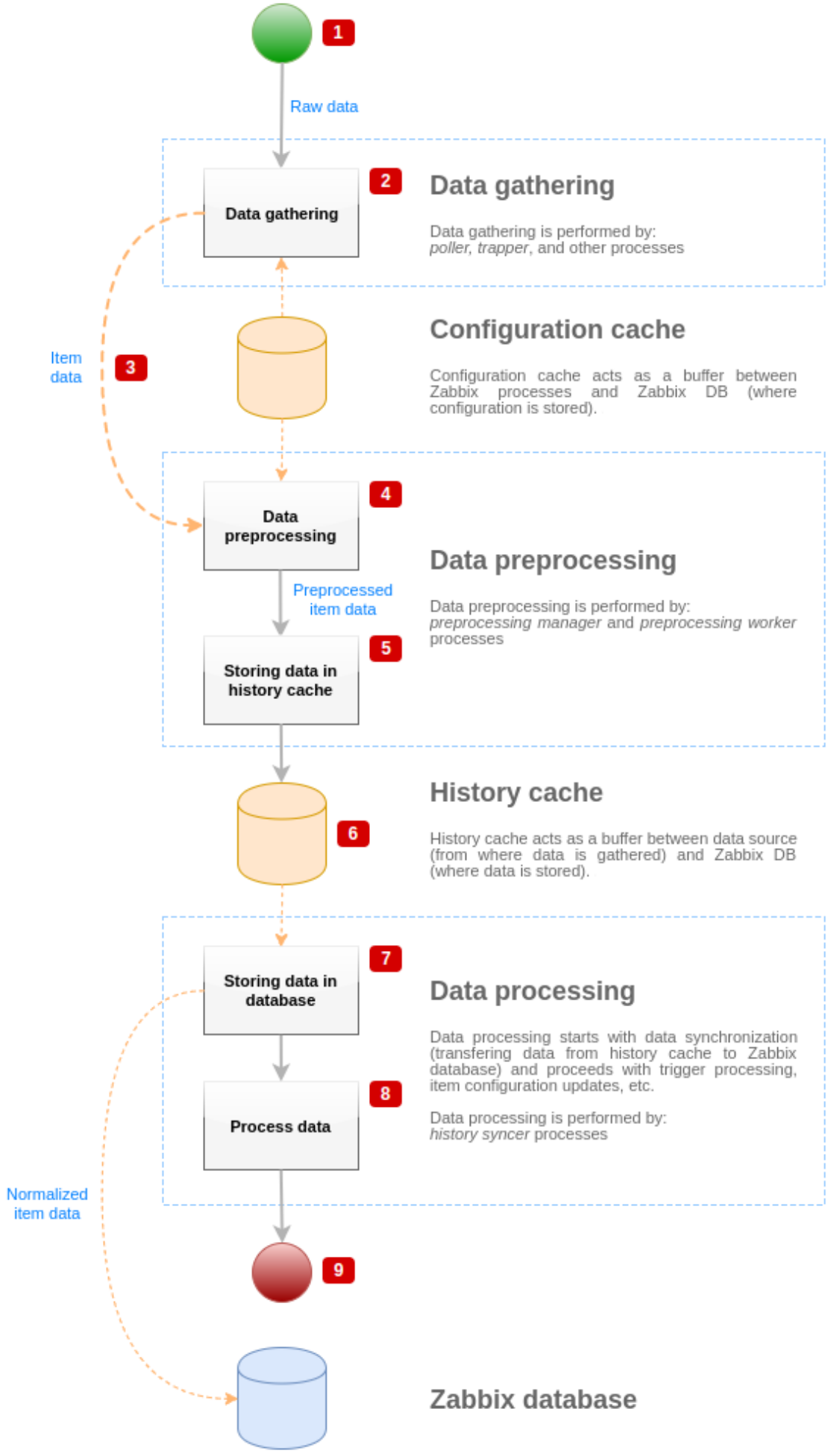
Panoramica

Questa sezione fornisce dettagli sulla pre-elaborazione dei valori degli item. La pre-elaborazione dei valori degli item consente di definire ed eseguire regole di trasformazione per i valori ricevuti dagli item.

La pre-elaborazione è gestita dal processo di gestione della pre-elaborazione insieme ai worker di pre-elaborazione che eseguono le relative operazioni. Tutti i valori sottoposti a pre-elaborazione, ricevuti da diversi raccoglitori di dati, passano attraverso il gestore della pre-elaborazione prima di essere aggiunti alla cache della cronologia. Per la comunicazione IPC tra i raccoglitori di dati (poller, trapper, ecc.) e il processo di pre-elaborazione vengono utilizzati socket. Le operazioni di pre-elaborazione vengono eseguite dal server Zabbix o dal proxy Zabbix (per gli item monitorati dal proxy).

Elaborazione del valore dell'item

Per visualizzare il flusso dei dati dalla sorgente dati al database di Zabbix, possiamo utilizzare il seguente diagramma semplificato:


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

Il diagramma sopra mostra solo i processi, gli oggetti e le azioni relativi all'elaborazione del valore dell'item in forma **semplificata**. Il diagramma non mostra i cambiamenti di direzione condizionali, la gestione degli errori o i cicli. Anche la cache dati locale del gestore di preprocessing non è mostrata, poiché non influisce direttamente sul flusso dei dati. Lo scopo di questo diagramma è mostrare i processi coinvolti nell'elaborazione del valore dell'item e il modo in cui interagiscono.

- La raccolta dei dati inizia con dati grezzi provenienti da una sorgente dati. A questo punto, i dati contengono solo ID, timestamp e valore (possono anche essere presenti più valori).
- Indipendentemente dal tipo di raccoglitore dati utilizzato, il concetto è lo stesso per controlli attivi o passivi, per item trapper, ecc., poiché cambia solo il formato dei dati e chi avvia la comunicazione (il raccoglitore dati è in attesa di una connessione e dei dati, oppure il raccoglitore dati avvia la comunicazione e richiede i dati). I dati grezzi vengono convalidati, la configurazione dell'item viene recuperata dalla cache di configurazione (i dati vengono arricchiti con i dati di configurazione).
- Per trasferire i dati dai raccoglitori dati al gestore di preprocessing viene utilizzato un meccanismo IPC basato su socket. A questo punto il raccoglitore dati continua a raccogliere dati senza attendere la risposta del gestore di preprocessing.
- Viene eseguito il preprocessing dei dati. Questo include l'esecuzione dei passaggi di preprocessing e l'elaborazione degli item dipendenti.

Note:
Un item può cambiare il proprio stato in NOT SUPPORTED durante l'esecuzione del preprocessing se uno qualsiasi dei passaggi di preprocessing fallisce.

- I dati di history dalla cache dati locale del gestore di preprocessing vengono scaricati nella cache di history.
- A questo punto il flusso dei dati si interrompe fino alla successiva sincronizzazione della cache di history (quando il processo history syncer esegue la sincronizzazione dei dati).
- Il processo di sincronizzazione inizia con la normalizzazione dei dati prima di memorizzarli nel database di Zabbix. La normalizzazione dei dati esegue conversioni nel tipo di item desiderato (tipo definito nella configurazione dell'item), inclusa la troncatura dei dati testuali in base alle dimensioni predefinite consentite per tali tipi (HISTORY_STR_VALUE_LEN per string, HISTORY_TEXT_VALUE_LEN per text e HISTORY_LOG_VALUE_LEN per valori log). I dati vengono inviati al database di Zabbix dopo il completamento della normalizzazione.

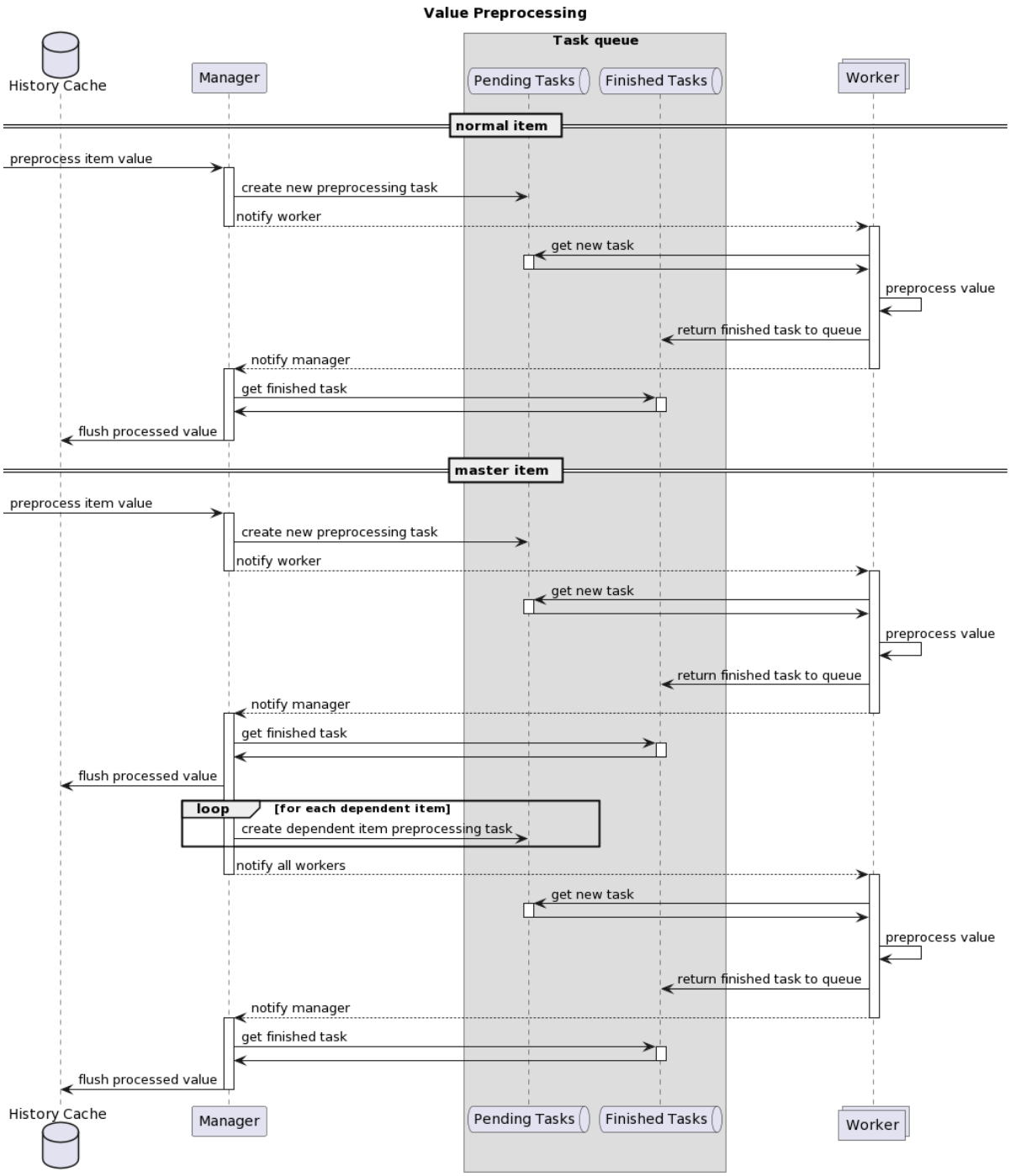
Note:
Un item può cambiare il proprio stato in NOT SUPPORTED se la normalizzazione dei dati fallisce (ad esempio, quando un valore testuale non può essere convertito in numero).

- I dati raccolti vengono elaborati: i trigger vengono controllati, la configurazione dell'item viene aggiornata se l'item diventa NOT SUPPORTED, ecc.
- Questo è considerato il termine del flusso dei dati dal punto di vista dell'elaborazione del valore dell'item.

Preelaborazione del valore dell'item

La preelaborazione dei dati viene eseguita nei seguenti passaggi:

- Se l'item non ha né preelaborazione né item dipendenti, il suo valore viene aggiunto alla cache della cronologia oppure inviato al gestore LLD. Altrimenti, il valore dell'item viene passato al gestore della preelaborazione utilizzando un meccanismo IPC basato su socket UNIX.
- Viene creato un task di preelaborazione, aggiunto alla coda e i worker di preelaborazione vengono notificati del nuovo task.
- A questo punto il flusso dei dati si interrompe finché non è disponibile almeno un worker di preelaborazione non occupato (cioè che non sta eseguendo alcun task).
- Quando un worker di preelaborazione è disponibile, preleva il task successivo dalla coda.
- Al termine della preelaborazione (sia in caso di esecuzione non riuscita che riuscita dei passaggi di preelaborazione), il valore preelaborato viene aggiunto alla coda dei task completati e il gestore viene notificato della presenza di un nuovo task completato.
- Il gestore della preelaborazione converte il risultato nel formato desiderato (definito dal tipo di valore dell'item) e lo aggiunge alla cache della cronologia oppure lo invia al gestore LLD.
- Se esistono item dipendenti per l'item elaborato, gli item dipendenti vengono aggiunti alla coda di preelaborazione con il valore preelaborato del master item. Gli item dipendenti vengono accodati bypassando le normali richieste di preelaborazione dei valori, ma solo per i master item con il valore impostato e non nello stato NOT SUPPORTED.



Si noti che nel diagramma la preelaborazione del master item è leggermente semplificata, omettendo la cache di preelaborazione.

Coda di preprocessing

La coda di preprocessing è organizzata come segue:

- l'elenco delle attività in sospeso:
  - attività create direttamente dalle richieste di preprocessing dei valori nell'ordine in cui sono state ricevute
- l'elenco delle attività immediate (elaborate prima delle attività in sospeso):
  - attività di test (create in risposta alle richieste di test di item/preprocessing dal frontend)
  - attività degli item dipendenti
  - attività di sequenza (attività che devono essere eseguite in un ordine rigoroso):
    - con passaggi di preprocessing che utilizzano l'ultimo valore:
      - modifica
      - throttling
      - JavaScript (cache del bytecode)
    - cache del preprocessing degli item dipendenti
- l'elenco delle attività completate

Cache di preprocessing

La cache di preprocessing è stata introdotta per migliorare le prestazioni del preprocessing per più item dipendenti che hanno passaggi di preprocessing simili (un risultato comune di LLD).

Il caching viene eseguito effettuando il preprocessing di un item dipendente e riutilizzando parte dei dati interni di preprocessing per il resto degli item dipendenti. La cache di preprocessing è supportata solo per il primo passaggio di preprocessing dei seguenti tipi:

- Pattern Prometheus (indicizza l'input in base alle metriche)
- JSONPath (analizza i dati in un albero di oggetti e indicizza la prima espressione `[?(@.path == "value")]`)

Worker di preprocessing

Il file di configurazione di Zabbix server consente agli utenti di impostare il numero di thread worker di preprocessing. Il parametro di configurazione StartPreprocessors deve essere usato per impostare il numero di istanze pre-avviate dei worker di preprocessing, che dovrebbe essere almeno pari al numero di core CPU disponibili.

Se le attività di preprocessing non sono vincolate alla CPU e comportano frequenti richieste di rete, è consigliabile configurare worker aggiuntivi. Il numero ottimale di worker di preprocessing può essere determinato da molti fattori, tra cui il numero di item "preprocessable" (item che richiedono l'esecuzione di qualsiasi fase di preprocessing), il numero di processi di raccolta dati, il numero medio di fasi di preprocessing per item, ecc. Un numero insufficiente di worker può portare a un elevato utilizzo della memoria. Per la risoluzione dei problemi relativi a un uso eccessivo della memoria nella tua installazione di Zabbix, consulta Profiling excessive memory usage with tcmalloc.

Tuttavia, supponendo che non vi siano operazioni di preprocessing pesanti come l'analisi di grandi blocchi XML/JSON, il numero di worker di preprocessing può corrispondere al numero totale di data gatherers. In questo modo, nella maggior parte dei casi (tranne quando i dati provenienti dal gatherer arrivano in blocco) ci sarà almeno un worker di preprocessing libero per i dati raccolti.

Warning:
Troppi processi di raccolta dati (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) insieme a IPMI manager, SNMP trapper e worker di preprocessing possono esaurire il limite di file descriptor per processo per il preprocessing manager.

 L'esaurimento del limite di file descriptor per processo causerà l'arresto di Zabbix server, in genere poco dopo l'avvio ma talvolta dopo più tempo. Per evitare tali problemi, esamina il file di configurazione di Zabbix server per ottimizzare il numero di controlli e processi concorrenti. Inoltre, se necessario, assicurati che il limite di file descriptor sia impostato sufficientemente in alto controllando e regolando i limiti di sistema.

Pipeline di elaborazione dei valori

L'elaborazione dei valori degli item viene eseguita in più passaggi (o fasi) da più processi. Questo può causare:

- Un item dipendente può ricevere valori, mentre il valore master NON può. Questo può essere ottenuto utilizzando il seguente caso d'uso:
  - L'item master ha il tipo di valore `UINT` (può essere utilizzato un item trapper), l'item dipendente ha il tipo di valore `TEXT`.
  - Non sono richiesti passaggi di preprocessing né per l'item master né per l'item dipendente.
  - Un valore testuale (ad esempio, "abc") deve essere passato all'item master.
  - Poiché non ci sono passaggi di preprocessing da eseguire, il gestore del preprocessing verifica che l'item master non sia nello stato NOT SUPPORTED e che il valore sia impostato (entrambe le condizioni sono vere) e mette in coda l'item dipendente con lo stesso valore dell'item master (poiché non ci sono passaggi di preprocessing).
  - Quando sia l'item master sia l'item dipendente raggiungono la fase di sincronizzazione della history, l'item master diventa NOT SUPPORTED a causa dell'errore di conversione del valore (i dati testuali non possono essere convertiti in un intero senza segno).

Di conseguenza, l'item dipendente riceve un valore, mentre l'item master cambia il proprio stato in NOT SUPPORTED.

- Un item dipendente riceve un valore che non è presente nella history dell'item master. Il caso d'uso è molto simile al precedente, tranne per il tipo dell'item master. Ad esempio, se per l'item master viene utilizzato il tipo `CHAR`, allora il valore dell'item master verrà troncato nella fase di sincronizzazione della history, mentre gli item dipendenti riceveranno i propri valori dal valore iniziale (non troncato) dell'item master.

Esempi di preprocessing

Panoramica

Questa sezione presenta esempi di utilizzo dei passaggi di preprocessing per svolgere alcune attività pratiche.

Filtraggio dei record del log eventi VMware

Questo esempio utilizza il passaggio di preprocessing Matches regular expression per filtrare gli eventi non necessari dal log eventi VMware.

1. Su un host VMware Hypervisor funzionante, verificare che l'item vmware.eventlog sia presente e funzioni correttamente. Si noti che l'item del log eventi potrebbe essere già presente sull'hypervisor se durante la creazione dell'host è stato collegato un template VMware.

2. Sull'host VMware Hypervisor, creare un item dipendente di tipo *Log* e impostare l'item del log eventi come item principale.

3. Nella scheda *Preprocessing* dell'item dipendente, fare clic su *Add* per creare un passaggio di preprocessing e selezionare *Matches regular expression* dal menu a discesa. Quindi, specificare uno dei seguenti pattern:

- Per filtrare tutti gli eventi del log:

```
.* logged in .*
```

- Per filtrare le righe che contengono nomi utente dopo "User":

```
\bUser\s+\K\S+
```

> **Attention:**
> Se l'espressione regolare non trova corrispondenza, l'item dipendente diventa non supportato con un messaggio di errore corrispondente. Per evitare questo comportamento, selezionare la casella *Custom on fail* e scegliere un'opzione, ad esempio scartare il valore o impostarne uno personalizzato. Si noti che i valori scartati non vengono memorizzati nel database; di conseguenza, i trigger non vengono valutati e i dati di trend non vengono generati.

In alternativa, è possibile utilizzare il passaggio di preprocessing Regular expression per estrarre i gruppi corrispondenti e controllare l'output:

- Per estrarre e visualizzare l'intero evento del log contenente "logged in", specificare i seguenti parametri:

```
Pattern: .*logged in.*
Output: \0
```

- Per estrarre e visualizzare i nomi utente che seguono "User":

```
Pattern: User (.*?)(?=\ )
Output: \1
```

Verifica del tipo di valore recuperato

Questo esempio utilizza il passaggio di preprocessing Moltiplicatore personalizzato per verificare se il tipo di valore dell'item recuperato è numerico.

Nella scheda *Preprocessing* di un item, seleziona il passaggio di preprocessing *Moltiplicatore personalizzato* e specifica il seguente parametro (moltiplica il valore recuperato per 1):

```
1
```

> **Attention:**
> Se il preprocessing fallisce (ad esempio, l'input non è numerico), l'item diventa non supportato con un messaggio di errore corrispondente. Per evitarlo, seleziona la casella *Personalizzato in caso di errore* e scegli un'opzione, ad esempio scartare il valore o impostarne uno personalizzato. Tieni presente che i valori scartati non vengono memorizzati nel database; di conseguenza, i trigger non vengono valutati e i dati di trend non vengono generati.

Verifica del valore non supportato

Questo esempio utilizza il passaggio di preprocessing Check for not supported value per verificare se il valore dell'item non può essere recuperato.

Quando un processo poller di Zabbix server/proxy tenta di raccogliere il valore di un item, può:

- Restituire un risultato valido.
- Restituire un risultato che inizialmente sembra valido ma che potrebbe diventare non supportato in seguito (ad esempio, a causa di una mancata corrispondenza del tipo di valore dopo il preprocessing).
- Restituire un errore durante la raccolta del valore, causando il passaggio dell'item a non supportato. Le cause comuni includono:
  - Chiave item sconosciuta (per Zabbix agent, Simple check o item interni di Zabbix)
  - OID sconosciuto (SNMP agent), sensore sconosciuto (IPMI agent) o nessuna metrica JMX (JMX agent)
  - Impossibile leggere il file trap (SNMP trap)
  - Script non trovato (External check)
  - URL inesistente (HTTP agent, Browser)
  - Accesso non riuscito (SSH agent, TELNET agent)

- Sintassi della formula non valida (Calculated), errore di sintassi JavaScript (Script) o SQL non valido (Database monitor)

Per rilevare e gestire gli errori nella raccolta dei valori degli item, è possibile utilizzare il passaggio di preprocessing *Check for not supported value*. Si noti che questo passaggio viene sempre eseguito per primo e rileva solo gli errori che si verificano prima dell'inizio del preprocessing.

Nella scheda *Preprocessing* di un item, selezionare il passaggio di preprocessing *Check for not supported value* e specificare uno dei seguenti parametri:

- Per qualsiasi errore:

```
Parameter: any error
```

- Per errori contenenti "cannot connect":

```
Parameter: error matches
Pattern: (?i)cannot connect
```

Quindi, utilizzare l'opzione *Custom on fail* per scartare il valore (in questo caso, l'errore), impostare un valore personalizzato o restituire un messaggio di errore personalizzato. Si noti che i valori discarded non vengono memorizzati nel database; di conseguenza, i trigger non vengono valutati e i dati di trend non vengono generati.

Funzionalità JSONPath

Panoramica

Questa sezione descrive le funzionalità JSONPath supportate nei passaggi di preprocessing dei valori degli item.

JSONPath è composto da segmenti separati da punti. Un segmento può assumere la forma di una semplice parola, che rappresenta il nome di un valore JSON, del carattere jolly (`*`) oppure di un costrutto più complesso racchiuso tra parentesi quadre. Il punto prima di un segmento tra parentesi quadre è facoltativo e può essere omesso.

Esempio JSONPath	Descrizione
`$.object.name`	Restituisce il contenuto di `object.name`.
`$.object['name']`	Restituisce il contenuto di `object.name`.
`$.object.['name']`	Restituisce il contenuto di `object.name`.
`$["object"]['name']`	Restituisce il contenuto di `object.name`.
`$.['object'].["name"]`	Restituisce il contenuto di `object.name`.
`$.object.history.length()`	Restituisce il numero di elementi dell'array `object.history`.
`$[?(@.name == 'Object')].price.first()`	Restituisce il valore della proprietà `price` del primo oggetto denominato "Object".
`$[?(@.name == 'Object')].history.first().length()`	Restituisce il numero di elementi dell'array history del primo oggetto denominato "Object".
`$[?(@.price > 10)].length()`	Restituisce il numero di oggetti con un prezzo maggiore di 10.

Vedi anche: Escape dei caratteri speciali dai valori delle macro LLD in JSONPath.

Segmenti supportati

Segmento	Descrizione
`<name>`	Corrisponde a una proprietà dell'oggetto in base al nome.
`*`	Corrisponde a tutte le proprietà dell'oggetto.
`['<name>']`	Corrisponde a una proprietà dell'oggetto in base al nome.
`['<name>', '<name>', ...]`	Corrisponde a una proprietà dell'oggetto in base a uno qualsiasi dei nomi elencati.
`[<index>]`	Corrisponde a un elemento dell'array in base all'indice.
`[<number>, <number>, ...]`	Corrisponde a un elemento dell'array in base a uno qualsiasi degli indici elencati.
`[*]`	Corrisponde a tutte le proprietà dell'oggetto o a tutti gli elementi dell'array.

Segmento	Descrizione
`[<start>:<end>]`	Corrisponde agli elementi dell'array in base all'intervallo definito: **<start>** - il primo indice da corrispondere (incluso); se non specificato, corrisponde a tutti gli elementi dell'array dall'inizio; se negativo, specifica l'offset iniziale dalla fine dell'array; **<end>** - l'ultimo indice da corrispondere (escluso); se non specificato, corrisponde a tutti gli elementi dell'array fino alla fine; se negativo, specifica l'offset iniziale dalla fine dell'array.
`[?(<expression>)]`	Corrisponde a oggetti/elementi dell'array applicando un'espressione di filtro.

Per trovare un segmento corrispondente ignorandone l'ascendenza (segmento scollegato), deve essere preceduto da due punti (`..`). Ad esempio, `$..name` o `$..['name']` restituiscono i valori di tutte le proprietà `name`.

I nomi degli elementi corrispondenti possono essere estratti aggiungendo un suffisso tilde (`~`) al JSONPath. Restituisce il nome dell'oggetto corrispondente o un indice in formato stringa dell'item dell'array corrispondente. Il formato di output segue le stesse regole delle altre query JSONPath: i risultati di un percorso definito vengono restituiti "così come sono", mentre i risultati di un percorso indefinito vengono restituiti in un array. Tuttavia, estrarre il nome di un elemento che corrisponde a un percorso definito ha un'utilità minima, poiché è già noto.

Espressione di filtro

L'espressione di filtro è un'espressione aritmetica in notazione infissa.

Operandi supportati:

Operand	Description

`"<text>"`
`'<text>'`
	Costante di testo.
	
	Esempio:
	`'value: \'1\''`
	`"value: '1'"`

`<number>`	Costante numerica che supporta la notazione scientifica. Esempio: 123
`<jsonpath starting with $>`	Valore a cui si fa riferimento tramite JSONPath dal nodo radice del documento di input; sono supportati solo percorsi definiti. Esempio: `$.object.name`
`<jsonpath starting with @>`	Valore a cui si fa riferimento tramite JSONPath dall'oggetto/elemento corrente; sono supportati solo percorsi definiti. Esempio: `@.name`

Operatori supportati: |Operator|Type|Description|Result| |-|---|---|---| |−|Binary|Sottrazione|Number| |+|Binary|Addizione|Number| |/|Binary|Divisione|Number| |*|Binary|Moltiplicazione|Number| |==|Binary|Uguaglianza|Boolean (1/0)| |!=|Binary|Disuguaglianza|Boolean (1/0)| |<|Binary|Minore di|Boolean (1/0)| |<=|Binary|Minore o uguale a|Boolean (1/0)| |>|Binary|Maggiore di|Boolean (1/0)| |>=|Binary|Maggiore o uguale a|Boolean (1/0)| |=~|Binary|Corrisponde all'espressione regolare|Boolean (1/0)| |!|Unary|NOT booleano|Boolean (1/0)| |||Binary|OR booleano|Boolean (1/0)| |&&|Binary|AND booleano|Boolean (1/0)|

Funzioni

Le funzioni possono essere utilizzate alla fine di JSONPath. È possibile concatenare più funzioni se la funzione precedente restituisce un valore accettato dalla funzione successiva.

Funzioni supportate:

Funzione	Descrizione	Input	Output
`avg`	Valore medio dei numeri in un array di input	Array di numeri	Numero
`min`	Valore minimo dei numeri in un array di input	Array di numeri	Numero
`max`	Valore massimo dei numeri in un array di input	Array di numeri	Numero
`sum`	Somma dei numeri in un array di input	Array di numeri	Numero
`length`	Numero di elementi in un array di input	Array	Numero

Funzione	Descrizione	Input	Output
`first`	Il primo elemento di un array	Array	Un costrutto JSON (oggetto, array, valore) in base al contenuto dell'array di input

Le funzioni aggregate di JSONPath accettano valori numerici tra virgolette. Questi valori vengono convertiti automaticamente da stringhe a tipi numerici quando è necessaria un'aggregazione. Un input incompatibile farà sì che la funzione generi un errore.

Valore di output

I JSONPath possono essere suddivisi in percorsi definiti e indefiniti. Un percorso definito può restituire solo null oppure una singola corrispondenza. Un percorso indefinito può restituire più corrispondenze: JSONPath con elenchi di nomi/indici separati, multipli, slice di array o segmenti di espressione. Tuttavia, quando viene utilizzata una funzione, il JSONPath diventa definito, poiché le funzioni producono sempre un singolo valore.

Un percorso definito restituisce l'oggetto/l'array/il valore a cui fa riferimento. Al contrario, un percorso indefinito restituisce un array degli oggetti/array/valori corrispondenti.

> **Attention:**
> L'ordine delle proprietà nei risultati delle query JSONPath potrebbe non corrispondere all'ordine originale delle proprietà nel JSON a causa dei metodi di ottimizzazione interni. Ad esempio, il JSONPath `$.books[1]["author", "title"]` può restituire `["title", "author"]`. Se è essenziale preservare l'ordine originale delle proprietà, è opportuno considerare metodi alternativi di elaborazione successiva alla query.

Regole di formattazione del percorso

Gli spazi bianchi (spazio, carattere di tabulazione) possono essere utilizzati nei segmenti con notazione tra parentesi quadre e nelle espressioni, ad esempio: `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`.

Le stringhe devono essere racchiuse tra apici singoli (') o doppi ("). All'interno delle stringhe, gli apici singoli o doppi (a seconda di quelli usati per racchiuderle) e le barre rovesciate (\) vengono preceduti dal carattere barra rovesciata (\).

Esempio

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Detti del secolo",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "La spada dell'onore",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "Il Signore degli Anelli",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
```

```
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Consegna il giorno successivo nella città locale",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Stampa e rilegatura del libro in formato A5",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Vari metodi di restauro",
      "active": false,
      "methods": [
        {
          "description": "Pulizia chimica",
          "price": 46
        },
        {
          "description": "Pressatura delle pagine danneggiate dall'umidità",
          "price": 24.5
        },
        {
          "description": "Rilegatura di un libro strappato",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "Il negozio è chiuso",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	no ”filters”

`$.filters`	definite	{
		”price”: 10,
		”category”: ”fiction”,
		”no filters”: ”no \”filters\””
		}

`$.books[1].title`	definite	Sword of Honour
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4

JSONPath	Type	Result
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sayings of the Century”]
`$.books[1]['author', "title"]`	indefinite	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] Nota: Questa query mostra che nelle query si possono usare operazioni aritmetiche; può essere semplificata in `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Result

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Stampa e assemblaggio del libro in formato A5”, ”Rilegatura di un libro strappato”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour
`$..tags.first().length()`	definite	5 Nota: `$..tags` è un percorso indefinito, quindi restituisce un array di elementi corrispondenti, cioè `[["a", "b", "c", "d", "e" ]]`; `first()` restituisce il primo elemento, cioè `["a", "b", "c", "d", "e"]`; `length()` calcola la lunghezza dell’elemento, cioè 5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Nota: Una query senza corrispondenze restituisce NULL per i percorsi definiti e indefiniti.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Nota: Nelle comparazioni di valori booleani devono essere usate costanti testuali.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Nota: Nelle comparazioni di valori booleani devono essere usate costanti testuali.
`$.services[?(@.servicegroup=="1002")].description`	indefinite	Restauro

Escape dei caratteri speciali dai valori delle macro LLD in JSONPath

Quando le macro di discovery a basso livello vengono usate nel preprocessing JSONPath e i loro valori vengono risolti, si applicano le seguenti regole di escape dei caratteri speciali:

- solo il backslash (\) e il carattere di doppia virgoletta (") vengono considerati per l'escape;
- se il valore della macro risolto contiene questi caratteri, ciascuno di essi viene preceduto da un backslash;
- se sono già preceduti da un backslash, questo non viene considerato un escape e sia il backslash sia i caratteri speciali successivi vengono nuovamente escapati una sola volta.

Ad esempio:

JSONPath	LLD macro value	After substitution
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]
	c:\temp	$.[?(@.value == "c:\\temp")]
	a\\b	$.[?(@.value == "a\\\\b")]

Quando viene usata nell'espressione, la macro che può contenere caratteri speciali deve essere racchiusa tra doppie virgolette:

JSONPath	LLD macro value	After substitution	Result
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	**Bad JSONPath expression**

Quando viene usata nel percorso, la macro che può contenere caratteri speciali deve essere racchiusa tra parentesi quadre **e** doppie virgolette:

JSONPath	LLD macro value	After substitution	Result
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Bad JSONPath expression**

Pre-elaborazione JavaScript

Panoramica

Questa sezione fornisce dettagli sulla pre-elaborazione tramite JavaScript.

> **Attention:**
> Non utilizzare assegnazioni non dichiarate nel JavaScript di pre-elaborazione. Usa `var` per dichiarare le variabili locali.

Preprocessing JavaScript

La preprocessing JavaScript viene eseguita richiamando una funzione JavaScript con un singolo parametro 'value' e un corpo di funzione fornito dall'utente. Il risultato del passaggio di preprocessing è il valore restituito da questa funzione; ad esempio, per eseguire la conversione da Fahrenheit a Celsius, inserisci:

```
return (value - 32)  * 5 / 9
```

nei parametri di preprocessing JavaScript, che verranno racchiusi in una funzione JavaScript da server:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Il parametro di input 'value' viene sempre passato come stringa. Il valore restituito viene automaticamente convertito in stringa tramite il metodo toString() (se l'operazione fallisce, l'errore viene restituito come valore stringa), con alcune eccezioni:

- la restituzione del valore undefined genererà un errore;

- la restituzione del valore null farà sì che il valore di input venga scartato, in modo simile alla preprocessing 'Discard value' nell'azione 'Custom on fail'.

Gli errori possono essere restituiti lanciando valori/oggetti (normalmente stringhe oppure oggetti Error).

Ad esempio:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Ogni script ha un timeout di esecuzione di 10 secondi (a seconda dello script, potrebbe volerci più tempo prima che il timeout venga attivato); il superamento di questo limite restituirà un errore. Viene imposto un limite di heap di 512 megabyte.

Il bytecode del passaggio di preprocessing JavaScript viene memorizzato nella cache e riutilizzato quando il passaggio viene applicato la volta successiva. Qualsiasi modifica ai passaggi di preprocessing dell'item causerà il reset dello script memorizzato nella cache e la sua ricompilazione in un secondo momento.

Errori di runtime consecutivi (3 di fila) causeranno la reinizializzazione del motore per mitigare la possibilità che uno script comprometta l'ambiente di esecuzione per gli script successivi (questa azione viene registrata con DebugLevel 4 e superiore).

La preprocessing JavaScript è implementata con il motore JavaScript Duktape.

Note:
Le definizioni dei metodi degli oggetti JavaScript integrati non possono essere ridefinite negli script di preprocessing dell'item.

Vedi anche: Additional JavaScript objects and global functions

Utilizzo delle macro negli script

È possibile utilizzare le macro utente (così come le macro LLD nel contesto del low-level discovery) nel codice JavaScript. Se uno script contiene macro utente, queste macro vengono risolte da server/proxy prima dell'esecuzione di specifici passaggi di preprocessing. Si noti che durante il test dei passaggi di preprocessing nel frontend, i valori delle macro non verranno recuperati e dovranno essere inseriti manualmente.

Note:
Il contesto viene ignorato quando una macro viene sostituita con il suo valore. Il valore della macro viene inserito nel codice così com'è; non è possibile aggiungere un escaping aggiuntivo prima di inserire il valore nel codice JavaScript. Si tenga presente che in alcuni casi ciò può causare errori JavaScript.

Nell'esempio seguente, se il valore ricevuto supera il valore della macro {$THRESHOLD}, verrà restituito invece il valore di soglia (se presente):

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

Esempi

I seguenti esempi illustrano come è possibile utilizzare il preprocessing JavaScript.

Ogni esempio contiene una breve descrizione, il corpo di una funzione per i parametri di preprocessing JavaScript e il risultato del passaggio di preprocessing, ovvero il valore restituito dalla funzione.

Esempio 1: Convertire un numero (notazione scientifica in intero)

Convertire il numero "2.62128e+07" dalla notazione scientifica a un intero.

```
return (Number(value))
```

Valore restituito dalla funzione: 26212800.

Esempio 2: Convertire un numero (da binario a decimale)

Convertire il numero binario "11010010" in un numero decimale.

```
return(parseInt(value,2))
```

Valore restituito dalla funzione: 210.

Esempio 3: Arrotondare un numero

Arrotonda il numero "18.2345" a 2 cifre.

```
return(Math.round(value* 100) / 100)
```

Valore restituito dalla funzione: 18.23.

Esempio 4: contare le lettere in una stringa

Conta il numero di lettere nella stringa ”Zabbix”.

```
return (value.length)
```

Valore restituito dalla funzione: 6.

Esempio 5: Ottenere il tempo rimanente

Ottieni il tempo rimanente (in secondi) fino alla data di scadenza di un certificato (Feb 12 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);
```

Valore restituito dalla funzione: 44380233.

Esempio 6: Rimuovere proprietà JSON

Modificare la struttura dei dati JSON rimuovendo tutte le proprietà con la chiave "data_size" o "index_size".

```
var obj=JSON.parse(value);

for (i = 0; i < Object.keys(obj).length; i++) {
    delete obj[i]["data_size"];
    delete obj[i]["index_size"];
}

return JSON.stringify(obj)
```

Valore accettato dalla funzione:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

Valore restituito dalla funzione:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

Esempio 7: Convertire lo stato di Apache in JSON

Convertire il valore ricevuto da un item Zabbix agent web.page.get (ad esempio, web.page.get[http://127.0.0.1:80/server-status?auto]) in un oggetto JSON.

```
// Convert Apache status to JSON

// Split the value into substrings and put these substrings into an array
var lines = value.split('\n');

// Create an empty object "output"
var output = {};

// Create an object "workers" with predefined properties
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Add the substrings from the "lines" array to the "output" object as properties (key-value pairs)
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]);
    }
}

// Multiversion metrics
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Parse "Scoreboard" property to get the worker count
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Add worker data to the "output" object
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['.']
};

// Return JSON string
return JSON.stringify(output);
```

Valore accettato dalla funzione:

```
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__.....................................................................................
```

Valore restituito dalla funzione:


```
{
    "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
    }
}
```

Oggetti JavaScript aggiuntivi

Panoramica

Questa sezione descrive le estensioni di Zabbix al linguaggio JavaScript implementato con Duktape e le funzioni JavaScript globali supportate.

Attention:
Non utilizzare assegnazioni a variabili non dichiarate nel JavaScript di preprocessing. Usa `var` per dichiarare le variabili locali.

Oggetti incorporati

Zabbix

L'oggetto Zabbix consente l'interazione con le funzionalità interne di Zabbix.

Method	Description
`log(loglevel, message)`	Scrive <message> nel log di Zabbix utilizzando il livello di log <loglevel> (vedere il parametro DebugLevel nel file di configurazione).

Esempio:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

È possibile utilizzare i seguenti alias:

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:
La dimensione totale di tutti i messaggi registrati è limitata a 8 MB per ogni esecuzione dello script.

Method	Description
`sleep(delay)`	Ritarda l'esecuzione di JavaScript di `delay` millisecondi.

Esempio (ritarda l'esecuzione di 15 secondi):

```
Zabbix.sleep(15000)
```

HttpRequest

Questo oggetto incapsula un handle cURL e consente di effettuare semplici richieste HTTP. Gli errori vengono generati come eccezioni.

Attention:
L'inizializzazione di più oggetti `HttpRequest` è limitata a 10 per esecuzione dello script.

Metodo	Descrizione
`addHeader(value)`	Aggiunge un campo di intestazione HTTP. Questo campo viene utilizzato per tutte le richieste successive finché non viene cancellato con il metodo `clearHeader()`. La lunghezza totale dei campi di intestazione che possono essere aggiunti a un singolo oggetto `HttpRequest` è limitata a 128 Kbyte (inclusi i caratteri speciali e i nomi delle intestazioni).
`clearHeader()`	Cancella l'intestazione HTTP. Se non sono impostati campi di intestazione, `HttpRequest` imposta Content-Type su application/json se i dati inviati sono in formato JSON; altrimenti su text/plain.
`connect(url)`	Invia una richiesta HTTP CONNECT all'URL e restituisce la risposta.
`customRequest(method, url, data)`	Consente di specificare qualsiasi metodo HTTP nel primo parametro. Invia la richiesta del metodo all'URL con un payload *data* facoltativo e restituisce la risposta.
`delete(url, data)`	Invia una richiesta HTTP DELETE all'URL con un payload *data* facoltativo e restituisce la risposta.
`getHeaders(<asArray>)`	Restituisce l'oggetto contenente i campi delle intestazioni HTTP ricevute. Il parametro `asArray` può essere impostato su "true" (ad esempio, `getHeaders(true)`), su "false" oppure può essere non definito. Se impostato su "true", i valori dei campi delle intestazioni HTTP ricevute vengono restituiti come array; questa modalità deve essere utilizzata per recuperare i valori dei campi di più intestazioni con lo stesso nome. Se non è impostato o è impostato su "false", i valori dei campi delle intestazioni HTTP ricevute vengono restituiti come stringhe.
`get(url, data)`	Invia una richiesta HTTP GET all'URL con un payload *data* facoltativo e restituisce la risposta.
`head(url)`	Invia una richiesta HTTP HEAD all'URL e restituisce la risposta.
`options(url)`	Invia una richiesta HTTP OPTIONS all'URL e restituisce la risposta.
`patch(url, data)`	Invia una richiesta HTTP PATCH all'URL con un payload *data* facoltativo e restituisce la risposta.
`put(url, data)`	Invia una richiesta HTTP PUT all'URL con un payload *data* facoltativo e restituisce la risposta.
`post(url, data)`	Invia una richiesta HTTP POST all'URL con un payload *data* facoltativo e restituisce la risposta.
`getStatus()`	Restituisce il codice di stato dell'ultima richiesta HTTP.
`setProxy(proxy)`	Imposta il proxy HTTP sul valore "proxy". Se questo parametro è vuoto, non viene utilizzato alcun proxy.

`setHttpAuth(bitmask, username, password)`	Imposta i metodi di autenticazione HTTP abilitati (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) nel parametro 'bitmask'.
	Il flag HTTPAUTH_NONE consente di disabilitare l'autenticazione HTTP.
	Esempi:
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	Invia una richiesta HTTP TRACE all'URL con un payload *data* facoltativo e restituisce la risposta.

Esempio:

```
try {
    Zabbix.log(4, 'jira webhook script value='+value);

    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new HttpRequest(),
    fields = {},
    resp;

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic '+params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
```

```
    fields.issuetype = {"id": params.issue_id};
    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: '+req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
} catch (error) {
    Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
    Zabbix.log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);
```

XML

L'oggetto XML consente l'elaborazione dei dati XML nel preprocessing di item e low-level discovery e nei webhook.

Attention:
Per utilizzare l'oggetto XML, server/proxy deve essere compilato con il supporto libxml2.

Method	Description
`XML.query(data, expression)`	Recupera il contenuto del nodo utilizzando XPath. Restituisce null se il nodo non viene trovato. **expression** - un'espressione XPath; **data** - dati XML come stringa.
`XML.toJson(data)`	Converte i dati in formato XML in JSON.
`XML.fromJson(object)`	Converte i dati in formato JSON in XML.

Esempio:

Input:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

Output:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

Regole di serializzazione

La conversione da XML a JSON verrà elaborata secondo le seguenti regole (per le conversioni da JSON a XML si applicano le regole inverse):

1. Gli attributi XML verranno convertiti in chiavi con i loro nomi preceduti da '@'.

Esempio:

Input:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Output:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Gli elementi autochiudenti (<foo/>) verranno convertiti come aventi valore 'null'.

Esempio:

Input:

```
<xml>
  <foo/>
</xml>
```

Output:

```
{
  "xml": {
    "foo": null
  }
}
```

3. Gli attributi vuoti (con valore "") verranno convertiti come aventi valore stringa vuota ("").

Esempio:

Input:

```
<xml>
  <foo bar="" />
</xml>
```

Output:

```
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. Più nodi figli con lo stesso nome di elemento verranno convertiti in un'unica chiave che avrà come valore un array di valori.

Esempio:

Input:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
```

```
  <foo>QUX</foo>
</xml>
```

Output:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Se un elemento di testo non ha attributi né figli, verrà convertito come stringa.

Esempio:

Input:

```
<xml>
    <foo>BAZ</foo>
</xml>
```

Output:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Se un elemento di testo non ha figli ma ha attributi, il contenuto testuale verrà convertito in un elemento con la chiave '#text' e il contenuto come valore; gli attributi verranno convertiti come descritto nella regola di serializzazione 1.

Esempio:

Input:

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Output:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Funzioni JavaScript globali

Con Duktape sono state implementate ulteriori funzioni JavaScript globali:

- `btoa(data)` - codifica i dati in una stringa base64.
- `atob(base64_string)` - decodifica una stringa base64 come buffer Uint8Array.

```
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)` - calcola l'hash MD5 dei dati.
- `sha256(data)` - calcola l'hash SHA256 dei dati.
- `hmac('<hash type>',key,data)` - restituisce un hash HMAC come stringa in formato esadecimale; `md5` e `sha256` sono supportati come `hash type`; i parametri `key` e `data` supportano dati binari.

  Esempi:

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`
- `sign(hash,key,data)` - restituisce la firma calcolata (firma RSA con SHA-256) come stringa, dove:
 hash - è consentito solo `sha256`, altrimenti viene generato un errore.
 key - la chiave privata. Deve essere conforme allo standard PKCS#1 o PKCS#8. La chiave può essere fornita in diverse forme:

  - con spazi al posto delle nuove righe
  - con `\n` con escape o senza escape al posto delle nuove righe
  - senza alcuna nuova riga, come stringa su una sola riga
  - come stringa in formato JSON

  La chiave può anche essere caricata da una macro utente/macro segreta/vault.

  data - i dati che verranno firmati. Possono essere una stringa (sono supportati anche dati binari) o un buffer (Uint8Array/ArrayBuffer).

  Esempio:

  - `sign('sha256',key,data)`

  OpenSSL o GnuTLS vengono utilizzati per calcolare le firme. Se Zabbix è stato compilato senza una di queste librerie di crittografia, verrà generato un errore ('missing OpenSSL or GnuTLS library').

Oggetti JavaScript dell'item del browser

Panoramica

Questa sezione descrive le estensioni di Zabbix al linguaggio JavaScript implementate con Duktape per l'uso nello script dell'item Browser. Queste estensioni integrano gli oggetti JavaScript descritti nella pagina Oggetti JavaScript aggiuntivi.

Browser

L'oggetto `Browser` gestisce le sessioni WebDriver, inizializzando una sessione al momento della creazione e terminandola al momento della distruzione. Un singolo script può supportare fino a quattro oggetti `Browser`.

Per costruire un oggetto `Browser`, utilizzare la sintassi `new Browser(options)`. Il parametro `options` (*oggetto JSON*) specifica le opzioni del browser, solitamente il risultato del metodo delle opzioni WebDriver (ad esempio, `Browser.chromeOptions()`).

Con l'oggetto `Browser` sono supportati i seguenti metodi.

Method	Description
`navigate(url)`	Passa all'URL specificato. Parametri: `url` - (string) URL a cui passare.
`getUrl()`	Restituisce una stringa con l'URL della pagina aperta.
`getPageSource()`	Restituisce una stringa con il sorgente della pagina aperta.
`findElement(strategy, selector)`	Restituisce un oggetto `Element` con un elemento nella pagina aperta (oppure restituisce `null` se nessun elemento corrisponde a `strategy` e `selector`). Parametri: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Strategia di individuazione; `selector` - (string) Selettore dell'elemento che utilizza la strategia di individuazione specificata.

Method	Description
`findElements(strategy, target)`	Restituisce un array di oggetti `Element` con più elementi nella pagina aperta (oppure restituisce un array vuoto se nessun elemento corrisponde alla strategia di individuazione e al target). Parametri: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Strategia di individuazione; `target` - (string) Selettore dell'elemento che utilizza la strategia di individuazione specificata.
`getCookies()`	Restituisce un array di oggetti `Cookie`.
`addCookie(cookie)`	Imposta i cookie. Parametri: `cookie` - (oggetto `Cookie`) Cookie da impostare.
`getScreenshot()`	Restituisce una stringa codificata in base64 che rappresenta un'immagine della viewport del browser.
`setScreenSize(x,y)`	Imposta la dimensione della viewport del browser. Parametri: `x` - (string) Larghezza in pixel; `y` - (string) Altezza in pixel.
`setScriptTimeout(timeout)`	Imposta il timeout di caricamento dello script. Parametri: `timeout` - (integer) Valore del timeout in millisecondi.
`setSessionTimeout(timeout)`	Imposta il timeout della sessione (caricamento della pagina). Parametri: `timeout` - (integer) Valore del timeout in millisecondi.
`setElementWaitTimeout(timeout)`	Imposta il timeout (implicito) della strategia di individuazione degli elementi. Parametri: `timeout` - (integer) Valore del timeout in millisecondi.
`collectPerfEntries(mark)`	Raccoglie le voci di prestazione da recuperare con il metodo `getResult()`. Parametri: `mark` - (string, optional) Contrassegno dell'istantanea delle prestazioni.
`getRawPerfEntries()`	Restituisce un array di oggetti voce di prestazione.
`getResult()`	Restituisce un oggetto `Result` con le statistiche della sessione del browser (informazioni sugli errori, istantanee delle prestazioni, ecc.).
`getError()`	Restituisce un oggetto `BrowserError` con gli errori del browser (oppure restituisce `null` se non ci sono errori del browser).
`setError(message)`	Imposta un messaggio di errore personalizzato da includere nell'oggetto `Result`. Parametri: `message` - (string) Messaggio di errore.
`discardError()`	Scarta l'errore da restituire nell'oggetto `Result`.
`getAlert()`	Restituisce un oggetto `Alert` con gli avvisi del browser (oppure restituisce `null` se non ci sono avvisi del browser).
`chromeOptions()`	Restituisce un oggetto `chromeOptions` con opzioni predefinite del browser Chrome.
`firefoxOptions()`	Restituisce un oggetto `firefoxOptions` con opzioni predefinite del browser Firefox.
`safariOptions()`	Restituisce un oggetto `safariOptions` con opzioni predefinite del browser Safari.
`edgeOptions()`	Restituisce un oggetto `edgeOptions` con opzioni predefinite del browser Edge.
`switchFrame(target)`	Passa al frame specificato. Parametri: `target` - (elemento del browser o integer, optional) Frame di destinazione. Per selezionare un frame tramite elemento, passare l'elemento. Per selezionare un frame tramite indice, passare il numero. Se lasciato vuoto, passerà al contesto di navigazione di livello superiore.

Tutti i metodi di Browser possono generare i seguenti errori:

- `BrowserError` - derivato dall'oggetto `Error`, viene generato se il costruttore `Browser` non riesce; contiene una proprietà aggiuntiva `browser` con un oggetto `Browser` che ha generato questo `BrowserError`.
- `WebdriverError` - derivato da `BrowserError`; contiene le stesse proprietà dell'oggetto `BrowserError`, che indicano se l'errore è stato generato in risposta a un errore nella risposta WebDriver.

Element

L'oggetto `Element` viene restituito dai metodi `findElement()`/`findElements()` dell'oggetto `Browser` e non può essere costruito direttamente.

L'oggetto `Element` rappresenta un elemento nella pagina web e fornisce metodi per interagire con esso.

I seguenti metodi sono supportati con l'oggetto `Element`.

Method	Description
`getAttribute(name)`	Restituisce una stringa con il valore dell'attributo dell'elemento (oppure restituisce `null` se l'attributo specificato non è stato trovato). Parameters: `name` - (string) Nome dell'attributo.
`getProperty(name)`	Restituisce una stringa con il valore della proprietà dell'elemento (oppure restituisce `null` se la proprietà specificata non è stata trovata). Parameters: `name` - (string) Nome della proprietà.
`getText()`	Restituisce una stringa con il valore del testo dell'elemento.
`click()`	Fa clic su un elemento.
`clear()`	Cancella il contenuto di un elemento modificabile.
`sendKeys(keys)`	Invia tasti. Parameters: `keys` - (string) Tasti da inviare.

Cookie

L'oggetto `Cookie` viene restituito dal metodo `getCookies()` dell'oggetto `Browser` e passato al metodo `addCookie()`.

Sebbene l'oggetto `Cookie` non abbia metodi, può contenere le seguenti proprietà:

Proprietà	Tipo	Descrizione
name	string	Nome del cookie.
value	string	Valore del cookie.
path	string	Percorso per cui il cookie è valido. Per impostazione predefinita è `"/"` se omesso durante l'aggiunta di un cookie.
domain	string	Dominio a cui il cookie è visibile. Per impostazione predefinita è il dominio URL del documento attivo del contesto di navigazione corrente della sessione se omesso durante l'aggiunta di un cookie.
secure	boolean	Indica se il cookie è sicuro. Per impostazione predefinita è `false` se omesso durante l'aggiunta di un cookie.
httpOnly	boolean	Indica se il cookie è solo HTTP. Per impostazione predefinita è `false` se omesso durante l'aggiunta di un cookie.
expiry	integer	Tempo di scadenza del cookie (in secondi dall'epoca Unix). Non deve essere impostato se omesso durante l'aggiunta di un cookie.
sameSite	string	L'attributo `sameSite` del cookie, che controlla se il cookie deve essere limitato a un contesto di prima parte o dello stesso sito. Può essere impostato su `"Lax"` o `"Strict"`. Per impostazione predefinita è `"None"` se omesso durante l'aggiunta di un cookie.

Avviso

L'oggetto `Alert` rappresenta un avviso di una pagina web, viene restituito dal metodo `getAlert()` dell'oggetto `Browser` e non può essere creato direttamente.

L'oggetto `Alert` contiene la proprietà `text` con il testo dell'avviso (oppure `null` se non ci sono avvisi).

I seguenti metodi sono supportati con l'oggetto `Alert`.

Method	Description
`accept()`	Accetta l'avviso.
`dismiss()`	Ignora l'avviso.

Risultato

L'oggetto `Result` contiene le statistiche della sessione ed è restituito dal metodo `getResult()` dell'oggetto `Browser`.

In genere, l'oggetto `Result` viene convertito in stringa e restituito dallo script, quindi analizzato nei valori degli item dipendenti tramite preprocessing.

Sebbene l'oggetto `Result` non abbia metodi, può contenere le seguenti proprietà.

Proprietà		Type	Descrizione
duration		string	Durata della sessione dalla creazione della sessione al recupero del risultato.
error		object	Informazioni sull'errore.
	http_status	integer	Stato HTTP restituito dal WebDriver (oppure 0 se non sono presenti errori WebDriver).
	error_code	string	Errore restituito dal WebDriver (oppure stringa vuota se non sono presenti errori WebDriver).
	message	string	Messaggio di errore del WebDriver (oppure stringa vuota se non sono presenti errori WebDriver).
performance_data		object	Statistiche delle prestazioni.
	summary	object	Riepilogo delle prestazioni.
		navigation object	Riepilogo della navigazione.
		resource object	Riepilogo delle risorse.
	details	array of objects	Statistiche delle prestazioni dopo ogni operazione che potrebbe aver causato una navigazione.
		mark string	*(facoltativo)* Contrassegno dell'istantanea delle prestazioni specificato con il metodo `collectPerfEntries()`.
		navigation object	Statistiche di navigazione.
		resource object	Riepilogo delle risorse per questo passaggio.
		user array of objects	Array di statistiche di tipo mark/measure.
	marks	array of objects	Indici delle istantanee delle prestazioni contrassegnate.
		name string	Nome del contrassegno dell'istantanea delle prestazioni.
		index integer	Indice dell'istantanea delle prestazioni nell'array details.

Pre-elaborazione da CSV a JSON

Panoramica

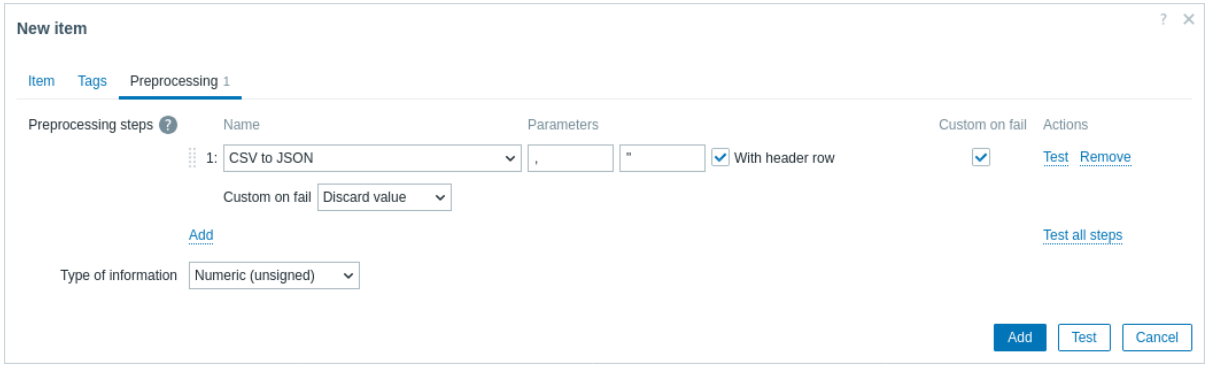
In questo passaggio di pre-elaborazione è possibile convertire i dati di un file CSV nel formato JSON. È supportato in:

- item (prototipi di item)
- regole di discovery di basso livello

Configurazione

Per configurare un passaggio di pre-elaborazione da CSV a JSON:

1. Vai alla scheda *Preprocessing* nella configurazione di item/discovery rule.
2. Fai clic su *Add*.
3. Seleziona l'opzione *CSV to JSON*.



Il primo parametro consente di impostare un delimitatore personalizzato. Se la prima riga dell'input CSV inizia con "Sep=" ed è seguita da un singolo carattere UTF-8, tale carattere verrà utilizzato come delimitatore nel caso in cui il primo parametro non sia impostato. Se il primo parametro non è impostato e non viene recuperato alcun delimitatore dalla riga "Sep=", verrà utilizzata una virgola come separatore.

Il secondo parametro opzionale consente di impostare un simbolo di virgolette.

Se la casella di controllo *With header row* è selezionata, i valori della riga di intestazione verranno interpretati come nomi delle colonne (per ulteriori informazioni, vedere Elaborazione dell'intestazione).

Se la casella di controllo *Custom on fail* è selezionata, l'item non passerà allo stato non supportato in caso di errore durante un passaggio di pre-elaborazione. È inoltre possibile impostare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

Header processing

The CSV file header line can be processed in two different ways:

- If the *With header row* checkbox is marked - header line values are interpreted as column names. In this case the column names must be unique and the data row should not contain more columns than the header row;
- If the *With header row* checkbox is not marked - the header line is interpreted as data. Column names are generated automatically (1,2,3,4...)

CSV file example:

```
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:
A quotation character within a quoted field in the input must be escaped by preceding it with another quotation character.

Processing header line

JSON output when a header line is expected:

```
[
   {
      "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"
```

```
    }
]
```

No header line processing

JSON output when a header line is not expected:

```
[
    {
        "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"
    }
]
```

Tipi di item

Panoramica

I tipi di item comprendono vari metodi per acquisire dati dal sistema. Ogni tipo di item dispone del proprio insieme di chiavi item supportate e dei parametri richiesti.

Zabbix offre attualmente i seguenti tipi di item:

- Zabbix agent
  - Monitoraggio dei file di log
- Controllo semplice
  - Monitoraggio VMware
- SNMP agent
- Trap SNMP
- Zabbix interno
- Zabbix trapper
- Controllo esterno
- Monitoraggio del database
- HTTP agent
  - Controllo Prometheus
- IPMI agent
- SSH agent
- Telnet agent
- JMX agent
- Item calcolato
- Item dipendente
- Item script
- Item browser
- Query di telemetria

I dettagli relativi a tutti i tipi di item sono inclusi nelle sottopagine di questa sezione. Sebbene i tipi di item offrano numerose opzioni per la raccolta dei dati, sono disponibili ulteriori opzioni tramite i parametri utente o i moduli caricabili.

Alcuni controlli vengono eseguiti esclusivamente dal server Zabbix (come monitoraggio senza agent), mentre altri richiedono Zabbix agent o persino Zabbix Java gateway (con il monitoraggio JMX).

Attention:
Se un determinato tipo di item richiede una specifica interfaccia (ad esempio, un controllo IPMI richiede un'interfaccia IPMI sul host), tale interfaccia deve esistere nella definizione del host.

Nella definizione del host è possibile impostare più interfacce: Zabbix agent, SNMP agent, JMX e IPMI. Se un item può utilizzare più di un'interfaccia, cercherà tra le interfacce disponibili del host (nell'ordine: Agent→SNMP→JMX→IPMI) la prima appropriata a cui collegarsi.

Tutti gli item che restituiscono testo (tipi di informazione carattere, log e testo) possono restituire anche solo spazi vuoti, se applicabile, impostando il valore restituito su una stringa vuota (supportato dalla versione 2.0).

Zabbix agent

Panoramica

Questa sezione fornisce dettagli sulle chiavi degli item che utilizzano la comunicazione con l'agent Zabbix per la raccolta dei dati.

Sono disponibili controlli agent passivi e attivi. Quando configuri un item, puoi selezionare il tipo richiesto:

- *Zabbix agent* - per controlli passivi
- *Zabbix agent (active)* - per controlli attivi

Nota che tutte le chiavi degli item supportate da Zabbix agent sono supportate anche dal nuovo Zabbix agent 2. Consulta le chiavi aggiuntive degli item che puoi usare solo con agent 2.

Chiavi degli item supportate

Di seguito sono elencate le chiavi degli item che è possibile utilizzare con Zabbix agent.

Le chiavi degli item sono elencate senza parametri e informazioni aggiuntive. Fare clic sulla chiave dell'item per visualizzare tutti i dettagli.

Chiave dell'item	Descrizione	Gruppo di item
kernel.maxfiles	Il numero massimo di file aperti supportato dal sistema operativo.	Kernel
kernel.maxproc	Il numero massimo di processi supportato dal sistema operativo.	
kernel.openfiles	Il numero di descrittori di file attualmente aperti.	
log	Il monitoraggio di un file di log.	Monitoraggio dei log
log.count	Il numero di righe corrispondenti in un file di log monitorato.	
logrt	Il monitoraggio di un file di log sottoposto a rotazione.	
logrt.count	Il numero di righe corrispondenti in un file di log monitorato sottoposto a rotazione.	
modbus.get	Legge i dati Modbus.	Modbus
net.dns	Verifica lo stato di un servizio DNS.	Rete
net.dns.perf	Verifica le prestazioni di un servizio DNS.	
net.dns.record	Esegue una query DNS.	
net.if.collisions	Il numero di collisioni fuori finestra.	
net.if.discovery	L'elenco delle interfacce di rete.	
net.if.get	L'elenco delle interfacce di rete con informazioni dettagliate.	
net.if.in	Le statistiche sul traffico in entrata su un'interfaccia di rete.	
net.if.out	Le statistiche sul traffico in uscita su un'interfaccia di rete.	
net.if.total	La somma delle statistiche sul traffico in entrata e in uscita su un'interfaccia di rete.	
net.tcp.listen	Verifica se questa porta TCP è nello stato LISTEN.	
net.tcp.port	Verifica se è possibile stabilire una connessione TCP alla porta specificata.	
net.tcp.service	Verifica se un servizio è in esecuzione e accetta connessioni TCP.	
net.tcp.service.perf	Verifica le prestazioni di un servizio TCP.	
net.tcp.socket.count	Restituisce il numero di socket TCP che corrispondono ai parametri.	
net.udp.listen	Verifica se questa porta UDP è nello stato LISTEN.	
net.udp.service	Verifica se un servizio è in esecuzione e risponde alle richieste UDP.	

Chiave dell'item	Descrizione	Gruppo di item
net.udp.service.perf	Verifica le prestazioni di un servizio UDP.	
net.udp.socket.count	Restituisce il numero di socket UDP che corrispondono ai parametri.	
proc.cpu.util	La percentuale di utilizzo della CPU del processo.	Processi
proc.get	L'elenco dei processi del sistema operativo e dei relativi parametri.	
proc.mem	La memoria utilizzata dal processo, in byte.	
proc.num	Il numero di processi.	
sensor	La lettura del sensore hardware.	Sensori
system.boottime	L'ora di avvio del sistema.	Sistema
system.cpu.discovery	L'elenco delle CPU/dei core CPU rilevati.	
system.cpu.intr	Gli interrupt del dispositivo.	
system.cpu.load	Il carico della CPU.	
system.cpu.num	Il numero di CPU.	
system.cpu.switches	Il numero di cambi di contesto.	
system.cpu.util	La percentuale di utilizzo della CPU.	
system.hostname	Il nome host del sistema.	
system.hw.chassis	Le informazioni sullo chassis.	
system.hw.cpu	Le informazioni sulla CPU.	
system.hw.devices	L'elenco dei dispositivi PCI o USB.	
system.hw.macaddr	L'elenco degli indirizzi MAC.	
system.localtime	L'ora del sistema.	
system.run	Esegue il comando specificato sull'host.	
system.stat	Le statistiche del sistema.	
system.sw.arch	Le informazioni sull'architettura software.	
system.sw.os	Le informazioni sul sistema operativo.	
system.sw.os.get	Informazioni dettagliate sul sistema operativo (versione, tipo, nome della distribuzione, versione principale e secondaria, ecc.).	
system.sw.packages	L'elenco dei pacchetti installati.	
system.sw.packages.get	Un elenco dettagliato dei pacchetti installati.	
system.swap.in	Le statistiche dello swap-in (dal dispositivo alla memoria).	
system.swap.out	Le statistiche dello swap-out (dalla memoria al dispositivo).	
system.swap.size	La dimensione dello spazio di swap in byte o in percentuale del totale.	
system.uname	L'identificazione del sistema.	
system.uptime	Il tempo di attività del sistema in secondi.	
system.users.num	Il numero di utenti connessi.	
vfs.dev.discovery	L'elenco dei dispositivi a blocchi e del relativo tipo.	File system virtuali
vfs.dev.get	L'elenco dei dispositivi a blocchi con informazioni dettagliate.	
vfs.dev.read	Le statistiche di lettura del disco.	
vfs.dev.write	Le statistiche di scrittura del disco.	
vfs.dir.count	Il numero di voci della directory.	
vfs.dir.get	L'elenco delle voci della directory.	
vfs.dir.size	La dimensione della directory.	
vfs.file.cksum	Il checksum del file, calcolato mediante l'algoritmo UNIX cksum.	
vfs.file.contents	Recupera il contenuto di un file.	
vfs.file.exists	Verifica se il file esiste.	
vfs.file.get	Restituisce informazioni su un file.	
vfs.file.md5sum	Il checksum MD5 del file.	
vfs.file.owner	Recupera il proprietario di un file.	
vfs.file.permissions	Restituisce una stringa di 4 cifre contenente il numero ottale con i permessi UNIX.	
vfs.file.regexp	Recupera una stringa dal file.	
vfs.file.regmatch	Trova una stringa nel file.	
vfs.file.size	La dimensione del file.	
vfs.file.time	Le informazioni sull'ora del file.	
vfs.fs.discovery	L'elenco dei file system montati con il relativo tipo e le opzioni di montaggio.	
vfs.fs.get	L'elenco dei file system montati con il relativo tipo e le opzioni di montaggio e, nella modalità *full*, lo spazio su disco disponibile e le statistiche sugli inode.	
vfs.fs.inode	Il numero o la percentuale di inode.	
vfs.fs.size	Lo spazio su disco in byte o in percentuale del totale.	
vm.memory.size	La dimensione della memoria in byte o in percentuale del totale.	Memoria virtuale

Chiave dell'item	Descrizione	Gruppo di item
web.page.get	Ottiene il contenuto di una pagina web.	Monitoraggio web
web.page.perf	Il tempo di caricamento di una pagina web completa.	
web.page.regexp	Trova una stringa nella pagina web.	
agent.hostmetadata	I metadati dell'host dell'agent.	Zabbix
agent.hostname	Il nome host dell'agent.	
agent.ping	Il controllo della disponibilità dell'agent.	
agent.variant	La variante di Zabbix agent (Zabbix agent o Zabbix agent 2).	
agent.version	La versione di Zabbix agent.	
zabbix.stats	Restituisce un insieme di metriche interne del server o del proxy Zabbix. Se vengono specificati `ip` e `port`, le metriche vengono recuperate dall'istanza remota; in caso contrario, dall'istanza locale.	
zabbix.stats (queue)	Restituisce il numero di item monitorati nella coda che sono in ritardo sul server o sul proxy Zabbix. Se vengono specificati `ip` e `port`, le metriche vengono recuperate dall'istanza remota; in caso contrario, dall'istanza locale.	

Piattaforme supportate

Salvo diversa indicazione nei dettagli dell'item, gli item dell'agent (e tutti i parametri) sono supportati su:

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

Molti item dell'agent sono supportati anche su **Windows**. Per i dettagli, vedere la pagina Windows agent item.

Dettagli della chiave item

I parametri senza parentesi angolari sono obbligatori. I parametri contrassegnati dalle parentesi angolari **< >** sono facoltativi.

kernel.maxfiles

 Il numero massimo di file aperti supportato dal sistema operativo.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD.

kernel.maxproc

 Il numero massimo di processi supportati dal sistema operativo.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux 2.6 e successivi, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

kernel.openfiles

 Il numero di descrittori di file attualmente aperti.
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux (l'item potrebbe funzionare su altre piattaforme di tipo UNIX).

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Il monitoraggio di un file di log.
 Valore restituito: *Log*.
 Vedere le piattaforme supportate.

Parametri:

- **file** - il percorso completo e il nome di un file di log;
- **regexp** - una espressione regolare che descrive il pattern richiesto;
- **encoding** - l'identificatore della code page identifier;
- **maxlines** - il numero massimo di nuove righe al secondo che l'agent invierà a Zabbix server o proxy. Questo parametro sostituisce il valore di `MaxLinesPerSecond` in `zabbix_agentd.conf`;
- **mode** - valori possibili:
  - *all* (predefinito);
  - *skip* - salta l'elaborazione dei dati più vecchi (influisce solo sugli item creati di recente);
- **output** - un template opzionale per la formattazione dell'output. La sequenza di escape **\0** viene sostituita con la parte di testo corrispondente (dal primo carattere in cui inizia la corrispondenza fino al carattere in cui termina la corrispondenza),

mentre una sequenza di escape **\N** (dove N=1...9) viene sostituita con l'N-esimo gruppo corrispondente (o con una stringa vuota se N supera il numero di gruppi acquisiti);
- **maxdelay** - il ritardo massimo in secondi. Tipo: float. Valori:
  - 0 - (predefinito) non ignorare mai le righe del file di log;
  - > 0.0 - ignorare le righe più vecchie per ottenere l'analisi delle righe più recenti entro i secondi di "maxdelay". Leggere le note su maxdelay prima di utilizzarlo!
- **options** - opzioni aggiuntive:
mtime-noreread - record non univoci, rileggi solo se la dimensione del file cambia (ignora la modifica del timestamp di modifica). (Questo parametro è deprecato dalla versione 5.0.2, perché ora mtime viene ignorato.)
- **persistent dir** (solo in zabbix_agentd su sistemi Unix; non supportato in Zabbix agent 2) - il percorso assoluto della directory in cui memorizzare i file persistenti. Vedere anche le note aggiuntive sui file persistenti.

Commenti:

- L'item deve essere configurato come active check.
- Se il file è mancante o i permessi non consentono l'accesso, l'item diventa non supportato.
- Se `output` viene lasciato vuoto, viene restituita l'intera riga contenente il testo corrispondente. Si noti che tutti i tipi di espressione regolare globali, tranne *Matches regular expression*, restituiscono sempre l'intera riga corrispondente e il parametro `output` viene ignorato.
- L'estrazione del contenuto tramite il parametro `output` avviene sull'agent.

Esempi:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Esempio di utilizzo del parametro `output` per estrarre un numero da un record di log:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #this item will
```

Esempio di utilizzo del parametro `output` per riscrivere un record di log prima dell'invio al server:

```
log[/app1/app.log,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",,,,"\1
```

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 Il conteggio delle righe corrispondenti in un file di log monitorato.
 Valore restituito: *Integer*.
 Vedere le piattaforme supportate.

Parametri:

- **file** - il percorso completo e il nome del file di log;
- **regexp** - una espressione regolare che descrive il pattern richiesto;
- **encoding** - l'identificatore della code page;
- **maxproclines** - il numero massimo di nuove righe al secondo che l'agent analizzerà (non può superare 10000). Il valore predefinito è 10*`MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - valori possibili:
  - *all* (predefinito);
  - *skip* - salta l'elaborazione dei dati più vecchi (influisce solo sugli item creati di recente).
- **maxdelay** - il ritardo massimo in secondi. Tipo: float. Valori:
  - 0 - (predefinito) non ignorare mai le righe del file di log;
  - > 0.0 - ignora le righe più vecchie per ottenere l'analisi delle righe più recenti entro i secondi "maxdelay". Leggere le note su maxdelay prima di utilizzarlo!
- **options** - opzioni aggiuntive:
mtime-noreread - record non univoci, rileggi solo se la dimensione del file cambia (ignora la modifica del timestamp di modifica). (Questo parametro è deprecato dalla versione 5.0.2, perché ora mtime viene ignorato.)
- **persistent dir** (solo in `zabbix_agentd` sui sistemi Unix; non supportato in Zabbix agent 2) - il percorso assoluto della directory in cui memorizzare i file persistenti. Vedere anche le note aggiuntive sui file persistenti.

Commenti:

- L'item deve essere configurato come controllo attivo.
- Le righe corrispondenti vengono conteggiate nelle nuove righe dall'ultimo controllo del log da parte dell'agent e quindi dipendono dall'intervallo di aggiornamento dell'item.
- Se il file manca o i permessi non consentono l'accesso, l'item diventa non supportato.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Il monitoraggio di un file di log ruotato.
 Valore restituito: *Log*.
 Vedere le piattaforme supportate.

Parametri:

- **file regexp** - il percorso assoluto del file, con il nome del file specificato tramite una espressione regolare. Si noti che l'espressione regolare si applica solo al nome del file e non deve corrispondere all'intero nome (ad esempio, `/path/to/agent` corrisponderà a `zabbix_agentd.log`).
- **regexp** - una espressione regolare che descrive il pattern di contenuto richiesto.
- **encoding** - l'identificatore della code page identifier.
- **maxlines** - il numero massimo di nuove righe al secondo che l'agent invierà a Zabbix server o proxy. Questo parametro sovrascrive il valore di `MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - i valori possibili sono:
  - *all* (predefinito);
  - *skip* - salta l'elaborazione dei dati più vecchi (influisce solo sugli item creati di recente).
- **output** - un template di formattazione dell'output opzionale. La sequenza di escape **\0** viene sostituita con la parte di testo corrispondente (dal primo carattere in cui inizia la corrispondenza fino al carattere in cui termina la corrispondenza), mentre una sequenza di escape **\N** (dove N=1...9) viene sostituita con il N-esimo gruppo corrispondente (o con una stringa vuota se N supera il numero di gruppi acquisiti).
- **maxdelay** - il ritardo massimo in secondi. Tipo: float. Valori:
  - 0 - (predefinito) non ignorare mai le righe del file di log;
  - > 0.0 - ignorare le righe più vecchie per ottenere le righe più recenti analizzate entro i secondi di "maxdelay". Leggere le note su maxdelay prima di utilizzarlo!
- **options** - il tipo di rotazione del file di log e altre opzioni. Valori possibili:
  - *rotate* (predefinito).
  - *copytruncate* - si noti che *copytruncate* non può essere usato insieme a *maxdelay*. In questo caso *maxdelay* deve essere 0 o non specificato; vedere le note su copytruncate.
  - *mtime-reread* - record non univoci, riletti se il tempo di modifica o la dimensione cambia (usato per impostazione predefinita se nessuna opzione `mtime-*` è impostata esplicitamente).
  - *mtime-noreread* - record non univoci, riletti solo se la dimensione cambia (ignora la modifica del tempo di modifica).
- **persistent dir** (solo in `zabbix_agentd` su sistemi Unix; non supportato in Zabbix agent 2) - il percorso assoluto della directory in cui memorizzare i file persistenti. Vedere anche le note aggiuntive sui file persistenti.

Commenti:

- L'item deve essere configurato come controllo attivo.
- La rotazione del log si basa sull'ultimo tempo di modifica dei file.
- Si noti che `logrt` è progettato per funzionare con un solo file di log attualmente attivo, con diversi altri file inattivi corrispondenti ruotati. Se, ad esempio, una directory contiene molti file di log attivi, è necessario creare un item `logrt` separato per ciascuno di essi. Altrimenti, se un item `logrt` rileva troppi file, ciò può portare all'esaurimento della memoria e a un arresto del monitoraggio.
- Se `output` viene lasciato vuoto, viene restituita l'intera riga contenente il testo corrispondente. Si noti che tutti i tipi di espressioni regolari globali, tranne *Matches regular expression*, restituiscono sempre l'intera riga corrispondente e il parametro `output` viene ignorato.
- L'estrazione del contenuto tramite il parametro `output` avviene sull'agent.
- Nel parametro `file regexp`, il percorso della directory del log e l'espressione regolare del file di log devono essere separati dal corretto separatore di directory:
  - Su Windows, il separatore deve essere una barra rovesciata (\); le barre oblique possono essere tollerate in altre posizioni, tranne quella che separa il percorso della directory del log e l'espressione regolare del file di log (vedere gli esempi sotto).
  - Sugli altri sistemi, deve essere una barra obliqua (/).

Esempi per Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will collect data from log files
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will collect dat
```

Esempi per altri sistemi:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #this item will match a file like "logfile1" (will no
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #this item will collect data fr
```

Esempio di utilizzo del parametro output per estrarre un numero da un record di log:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #this ite
```

Esempio di utilizzo del parametro output per riscrivere un record di log prima dell'invio a server:

```
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>]

 Il conteggio delle righe corrispondenti in un file di log monitorato che viene ruotato.
 Valore restituito: *Integer*.
 Vedere le piattaforme supportate.

Parametri:

- **file regexp** - il percorso assoluto del file, con il nome del file specificato tramite una espressione regolare. Si noti che l'espressione regolare si applica solo al nome del file e non deve corrispondere all'intero nome (ad esempio, `/path/to/agent` corrisponderà a `zabbix_agentd.log`).
- **regexp** - una espressione regolare che descrive il pattern richiesto.
- **encoding** - l'identificatore della code page.
- **maxproclines** - il numero massimo di nuove righe al secondo che l'agent analizzerà (non può superare 10000). Il valore predefinito è 10*`MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - valori possibili:
  - *all* (predefinito);
  - *skip* - salta l'elaborazione dei dati più vecchi (influisce solo sugli item creati di recente).
- **maxdelay** - il ritardo massimo in secondi. Tipo: float. Valori:
  - 0 - (predefinito) non ignorare mai le righe del file di log;
  - > 0.0 - ignora le righe più vecchie per ottenere che le righe più recenti vengano analizzate entro "maxdelay" secondi. Leggere le note su `maxdelay` prima di utilizzarlo!
- **options** - il tipo di rotazione del file di log e altre opzioni. Valori possibili:
  - *rotate* (predefinito).
  - *copytruncate* - si noti che `copytruncate` non può essere usato insieme a `maxdelay`. In questo caso `maxdelay` deve essere 0 o non specificato; vedere le note su `copytruncate`.
  - *mtime-reread* - record non univoci, riletti se il tempo di modifica o la dimensione cambia (predefinito).
  - *mtime-noreread* - record non univoci, riletti solo se la dimensione cambia (ignora la modifica del tempo di modifica).
- **persistent dir** (solo in `zabbix_agentd` su sistemi Unix; non supportato in Zabbix agent 2) - il percorso assoluto della directory in cui memorizzare i file persistenti. Vedere anche le note aggiuntive sui file persistenti.

Commenti:

- L'item deve essere configurato come active check.
- Le righe corrispondenti vengono conteggiate nelle nuove righe dall'ultimo controllo del log da parte dell'agent e quindi dipendono dall'intervallo di aggiornamento dell'item.
- La rotazione del log si basa sull'ultimo tempo di modifica dei file.
- Nel parametro `file regexp`, il percorso della directory del log e l'espressione regolare del file di log devono essere separati dal separatore di directory corretto:
  - Su Windows, il separatore deve essere una barra rovesciata (\); le barre oblique (/) possono essere tollerate in altre posizioni, tranne quella che separa il percorso della directory del log e l'espressione regolare del file di log (vedere gli esempi sotto).
  - Sugli altri sistemi, deve essere una barra obliqua (/).

Esempi per Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #questo item conterà il numero di corr
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #questo item conterà
```

Esempi per altri sistemi:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #questo item conterà il numero di corrispon
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #questo item conterà il n
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Legge i dati Modbus.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, Windows.

Parametri:

- **endpoint** - l'endpoint definito come `protocol://connection_string`;
- **slave id** - l'ID dello slave;
- **function** - la funzione Modbus;
- **address** - l'indirizzo del primo registro, coil o input;
- **count** - il numero di record da leggere;
- **type** - il tipo di dati;
- **endianness** - la configurazione di endianness;

- **offset** - il numero di registri, a partire da `address`, i cui risultati verranno scartati.

Su Windows, questo item segue i percorsi UNC nel parametro `endpoint`, il che può rappresentare un rischio per la sicurezza.

Vedere una descrizione dettagliata dei parametri.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Controlla lo stato di un servizio DNS.
 Valori restituiti: - 0 - la risoluzione DNS è fallita (il server DNS non ha risposto o ha restituito un errore); - 1 - la risoluzione DNS è riuscita.
 Vedi le piattaforme supportate.

Parametri:

- **ip** (ignorato su Windows a meno che non si usi Zabbix agent 2) - l'indirizzo IP del server DNS (lasciare vuoto per usare il server DNS predefinito);
- **name** - il nome DNS da interrogare;
- **type** - il tipo di record da interrogare (il valore predefinito è `SOA`);
- **timeout** (ignorato su Windows a meno che non si usi Zabbix agent 2) - il timeout della richiesta in secondi (il valore predefinito è 1 secondo);
- **count** (ignorato su Windows a meno che non si usi Zabbix agent 2) - il numero di tentativi per la richiesta (il valore predefinito è 2);
- **protocol** - il protocollo usato per eseguire le query DNS: *udp* (predefinito) o *tcp*.

Commenti:

- I valori possibili per `type` sono: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (non supportato per Zabbix agent su Windows, Zabbix agent 2 su tutti i sistemi operativi), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Per le ricerche DNS inverse (quando `type` è impostato su *PTR*), è possibile fornire il nome DNS sia in formato invertito sia non invertito (vedi gli esempi sotto). Nota che quando viene richiesto un record PTR, il nome DNS è in realtà un indirizzo IP.
- I nomi di dominio internazionalizzati non sono supportati; usare invece nomi codificati in IDNA.

Esempi:

```
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>]

 Controlla le prestazioni di un servizio DNS.
 Valore restituito: *Float* (0 - il servizio non è disponibile; secondi - il numero di secondi trascorsi in attesa di una risposta dal servizio).
 Vedi piattaforme supportate.

Parametri:

- **ip** (ignorato su Windows a meno che non si utilizzi Zabbix agent 2) - l'indirizzo IP del server DNS (lasciare vuoto per il server DNS predefinito);
- **name** - il nome DNS da interrogare;
- **type** - il tipo di record da interrogare (il valore predefinito è `SOA`);
- **timeout** (ignorato su Windows a meno che non si utilizzi Zabbix agent 2) - il timeout della richiesta in secondi (il valore predefinito è 1 secondo);
- **count** (ignorato su Windows a meno che non si utilizzi Zabbix agent 2) - il numero di tentativi per la richiesta (il valore predefinito è 2);
- **protocol** - il protocollo usato per eseguire le query DNS: *udp* (predefinito) oppure *tcp*.

Commenti:

- I valori possibili per `type` sono:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (non supportato per Zabbix agent su Windows, Zabbix agent 2 su tutti i sistemi operativi), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Per le ricerche DNS inverse (quando `type` è impostato su *PTR*), è possibile fornire il nome DNS sia in formato invertito sia non invertito (vedere gli esempi sotto). Si noti che quando viene richiesto un record PTR, il nome DNS è in realtà un indirizzo IP.
- I nomi di dominio internazionalizzati non sono supportati; utilizzare invece nomi codificati IDNA.
- L'item restituisce un tempo di risposta invece di 0 quando il server DNS risponde con un codice di errore (ad esempio, `NXDOMAIN` o `SERVFAIL`).

Esempi:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Esegue una query DNS.

Zabbix agent 2 fornisce anche net.dns.get, che offre funzionalità aggiuntive come un maggior numero di tipi di record e un controllo più preciso del monitoraggio DNS. Valore restituito: una stringa di caratteri con il tipo di informazione richiesto.
 Vedere le piattaforme supportate.

Parametri:

- **ip** (ignorato su Windows a meno che non si utilizzi Zabbix agent 2) - l'indirizzo IP del server DNS (lasciare vuoto per usare il server DNS predefinito);
- **name** - il nome DNS da interrogare;
- **type** - il tipo di record da interrogare (il valore predefinito è SOA);
- **timeout** (ignorato su Windows a meno che non si utilizzi Zabbix agent 2) - il timeout della richiesta in secondi (il valore predefinito è 1 secondo);
- **count** (ignorato su Windows a meno che non si utilizzi Zabbix agent 2) - il numero di tentativi per la richiesta (il valore predefinito è 2);
- **protocol** - il protocollo usato per eseguire le query DNS: *udp* (predefinito) oppure *tcp*.

Commenti:

- I valori possibili per `type` sono:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (non supportato per Zabbix agent su Windows, Zabbix agent 2 su tutti i sistemi operativi), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Per le ricerche DNS inverse (quando `type` è impostato su *PTR*), è possibile fornire il nome DNS in formato invertito o non invertito (vedere gli esempi sotto). Si noti che quando viene richiesto un record PTR, il nome DNS è in realtà un indirizzo IP.
- I nomi di dominio internazionalizzati non sono supportati, utilizzare invece nomi codificati IDNA.

Esempi:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 Il numero di collisioni fuori finestra.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. Su NetBSD sono richiesti i privilegi di root.

Parametro:

- **if** - nome dell'interfaccia di rete

net.if.discovery

 L'elenco delle interfacce di rete. Utilizzato per il low-level discovery.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.get[<regexp>]

 Elenco delle interfacce di rete con informazioni dettagliate.
 Utilizzato per la discovery di basso livello.
 Questo item è una versione estesa dell'item Zabbix agent `net.if.discovery`.
 Valore restituito: *oggetto JSON*.
 Piattaforme supportate: Linux.

Parametri:

- **regexp** - una espressione regolare che corrisponde ai nomi delle interfacce di rete.

Commenti:

- Restituisce un oggetto JSON con le interfacce di rete i cui nomi corrispondono alla regular expression specificata (vedere maggiori dettagli).

net.if.in[if,<mode>]

 Le statistiche del traffico in ingresso su un'interfaccia di rete.
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Su NetBSD sono richiesti privilegi di root.

Parametri:

- **if** - nome dell'interfaccia di rete (Unix); descrizione completa dell'interfaccia di rete o indirizzo IPv4; oppure, se tra parentesi graffe, GUID dell'interfaccia di rete (Windows);
- **mode** - valori possibili:
  - *bytes* - numero di byte (predefinito);
  - *packets* - numero di pacchetti;
  - *errors* - numero di errori;
  - *dropped* - numero di pacchetti scartati;
  - *overruns (fifo)* - numero di errori del buffer FIFO;
  - *frame* - numero di errori di framing dei pacchetti;
  - *compressed* - numero di pacchetti compressi ricevuti dal driver del dispositivo;
  - *multicast* - numero di frame multicast ricevuti dal driver del dispositivo.

Commenti:

- È possibile usare questa chiave con il passaggio di preprocessing *Change per second* per ottenere le statistiche in byte al secondo.
- La modalità *dropped* è supportata solo su Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Le modalità *overruns*, *frame*, *compressed*, *multicast* sono supportate solo su Linux.
- Su HP-UX questo item non fornisce dettagli sulle interfacce loopback (ad esempio lo0).

Esempi:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 Le statistiche del traffico in uscita su un'interfaccia di rete.
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Su NetBSD sono richiesti privilegi di root.

Parametri:

- **if** - nome dell'interfaccia di rete (Unix); descrizione completa dell'interfaccia di rete o indirizzo IPv4; oppure, se tra parentesi graffe, GUID dell'interfaccia di rete (Windows);
- **mode** - valori possibili:
  - *bytes* - numero di byte (predefinito);
  - *packets* - numero di pacchetti;
  - *errors* - numero di errori;
  - *dropped* - numero di pacchetti scartati;
  - *overruns (fifo)* - numero di errori del buffer FIFO;
  - *collisions (colls)* - numero di collisioni rilevate sull'interfaccia;
  - *carrier* - numero di perdite di carrier rilevate dal driver del dispositivo;
  - *compressed* - numero di pacchetti compressi trasmessi dal driver del dispositivo.

Commenti:

- È possibile usare questa chiave con il passo di preprocessing *Change per second* per ottenere le statistiche in byte al secondo.
- La modalità *dropped* è supportata solo su Linux, HP-UX.
- Le modalità *overruns*, *collision*, *carrier*, *compressed* sono supportate solo su Linux.
- Su HP-UX questo item non fornisce dettagli sulle interfacce loopback (ad esempio lo0).

Esempi:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 La somma delle statistiche del traffico in ingresso e in uscita su un'interfaccia di rete.
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Su NetBSD sono richiesti privilegi di root.

Parametri:

- **if** - nome dell'interfaccia di rete (Unix); descrizione completa dell'interfaccia di rete o indirizzo IPv4; oppure, se tra parentesi graffe, GUID dell'interfaccia di rete (Windows);
- **mode** - valori possibili:
  - *bytes* - numero di byte (predefinito);
  - *packets* - numero di pacchetti;
  - *errors* - numero di errori;
  - *dropped* - numero di pacchetti scartati;
  - *overruns (fifo)* - numero di errori del buffer FIFO;
  - *collisions (colls)* - numero di collisioni rilevate sull'interfaccia;
  - *compressed* - numero di pacchetti compressi trasmessi o ricevuti dal driver del dispositivo.

Commenti:

- È possibile usare questa chiave con il passaggio di preprocessing *Change per second* per ottenere le statistiche in byte al secondo.
- La modalità *dropped* è supportata solo su Linux, HP-UX. I pacchetti scartati sono supportati solo se sia `net.if.in` sia `net.if.out` funzionano per i pacchetti scartati sulla tua piattaforma.
- Le modalità *overruns*, *collision*, *compressed* sono supportate solo su Linux.
- Su HP-UX questo item non fornisce dettagli sulle interfacce loopback (ad esempio lo0).

Esempi:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Verifica se questa porta TCP è nello stato LISTEN.
 Valori restituiti: 0 - non è nello stato LISTEN; 1 - è nello stato LISTEN.
 Piattaforme supportate: Linux, FreeBSD, Solaris, MacOS X, Windows.

Parametro:

- **port** - numero della porta TCP

Nei kernel Linux 2.6.14 e successivi, le informazioni sui socket TCP in ascolto vengono ottenute dall'interfaccia NETLINK del kernel, se possibile. In caso contrario, le informazioni vengono recuperate dai file `/proc/net/tcp` e `/roc/net/tcp6`.

Esempio:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Verifica se è possibile stabilire una connessione TCP alla porta specificata.
 Valori restituiti: 0 - impossibile connettersi; 1 - connessione possibile.
 Vedere le piattaforme supportate.

Parametri:

- **ip** - l'indirizzo IP o il nome DNS (il valore predefinito è `127.0.0.1`);
- **port** - il numero della porta.

Commenti:

- Per un semplice test delle prestazioni TCP, usare `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Questi controlli possono generare messaggi aggiuntivi nei file di log dei daemon di sistema (di solito vengono registrate le sessioni SMTP e SSH).

Esempio:

```
net.tcp.port[,80] #questo item può essere usato per testare la disponibilità del server web in esecuzione
```

net.tcp.service[service,<ip>,<port>]

 Verifica se un servizio è in esecuzione e accetta connessioni TCP.
 Valori restituiti: 0 - il servizio è inattivo; 1 - il servizio è in esecuzione.
 Vedere le piattaforme supportate.

Parametri:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* o *telnet* (vedere i dettagli);
- **ip** - l'indirizzo IP o il nome DNS (il valore predefinito è `127.0.0.1`);
- **port** - il numero di porta (per impostazione predefinita viene usato il numero di porta standard del servizio).

Commenti:

- Questi controlli possono generare messaggi aggiuntivi nei file di log dei daemon di sistema (di solito vengono registrate le sessioni SMTP e SSH).
- Il controllo dei protocolli crittografati (come IMAP sulla porta 993 o POP sulla porta 995) al momento non è supportato. Come soluzione alternativa, usare `net.tcp.port[]` per controlli di questo tipo.
- Il controllo di LDAP e HTTPS su Windows è supportato solo da Zabbix agent 2.
- Il controllo telnet cerca un prompt di accesso (':' alla fine).

Esempio:

```
net.tcp.service[ftp,,45] #this item can be used to test the availability of FTP server on TCP port 45
```

net.tcp.service.perf[service,<ip>,<port>]

 Controlla le prestazioni di un servizio TCP.
 Valori restituiti: *Float* (0 - il servizio non è disponibile; secondi - il numero di secondi trascorsi in attesa di una risposta dal servizio).
 Vedi piattaforme supportate.

Parametri:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* o *telnet* (vedi dettagli);
- **ip** - l'indirizzo IP o il nome DNS (predefinito `127.0.0.1`);
- **port** - il numero della porta (per impostazione predefinita viene usato il numero di porta standard del servizio).

Commenti:

- Il controllo dei protocolli cifrati (come IMAP sulla porta 993 o POP sulla porta 995) al momento non è supportato. Come soluzione alternativa, usa `net.tcp.service.perf[tcp,<ip>,<port>]` per controlli di questo tipo.
- Il controllo telnet cerca un prompt di login (':' alla fine).

Esempio:

```
net.tcp.service.perf[ssh] #this item can be used to test the speed of initial response from the SSH server
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Restituisce il numero di socket TCP che corrispondono ai parametri.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, Windows.

Parametri:

- **laddr** - l'indirizzo IPv4/6 locale o la sottorete CIDR;
- **lport** - il numero di porta locale o il nome del servizio;
- **raddr** - l'indirizzo IPv4/6 remoto o la sottorete CIDR;
- **rport** - il numero di porta remota o il nome del servizio;
- **state** - lo stato della connessione (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Esempio:

```
net.tcp.socket.count[,80,,,established] #il numero di connessioni alla porta TCP locale 80 nello stato est
```

net.udp.listen[port]

 Controlla se questa porta UDP è nello stato LISTEN.
 Valori restituiti: 0 - non è nello stato LISTEN; 1 - è nello stato LISTEN.
 Piattaforme supportate: Linux, FreeBSD, Solaris, MacOS X.

Parametro:

- **port** - numero della porta UDP

Esempio:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Verifica se un servizio è in esecuzione e risponde alle richieste UDP.
 Valori restituiti: 0 - il servizio non è attivo; 1 - il servizio è in esecuzione.
 Vedi le piattaforme supportate.

Parametri:

- **service** - *ntp* (vedi dettagli);
- **ip** - l'indirizzo IP o il nome DNS (il valore predefinito è `127.0.0.1`);
- **port** - il numero di porta (per impostazione predefinita viene usato il numero di porta standard del servizio).

Esempio:

```
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>]

 Controlla le prestazioni di un servizio UDP.
 Valori restituiti: *Float* (0 - il servizio non è attivo; secondi - il numero di secondi trascorsi in attesa di una risposta dal servizio).
 Vedi piattaforme supportate.

Parametri:

- **service** - *ntp* (vedi dettagli);
- **ip** - l'indirizzo IP o il nome DNS (il valore predefinito è `127.0.0.1`);
- **port** - il numero di porta (per impostazione predefinita viene usato il numero di porta standard del servizio).

Esempio:

```
net.udp.service.perf[ntp] #this item can be used to test response time from NTP service
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Restituisce il numero di socket UDP che corrispondono ai parametri.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, Windows.

Parametri:

- **laddr** - l'indirizzo IPv4/6 locale o la subnet CIDR;
- **lport** - il numero di porta locale o il nome del servizio;
- **raddr** - l'indirizzo IPv4/6 remoto o la subnet CIDR;
- **rport** - il numero di porta remota o il nome del servizio;
- **state** - lo stato della connessione (*established*, *unconn*).

Esempio:

```
net.udp.socket.count[,,,,established] #restituisce il numero di socket UDP nello stato connesso
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 La percentuale di utilizzo CPU del processo.
 Valore restituito: *Float*.
 Piattaforme supportate: Linux, Solaris[6].

Parametri:

- **name** - il nome del processo (predefinito: *all processes*);
- **user** - il nome dell'utente (predefinito: *all users*);
- **type** - il tipo di utilizzo CPU: *total* (predefinito), *user* o *system*;
- **cmdline** - filtro per riga di comando (è una espressione regolare);
- **mode** - la modalità di raccolta dei dati: *avg1* (predefinito), *avg5* o *avg15*;
- **zone** - la zona di destinazione: *current* (predefinito) o *all*. Questo parametro è supportato solo su Solaris.

Commenti:

- Il valore restituito si basa sulla percentuale di utilizzo di un singolo core CPU. Ad esempio, l'utilizzo CPU di un processo che usa completamente due core è 200%.
- I dati di utilizzo CPU del processo vengono raccolti da un collector che supporta un massimo di 1024 query univoche (per nome, utente e riga di comando). Le query non accessibili durante le ultime 24 ore vengono rimosse dal collector.
- Quando si imposta il parametro zone su *current* (o predefinito) nel caso in cui l'agent sia stato compilato su un Solaris senza supporto per le zone, ma venga eseguito su un Solaris più recente in cui le zone sono supportate, l'agent restituirà NOTSUPPORTED (l'agent non può limitare i risultati alla sola zona corrente). Tuttavia, *all* è supportato in questo caso.

Esempi:

```
proc.cpu.util[,root] #utilizzo CPU di tutti i processi in esecuzione sotto l'utente "root"
proc.cpu.util[zabbix_server,zabbix] #utilizzo CPU di tutti i processi zabbix_server in esecuzione sotto l'
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 L'elenco dei processi del sistema operativo e dei loro parametri. Può essere usato per la discovery a basso livello.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, FreeBSD, Windows, OpenBSD, NetBSD.

Parametri:

- **name** - il nome del processo (predefinito *tutti i processi*).
- **user** - il nome dell'utente (predefinito *tutti gli utenti*).
- **cmdline** - filtro in base alla riga di comando (è una espressione regolare). Questo parametro non è supportato per Windows; sulle altre piattaforme non è supportato se mode è impostato su 'summary'.

- **mode** - valori possibili:
  - *process* (predefinito);
  - *thread* (non supportato per NetBSD);
  - *summary*. Vedi un elenco dei parametri del processo restituiti per ciascun mode e sistema operativo.

Commenti:

- Se un valore non può essere recuperato, ad esempio a causa di un errore (il processo è già terminato, mancanza di permessi, fallimento della chiamata di sistema), verrà restituito −1.
- Vedi le note sulla selezione dei processi con i parametri `name` e `cmdline` (specifico per Linux).

Esempi:

```
proc.get[zabbix_server,zabbix,,process] #elenco di tutti i processi zabbix_server in esecuzione sotto l'ut
proc.get[java,,,thread] #elenco di tutti i processi Java, restituisce una voce per thread
proc.get[,zabbix,,summary] #dati combinati per i processi di ciascun tipo in esecuzione sotto l'utente zab
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 La memoria utilizzata dal processo in byte.
 Valore restituito: - *Integer* - con `mode` impostato su *max*, *min*, *sum*; - *Float* - con `mode` impostato su *avg*
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Parametri:

- **name** - il nome del processo (predefinito: *all processes*);
- **user** - il nome dell'utente (predefinito: *all users*);
- **mode** - valori possibili: *avg*, *max*, *min* o *sum* (predefinito);
- **cmdline** - filtro in base alla riga di comando (è una espressione regolare);
- **memtype** - il tipo di memoria utilizzato dal processo

Commenti:

- Il parametro `memtype` è supportato solo su Linux, FreeBSD, Solaris[6], AIX.
- Quando più processi utilizzano memoria condivisa, la somma della memoria utilizzata dai processi può produrre valori elevati e non realistici.

Vedi le note sulla selezione dei processi con i parametri `name` e `cmdline` (specifiche di Linux).

Quando questo item viene richiamato dalla riga di comando e contiene un parametro della riga di comando (ad esempio usando la modalità di test dell'agent: `zabbix_agentd -t proc.mem[,,,apache2]`), verrà conteggiato un processo aggiuntivo, poiché l'agent conterà se stesso.

Esempi:

```
proc.mem[,root] #la memoria utilizzata da tutti i processi in esecuzione sotto l'utente "root"
proc.mem[zabbix_server,zabbix] #la memoria utilizzata da tutti i processi zabbix_server in esecuzione sott
proc.mem[,oracle,max,oracleZABBIX] #la memoria utilizzata dal processo che consuma più memoria tra quelli
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Il numero di processi.
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parametri:

- **name** - il nome del processo (il valore predefinito è *all processes*).
- **user** - il nome utente (il valore predefinito è *all users*).
- **state** - valori possibili:
  - *all* (predefinito);
  - *disk* - sospensione non interrompibile;
  - *run* - in esecuzione;
  - *sleep* - sospensione interrompibile;
  - *trace* - arrestato;
  - *zomb* - zombie.
- **cmdline** - filtro per la riga di comando (è una espressione regolare).
- **zone** - la zona di destinazione: *current* (predefinito) oppure *all*. Questo parametro è supportato solo su Solaris.

Commenti:

- I parametri di stato *disk* e *trace* sono supportati solo su Linux, FreeBSD, OpenBSD, NetBSD.
- Su Windows, sono supportati solo i parametri `name` e `user`.
- Quando questo item viene richiamato dalla riga di comando e contiene un parametro della riga di comando (ad esempio usando la modalità di test dell'agent: `zabbix_agentd -t proc.num[,,,apache2]`), verrà conteggiato un processo aggiuntivo, poiché l'agent conterà se stesso.

- Quando si imposta il parametro `zone` su `current` (o predefinito) nel caso in cui l'agent sia stato compilato su un Solaris senza supporto per le zone, ma venga eseguito su una versione più recente di Solaris in cui le zone sono supportate, l'agent restituirà NOTSUPPORTED (l'agent non può limitare i risultati alla sola zona corrente). Tuttavia, *all* è supportato in questo caso.
- Vedere le note sulla selezione dei processi con i parametri `name` e `cmdline` (specifiche per Linux).

Esempi:

```
proc.num[,mysql] #il numero di processi in esecuzione sotto l'utente mysql
proc.num[apache2,www-data] #il numero di processi apache2 in esecuzione sotto l'utente www-data
proc.num[,oracle,sleep,oracleZABBIX] #il numero di processi nello stato sleep in esecuzione sotto Oracle c
```

sensor[device,sensor,<mode>]

 Lettura di un sensore hardware.
 Valore restituito: *Float*.
 Piattaforme supportate: Linux, OpenBSD.

Parametri:

- **device** - il nome del dispositivo;
- **sensor** - il nome del sensore;
- **mode** - valori possibili: *avg*, *max* o *min*.

Commenti:

- Legge `/proc/sys/dev/sensors` su Linux 2.4.
- Legge `/sys/class/hwmon` su Linux 2.6+. Vedi una descrizione più dettagliata dell'item sensor su Linux.
- Legge il MIB `hw.sensors` su OpenBSD.
- Su Linux 2.4, se specifichi mode, allora device e sensor sono trattati come espressioni regolari.
- Su Linux 2.6+ e OpenBSD, oppure se mode viene omesso, device e sensor sono trattati come valori letterali.

Esempi:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #la temperatura di una CPU
sensor["cpu[0-2]$",temp,avg] #la temperatura media delle prime tre CPU
```

system.boottime

 L'ora di avvio del sistema.
 Valore restituito: *Intero (timestamp Unix)*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

system.cpu.discovery

 L'elenco delle CPU/core CPU rilevate. Utilizzato per il low-level discovery.
 Valore restituito: *stringa JSON*.
 Vedi piattaforme supportate.

system.cpu.intr

 Le interruzioni del dispositivo.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<mode>]

 Il carico della CPU.
 Valore restituito: *Float*.
 Vedere piattaforme supportate.

Parametri:

- **cpu** - valori possibili: *all* (predefinito) oppure *percpu* (il carico totale diviso per il numero di CPU online);
- **mode** - valori possibili: *avg1* (media di un minuto, predefinito), *avg5* oppure *avg15*.

Esempio:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 Il numero di CPU.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parametro:

- **type** - valori possibili: *online* (predefinito) oppure *max*

Il parametro di tipo *max* è supportato solo su Linux, FreeBSD, Solaris, MacOS X.

Esempio:

```
system.cpu.num
```

system.cpu.switches

 Il numero di cambi di contesto.
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 La percentuale di utilizzo della CPU.
 Valore restituito: *Float*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parametri:

- **cpu** - <*numero CPU*> oppure *all* (predefinito);
- **type** - valori possibili: *user* (predefinito), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *spin* (su OpenBSD), *guest* (su kernel Linux 2.6.24 e successivi), oppure *guest_nice* (su kernel Linux 2.6.33 e successivi);
- **mode** - valori possibili: *avg1* (media di un minuto, predefinito), *avg5* oppure *avg15*;
- **logical or physical** - valori possibili: *logical* (predefinito) oppure *physical*. Questo parametro è supportato solo su AIX.

Commenti:

- Il parametro di tipo *nice* è supportato solo su Linux, FreeBSD, HP-UX, OpenBSD, NetBSD.
- Il parametro di tipo *iowait* è supportato solo su Linux 2.6 e successivi, Solaris, AIX.
- Il parametro di tipo *interrupt* è supportato solo su Linux 2.6 e successivi, FreeBSD, OpenBSD.
- I parametri di tipo *softirq*, *steal*, *guest*, *guest_nice* sono supportati solo su Linux 2.6 e successivi.
- I parametri di modalità *avg5* e *avg15* sono supportati su Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD.

Esempio:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 Il nome host del sistema.
 Valore restituito: *String*.
 Vedi piattaforme supportate.

Parametri:

- **type** - valori possibili:
  - *netbios* (predefinito su Windows);
  - *host* (predefinito su Linux);
  - *shorthost* (restituisce la parte del nome host prima del primo punto, una stringa completa per i nomi senza punti);
  - *fqdn* (restituisce il Fully Qualified Domain Name).
- **transform** - valori possibili:
  - *none* (predefinito);
  - *lower* (converte in minuscolo).

Il valore viene acquisito prendendo `nodename` dall'output dell'API di sistema uname().

Esempi di valori restituiti:

```
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>]

 Le informazioni sullo chassis.
 Valore restituito: *String*.
 Piattaforme supportate: Linux.

Parametro:

- **info** - valori possibili: *full* (predefinito), *model*, *serial*, *type* oppure *vendor*

Commenti:

- Questa chiave item dipende dalla disponibilità della tabella SMBIOS.
- Tenterà di leggere la tabella DMI da sysfs; se l'accesso a sysfs fallisce, proverà quindi a leggere direttamente dalla memoria.
- Sono richiesti i **permessi di root** perché il valore viene acquisito leggendo da sysfs o dalla memoria.

Esempio:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 Le informazioni sulla CPU.
 Valore restituito: *String* o *Integer*.
 Piattaforme supportate: Linux.

Parametri:

- **cpu** - <*numero CPU*> oppure *all* (predefinito);
- **info** - valori possibili: *full* (predefinito), *curfreq*, *maxfreq*, *model* oppure *vendor*.

Commenti:

- Raccoglie le informazioni da `/proc/cpuinfo` e `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq`.
- Se viene specificato un numero di CPU e *curfreq* o *maxfreq*, viene restituito un valore numerico (Hz).

Esempio:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 L'elenco dei dispositivi PCI o USB.
 Valore restituito: *Testo*.
 Piattaforme supportate: Linux.

Parametro:

- **type** - *pci* (predefinito) oppure *usb*

Restituisce l'output dell'utilità lspci o lsusb (eseguita senza alcun parametro).

Esempio:

```
system.hw.devices → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge
```

system.hw.macaddr[<interface>,<format>]

 Elenco degli indirizzi MAC.
 Valore restituito: *String*.
 Piattaforme supportate: Linux.

Parametri:

- **interface** - *all* (predefinito) oppure una espressione regolare;
- **format** - *full* (predefinito) oppure *short*.

Commenti:

- Elenca gli indirizzi MAC delle interfacce il cui nome corrisponde alla espressione regolare `interface` specificata (*all* elenca tutte le interfacce).
- Se `format` è specificato come *short*, i nomi delle interfacce e gli indirizzi MAC identici non vengono elencati.

Esempio:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 L'ora di sistema.
 Valore restituito: *Integer* - con `type` impostato su *utc*; *String* - con `type` impostato su *local*.
 Vedere le piattaforme supportate.

Parametri:

- **type** - valori possibili:
  - *utc* - (predefinito) il tempo trascorso dall'Epoch (00:00:00 UTC, 1 gennaio 1970), misurato in secondi;
  - *local* - l'ora nel formato 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'.

Deve essere usato come controllo passivo per Zabbix agent; su Zabbix agent 2 può essere configurato come controllo attivo.

Esempio:

```
system.localtime[local] #create an item using this key and then use it to display the host time in the *Cl
```

system.run[command,<mode>]

 Esegue il comando specificato sull'host.
 Valore restituito: risultato *Text* del comando oppure 1 - con `mode` impostato su *nowait* (indipendentemente dal risultato del comando).
 Vedi piattaforme supportate.

Parametri:

- **command** - comando da eseguire;
- **mode** - valori possibili:
  - *wait* - attende la fine dell'esecuzione (predefinito);
  - *nowait* - non attende.

Commenti:

- Questo item è disabilitato per impostazione predefinita. Scopri come abilitarli.
- Il valore restituito dell'item è l'output standard insieme all'errore standard prodotto dal comando. Il controllo del codice di uscita non viene eseguito.
- Per essere elaborato correttamente, il valore restituito del comando deve essere di tipo dati `text`. È consentito anche un risultato vuoto.
- Il valore restituito è limitato a 16MB (inclusi gli spazi finali che vengono troncati); si applicano anche i limiti del database.
- Vedi anche: Esecuzione dei comandi.

Esempio:

```
system.run[ls -l /] #restituisce un elenco dettagliato dei file della directory root
```

system.stat[resource,<type>]

 Le statistiche di sistema.
 Valore restituito: *Integer* o *float*.
 Piattaforme supportate: AIX.

Parametri:

- **ent** - il numero di unità di processore che questa partizione ha diritto a ricevere (float).
- **kthr,<type>** - informazioni sugli stati dei thread del kernel:
  - *r* - numero medio di thread del kernel eseguibili (float);
  - *b* - numero medio di thread del kernel inseriti nella coda di attesa del Virtual Memory Manager (float).
- **memory,<type>** - informazioni sull'utilizzo della memoria virtuale e reale:
  - *avm* - pagine virtuali attive (integer);
  - *fre* - dimensione della lista libera (integer).
- **page,<type>** - informazioni sui page fault e sull'attività di paging:
  - *fi* - page-in da file al secondo (float);
  - *fo* - page-out su file al secondo (float);
  - *pi* - pagine caricate dallo spazio di paging (float);
  - *po* - pagine scaricate nello spazio di paging (float);
  - *fr* - pagine liberate (sostituzione di pagine) (float);
  - *sr* - pagine analizzate dall'algoritmo di sostituzione delle pagine (float).
- **faults,<type>** - tasso di trap e interrupt:
  - *in* - interrupt dei dispositivi (float);
  - *sy* - chiamate di sistema (float);
  - *cs* - context switch dei thread del kernel (float).
- **cpu,<type>** - suddivisione percentuale dell'utilizzo del tempo di processore:
  - *us* - tempo utente (float);
  - *sy* - tempo di sistema (float);
  - *id* - tempo di inattività (float);
  - *wa* - tempo di inattività durante il quale il sistema aveva richieste I/O disco/NFS in sospeso (float);
  - *pc* - numero di processori fisici consumati (float);
  - *ec* - percentuale di capacità assegnata consumata (float);
  - *lbusy* - indica la percentuale di utilizzo del/i processore/i logico/i avvenuta durante l'esecuzione a livello utente e di sistema (float);
  - *app* - indica i processori fisici disponibili nel pool condiviso (float).
- **disk,<type>** - statistiche del disco:
  - *bps* - indica la quantità di dati trasferiti (letti o scritti) sull'unità in byte al secondo (integer);
  - *tps* - indica il numero di trasferimenti al secondo inviati al disco/nastro fisico (float).

Commenti:

- Si notino le seguenti limitazioni in questi item:
  - `system.stat[cpu,app]` - supportato solo su AIX LPAR di tipo "Shared";
  - `system.stat[cpu,ec]` - supportato su AIX LPAR di tipo "Shared" e "Dedicated" ("Dedicated" restituisce sempre 100 (percentuale));
  - `system.stat[cpu,lbusy]` - supportato solo su AIX LPAR di tipo "Shared";
  - `system.stat[cpu,pc]` - supportato su AIX LPAR di tipo "Shared" e "Dedicated";
  - `system.stat[ent]` - supportato su AIX LPAR di tipo "Shared" e "Dedicated".

system.sw.arch

 Le informazioni sull'architettura software.
 Valore restituito: *String*.
 Vedere piattaforme supportate.

Le informazioni vengono ottenute dalla funzione `uname()`.

Esempio:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Informazioni sul sistema operativo.
 Valore restituito: *String*.
 Piattaforme supportate: Linux, Windows.

Parametro:

- **info** - valori possibili: *full* (predefinito), *short* oppure *name*

Le informazioni di info vengono acquisite da (si noti che non tutti i file e le opzioni sono presenti in tutte le distribuzioni):

- `/proc/version` (*full*) su Linux;
- `/proc/version_signature` (*short*) su Linux;
- il parametro PRETTY_NAME da `/etc/os-release` sui sistemi Linux che lo supportano oppure `/etc/issue.net` (*name*);
- la chiave di registro `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` su Windows.

Esempi:

```
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

 Informazioni dettagliate sul sistema operativo (versione, tipo, nome della distribuzione, versione minore e maggiore, ecc.).
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, Windows.

system.sw.packages[<regexp>,<manager>,<format>]

 Elenco dei pacchetti installati.
 Valore restituito: *Text*.
 Piattaforme supportate: Linux.

Parametri:

- **regexp** - *all* (predefinito) oppure una espressione regolare;
- **manager** - *all* (predefinito) oppure un gestore di pacchetti;
- **format** - *full* (predefinito) oppure *short*.

Commenti:

- Elenca in ordine alfabetico i pacchetti installati il cui nome corrisponde alla espressione regolare specificata (*all* li elenca tutti).
- Gestori di pacchetti supportati (comando eseguito):
  - pkg (`dpkg --get-selections`);
  - pkgtool (`ls /var/log/packages`);
  - rpm (`rpm -qa`);
  - pacman (`pacman -Q`);
  - portage.
- Se `format` è specificato come *full*, i pacchetti sono raggruppati per gestore di pacchetti (ogni gestore su una riga separata che inizia con il proprio nome tra parentesi quadre).
- Se `format` è specificato come *short*, i pacchetti non sono raggruppati e sono elencati su una sola riga.

Esempio:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Un elenco dettagliato dei pacchetti installati.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux.

Parametri:

- **regexp** - *all* (predefinito) oppure un'espressione regolare;
- **manager** - *all* (predefinito) oppure un gestore di pacchetti (valori possibili: *rpm*, *dpkg*, *pkgtool*, *pacman* o *portage*).

Commenti:

- Restituisce JSON non formattato con i pacchetti installati il cui nome corrisponde all'espressione regolare specificata.
- L'output è un array di oggetti, ciascuno contenente le seguenti chiavi: name, manager, version, size, architecture, buildtime e installtime (vedi maggiori dettagli).

system.swap.in[<device>,<type>]

 Le statistiche di swap-in (dal dispositivo alla memoria).
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - specifica il dispositivo usato per lo swapping (solo Linux) oppure *all* (predefinito);
- **type** - valori possibili:
  - *count* (numero di swap-in, predefinito sulle piattaforme non Linux);
  - *sectors* (settori scambiati in);
  - *pages* (pagine scambiate in, predefinito su Linux).

Commenti:

- La fonte di queste informazioni è:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Tieni presente che *pages* funzionerà solo se non è stato specificato device.
- Il parametro di tipo *sectors* è supportato solo su Linux.

Esempio:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Le statistiche di swap-out (dalla memoria al dispositivo).
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - specifica il dispositivo usato per lo swapping (solo Linux) oppure *all* (predefinito);
- **type** - valori possibili:
  - *count* (numero di swapout, predefinito sulle piattaforme non Linux);
  - *sectors* (settori scritti in swap);
  - *pages* (pagine scritte in swap, predefinito su Linux).

Commenti:

- La fonte di queste informazioni è:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Nota che *pages* funzionerà solo se non è stato specificato device.
- Il parametro di tipo *sectors* è supportato solo su Linux.

Esempio:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 La dimensione dello spazio di swap in byte o in percentuale sul totale.
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parametri:

- **device** - specifica il dispositivo usato per lo swap (solo FreeBSD) oppure *all* (predefinito);
- **type** - valori possibili:
  - *free* (spazio di swap libero, predefinito);
  - *pfree* (spazio di swap libero, in percentuale);
  - *pused* (spazio di swap usato, in percentuale);
  - *total* (spazio di swap totale);
  - *used* (spazio di swap usato).

Commenti:

- Nota che *pfree*, *pused* non sono supportati su Windows se la dimensione dello swap è 0.
- Se device non è specificato, Zabbix agent prenderà in considerazione solo i dispositivi di swap (file), la memoria fisica verrà ignorata. Ad esempio, sui sistemi Solaris il comando `swap -s` include una parte di memoria fisica e dispositivi di swap (a differenza di `swap -l`).

Esempio:

```
system.swap.size[,pfree] → percentuale di spazio di swap libero
```

system.uname

 Identificazione del sistema.
 Valore restituito: *Stringa*.
 Vedere piattaforme supportate.

Commenti:

- Su UNIX il valore di questo item viene ottenuto con la chiamata di sistema uname().
- Su Windows l'item restituisce l'architettura del sistema operativo, mentre su UNIX restituisce l'architettura della CPU.

Esempi:

```
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

 Il tempo di attività del sistema in secondi.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Nella configurazione degli item, utilizzare le unità **s** o **uptime** per ottenere valori leggibili.

system.users.num

 Il numero di utenti connessi.
 Valore restituito: *Integer*.
 Vedere le piattaforme supportate.

Il comando **who** viene utilizzato lato agent per ottenere il valore.

vfs.dev.discovery

 L'elenco dei dispositivi a blocchi e del loro tipo. Utilizzato per il low-level discovery.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux.

vfs.dev.get[<mode>,<regexp>]

 Elenco dei dispositivi a blocchi con informazioni dettagliate. Usato per la discovery a basso livello.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, AIX.

Parametri:

- **mode** - valori possibili:
  - *disks* (predefinito) - restituisce l'elenco dei dischi;
  - *disk_stats* - restituisce le statistiche per ciascun disco;
  - *devices* (solo Linux) - restituisce l'elenco dei dispositivi a blocchi (dischi e partizioni);
  - *device_stats* (solo Linux) - restituisce le statistiche per ciascun dispositivo (dischi e partizioni).
- **regexp** - una espressione regolare usata per filtrare i dispositivi a blocchi in base al loro nome.

vfs.dev.read[<device>,<type>,<mode>]

 Le statistiche di lettura del disco.
 Valore restituito: - *Integer* - con `type` in *sectors*, *operations*, *bytes*; - *Float* - con `type` in *sps*, *ops*, *bps*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parametri:

- **device** - dispositivo disco (il valore predefinito è *all* [3]);
- **type** - valori possibili: *sectors*, *operations*, *bytes*, *sps*, *ops* o *bps* (*sps*, *ops*, *bps* indicano rispettivamente: settori, operazioni, byte al secondo);
- **mode** - valori possibili: *avg1* (media su un minuto, predefinito), *avg5* o *avg15*. Questo parametro è supportato solo con `type` impostato su: sps, ops, bps.

Commenti:

- Se si utilizza un intervallo di aggiornamento di tre ore o più[2], questo item restituirà sempre 0.
- I parametri di tipo *sectors* e *sps* sono supportati solo su Linux.
- Il parametro di tipo *ops* è supportato solo su Linux e FreeBSD.
- Il parametro di tipo *bps* è supportato solo su FreeBSD.
- Il parametro di tipo *bytes* è supportato solo su FreeBSD, Solaris, AIX, OpenBSD.
- Il parametro `mode` è supportato solo su Linux, FreeBSD.
- È possibile usare nomi relativi dei dispositivi (ad esempio, `sda`) così come un prefisso opzionale `/dev/` (ad esempio, `/dev/sda`).
- Sono supportati i volumi logici LVM.
- I valori predefiniti del parametro 'type' per i diversi sistemi operativi:
  - *AIX* - operations;

    - *FreeBSD* - bps;
    - *Linux* - sps;
    - *OpenBSD* - operations;
    - *Solaris* - bytes.
- *sps*, *ops* e *bps* sulle piattaforme supportate sono limitati a 1024 dispositivi (1023 singoli e uno per *all*).

Esempio:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 Le statistiche di scrittura del disco.
 Valore restituito: *Integer* - con `type` in *sectors*, *operations*, *bytes*; *Float* - con `type` in *sps*, *ops*, *bps*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parametri:

- **device** - dispositivo disco (il valore predefinito è *all* [3]);
- **type** - valori possibili: *sectors*, *operations*, *bytes*, *sps*, *ops* o *bps* (*sps*, *ops*, *bps* stanno per: rispettivamente settori, operazioni, byte al secondo);
- **mode** - valori possibili: *avg1* (media su un minuto, predefinito), *avg5* o *avg15*. Questo parametro è supportato solo con `type` impostato su: sps, ops, bps.

Commenti:

- Se si utilizza un intervallo di aggiornamento di tre ore o più[2], questo item restituirà sempre 0.
- I parametri di tipo *sectors* e *sps* sono supportati solo su Linux.
- Il parametro di tipo *ops* è supportato solo su Linux e FreeBSD.
- Il parametro di tipo *bps* è supportato solo su FreeBSD.
- Il parametro di tipo *bytes* è supportato solo su FreeBSD, Solaris, AIX, OpenBSD.
- Il parametro `mode` è supportato solo su Linux, FreeBSD.
- È possibile usare nomi relativi dei dispositivi (ad esempio, `sda`) così come un prefisso opzionale `/dev/` (ad esempio, `/dev/sda`).
- Sono supportati i volumi logici LVM.
- I valori predefiniti del parametro `type` per i diversi sistemi operativi:
    - *AIX* - operations
    - *FreeBSD* - bps
    - *Linux* - sps
    - *OpenBSD* - operations
    - *Solaris* - bytes
- *sps*, *ops* e *bps* sulle piattaforme supportate sono limitati a 1024 dispositivi (1023 singoli e uno per *all*).

Esempio:

```
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>]

 Il numero di voci della directory.
 Valore restituito: *Integer*.
 Vedere le piattaforme supportate.

Parametri:

- **dir** - il percorso assoluto della directory;
- **regex incl** - un'espressione regolare che descrive il modello del nome dell'entità (file, directory, collegamento simbolico) da includere; include tutto se vuoto (valore predefinito);
- **regex excl** - un'espressione regolare che descrive il modello del nome dell'entità (file, directory, collegamento simbolico) da escludere; non esclude nulla se vuoto (valore predefinito);
- **types incl** - i tipi di voci della directory da contare, valori possibili:
    - *file* - file normale;
    - *dir* - sottodirectory;
    - *sym* - collegamento simbolico;
    - *sock* - socket;
    - *bdev* - dispositivo a blocchi;
    - *cdev* - dispositivo a caratteri;
    - *fifo* - FIFO;
    - *dev* - sinonimo di "bdev,cdev";
    - *all* - tutti i tipi (predefinito), ovvero "file,dir,sym,sock,bdev,cdev,fifo". Più tipi devono essere separati da virgole e racchiusi tra virgolette.

- **types excl** - i tipi di voci della directory (vedere `types incl`) da NON contare. Se un tipo di voce è presente sia in `types incl` sia in `types excl`, le voci della directory di questo tipo NON vengono contate.
- **max depth** - la profondità massima delle sottodirectory da attraversare:
  - **-1** (predefinito) - illimitata;
  - **0** - non entra nelle sottodirectory.
- **min size** - la dimensione minima (in byte) del file da contare. I file più piccoli non vengono contati. È possibile utilizzare i suffissi di memoria.
- **max size** - la dimensione massima (in byte) del file da contare. I file più grandi non vengono contati. È possibile utilizzare i suffissi di memoria.
- **min age** - l'età minima (in secondi) della voce della directory da contare. Le voci più recenti non vengono contate. È possibile utilizzare i suffissi temporali.
- **max age** - l'età massima (in secondi) della voce della directory da contare. Le voci con un'età pari o superiore non vengono contate (in base all'ora di modifica). È possibile utilizzare i suffissi temporali.
- **regex excl dir** - un'espressione regolare che descrive il modello del nome della directory da escludere. Tutto il contenuto della directory verrà escluso (a differenza di `regex_excl`).

Commenti:

- In Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Le variabili d'ambiente, ad esempio `%APP_HOME%`, `$HOME` e `%TEMP%`, non sono supportate.
- Le pseudo-directory "." e ".." non vengono mai conteggiate.
- I collegamenti simbolici non vengono mai seguiti durante l'attraversamento delle directory.
- Sia `regex incl` sia `regex excl` vengono applicate a file e directory durante il calcolo del numero di voci, ma vengono ignorate nella selezione delle sottodirectory da attraversare (se `regex incl` è `(?i)^.+\.zip$` e `max depth` non è impostato, verranno attraversate tutte le sottodirectory, ma verranno contati solo i file di tipo zip).
- Il tempo di esecuzione è limitato dal valore di timeout nella configurazione dell'agent (3 secondi). Poiché l'attraversamento di directory di grandi dimensioni può richiedere più tempo, non verranno restituiti dati e l'item diventerà non supportato. Non verrà restituito un conteggio parziale.
- Durante il filtraggio per dimensione, solo i file normali hanno dimensioni significative. In Linux e BSD, anche le directory hanno dimensioni diverse da zero (in genere alcuni KB). I dispositivi hanno dimensione zero; ad esempio, la dimensione di `/dev/sda1` non riflette quella della partizione corrispondente. Pertanto, quando si utilizzano `<min_size>` e `<max_size>`, è consigliabile specificare `<types_incl>` come "*file*" per evitare risultati imprevisti.

Esempi:

```
vfs.dir.count[/dev] #monitora il numero di dispositivi in /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #monitora il numero di file in una directory tempora
```

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>]

 L'elenco delle voci della directory.
 Valore restituito: *stringa JSON*.
 Vedere piattaforme supportate.

Parametri:

- **dir** - il percorso assoluto della directory;
- **regex incl** - un'espressione regolare che descrive il modello del nome dell'entità (file, directory, collegamento simbolico) da includere; include tutto se vuoto (valore predefinito);
- **regex excl** - un'espressione regolare che descrive il modello del nome dell'entità (file, directory, collegamento simbolico) da escludere; non esclude nulla se vuoto (valore predefinito);
- **types incl** - i tipi di voci della directory da elencare, valori possibili:
  - *file* - file normale;
  - *dir* - sottodirectory;
  - *sym* - collegamento simbolico;
  - *sock* - socket;
  - *bdev* - dispositivo a blocchi;
  - *cdev* - dispositivo a caratteri;
  - *fifo* - FIFO;
  - *dev* - sinonimo di "bdev,cdev";
  - *all* - tutti i tipi (predefinito), ovvero "file,dir,sym,sock,bdev,cdev,fifo". Più tipi devono essere separati da virgole e racchiusi tra virgolette.
- **types excl** - i tipi di voci della directory (vedere `types incl`) da NON elencare. Se un tipo di voce è presente sia in `types incl` sia in `types excl`, le voci della directory di questo tipo NON vengono elencate.
- **max depth** - la profondità massima delle sottodirectory da attraversare:
  - **-1** (predefinito) - illimitata;
  - **0** - non scende nelle sottodirectory.

- **min size** - la dimensione minima (in byte) del file da elencare. I file più piccoli non vengono elencati. È possibile utilizzare i suffissi di memoria.
- **max size** - la dimensione massima (in byte) del file da elencare. I file più grandi non vengono elencati. È possibile utilizzare i suffissi di memoria.
- **min age** - l'età minima (in secondi) della voce della directory da elencare. Le voci più recenti non vengono elencate. È possibile utilizzare i suffissi temporali.
- **max age** - l'età massima (in secondi) della voce della directory da elencare. Le voci con questa età o più vecchie non vengono elencate (in base all'ora di modifica). È possibile utilizzare i suffissi temporali.
- **regex excl dir** - un'espressione regolare che descrive il modello del nome della directory da escludere. Tutto il contenuto della directory verrà escluso (a differenza di `regex excl`).

Commenti:

- In Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Le variabili d'ambiente, ad esempio `%APP_HOME%`, `$HOME` e `%TEMP%`, non sono supportate.
- Le pseudo-directory "." e ".." non vengono mai elencate.
- I collegamenti simbolici non vengono mai seguiti durante l'attraversamento delle directory.
- Sia `regex incl` sia `regex excl` vengono applicate a file e directory durante la generazione dell'elenco delle voci, ma vengono ignorate nella selezione delle sottodirectory da attraversare (se `regex incl` è `(?i)^.+\.zip$` e `max depth` non è impostato, verranno attraversate tutte le sottodirectory, ma verranno conteggiati solo i file di tipo zip).
- Il tempo di esecuzione è limitato dal valore di timeout nella configurazione dell'agent. Poiché l'attraversamento di directory di grandi dimensioni può richiedere più tempo, non verranno restituiti dati e l'item diventerà non supportato. Non verrà restituito un elenco parziale.
- Quando si filtra per dimensione, solo i file normali hanno dimensioni significative. In Linux e BSD, anche le directory hanno dimensioni diverse da zero (in genere alcuni KB). I dispositivi hanno dimensione zero; ad esempio, la dimensione di `/dev/sda1` non riflette quella della partizione corrispondente. Pertanto, quando si utilizzano `min size` e `max size`, è consigliabile specificare `types incl` come "*file*" per evitare risultati imprevisti.

Esempi:

```
vfs.dir.get[/dev] #recupera l'elenco dei dispositivi in /dev (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #recupera l'elenco dei file in una directory temporane
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 La dimensione della directory (in byte).
 Valore restituito: *Integer*.
 Piattaforme supportate: Linux, Windows. L'item può funzionare anche su altre piattaforme simili a UNIX.

Parametri:

- **dir** - il percorso assoluto della directory;
- **regex incl** - una espressione regolare che descrive il pattern del nome dell'entità (file, directory, collegamento simbolico) da includere; include tutto se vuoto (valore predefinito);
- **regex excl** - una espressione regolare che descrive il pattern del nome dell'entità (file, directory, collegamento simbolico) da escludere; non esclude nulla se vuoto (valore predefinito);
- **mode** - valori possibili:
  - *apparent* (predefinito) - ottiene le dimensioni apparenti dei file invece dell'utilizzo su disco (si comporta come `du -sb dir`);
  - *disk* - ottiene l'utilizzo su disco (si comporta come `du -s -B1 dir`). A differenza del comando du, l'item vfs.dir.size tiene conto dei file nascosti quando calcola la dimensione della directory (si comporta come `du -sb .[^.]* *` all'interno di dir).
- **max depth** - la profondità massima delle sottodirectory da attraversare:
  - **-1** (predefinito) - illimitata;
  - **0** - nessuna discesa nelle sottodirectory.
- **regex excl dir** - una espressione regolare che descrive il pattern del nome della directory da escludere. Tutto il contenuto della directory verrà escluso (a differenza di `regex excl`)

Commenti:

- Su Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.
- Vengono calcolate solo le directory con almeno il permesso di lettura per l'utente *zabbix*. Per le directory con solo il permesso di lettura, viene calcolata la dimensione della directory stessa. Le directory con permessi di lettura ed esecuzione vengono calcolate includendo il contenuto.
- Con directory di grandi dimensioni o unità lente, questo item può andare in timeout a causa dell'impostazione Timeout nei file di configurazione di agent e server/proxy. Aumentare i valori di timeout secondo necessità.
- Il limite della dimensione dei file dipende dal supporto per file di grandi dimensioni.

Esempi:

```
vfs.dir.size[/tmp,log] #calcola la dimensione di tutti i file in /tmp che contengono 'log' nei loro nomi
vfs.dir.size[/tmp,log,^.+\.old$] #calcola la dimensione di tutti i file in /tmp che contengono 'log' nei l
```

vfs.file.cksum[file,<mode>]

 Il checksum del file, calcolato con l'algoritmo UNIX cksum.
 Valore restituito: *Integer* - con `mode` impostato su *crc32*, *String* - con `mode` impostato su *md5*, *sha256*.
 Vedere le piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **mode** - *crc32* (predefinito), *md5* o *sha256*.

Commenti:

- Su Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempio:

```
vfs.file.cksum[/etc/passwd]
```

Esempi di valori restituiti (rispettivamente crc32/md5/sha256):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Recupera il contenuto di un file[7].
 Valore restituito: *Text*.
 Vedi piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **encoding** - l'identificatore della code page identifier.

Commenti:

- Su Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.
- Il valore restituito è limitato a 16 MB (inclusi gli spazi finali, che vengono troncati); si applicano anche i limiti del database.
- Viene restituita una stringa vuota se il file è vuoto o contiene solo caratteri LF/CR.
- Il byte order mark (BOM) è escluso dall'output.

Esempio:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Verifica se il file esiste.
 Valore restituito: 0 - non trovato; 1 - esiste un file del tipo specificato.
 Vedi piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **types incl** - l'elenco dei tipi di file da includere, valori possibili:
  - *file* (file regolare, predefinito (se `types_excl` non è impostato));
  - *dir* (directory);
  - *sym* (collegamento simbolico);
  - *sock* (socket);
  - *bdev* (dispositivo a blocchi);
  - *cdev* (dispositivo a caratteri);
  - *fifo* (FIFO);
  - *dev* (sinonimo di "bdev,cdev");
  - *all* (tutti i tipi menzionati, predefinito se `types_excl` è impostato).
- **types excl** - l'elenco dei tipi di file da escludere, vedere `types_incl` per i valori possibili (per impostazione predefinita non viene escluso alcun tipo)

Commenti:

- Su Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.
- Più tipi devono essere separati da una virgola e l'intero insieme racchiuso tra virgolette "".
- Se lo stesso tipo è presente sia in <types_incl> sia in <types_excl>, i file di questo tipo vengono esclusi.

- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempi:

```
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]

 Restituisce informazioni su un file.
 Valore restituito: *stringa JSON*.
 Vedi piattaforme supportate.

Parametro:

- **file** - il percorso completo del file

Commenti:

- Su Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Tipi di file supportati sui sistemi simili a UNIX: file regolare, directory, collegamento simbolico, socket, dispositivo a blocchi, dispositivo a caratteri, FIFO.
- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempio:

```
vfs.file.get[/etc/passwd] #return a JSON with information about the /etc/passwd file (type, user, permissi
```

vfs.file.md5sum[file]

 Il checksum MD5 del file.
 Valore restituito: stringa di caratteri (hash MD5 del file).
 Vedere le piattaforme supportate.

Parametro:

- **file** - il percorso completo del file

Commenti:

- Su Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempio:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Esempio di valore restituito:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Recupera il proprietario di un file.
 Valore restituito: *String*.
 Vedi piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **ownertype** - *user* (predefinito) oppure *group* (solo Unix);
- **resulttype** - *name* (predefinito) oppure *id*; per id - restituisce uid/gid su Unix, SID su Windows.

Commenti:

- Su Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.
- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempi:

```
vfs.file.owner[/tmp/zabbix_server.log] #restituisce il proprietario del file /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #restituisce l'ID del proprietario del file /tmp/zabbix_server.
```

vfs.file.permissions[file]

 Restituisce una stringa di 4 cifre contenente il numero ottale con i permessi UNIX.
 Valore restituito: *String*.
 Piattaforme supportate: Linux. L'item può funzionare anche su altre piattaforme simili a UNIX.

Parametri:

- **file** - il percorso completo del file

Il limite della dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempio:

```
vfs.file.permissions[/etc/passwd] #restituisce i permessi di /etc/passwd, ad esempio, '0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 Recupera una stringa nel file[7].
 Valore restituito: la riga contenente la stringa corrispondente, oppure quanto specificato dal parametro opzionale `output`.
 Vedi piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **regexp** - una espressione regolare che descrive il pattern richiesto;
- **encoding** - l'identificatore della code page;
- **start line** - il numero della prima riga da cercare (per impostazione predefinita, la prima riga del file);
- **end line** - il numero dell'ultima riga da cercare (per impostazione predefinita, l'ultima riga del file);
- **output** - un template di formattazione dell'output opzionale. La sequenza di escape **\0** viene sostituita con la parte di testo corrispondente (dal primo carattere in cui inizia la corrispondenza fino al carattere in cui termina), mentre una sequenza di escape **\N** (dove N=1...9) viene sostituita con l'N-esimo gruppo corrispondente (o con una stringa vuota se N supera il numero di gruppi catturati).

Commenti:

- Su Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.
- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.
- Viene restituita solo la prima riga corrispondente.
- Viene restituita una stringa vuota se nessuna riga corrisponde all'espressione.
- Il byte order mark (BOM) è escluso dall'output.
- L'estrazione del contenuto tramite il parametro `output` avviene sull'agent.

Esempi:

```
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] → ottenimento dell'ID dell'utente *zabbix*
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Trova una stringa nel file[7].
 Valori restituiti: 0 - corrispondenza non trovata; 1 - trovata.
 Vedi piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **regexp** - una espressione regolare che descrive il pattern richiesto;
- **encoding** - l'identificatore della code page identifier;
- **start line** - il numero della prima riga da cercare (per impostazione predefinita, la prima riga del file);
- **end line** - il numero dell'ultima riga da cercare (per impostazione predefinita, l'ultima riga del file).

Commenti:

- Su Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Il limite della dimensione del file dipende dal supporto per file di grandi dimensioni.
- Il byte order mark (BOM) viene ignorato.

Esempio:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 La dimensione del file (in byte).
 Valore restituito: *Integer*.
 Vedi piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **mode** - valori possibili:
  - *bytes* (predefinito);
  - *lines* (anche le righe vuote vengono conteggiate).

Commenti:

- Su Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.

- Il file deve avere i permessi di lettura per l'utente *zabbix*.
- Il limite della dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempio:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Informazioni sull'orario del file.
 Valore restituito: *Integer* (timestamp Unix).
 Vedere le piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **mode** - valori possibili:
  - *modify* (predefinito) - l'ultima volta in cui è stato modificato il contenuto del file (predefinito);
  - *access* - l'ultima volta in cui il file è stato letto;
  - *change* - l'ultima volta in cui sono state modificate le proprietà del file.

Commenti:

- Su Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Il limite di dimensione del file dipende dal supporto per file di grandi dimensioni.

Esempio:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 L'elenco dei filesystem montati con il relativo tipo e le opzioni di montaggio. Utilizzato per la low-level discovery.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get[<mode>,<mountpoint>]

 L'elenco dei filesystem montati con il relativo tipo e le opzioni di montaggio e, nella modalità *full*, lo spazio su disco disponibile e le statistiche sugli inode. Può essere utilizzato per il low-level discovery.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parametri:

- **mode** - valori possibili:
  - *full* (predefinita) - proprietà del filesystem con lo spazio su disco e le statistiche sugli inode;
  - *short* - solo le proprietà del filesystem, senza lo spazio su disco e le statistiche sugli inode;
- **mountpoint** - il punto di montaggio per il quale restituire i dati (ad esempio, `/var` o `C:`). Se non specificato, vengono restituiti tutti i filesystem montati.

Commenti:

- Nella modalità *short* vengono restituiti solo il nome, il tipo e le opzioni di montaggio del filesystem (su Windows: nome, tipo, etichetta e tipo di unità). I filesystem non vengono interrogati per ottenere le statistiche; sui sistemi Unix-like viene letto solo l'elenco dei filesystem montati (ad esempio, `/proc/mounts` su Linux), quindi il controllo non attiva l'automount e non è influenzato da filesystem di rete che non rispondono. Questo rende la modalità *short* adatta al low-level discovery.
- Il valore **mountpoint** deve corrispondere esattamente alla proprietà `fsname`. Su Windows, il confronto non distingue tra maiuscole e minuscole; specificare il punto di montaggio senza la barra rovesciata finale (ad esempio, `C:`).
- Se **mountpoint** è specificato, i dati vengono restituiti solo per quel punto di montaggio. Se nessun filesystem montato corrisponde, viene restituito un array vuoto (`[]`).
- L'output può contenere più voci con lo stesso `fsname` se più di un filesystem è montato sulla stessa directory.
- Nella modalità *full*, lo spazio su disco disponibile e le statistiche sugli inode vengono restituite solo se l'utente del Zabbix agent dispone dell'autorizzazione di attraversamento su ogni directory nel percorso del punto di montaggio; in caso contrario, il filesystem viene omesso dall'output. Questo non si applica alla modalità *short*, che può quindi elencare filesystem assenti dall'output della modalità *full*.
- Le proprietà `fslabel` e `fsdrivetype` vengono restituite solo su Windows; le proprietà `inodes` e `options` non vengono restituite su Windows.
- Vengono segnalati anche i filesystem con un numero di inode pari a zero, come può accadere per i filesystem con inode dinamici (ad esempio, btrfs).
- Vedere anche: Discovery dei filesystem montati.

Esempi:

```
vfs.fs.get #tutti i filesystem montati con statistiche; equivalente a vfs.fs.get[full]
vfs.fs.get[short] #tutti i filesystem montati senza statistiche
```

```
vfs.fs.get[full,/var] #statistiche solo per il punto di montaggio /var
vfs.fs.get[short,C:] #solo proprietà del volume C: (Windows)
```

vfs.fs.inode[fs,<mode>]

 Il numero o la percentuale di inode.
 Valore restituito: *Integer* - per il numero; *Float* - per la percentuale.
 Vedere piattaforme supportate.

Parametri:

- **fs** - il filesystem;
- **mode** - valori possibili: *total* (predefinito), *free*, *used*, *pfree* (liberi, percentuale) oppure *pused* (utilizzati, percentuale).

Se il conteggio degli inode è uguale a zero, come può accadere per i file system con inode dinamici (ad esempio btrfs), i valori pfree/pused verranno riportati rispettivamente come ”100” e ”0”.

Esempio:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 Lo spazio su disco in byte o in percentuale del totale.
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.
 Vedere le piattaforme supportate.

Parametri:

- **fs** - il filesystem;
- **mode** - valori possibili: *total* (predefinito), *free*, *used*, *pfree* (libero, percentuale) oppure *pused* (usato, percentuale).

Commenti:

- Se il filesystem non è montato, restituisce la dimensione di un filesystem locale in cui si trova il punto di mount.
- Lo spazio riservato di un file system viene preso in considerazione e non incluso quando si utilizza la modalità *free*.

Esempio:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 La dimensione della memoria in byte o in percentuale del totale.
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.
 Vedere le piattaforme supportate.

Parametro:

- **mode** - valori possibili: *total* (predefinito), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (utilizzata, percentuale), *available* oppure *pavailable* (disponibile, percentuale).

Commenti:

- Questo item accetta tre categorie di parametri:
  - *total* - quantità totale di memoria;
  - tipi di memoria specifici della piattaforma: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - stime a livello utente della quantità di memoria utilizzata e disponibile: *used*, *pused*, *available*, *pavailable*.
- Il parametro della modalità *active* è supportato solo su FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- I parametri della modalità *anon*, *exec*, *file* sono supportati solo su NetBSD.
- Il parametro della modalità *buffers* è supportato solo su Linux, FreeBSD, OpenBSD, NetBSD.
- Il parametro della modalità *cached* è supportato solo su Linux, FreeBSD, AIX, OpenBSD, NetBSD.
- I parametri della modalità *inactive*, *wired* sono supportati solo su FreeBSD, MacOS X, OpenBSD, NetBSD.
- Il parametro della modalità *pinned* è supportato solo su AIX.
- Il parametro della modalità *shared* è supportato solo su Linux 2.4, FreeBSD, OpenBSD, NetBSD.
- Vedere anche gli ulteriori dettagli per questo item.

Esempio:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>,<redirect_limit>]

 Ottiene il contenuto di una pagina web.
 Valore restituito: sorgente della pagina web come testo (incluse le intestazioni).
 Vedere piattaforme supportate.

Parametri:

- **host** - il nome host o l'URL (nel formato `scheme://host:port/path`, dove solo *host* è obbligatorio). Schemi URL consentiti: *http*, *https*[4]. Se lo schema è omesso, verrà utilizzato *http*. Se viene specificato un URL, `path` e `port` devono essere vuoti. La specifica di nome utente/password per la connessione a server che richiedono l'autenticazione, ad esempio: `http://user:password@www.example.com`, è possibile solo con il supporto cURL [4]. Punycode è supportato nei nomi host.
- **path** - il percorso a un documento HTML (il valore predefinito è /);
- **port** - il numero di porta (il valore predefinito è 80 per HTTP)
- **redirect_limit** - il numero massimo di reindirizzamenti da seguire [4], da 0 a 1000 (il valore predefinito è 0: i reindirizzamenti non vengono seguiti).

Commenti:

- Questo item diventa non supportato se la risorsa specificata in `host` non esiste o non è disponibile, oppure se `redirect_limit` è maggiore di 0 e il numero di reindirizzamenti lo supera.
- `host` può essere un nome host, un nome di dominio, un indirizzo IPv4 o IPv6. Tuttavia, per un indirizzo IPv6, l'agent Zabbix deve essere compilato con il supporto IPv6 abilitato.
- Se i reindirizzamenti vengono seguiti, il valore restituito contiene le intestazioni di ogni risposta di reindirizzamento, seguite dalle intestazioni e dal corpo della risposta finale.

Esempi:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
web.page.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 Il tempo di caricamento di una pagina web completa (in secondi).
 Valore restituito: *Float*.
 Vedere piattaforme supportate.

Parametri:

- **host** - il nome host o l'URL (nel formato `scheme://host:port/path`, dove solo *host* è obbligatorio). Schemi URL consentiti: *http*, *https*[4]. Se lo schema è omesso, verrà utilizzato *http*. Se viene specificato un URL, `path` e `port` devono essere vuoti. La specifica di nome utente/password per la connessione a server che richiedono l'autenticazione, ad esempio: `http://user:password@www.example.com`, è possibile solo con il supporto cURL [4]. Punycode è supportato nei nomi host.
- **path** - il percorso di un documento HTML (il valore predefinito è /);
- **port** - il numero di porta (il valore predefinito è 80 per HTTP)
- **redirect_limit** - il numero massimo di reindirizzamenti da seguire [4], da 0 a 1000 (il valore predefinito è 0: i reindirizzamenti non vengono seguiti).

Commenti:

- Questo item diventa non supportato se la risorsa specificata in `host` non esiste o non è disponibile, oppure se `redirect_limit` è maggiore di 0 e il numero di reindirizzamenti lo supera.
- `host` può essere un nome host, un nome di dominio, un indirizzo IPv4 o IPv6. Tuttavia, per gli indirizzi IPv6, Zabbix agent deve essere compilato con il supporto IPv6 abilitato.
- Se i reindirizzamenti vengono seguiti, il tempo di caricamento include il tempo impiegato per tutti i reindirizzamenti.

Esempi:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Trova una stringa nella pagina web.
 Valore restituito: la stringa corrispondente oppure quella specificata dal parametro facoltativo `output`.
 Vedere piattaforme supportate.

Parametri:

- **host** - il nome host o l'URL (nel formato `scheme://host:port/path`, dove solo *host* è obbligatorio). Schemi URL consentiti: *http*, *https*[4]. Se lo schema è omesso, verrà utilizzato *http*. Se viene specificato un URL, `path` e `port` devono

essere vuoti. La specifica di nome utente/password per la connessione a server che richiedono l'autenticazione, ad esempio `http://user:password@www.example.com`, è possibile solo con il supporto cURL [4]. I nomi host in Punycode sono supportati.
- **path** - il percorso di un documento HTML (il valore predefinito è `/`).
- **port** - il numero di porta (il valore predefinito è 80 per HTTP).
- **regexp** - un'espressione regolare che descrive il modello richiesto.
- **length** - il numero massimo di caratteri da restituire;
- **output** - un template facoltativo per la formattazione dell'output. La sequenza di escape **\0** viene sostituita con la parte di testo corrispondente (dal primo carattere in cui inizia la corrispondenza fino al carattere in cui termina), mentre una sequenza di escape **\N** (dove N=1...9) viene sostituita con il gruppo corrispondente individuato (oppure con una stringa vuota se N supera il numero di gruppi acquisiti).
- **redirect_limit** - il numero massimo di reindirizzamenti da seguire [4], da 0 a 1000 (il valore predefinito è 0: i reindirizzamenti non vengono seguiti).

Note:

- Questo item diventa non supportato se la risorsa specificata in `host` non esiste o non è disponibile, oppure se `redirect_limit` è maggiore di 0 e il numero di reindirizzamenti lo supera.
- `host` può essere un nome host, un nome di dominio, un indirizzo IPv4 o IPv6. Tuttavia, per un indirizzo IPv6, l'agent Zabbix deve essere compilato con il supporto IPv6 abilitato.
- L'estrazione del contenuto tramite il parametro `output` viene eseguita sull'agent.
- Se vengono seguiti reindirizzamenti, vengono analizzati anche gli header di ogni risposta di reindirizzamento.

Esempi:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

agent.hostmetadata

 I metadati dell'host dell'agent.
 Valore restituito: *String*.
 Vedi piattaforme supportate.

Restituisce il valore dei parametri `HostMetadata` o `HostMetadataItem`, oppure una stringa vuota se non ne è definito alcuno.

agent.hostname

 Il nome host dell'agent.
 Valore restituito: *String*.
 Vedi piattaforme supportate.

Restituisce:

- Come controllo passivo - il nome del primo host elencato nel parametro Hostname del file di configurazione dell'agent;
- Come controllo attivo - il nome dell'hostname corrente.

agent.ping

 Controllo della disponibilità dell'agent.
 Valore restituito: Nessuno - non disponibile; 1 - disponibile.
 Vedi piattaforme supportate.

Usa la funzione trigger `nodata()` per verificare l'indisponibilità dell'host.

agent.variant

 La variante dell'agent Zabbix (Zabbix agent o Zabbix agent 2).
 Valore restituito: 1 - Zabbix agent; 2 - Zabbix agent 2.
 Vedere piattaforme supportate.

agent.version

 La versione di Zabbix agent.
 Valore restituito: *String*.
 Vedere piattaforme supportate.

Esempio di valore restituito:

```
6.0.3
```

zabbix.stats[<ip>,<port>]

 Restituisce un insieme di metriche interne di Zabbix server o proxy. Se `ip` e `port` sono specificati, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
 Valore restituito: *stringa JSON*.
 Vedi piattaforme supportate.

Parametri:

- **ip** - l'elenco di indirizzi IP/DNS/maschere di rete di server/proxy da interrogare in remoto (il valore predefinito è `127.0.0.1`);
- **port** - la porta di server/proxy da interrogare in remoto (il valore predefinito è 10051)

Commenti:

- Un insieme selezionato di metriche interne viene restituito da questo item. Per i dettagli, vedi Monitoraggio remoto delle statistiche di Zabbix.
- Nota che la richiesta delle statistiche sarà accettata solo dagli indirizzi elencati nel parametro `StatsAllowedIP` di server/proxy sull'istanza di destinazione.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Restituisce il numero di item monitorati nella coda che risultano in ritardo sul server Zabbix o sul proxy. Se `ip` e `port` sono specificati, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
 Valore restituito: *stringa JSON*.
 Vedi le piattaforme supportate.

Parametri:

- **ip** - l'elenco di IP/DNS/maschere di rete dei server/proxy da interrogare in remoto (il valore predefinito è `127.0.0.1`);
- **port** - la porta del server/proxy da interrogare in remoto (il valore predefinito è `10051`)
- **queue** - costante (da usare così com'è)
- **from** - in ritardo di almeno (il valore predefinito è 6 secondi)
- **to** - in ritardo di al massimo (il valore predefinito è infinito)

Nota che la richiesta stats sarà accettata solo dagli indirizzi elencati nel parametro 'StatsAllowedIP' server/proxy dell'istanza di destinazione.

Note a piè di pagina

[1] Nota specifica per Linux.
Zabbix agent deve disporre dell'accesso in sola lettura al filesystem `/proc`.
Le patch del kernel disponibili su www.grsecurity.org limitano i diritti di accesso degli utenti non privilegiati.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: Zabbix agent terminerà le connessioni ai dispositivi "stale" se i valori degli item non vengono consultati per più di 3 ore.
Ciò può verificarsi se un sistema dispone di dispositivi con percorsi che cambiano dinamicamente o se un dispositivo viene rimosso manualmente.
Si noti inoltre che questi item, se utilizzano un intervallo di aggiornamento pari o superiore a 3 ore, restituiranno sempre '0'.

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: se per il primo parametro viene utilizzato il valore predefinito `all`, la chiave restituirà statistiche riepilogative, inclusi tutti i dispositivi a blocchi come sda, sdb e le relative partizioni (sda1, sda2, sdb3...), nonché i dispositivi multipli (RAID MD) basati su tali dispositivi a blocchi/partizioni e i volumi logici (LVM) basati su tali dispositivi a blocchi/partizioni.
In questi casi, i valori restituiti devono essere considerati solo come valori relativi (dinamici nel tempo), non come valori assoluti.

[4] HTTPS e il seguente reindirizzamento (`redirect_limit` maggiore di 0) sono supportati solo se Zabbix agent è stato compilato con il supporto cURL; in caso contrario, l'item diventa non supportato. Zabbix agent 2 non richiede cURL per nessuna delle due funzionalità.

[5] I valori `bytes` ed `errors` non sono supportati per le interfacce loopback nei sistemi Solaris fino a Solaris 10 6/06 incluso, poiché le statistiche relative a byte, errori e utilizzo non vengono memorizzate e/o riportate dal kernel.
Tuttavia, se si monitora un sistema Solaris tramite net-snmp, i valori potrebbero essere restituiti, poiché net-snmp contiene codice legacy derivato da cmu-snmp, risalente anche al 1997, che, quando non riesce a leggere i valori dei byte dalle statistiche dell'interfaccia, restituisce il contatore dei pacchetti (presente sulle interfacce loopback) moltiplicato per un valore arbitrario pari a 308.
Ciò presuppone che la lunghezza media di un pacchetto sia di 308 ottetti, una stima molto approssimativa, poiché il limite MTU sui sistemi Solaris per le interfacce loopback è di 8892 byte.
Questi valori non devono essere considerati corretti o anche solo approssimativamente accurati.
Sono stime approssimative.
Zabbix agent non effettua alcuna stima, ma net-snmp restituirà un valore per questi campi.

[6] La riga di comando su Solaris, ottenuta da `/proc/pid/psinfo`, è limitata a 80 byte e contiene la riga di comando così come era al momento dell'avvio del processo.

[7] Gli item `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` possono essere utilizzati per recuperare il contenuto dei file.
Se si desidera limitare l'accesso a file specifici contenenti informazioni sensibili, eseguire Zabbix agent con un utente che non disponga delle autorizzazioni per visualizzare tali file.

Utilizzo con le utility da riga di comando

Si noti che, quando si testano o si utilizzano chiavi di item con zabbix_agentd o zabbix_get dalla riga di comando, è necessario considerare anche la sintassi della shell.

Ad esempio, se un determinato parametro della chiave deve essere racchiuso tra virgolette doppie, è necessario eseguire esplicitamente l'escape delle virgolette doppie; in caso contrario, la shell le eliminerà in quanto caratteri speciali e non verranno passate

all'utility Zabbix.

Esempi:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

Impostazioni di codifica

Per assicurarsi che i dati acquisiti non vengano danneggiati, è possibile specificare la codifica corretta per l'elaborazione del controllo (ad esempio, `vfs.file.contents`) nel parametro `encoding`. L'elenco delle codifiche supportate (identificatori delle pagine di codice) è disponibile nella documentazione di libiconv (GNU Project) o nella documentazione di Microsoft Windows SDK per gli "Identificatori delle pagine di codice". Si noti che Microsoft a volte contrassegna alcune pagine di codice come "disponibili solo per le applicazioni gestite", indicando che sono esposte solo tramite il runtime .NET e non sono disponibili attraverso le API native Win32. Zabbix agent implementa una propria logica di conversione della codifica; pertanto, tali pagine di codice sono supportate da Zabbix agent anche quando le funzioni native di Windows non le forniscono.

Se nel parametro `encoding` non viene specificata alcuna codifica, vengono applicate le seguenti strategie di risoluzione:

- Se la codifica non è specificata (o è una stringa vuota), si presume che sia UTF-8 e i dati vengono elaborati "così come sono";
- Analisi del BOM - applicabile agli item `vfs.file.contents`, `vfs.file.regexp`, `vfs.file.regmatch`. Si tenta di determinare la codifica corretta utilizzando il byte order mark (BOM) all'inizio del file. Se il BOM non è presente, viene applicata la risoluzione standard (vedere sopra).

Risoluzione dei problemi relativi agli item dell'agent

In caso di controlli passivi, per evitare che l'item non riceva alcun valore perché la richiesta del server all'agent scade prima, è necessario tenere presente quanto segue:

- Se la versione dell'agent è precedente a quella del server, il valore `Timeout` nella configurazione dell'item (o il timeout globale) potrebbe dover essere superiore al valore `Timeout` nel file di configurazione dell'agent.
- Se la versione dell'agent è più recente di quella del server, il valore `Timeout` nel file di configurazione del server potrebbe dover essere superiore al valore `Timeout` nel file di configurazione dell'agent.

Zabbix agent 2

Zabbix agent 2 supporta tutte le chiavi item supportate per Zabbix agent su Unix e Windows. Questa pagina fornisce i dettagli sulle chiavi item aggiuntive, utilizzabili solo con Zabbix agent 2, raggruppate in base al plugin a cui appartengono.

Le chiavi item sono elencate senza parametri e informazioni aggiuntive. Fare clic sulla chiave item per visualizzare i dettagli completi.

Item key	Description	Plugin
ceph.df.details	L'utilizzo dei dati del cluster e la distribuzione tra i pool.	Ceph (caricabile)
ceph.osd.stats	Statistiche aggregate e per OSD.	
ceph.osd.discovery	L'elenco degli OSD rilevati.	
ceph.osd.dump	Le soglie di utilizzo e gli stati degli OSD.	
ceph.ping	Verifica se è possibile stabilire una connessione a Ceph.	
ceph.pool.discovery	L'elenco dei pool rilevati.	
ceph.status	Lo stato complessivo del cluster.	
docker.container_info	Informazioni di basso livello su un container.	Docker
docker.container_stats	Le statistiche di utilizzo delle risorse del container.	
docker.containers	Restituisce l'elenco dei container.	
docker.containers.discovery	Restituisce l'elenco dei container. Usato per la discovery di basso livello.	
docker.data.usage	Informazioni sull'utilizzo corrente dei dati.	
docker.images	Restituisce l'elenco delle immagini.	
docker.images.discovery	Restituisce l'elenco delle immagini. Usato per la discovery di basso livello.	
docker.info	Le informazioni di sistema.	
docker.ping	Verifica se il daemon Docker è attivo o meno.	
ember.get	Restituisce il risultato del dispositivo richiesto.	Ember+ (caricabile)
memcached.ping	Verifica se una connessione è attiva o meno.	Memcached
memcached.stats	Ottiene l'output del comando STATS.	

Item key	Description	Plugin
mongodb.cfg.discovery	Restituisce un elenco dei server di configurazione rilevati.	MongoDB (caricabile)
mongodb.collection.stats	Restituisce varie statistiche di archiviazione per una determinata collection.	
mongodb.collections.discovery	Restituisce un elenco delle collection rilevate.	
mongodb.collections.usage	Restituisce le statistiche di utilizzo delle collection.	
mongodb.connpool.stats	Restituisce informazioni sulle connessioni in uscita aperte dall'istanza del database corrente verso altri membri del cluster shardato o del replica set.	
mongodb.db.stats	Restituisce le statistiche che riflettono lo stato di un determinato database.	
mongodb.db.discovery	Restituisce un elenco dei database rilevati.	
mongodb.jumbo_chunks.count	Restituisce il numero di jumbo chunk.	
mongodb.oplog.stats	Restituisce lo stato del replica set, usando i dati interrogati dall'oplog.	
mongodb.ping	Verifica se una connessione è attiva o meno.	
mongodb.rs.config	Restituisce la configurazione corrente del replica set.	
mongodb.rs.status	Restituisce lo stato del replica set dal punto di vista del membro su cui viene eseguito il metodo.	
mongodb.server.status	Restituisce lo stato del database.	
mongodb.sh.discovery	Restituisce l'elenco degli shard rilevati presenti nel cluster.	
mongodb.version	Restituisce la versione del server del database.	
mqtt.get	Si sottoscrive a uno o più topic specifici (con caratteri jolly) del broker fornito e attende le pubblicazioni.	MQTT
mssql.availability.group.get	Restituisce i gruppi di disponibilità.	MSSQL (caricabile)
mssql.custom.query	Restituisce il risultato di una query personalizzata.	
mssql.db.get	Restituisce tutti i database MSSQL disponibili.	
mssql.job.status.get	Restituisce lo stato dei job.	
mssql.last.backup.get	Restituisce l'ora dell'ultimo backup per tutti i database.	
mssql.local.db.get	Restituisce i database che partecipano a un gruppo di disponibilità Always On e a una replica (primaria o secondaria) e che si trovano sul server a cui è stata stabilita la connessione.	
mssql.mirroring.get	Restituisce le informazioni sul mirroring.	
mssql.nonlocal.db.get	Restituisce i database che partecipano a un gruppo di disponibilità Always On e a una replica (primaria o secondaria) situata su altri server (il database non è locale all'istanza di SQL Server a cui è stata stabilita la connessione).	
mssql.perfcounter.get	Restituisce i contatori delle prestazioni.	
mssql.ping	Verifica se una connessione è attiva o meno.	
mssql.quorum.get	Restituisce le informazioni sul quorum.	
mssql.quorum.member.get	Restituisce i membri del quorum.	
mssql.replica.get	Restituisce le repliche.	
mssql.version	Restituisce la versione MSSQL.	
mysql.custom.query	Restituisce il risultato di una query personalizzata.	MySQL
mysql.db.discovery	Restituisce l'elenco dei database MySQL.	
mysql.db.size	La dimensione del database in byte.	
mysql.get_status_variables	Valori delle variabili di stato globali.	
mysql.ping	Verifica se una connessione è attiva o meno.	
mysql.replication.discovery	Restituisce l'elenco delle repliche MySQL.	
mysql.replication.get_slave_status	Lo stato della replica.	
mysql.version	La versione MySQL.	
net.dns.get	Esegue una query DNS e restituisce informazioni dettagliate sui record DNS.	Network
nvml.device.count	Il numero di dispositivi GPU.	NVIDIA GPU (caricabile)
nvml.device.decoder.utilization	Utilizzo del decoder del dispositivo GPU.	
nvml.device.ecc.mode	Modalità ECC del dispositivo GPU.	
nvml.device.encoder.stats.get	Statistiche dell'encoder del dispositivo GPU.	
nvml.device.encoder.utilization	Utilizzo dell'encoder del dispositivo GPU.	
nvml.device.energy.consumption	Consumo energetico del dispositivo GPU.	
nvml.device.errors.memory	Informazioni sugli errori di memoria ECC del dispositivo GPU.	
nvml.device.errors.register	Informazioni sugli errori di registro ECC del dispositivo GPU.	
nvml.device.fan.speed.avg	Velocità media della ventola del dispositivo GPU.	
nvml.device.get	GPU rilevate con UUID e nome.	
nvml.device.graphics.frequency	Frequenza di clock grafica del dispositivo GPU.	

Item key	Description	Plugin
nvml.device.memory.bar1.info	Informazioni sulla memoria BAR1 del dispositivo GPU.	
nvml.device.memory.fb.get	Informazioni sulla memoria framebuffer del dispositivo GPU.	
nvml.device.memory.frequency	Frequenza di clock della memoria del dispositivo GPU.	
nvml.device.pci.utilization	Utilizzo PCI del dispositivo GPU.	
nvml.device.performance.state	Stato di prestazione del dispositivo GPU.	
nvml.device.power.limit	Limite di potenza del dispositivo GPU.	
nvml.device.power.usage	Utilizzo di potenza del dispositivo GPU.	
nvml.device.serial	Numero di serie del dispositivo GPU.	
nvml.device.sm.frequency	Frequenza di clock del multiprocessore in streaming del dispositivo GPU.	
nvml.device.temperature	Temperatura del dispositivo GPU.	
nvml.device.utilization	Statistiche di utilizzo del dispositivo GPU.	
nvml.device.video.frequency	Frequenza di clock video del dispositivo GPU.	
nvml.system.driver.version	La versione del driver NVIDIA.	
nvml.version	La versione della libreria NVML.	
oracle.diskgroups.stats	Restituisce le statistiche dei gruppi di dischi Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Restituisce l'elenco dei gruppi di dischi ASM.	
oracle.archive.info	Le statistiche dei log di archivio.	
oracle.cdb.info	Le informazioni sui Container Database (CDB).	
oracle.custom.query	Il risultato di una query personalizzata.	
oracle.datafiles.stats	Restituisce le statistiche dei data file.	
oracle.db.discovery	Restituisce l'elenco dei database.	
oracle.fra.stats	Restituisce le statistiche della Fast Recovery Area (FRA).	
oracle.instance.info	Le statistiche dell'istanza.	
oracle.pdb.info	Le informazioni sui Pluggable Database (PDB).	
oracle.pdb.discovery	Restituisce l'elenco dei PDB.	
oracle.pga.stats	Restituisce le statistiche della Program Global Area (PGA).	
oracle.ping	Verifica se è possibile stabilire una connessione a Oracle.	
oracle.proc.stats	Restituisce le statistiche dei processi.	
oracle.redolog.info	Le informazioni sui file di log dal control file.	
oracle.sga.stats	Restituisce le statistiche della System Global Area (SGA).	
oracle.sessions.stats	Restituisce le statistiche delle sessioni.	
oracle.sys.metrics	Restituisce un insieme di valori delle metriche di sistema.	
oracle.sys.params	Restituisce un insieme di valori dei parametri di sistema.	
oracle.ts.stats	Restituisce le statistiche dei tablespace.	
oracle.ts.discovery	Restituisce un elenco dei tablespace.	
oracle.user.info	Restituisce le informazioni sugli utenti Oracle.	
oracle.version	Restituisce la versione del server del database.	
pgsql.autovacuum.count	Il numero di worker autovacuum.	PostgreSQL (caricabile)
pgsql.archive	Le informazioni sui file archiviati.	
pgsql.bgwriter	Il numero complessivo di checkpoint per il cluster di database, suddiviso per tipo di checkpoint.	
pgsql.cache.hit	Il tasso di hit della cache buffer di PostgreSQL.	
pgsql.connections	Restituisce le connessioni per tipo.	
pgsql.custom.query	Restituisce il risultato di una query personalizzata.	
pgsql.db.age	L'età del FrozenXID più vecchio del database.	
pgsql.db.bloating_tables	Il numero di tabelle gonfiate per database.	
pgsql.db.discovery	L'elenco dei database PostgreSQL.	
pgsql.db.size	La dimensione del database in byte.	
pgsql.dbstat	Raccoglie le statistiche per database.	
pgsql.dbstat.sum	I dati riepilogati per tutti i database in un cluster.	
pgsql.locks	Le informazioni sui lock concessi per database.	
pgsql.oldest.xid	L'età dell'XID più vecchio.	
pgsql.ping	Verifica se una connessione è attiva o meno.	
pgsql.queries	Metriche delle query per tempo di esecuzione.	
pgsql.replication.count	Il numero di server standby.	
pgsql.replication.process	Il ritardo di flush, write e replay per ciascun processo sender.	
pgsql.replication.process.discovery	La discovery del nome del processo di replica.	
pgsql.replication.recovery_role	Lo stato di recovery.	
pgsql.replication.status	Lo stato della replica.	

Item key	Description	Plugin
pgsql.replication_lag.b	Il ritardo di replica in byte.	
pgsql.replication_lag.sec	Il ritardo di replica in secondi.	
pgsql.uptime	L'uptime di PostgreSQL in millisecondi.	
pgsql.version	Restituisce la versione di PostgreSQL.	
pgsql.wal.stat	Le statistiche WAL.	
redis.config	Ottiene i parametri di configurazione di un'istanza Redis che corrispondono al pattern.	Redis
redis.info	Ottiene l'output del comando INFO.	
redis.ping	Verifica se una connessione è attiva o meno.	
redis.slowlog.count	Il numero di voci del log lento da quando Redis è stato avviato.	
smart.attribute.discovery	Restituisce un elenco degli attributi dei dispositivi S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Restituisce un elenco dei dispositivi S.M.A.R.T.	
smart.disk.get	Restituisce tutte le proprietà disponibili dei dispositivi S.M.A.R.T.	
systemd.unit.get	Restituisce tutte le proprietà di un'unità systemd.	Systemd
systemd.unit.info	Informazioni sull'unità systemd.	
systemd.unit.discovery	L'elenco delle unità systemd e i relativi dettagli.	
web.certificate.get	Convalida i certificati e restituisce i dettagli del certificato.	Web certificates

Dettagli della chiave item

I parametri senza parentesi angolari sono obbligatori. I parametri contrassegnati con parentesi angolari **< >** sono facoltativi.

ceph.df.details[connString,<user>,<apikey>]

 L'utilizzo dei dati del cluster e la distribuzione tra i pool.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Statistiche aggregate e per OSD.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 L'elenco degli OSD rilevati. Utilizzato per il rilevamento di basso livello.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 Le soglie di utilizzo e gli stati degli OSD.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

ceph.ping[connString,<user>,<apikey>]

 Verifica se è possibile stabilire una connessione a Ceph.
 Valore restituito: *0* - la connessione è interrotta (se è presente qualsiasi errore, inclusi problemi di AUTH e di configurazione); *1* - la connessione è riuscita.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 L'elenco dei pool rilevati. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

ceph.status[connString,<user>,<apikey>]

 Lo stato complessivo del cluster.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, apikey** - il nome utente e la chiave API per la connessione a Ceph.

docker.container_info[<ID>,<info>]

 Informazioni di basso livello su un container.
 Valore restituito: l'output della chiamata API ContainerInspect serializzato come JSON.

Parametri:

- **ID** - l'ID o il nome del container;
- **info** - la quantità di informazioni restituite. Valori supportati: *short* (predefinito) oppure *full*.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al group 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.container_stats[<ID>]

 Le statistiche di utilizzo delle risorse del container.
 Valore restituito: l'output della chiamata API ContainerStats e la percentuale di utilizzo della CPU serializzati come JSON.

Parametro:

- **ID** - l'ID o il nome del container.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al gruppo 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.containers

 L'elenco dei container.
 Valore restituito: l'output della chiamata API ContainerList serializzato come JSON.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al group 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.containers.discovery[<options>]

 Restituisce l'elenco dei container. Utilizzato per il low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametro:

- **options** - specifica se devono essere rilevati tutti i container o solo quelli in esecuzione. Valori supportati: *true* - restituisce tutti i container; *false* - restituisce solo i container in esecuzione (predefinito).

L'utente di Agent 2 ('zabbix') deve essere aggiunto al group 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.data.usage

 Informazioni sull'utilizzo corrente dei dati.
 Valore restituito: l'output della chiamata API SystemDataUsage serializzato come JSON.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al group 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.images

 Restituisce l'elenco delle immagini.
 Valore restituito: l'output della chiamata API ImageList serializzato come JSON.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al gruppo 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.images.discovery

 Restituisce l'elenco delle immagini. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al gruppo 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.info

 Le informazioni di sistema.
 Valore restituito: l'output della chiamata API SystemInfo serializzato come JSON.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al group 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

docker.ping

 Verifica se il demone Docker è attivo oppure no.
 Valore restituito: *1* - la connessione è attiva; *0* - la connessione è interrotta.

L'utente di Agent 2 ('zabbix') deve essere aggiunto al gruppo 'docker' per disporre di privilegi sufficienti. In caso contrario, il controllo non riuscirà.

ember.get[<uri>,<path>]

 Restituisce il risultato del dispositivo richiesto.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - URI del dispositivo Ember+. Predefinito: 127.0.0.1:9998;
- **path** - percorso OID del dispositivo. Vuoto per impostazione predefinita, restituisce i dati della raccolta radice.

memcached.ping[connString,<user>,<password>]

 Verifica se una connessione è attiva oppure no.
 Valore restituito: *1* - la connessione è attiva; *0* - la connessione è interrotta (se è presente qualsiasi errore, inclusi problemi di AUTH e di configurazione).

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a Memcached.

memcached.stats[connString,<user>,<password>,<type>]

 Ottiene l'output del comando STATS.
 Valore restituito: *JSON* - l'output è serializzato come JSON.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a Memcached;
- **type** - il tipo di statistiche da restituire: *items*, *sizes*, *slabs* oppure *settings* (vuoto per impostazione predefinita, restituisce le statistiche generali).

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 Restituisce una varietà di statistiche di archiviazione per una determinata collection.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB;
- **database** - il nome del database (predefinito: admin);
- **collection** - il nome della collection.

mongodb.cfg.discovery[connString,<user>,<password>]

 Restituisce un elenco dei server di configurazione rilevati.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MongoDB.

mongodb.collections.discovery[connString,<user>,<password>]

 Restituisce un elenco delle collection individuate. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.collections.usage[connString,<user>,<password>]

 Restituisce le statistiche di utilizzo per le collection.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.connpool.stats[connString,<user>,<password>]

 Restituisce informazioni sulle connessioni in uscita aperte dall'istanza di database corrente verso altri membri del cluster shardato o del replica set.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MongoDB.

mongodb.db.stats[connString,<user>,<password>,<database>]

 Restituisce le statistiche che riflettono lo stato di un determinato database.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB;
- **database** - il nome del database (predefinito: admin).

mongodb.db.discovery[connString,<user>,<password>]

 Restituisce un elenco dei database individuati. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MongoDB.

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 Restituisce il numero di chunk jumbo.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.oplog.stats[connString,<user>,<password>]

 Restituisce lo stato del replica set, utilizzando i dati raccolti dall'oplog.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.ping[connString,<user>,<password>]

 Verifica se una connessione è attiva oppure no.
 Valore restituito: *1* - la connessione è attiva; *0* - la connessione è interrotta (se è presente qualsiasi errore, inclusi problemi di AUTH e di configurazione).

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.rs.config[connString,<user>,<password>]

 Restituisce la configurazione corrente del set di repliche.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.rs.status[connString,<user>,<password>]

 Restituisce lo stato del replica set dal punto di vista del membro su cui viene eseguito il metodo.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.server.status[connString,<user>,<password>]

 Restituisce lo stato del database.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mongodb.sh.discovery[connString,<user>,<password>]

 Restituisce l'elenco degli shard rilevati presenti nel cluster.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MongoDB.

mongodb.version[connString,<user>,<password>]

 Restituisce la versione del server di database.
 Valore restituito: *String*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MongoDB.

mqtt.get[<broker url>,topic,<user>,<password>]

 Si sottoscrive a un topic specifico o a più topic (con caratteri jolly) del broker specificato e attende le pubblicazioni.
 Valore restituito: dipende dal contenuto del topic. Se vengono usati caratteri jolly, restituisce il contenuto del topic in formato JSON.

Parametri:

- **broker url** - l'URL del broker MQTT nel formato `protocol://host:port` senza parametri di query (protocolli supportati: `tcp`, `ssl`, `ws`). Se non viene specificato alcun valore, l'agent userà `tcp://localhost:1883`. Se un protocollo o una porta vengono omessi, verranno usati il protocollo predefinito (`tcp`) o la porta predefinita (1883);
- **topic** - il topic MQTT (obbligatorio). Sono supportati i caratteri jolly (+,#);
- **user, password** - le credenziali di autenticazione (se richieste).

Commenti:

- L'item deve essere configurato come active check (tipo di item "Zabbix agent (active)");
- I certificati di crittografia TLS possono essere utilizzati salvandoli in una posizione predefinita (ad esempio, la directory `/etc/ssl/certs/` per Ubuntu). Per TLS, usare lo schema `tls://`.

mssql.availability.group.get[URI,<user>,<password>]

 Restituisce i gruppi di disponibilità.
 Valore restituito: *oggetto JSON*.

Parametri:

- **URI** - URI del server MSSQL (l'unico schema supportato è `sqlserver://`). Le credenziali incorporate verranno ignorate. È possibile specificare un nome di istanza come parte dell'URI, ad esempio: `sqlserver://localhost/InstanceName` (senza porta). Se viene specificata una porta, il nome dell'istanza viene ignorato;
- **user, password** - nome utente, password da inviare al server MSSQL protetto.

Per ulteriori informazioni, vedere il file readme del plugin MSSQL.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Restituisce il risultato di una query personalizzata.
 Valore restituito: *oggetto JSON*.

Parametri:

- **URI** - URI del server MSSQL (l'unico schema supportato è `sqlserver://`). Le credenziali incorporate verranno ignorate. È possibile specificare un nome di istanza come parte dell'URI, ad esempio: `sqlserver://localhost/InstanceName` (senza porta). Se viene specificata una porta, il nome dell'istanza viene ignorato;
- **user, password** - nome utente, password da inviare al server MSSQL protetto;
- **queryName** - nome di una query personalizzata configurata in `Plugins.MSSQL.CustomQueriesDir` senza l'estensione .sql;
- **args** - uno o più argomenti separati da virgole da passare a una query.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.db.get

 Restituisce tutti i database MSSQL disponibili.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.job.status.get

 Restituisce lo stato dei job.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.last.backup.get

 Restituisce l'ora dell'ultimo backup per tutti i database.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.local.db.get

 Restituisce i database che partecipano a un gruppo di disponibilità Always On e a una replica (primaria o secondaria) e che si trovano sul server a cui è stata stabilita la connessione.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.mirroring.get

 Restituisce informazioni sul mirroring.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.nonlocal.db.get

 Restituisce i database che partecipano a un gruppo di disponibilità Always On e una replica (primaria o secondaria) situata su altri server (il database non è locale all'istanza di SQL Server a cui è stata stabilita la connessione).
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.perfcounter.get

 Restituisce i contatori delle prestazioni.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.ping

 Esegue il ping del database. Verifica se la connessione è configurata correttamente.
 Valore restituito: *1* - attivo, *0* - non attivo.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.quorum.get

 Restituisce le informazioni sul quorum.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.quorum.member.get

 Restituisce i membri del quorum.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.replica.get

 Restituisce le repliche.
 Valore restituito: *oggetto JSON*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mssql.version

 Restituisce la versione di MSSQL.
 Valore restituito: *String*.

Per ulteriori informazioni, consultare il file readme del plugin MSSQL.

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Restituisce il risultato di una query personalizzata.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - URI o nome della sessione;
- **user, password** - credenziali di accesso MySQL;

- **queryName** - nome di una query personalizzata; deve corrispondere al nome del file SQL senza estensione;
- **args** - uno o più argomenti separati da virgole da passare a una query.

Per ulteriori informazioni, vedere il file readme del plugin MySQL, sezione *Query personalizzate*.

mysql.db.discovery[connString,<user>,<password>]

 Restituisce l'elenco dei database MySQL. Utilizzato per low-level discovery.
 Valore restituito: il risultato della query SQL "show databases" nel formato JSON LLD.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MySQL.

mysql.db.size[connString,<user>,<password>,<database name>]

 La dimensione del database in byte.
 Valore restituito: risultato della query SQL "select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?" per il database specificato, in byte.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MySQL;
- **database name** - il nome del database.

mysql.get_status_variables[connString,<user>,<password>]

 Valori delle variabili di stato globali.
 Valore restituito: risultato della query SQL "show global status" in formato JSON.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MySQL.

mysql.ping[connString,<user>,<password>]

 Verifica se una connessione è attiva oppure no.
 Valore restituito: *1* - la connessione è attiva; *0* - la connessione è interrotta (se è presente qualsiasi errore, inclusi problemi di AUTH e di configurazione).

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MySQL.

mysql.replication.discovery[connString,<user>,<password>]

 Restituisce l'elenco delle repliche MySQL. Utilizzato per il low-level discovery.
 Valore restituito: il risultato della query SQL `SHOW SLAVE STATUS` o `SHOW REPLICA STATUS` nel formato JSON LLD.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MySQL.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 Lo stato della replica.
 Valore restituito: risultato della query SQL `SHOW SLAVE STATUS` o `SHOW REPLICA STATUS` in formato JSON.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso MySQL;
- **master host** - il nome del host master della replica. Se non ne viene trovato nessuno, viene restituito un errore. Se questo parametro non è specificato, vengono restituiti tutti gli host.

mysql.version[connString,<user>,<password>]

 La versione di MySQL.
 Valore restituito: *String* (con la versione dell'istanza MySQL).

Parametri:

- **connString** - l'URI o il nome della sessione;
- **user, password** - le credenziali di accesso a MySQL.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Esegue una query DNS e restituisce informazioni dettagliate sui record DNS.
 Questo item è una versione estesa dell'item Zabbix agent `net.dns.record` con supporto per più tipi di record e flag personalizzabili.
 Valori restituiti: *oggetto JSON*

Parametri:

- **ip** - l'indirizzo IP del server DNS (lasciare vuoto per usare il server DNS predefinito);
- **name** - il nome DNS da interrogare;
- **type** - il tipo di record da interrogare (il valore predefinito è *SOA*);
- **timeout** - il timeout della richiesta in secondi (il valore predefinito è 1 secondo);
- **count** - il numero di tentativi per la richiesta (il valore predefinito è 2);
- **protocol** - il protocollo usato per eseguire le query DNS: *udp* (predefinito) oppure *tcp*;
- **flags** - uno o più argomenti separati da virgole da passare a una query.

Commenti:

- I valori possibili per `type` sono: *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*. Si noti che i valori devono essere solo in maiuscolo; i valori in minuscolo o misti non sono supportati.
- Per le ricerche DNS inverse (quando `type` è impostato su *PTR*), è possibile fornire il nome DNS sia in formato invertito sia non invertito (vedere gli esempi sotto). Si noti che quando viene richiesto un record PTR, il nome DNS è in realtà un indirizzo IP.
- I valori possibili per `flags` sono:
  - *cdflag* oppure *nocdflag* (predefinito);
  - *rdflag* (predefinito) oppure *nordflag*;
  - *dnssec* oppure *nodnssec* (predefinito);
  - *nsid* oppure *nonsid* (predefinito);
  - *edns0* (predefinito) oppure *noedns0*;
  - *aaflag* oppure *noaaflag* (predefinito);
  - *adflag* oppure *noadflag* (predefinito). I `flags` *dnssec* e *nsid* non possono essere usati insieme a *noedns0*, poiché entrambi richiedono *edns0*. Si noti che i valori devono essere solo in minuscolo; i valori in maiuscolo o misti non sono supportati.
- I nomi di dominio internazionalizzati non sono supportati; utilizzare invece nomi codificati in IDNA.
- L'output è un oggetto contenente le informazioni del record DNS in base ai parametri forniti (vedere maggiori dettagli).

Esempi:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 Il numero di dispositivi GPU.
 Valore restituito: *Intero*.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.decoder.utilization[<deviceUUID>]

 Utilizzo del decoder del dispositivo GPU in percentuale.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.ecc.mode[<deviceUUID>]

 Informazioni sulla modalità ECC del dispositivo GPU (corrente, in sospeso).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.encoder.stats.get[<deviceUUID>]

 Statistiche dell'encoder del dispositivo GPU.
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.encoder.utilization[<deviceUUID>]

 Utilizzo dell'encoder del dispositivo GPU in percentuale.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.energy.consumption[<deviceUUID>]

 Consumo energetico totale del dispositivo GPU in millijoule (mJ) dall'ultimo ricaricamento del driver.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.errors.memory[<deviceUUID>]

 Informazioni sugli errori ECC della memoria del dispositivo GPU (corretti, non corretti).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.errors.register[<deviceUUID>]

 Informazioni sugli errori ECC del registro del dispositivo GPU (corretti, non corretti).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.fan.speed.avg[<deviceUUID>]

 Velocità media della ventola del dispositivo GPU come percentuale della velocità massima.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.get

 GPU rilevate con UUID e nome.
 Valore restituito: *JSON*.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.graphics.frequency[<deviceUUID>]

 Velocità di clock grafica del dispositivo GPU in MHz.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.memory.fb.get[<deviceUUID>]

 Statistiche della memoria framebuffer del dispositivo GPU (totale, riservata, libera, utilizzata).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.memory.bar1.get[<deviceUUID>]

 Statistiche della memoria BAR1 del dispositivo GPU (totale, libera, utilizzata).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.memory.frequency[<deviceUUID>]

 Velocità di clock della memoria del dispositivo GPU in MHz.
 Valore restituito: *Intero*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.pci.utilization[<deviceUUID>]

 Utilizzo PCI del dispositivo GPU (throughput di trasmissione/ricezione in KBps).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.performance.state[<deviceUUID>]

 Stato delle prestazioni del dispositivo GPU.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.power.limit[<deviceUUID>]

 Limite di potenza del dispositivo GPU in milliwatt.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.power.usage[<deviceUUID>]

 Consumo energetico corrente del dispositivo GPU in milliwatt.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.serial[<deviceUUID>]

 Numero di serie del dispositivo GPU.
 Valore restituito: *String*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.sm.frequency[<deviceUUID>]

 Velocità di clock del multiprocessore di streaming del dispositivo GPU in MHz.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.temperature[<deviceUUID>]

 Temperatura del dispositivo GPU in gradi Celsius.
 Valore restituito: *Integer*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.utilization[<deviceUUID>]

 Statistiche di utilizzo del dispositivo GPU (utilizzo di GPU/memoria in percentuale).
 Valore restituito: *JSON*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.device.video.frequency[<deviceUUID>]

 Velocità del clock video del dispositivo GPU in MHz.
 Valore restituito: *Intero*.

Parametro:

- **deviceUUID** - UUID del dispositivo GPU.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.system.driver.version

 La versione del driver NVIDIA.
 Valore restituito: *String*.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

nvml.version

 La versione della libreria NVML.
 Valore restituito: *String*.

Per ulteriori informazioni, consultare il file readme del plugin NVIDIA GPU.

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Restituisce le statistiche dei gruppi di dischi Automatic Storage Management (ASM).
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **diskgroup** - il nome del gruppo di dischi ASM da interrogare.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Restituisce l'elenco dei gruppi di dischi ASM. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;

    - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.archive.info[connString,<user>,<password>,<service>,<destination>]

 Le statistiche dei log di archivio.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
    - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
    - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
    - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni sopra indicate si applica;
    - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **destination** - il nome della destinazione da interrogare.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Le informazioni sui Container Database (CDB).
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
    - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
    - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
    - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
    - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **database** - il nome del database da interrogare.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 Il risultato di una query personalizzata.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
    - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
    - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
    - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
    - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;

- **queryName** - il nome di una query personalizzata; deve corrispondere al nome del file SQL senza estensione;
- **args** - uno o più argomenti separati da virgole da passare alla query.

Commenti: - I dati restituiti vengono convertiti automaticamente in JSON. - Evitare di restituire JSON direttamente dalle query, poiché verrà danneggiato quando il plugin tenterà di convertirlo nuovamente in JSON.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Restituisce le statistiche dei file di dati.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni sopra indicate si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Restituisce l'elenco dei database. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (spazi iniziali ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Restituisce le statistiche della Fast Recovery Area (FRA).
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni sopra indicate si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.instance.info[connString,<user>,<password>,<service>]

 Le statistiche dell'istanza.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Le informazioni sui Pluggable Database (PDB).
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **database** - il nome del database da interrogare.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Restituisce l'elenco dei PDB. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Restituisce le statistiche della Program Global Area (PGA).
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni sopra indicate si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.

- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.ping[connString,<user>,<password>,<service>]

 Verifica se è possibile stabilire una connessione a Oracle.
 Valore restituito: *1* - la connessione è riuscita; *0* - la connessione è interrotta (se viene presentato qualsiasi errore, inclusi problemi di AUTH e di configurazione).

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

Esempi:

```
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>]

 Restituisce le statistiche dei processi.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.redolog.info[connString,<user>,<password>,<service>]

 Le informazioni sul file di log dal file di controllo.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni sopra indicate si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.

- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Restituisce le statistiche della System Global Area (SGA).
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 Restituisce le statistiche delle sessioni.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **lockMaxTime** - la durata massima del blocco della sessione, in secondi, per conteggiare la sessione come bloccata per un periodo prolungato. Valore predefinito: 600 secondi.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Restituisce un insieme di valori delle metriche di sistema.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e verrà comunque trattato come URI;
  - nome della sessione - se tale nome è presente nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e non si applica nessuna delle condizioni precedenti;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **duration** - l'intervallo di acquisizione (in secondi) dei valori delle metriche di sistema. Valori possibili: *60* - durata lunga (predefinita), *15* - durata breve.

oracle.sys.params[connString,<user>,<password>,<service>]

 Restituisce un insieme di valori dei parametri di sistema.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Restituisce le statistiche dei tablespace.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni sopra indicate si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **tablespace** - nome del tablespace da interrogare. Valore predefinito (se lasciato vuoto e `type` è impostato):
  - "TEMP" (se `type` è impostato su "TEMPORARY");
  - "USERS" (se `type` è impostato su "PERMANENT").
- **type** - il tipo di tablespace da interrogare. Valore predefinito (se `tablespace` è impostato): "PERMANENT".
- **conname** - nome del container per il quale sono richieste le informazioni.

Se `tablespace`, `type` o `conname` viene omesso, l'item restituirà le statistiche del tablespace per tutti i container corrispondenti (inclusi PDB e CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Restituisce un elenco di tablespace. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e verrà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

oracle.user.info[connString,<user>,<password>,<service>,<username>]

 Restituisce informazioni sull'utente Oracle.
 Valore restituito: *oggetto JSON*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` o `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle;
- **username** - il nome utente per il quale sono necessarie le informazioni. I nomi utente in minuscolo non sono supportati. Predefinito: utente corrente.

oracle.version[connString,<user>,<password>,<service>]

 Restituisce la versione del server di database.
 Valore restituito: *String*.

Parametri:

- **connString** - può essere:
  - l'URI - se non viene trovata alcuna sessione con il nome specificato e `Plugins.Oracle.ResolveTNS` è impostato su false. Se contiene uno schema (ad esempio, "tcp://"), una porta (ad esempio, 1521) o entrambi, l'opzione ResolveTNS non verrà presa in considerazione e sarà comunque trattato come URI;
  - nome della sessione - se tale nome viene trovato nel file di configurazione del plugin;
  - chiave TNS - se l'opzione ResolveTNS è impostata su true e nessuna delle condizioni precedenti si applica;
  - valore TNS - se inizia con la parentesi aperta "(" (gli spazi iniziali vengono ignorati). In questo caso, l'opzione ResolveTNS non viene presa in considerazione.
- **user** - il nome utente Oracle; supporta l'aggiunta di una delle opzioni di accesso `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oppure `as sysrac` nel formato `user as sysdba` (un'opzione di accesso non distingue tra maiuscole e minuscole e non deve contenere uno spazio finale).
- **password** - la password Oracle;
- **service** - il nome del servizio Oracle.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 Il numero di worker autovacuum.
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.archive[uri,<username>,<password>,<database name>]

 Informazioni sui file archiviati.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 Il numero complessivo di checkpoint per il cluster di database, suddiviso per tipo di checkpoint.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;

- **database name** - il nome del database (se omesso, viene utilizzato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.cache.hit[uri,<username>,<password>,<database name>]

 Il tasso di hit della cache dei buffer di PostgreSQL.
 Valore restituito: *Float*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali di PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.connections[uri,<username>,<password>,<database name>]

 Restituisce le connessioni per tipo.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 Restituisce il risultato di una query personalizzata.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di Plugins.PostgreSQL.Default.Database da postgresql.conf);
- **queryName** - il nome di una query personalizzata, deve corrispondere al nome del file SQL senza estensione;
- **args** - uno o più argomenti separati da virgole da passare a una query.

pgsql.db.age[uri,<username>,<password>,<database name>]

 L'età del più vecchio FrozenXID del database (numero di transazioni).
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.db.bloating_tables[uri,<username>,<password>,<database name>]

 Il numero di tabelle gonfiate per database.
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 Elenco dei database PostgreSQL. Usato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.db.size[uri,<username>,<password>,<database name>]

 La dimensione del database in byte.
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene utilizzato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.dbstat[uri,<username>,<password>,<database name>]

 Raccoglie le statistiche per database. Usato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.dbstat.sum[uri,<username>,<password>,<database name>]

 I dati riepilogativi per tutti i database in un cluster.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.locks[uri,<username>,<password>,<database name>]

 Informazioni sui lock concessi per database. Usato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 L'età dell'XID più vecchio (numero di transazioni).
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.ping[uri,<username>,<password>,<database name>]

 Verifica se una connessione è attiva o meno.
 Valore restituito: *1* - la connessione è attiva; *0* - la connessione è interrotta (se viene rilevato qualsiasi errore, inclusi problemi di AUTH e di configurazione).

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Interroga le metriche in base al tempo di esecuzione.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`);
- **time period** - il limite di tempo di esecuzione per il conteggio delle query lente (deve essere un intero positivo).

pgsql.replication.count[uri,<username>,<password>]

 Il numero di server standby.
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.replication.process[uri,<username>,<password>]

 Il flush lag, il write lag e il replay lag per ciascun processo sender.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.replication.process.discovery[uri,<username>,<password>]

 Il rilevamento del nome del processo di replica.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.replication.recovery_role[uri,<username>,<password>]

 Lo stato di recovery.
 Valore restituito: *0* - modalità master; *1* - la recovery è ancora in corso (modalità standby).

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.replication.status[uri,<username>,<password>]

 Lo stato della replica.
 Valore restituito: *0* - lo streaming non è attivo; *1* - lo streaming è attivo; *2* - modalità master.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.replication_lag.b[uri,<username>,<password>]

 Il ritardo di replica in byte.
 Valore restituito: *Intero*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.replication_lag.sec[uri,<username>,<password>]

 Il ritardo di replica in secondi.
 Valore restituito: *Integer*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL.

pgsql.uptime[uri,<username>,<password>,<database name>]

 L'uptime di PostgreSQL in millisecondi.
 Valore restituito: *Float*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.version[uri,<username>,<password>,<database name>]

 Restituisce la versione di PostgreSQL.
 Valore restituito: *String*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;

- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

pgsql.wal.stat[uri,<username>,<password>,<database name>]

 Le statistiche WAL.
 Valore restituito: *oggetto JSON*.

Parametri:

- **uri** - l'URI o il nome della sessione;
- **username, password** - le credenziali PostgreSQL;
- **database name** - il nome del database (se omesso, viene usato il valore di `Plugins.PostgreSQL.Default.Database` da `postgresql.conf`).

redis.config[connString,<password>,<pattern>]

 Ottiene i parametri di configurazione di un'istanza Redis che corrispondono al pattern.
 Valore restituito: *JSON* - se è stato utilizzato un pattern in stile glob; valore singolo - se il pattern non conteneva alcun carattere jolly.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **password** - la password Redis;
- **pattern** - un pattern in stile glob (* per impostazione predefinita).

Commento:

- Avviso di sicurezza: `redis.config` esegue il comando Redis `CONFIG GET` e può restituire parametri di configurazione sensibili (ad esempio `requirepass`, `masterauth`, impostazioni relative a TLS e altri segreti). L'uso del pattern predefinito (`*`) o di pattern estesi può esporre questi segreti a chiunque abbia il permesso di visualizzare i valori degli item in Zabbix.

redis.info[connString,<password>,<section>]

 Ottiene l'output del comando INFO.
 Valore restituito: *JSON* - l'output è serializzato come JSON.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **password** - la password di Redis;
- **section** - la sezione delle informazioni (*default* per impostazione predefinita).

redis.ping[connString,<password>]

 Verifica se una connessione è attiva oppure no.
 Valore restituito: *1* - la connessione è attiva; *0* - la connessione è interrotta (se è presente qualsiasi errore, inclusi problemi di AUTH e di configurazione).

Parametri:

- **connString** - l'URI o il nome della sessione;
- **password** - la password di Redis.

redis.slowlog.count[connString,<password>]

 Il numero di voci nel log lento da quando Redis è stato avviato.
 Valore restituito: *Integer*.

Parametri:

- **connString** - l'URI o il nome della sessione;
- **password** - la password di Redis.

smart.attribute.discovery

 Restituisce un elenco di attributi del dispositivo S.M.A.R.T.
 Valore restituito: *oggetto JSON*.

Commenti:

- Vengono restituiti i seguenti macro e i relativi valori: `{#NAME}`, `{#DISKTYPE}`, `{#ID}`, `{#ATTRNAME}`, `{#THRESH}`;
- Sono supportati i tipi di unità HDD, SSD e NVME. Le unità possono essere singole o combinate in un RAID. `{#NAME}` avrà un suffisso nel caso di RAID, ad esempio: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.discovery[<type>]

 Restituisce un elenco di dispositivi S.M.A.R.T.
 Valore restituito: *oggetto JSON*.

Parametro:

- **type** - specifica un valore da cercare per i dischi. Valori possibili: *id*, *name* (predefinito). *id* non è supportato su Windows e restituirà un errore se selezionato

Commenti:

- Vengono restituiti i seguenti macro e i relativi valori: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE};
- Sono supportati i tipi di unità HDD, SSD e NVME. Se un'unità non appartiene a un RAID, {#RAIDTYPE} sarà vuoto. {#NAME} avrà un'aggiunta in caso di RAID, ad esempio: {"{#NAME}": "/dev/sda cciss,2"}.

smart.disk.get[<path>,<raid type>]

 Restituisce tutte le proprietà disponibili dei dispositivi S.M.A.R.T.
 Valore restituito: *oggetto JSON*.

Parametri:

- **path** - il percorso del disco, la macro {#PATH} può essere usata come valore;
- **raid_type** - il tipo di RAID, la macro {#RAID} può essere usata come valore

Commenti:

- Sono supportati i tipi di unità HDD, SSD e NVME. Le unità possono essere singole o combinate in un RAID;
- I dati includono la versione di smartctl e gli argomenti della chiamata, oltre ai campi aggiuntivi:
disk_name - contiene il nome con i componenti aggiuntivi richiesti per la discovery del RAID, ad esempio: {"disk_name": "/dev/sda cciss,2"}
disk_type - contiene il tipo di disco HDD, SSD o NVME, ad esempio: {"disk_type": "ssd"};
- Se non vengono specificati parametri, l'item restituirà informazioni su tutti i dischi.

systemd.unit.get[unit name,<interface>]

 Restituisce tutte le proprietà di un'unità systemd.
 Valore restituito: *oggetto JSON*.

Parametri:

- **unit name** - il nome dell'unità (potresti voler usare la macro {#UNIT.NAME} nel prototype dell'item per individuare il nome);
- **interface** - il tipo di interfaccia dell'unità, valori possibili: *Unit* (predefinito), *Service*, *Socket*, *Device*, *Mount*, *Automount*, *Swap*, *Target*, *Path*.

Commenti:

- Questo item è supportato solo sulla piattaforma Linux;
- LoadState, ActiveState e UnitFileState per l'interfaccia Unit vengono restituiti come testo e intero: "ActiveState":{"state":1,"tex

systemd.unit.info[nome unità,<property>,<interface>]

 Informazioni su un'unità systemd.
 Valore restituito: *String*.

Parametri:

- **nome unità** - il nome dell'unità (è possibile utilizzare la macro {#UNIT.NAME} nel prototipo di item per individuare il nome);
- **property** - la property dell'unità (ad esempio, ActiveState (predefinita), LoadState, Description);
- **interface** - il tipo di interfaccia dell'unità (ad esempio, Unit (predefinita), Socket, Service).

Commenti:

- Questo item è supportato solo sulla piattaforma Linux;
- Questo item consente di recuperare una property specifica da un tipo specifico di interfaccia, come descritto nell'API dbus.

Esempi:

```
systemd.unit.info["{#UNIT.NAME}"] #raccoglie informazioni sullo stato attivo (active, reloading, inactive,
systemd.unit.info["{#UNIT.NAME}",LoadState] #raccoglie informazioni sullo stato di caricamento delle unità
systemd.unit.info[mysqld.service,Id] #recupera il nome tecnico del servizio (mysqld.service)
systemd.unit.info[mysqld.service,Description] #recupera la descrizione del servizio (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #recupera l'ultima volta in cui il servizio è entra
systemd.unit.info[dbus.socket,NConnections,Socket] #raccoglie il numero di connessioni da questa unità soc
```

systemd.unit.discovery[<type>]

 Elenco delle unità systemd e dei relativi dettagli. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametro:

- **type** - valori possibili: *all*, *automount*, *device*, *mount*, *path*, *service* (predefinito), *socket*, *swap*, *target*.

Questo item è supportato solo sulla piattaforma Linux.

web.certificate.get[hostname,<port>,<address>]

 Valida i certificati e restituisce i dettagli del certificato.
 Valore restituito: *oggetto JSON*.

Parametro:

- **hostname** - può essere un IP o un DNS.
Può contenere lo schema URL (*solo https*), il path (verrà ignorato) e la porta.
Se una porta è specificata sia nel primo sia nel secondo parametro, i loro valori devono coincidere.
Se è specificato address (il 3° parametro), hostname viene usato solo per SNI e per la verifica dell'hostname;
- **port** - il numero della porta (predefinito: 443 per HTTPS);
- **address** - può essere un IP o un DNS. Se specificato, verrà usato per la connessione, mentre hostname (il 1° parametro) verrà usato per SNI e per la verifica dell'host. Nel caso in cui il 1° parametro sia un IP e il 3° parametro sia un DNS, il 1° parametro verrà usato per la connessione e il 3° parametro per SNI e per la verifica dell'host.

Commenti:

- Questo item diventa non supportato se la destinazione specificata nella configurazione dell'host non esiste, non è disponibile oppure se il TLS handshake fallisce per qualsiasi errore tranne un certificato non valido;
- Attualmente non sono supportati l'estensione X.509 AIA (Authority Information Access), CRL e OCSP (incluso OCSP stapling) e Certificate Transparency;
- Campi della risposta JSON:
  - *x509*: contiene i dettagli del certificato X.509.
    - *version*: la versione X.509 (ad esempio, 3).
    - *serial_number*: il numero di serie del certificato.
    - *signature_algorithm*: l'algoritmo usato per firmare il certificato (ad esempio, `SHA256-RSA`).
    - *issuer*: l'emittente del certificato.
    - *not_before*: la data di inizio validità del certificato.
    - *not_after*: la data di scadenza del certificato.
    - *subject*: il soggetto del certificato.
    - *public_key_algorithm*: l'algoritmo usato per la chiave pubblica (ad esempio, `RSA`).
    - *alternative_names*: nomi alternativi del soggetto (se presenti), altrimenti `null`.
  - result: contiene il risultato della validazione.
    - *value*: lo stato della validazione (vedere i possibili valori sotto).
    - *message*: messaggio dettagliato di validazione (ad esempio, `"certificate verified successfully"`).
  - *sha1_fingerprint*: l'impronta SHA-1 del certificato.
  - *sha256_fingerprint*: l'impronta SHA-256 del certificato.
- Il campo `$.result.value` indica il risultato della validazione del certificato. I valori possibili includono:
  - *valid* - il certificato è valido e attendibile.
  - *valid-but-self-signed* - il certificato è valido ma autofirmato, il che significa che il suo subject corrisponde al suo issuer.
  - *invalid* - il certificato non è valido a causa di un problema come scadenza, hostname non corretto o autorità di firma sconosciuta.

Esempio:

```
web.certificate.get[example.com,443]
```

Risposta JSON:


```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
```

```
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```

Windows Zabbix agent

Panoramica

Gli item dell'agent Zabbix per Windows sono presentati in due elenchi:

- Item condivisi - le chiavi item condivise con l'agent Zabbix per UNIX;
- Item specifici di Windows - le chiavi item supportate **solo** su Windows.

Si noti che tutte le chiavi item supportate dall'agent Zabbix su Windows sono supportate anche dall'agent Zabbix 2 di nuova generazione. Vedere anche le chiavi item aggiuntive che possono essere utilizzate solo con agent 2.

Vedere anche: Permessi minimi per gli item Windows

Item condivisi

La tabella seguente elenca gli item di Zabbix agent supportati su Windows e condivisi con Zabbix agent UNIX:

- La chiave dell'item è un collegamento ai dettagli completi dell'item di Zabbix agent UNIX
- Sono inclusi i commenti sugli item rilevanti per Windows

Item key	Description	Item group
log	Il monitoraggio di un file di log. Questo item non è supportato per Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	Log monitoring
log.count	Il numero di righe corrispondenti in un file di log monitorato. Questo item non è supportato per Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	
logrt	Il monitoraggio di un file di log ruotato. Questo item non è supportato per Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	
logrt.count	Il numero di righe corrispondenti in un file di log monitorato ruotato. Questo item non è supportato per Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	
modbus.get	Legge i dati Modbus.	Modbus
net.dns	Verifica se il servizio DNS è attivo. I parametri `ip`, `timeout` e `count` vengono ignorati su Windows, a meno che non si utilizzi Zabbix agent 2.	Network
net.dns.perf	Verifica le prestazioni di un servizio DNS. I parametri `ip`, `timeout` e `count` vengono ignorati su Windows, a meno che non si utilizzi Zabbix agent 2.	
net.dns.record	Esegue una query DNS. I parametri `ip`, `timeout` e `count` vengono ignorati su Windows, a meno che non si utilizzi Zabbix agent 2.	
net.if.discovery	L'elenco delle interfacce di rete. Alcune versioni di Windows (ad esempio Server 2008) potrebbero richiedere l'installazione degli aggiornamenti più recenti per supportare i caratteri non ASCII nei nomi delle interfacce.	
net.if.in	Le statistiche del traffico in ingresso su un'interfaccia di rete. Su Windows, l'item ottiene i valori dai contatori a 64 bit, se disponibili. I contatori statistici delle interfacce a 64 bit sono stati introdotti in Windows Vista e Windows Server 2008. Se i contatori a 64 bit non sono disponibili, l'agent usa contatori a 32 bit. I nomi delle interfacce multibyte sono supportati su Windows. È possibile ottenere le descrizioni delle interfacce di rete su Windows con gli item net.if.discovery o net.if.list.	

Item key	Description	Item group
net.if.out	Le statistiche del traffico in uscita su un'interfaccia di rete. Su Windows, l'item ottiene i valori dai contatori a 64 bit, se disponibili. I contatori statistici delle interfacce a 64 bit sono stati introdotti in Windows Vista e Windows Server 2008. Se i contatori a 64 bit non sono disponibili, l'agent usa contatori a 32 bit. I nomi delle interfacce multibyte sono supportati su Windows. È possibile ottenere le descrizioni delle interfacce di rete su Windows con gli item net.if.discovery o net.if.list.	
net.if.total	La somma delle statistiche del traffico in ingresso e in uscita su un'interfaccia di rete. Su Windows, l'item ottiene i valori dai contatori a 64 bit, se disponibili. I contatori statistici delle interfacce a 64 bit sono stati introdotti in Windows Vista e Windows Server 2008. Se i contatori a 64 bit non sono disponibili, l'agent usa contatori a 32 bit. I nomi delle interfacce multibyte sono supportati su Windows. È possibile ottenere le descrizioni delle interfacce di rete su Windows con gli item net.if.discovery o net.if.list.	
net.tcp.listen	Verifica se questa porta TCP è nello stato LISTEN.	
net.tcp.port	Verifica se è possibile stabilire una connessione TCP alla porta specificata.	
net.tcp.service	Verifica se un servizio è in esecuzione e accetta connessioni TCP. Il controllo di LDAP e HTTPS su Windows è supportato solo da Zabbix agent 2.	
net.tcp.service.perf	Verifica le prestazioni di un servizio TCP. Il controllo di LDAP e HTTPS su Windows è supportato solo da Zabbix agent 2.	
net.tcp.socket.count	Restituisce il numero di socket TCP che corrispondono ai parametri. Questo item è supportato su Linux da Zabbix agent, ma su Windows è supportato solo da Zabbix agent 2 su Windows a 64 bit.	
net.udp.service	Verifica se un servizio è in esecuzione e risponde alle richieste UDP.	
net.udp.service.perf	Verifica le prestazioni di un servizio UDP.	
net.udp.socket.count	Restituisce il numero di socket UDP che corrispondono ai parametri. Questo item è supportato su Linux da Zabbix agent, ma su Windows è supportato solo da Zabbix agent 2 su Windows a 64 bit.	
proc.get	L'elenco dei processi del sistema operativo e dei relativi parametri. Il parametro `cmdline` non è supportato su Windows.	Processes
proc.num	Il numero di processi. Su Windows, sono supportati solo i parametri `name` e `user`.	
system.cpu.discovery	L'elenco delle CPU/nuclei CPU rilevati.	System
system.cpu.load	Il carico della CPU. Quando un processo collector viene avviato su Zabbix agent, i seguenti contatori di prestazioni vengono inizializzati e successivamente usati per questo item: `\System\Processor Queue Length`	
system.cpu.num	Il numero di CPU.	
system.cpu.util	La percentuale di utilizzo della CPU. Il valore viene acquisito usando il contatore di prestazioni *Processor Time*. Si noti che da Windows 8 il Task Manager mostra l'utilizzo della CPU in base al contatore di prestazioni *Processor Utility*, mentre nelle versioni precedenti era il contatore *Processor Time* (vedere maggiori dettagli). *system* è l'unico parametro `type` supportato su Windows.	
system.hostname	Il nome host del sistema. Il valore viene acquisito tramite le funzioni `GetComputerName()` (per **netbios**), `GetComputerNameExA()` (per **fqdn**) o `gethostname()` (per **host**) su Windows. Vedere anche una descrizione più dettagliata.	
system.localtime	L'ora del sistema.	
system.run	Esegue il comando specificato sull'host.	
system.sw.arch	Le informazioni sull'architettura software.	
system.sw.os	Le informazioni sul sistema operativo.	
system.sw.os.get	Informazioni dettagliate sul sistema operativo (versione, tipo, nome della distribuzione, versione minore e maggiore, ecc.).	

Item key	Description	Item group
system.swap.size	La dimensione dello spazio di swap in byte o in percentuale sul totale. Il parametro di tipo `pused` è supportato su Linux da Zabbix agent, ma su Windows è supportato solo da Zabbix agent 2. Si noti che questa chiave potrebbe riportare una dimensione/percentuale dello spazio di swap non corretta su piattaforme Windows virtualizzate (VMware ESXi, VirtualBox). In questo caso è possibile usare la chiave `perf_counter[\700(_Total)\702]` per ottenere la percentuale corretta dello spazio di swap.	
system.uname	Identificazione del sistema. Su Windows il valore di questo item viene ottenuto dalle classi WMI `Win32_OperatingSystem` e `Win32_Processor`. Il nome del sistema operativo (inclusa l'edizione) potrebbe essere tradotto nella lingua di visualizzazione dell'utente. In alcune versioni di Windows contiene simboli di marchio registrato e spazi aggiuntivi.	
system.uptime	Il tempo di attività del sistema in secondi.	
vfs.dir.count	Il numero di voci della directory. Su Windows, i collegamenti simbolici alle directory vengono ignorati e i collegamenti fisici vengono conteggiati una sola volta.	Virtual file systems
vfs.dir.get	L'elenco delle voci della directory. Su Windows, i collegamenti simbolici alle directory vengono ignorati e i collegamenti fisici vengono conteggiati una sola volta.	
vfs.dir.size	La dimensione della directory. Su Windows qualsiasi collegamento simbolico viene ignorato e i collegamenti fisici vengono presi in considerazione una sola volta.	
vfs.file.cksum	Il checksum del file, calcolato dall'algoritmo UNIX cksum.	
vfs.file.contents	Recupero del contenuto di un file.	
vfs.file.exists	Verifica se il file esiste. Su Windows, le virgolette doppie devono essere escape con backslash '\' e l'intera chiave dell'item deve essere racchiusa tra virgolette doppie quando si usa l'utility da riga di comando per chiamare `zabbix_get.exe` o `agent2`. Si noti che l'item può diventare non supportato su Windows se si cerca una directory all'interno di una directory inesistente, ad esempio `vfs.file.exists[C:\no\dir,dir]` (dove 'no' non esiste).	
vfs.file.get	Restituisce informazioni su un file. Tipi di file supportati su Windows: file regolare, directory, collegamento simbolico	
vfs.file.md5sum	Il checksum MD5 del file.	
vfs.file.owner	Recupera il proprietario di un file.	
vfs.file.regexp	Recupera una stringa nel file.	
vfs.file.regmatch	Trova una stringa nel file.	
vfs.file.size	La dimensione del file.	
vfs.file.time	Le informazioni sull'ora del file. Su Windows XP `vfs.file.time[file,change]` può essere uguale a `vfs.file.time[file,access]`.	
vfs.fs.discovery	L'elenco dei filesystem montati con il relativo tipo e le opzioni di mount. La macro `{#FSLABEL}` è supportata su Windows.	
vfs.fs.get	L'elenco dei filesystem montati con il relativo tipo, lo spazio disponibile su disco, le statistiche inode e le opzioni di mount. La macro `{#FSLABEL}` è supportata su Windows.	
vfs.fs.size	Lo spazio su disco in byte o in percentuale sul totale.	
vm.memory.size	La dimensione della memoria in byte o in percentuale sul totale.	Virtual memory
web.page.get	Ottiene il contenuto di una pagina web.	Web monitoring
web.page.perf	Il tempo di caricamento di una pagina web completa.	
web.page.regexp	Trova una stringa nella pagina web.	
agent.hostmetadata	I metadati dell'host dell'agent.	Zabbix
agent.hostname	Il nome host dell'agent.	
agent.ping	Il controllo di disponibilità dell'agent.	
agent.variant	La variante di Zabbix agent (Zabbix agent o Zabbix agent 2).	
agent.version	La versione di Zabbix agent.	
zabbix.stats	Restituisce da remoto un insieme di metriche interne di Zabbix server o proxy.	

Item key	Description	Item group
zabbix.stats (queue)	Restituisce da remoto il numero di item monitorati nella coda che sono in ritardo su Zabbix server o proxy.	

Item specifici di Windows

La tabella fornisce dettagli sulle chiavi item supportate **solo** dallo Zabbix agent per Windows.

Gli item specifici di Windows talvolta sono una controparte approssimativa di un item agent simile; ad esempio, `proc_info`, supportato in Windows, corrisponde approssimativamente all'item `proc.mem`, non supportato in Windows.

La chiave item è un collegamento ai dettagli completi della chiave item.

Chiave item	Descrizione	Gruppo di item
eventlog	Il monitoraggio del registro eventi di Windows.	Monitoraggio dei log
eventlog.count	Il conteggio delle righe nel registro eventi di Windows.	
net.if.list	L'elenco delle interfacce di rete (include tipo di interfaccia, stato, indirizzo IPv4, descrizione).	Rete
perf_counter	Il valore di qualsiasi contatore delle prestazioni di Windows.	Contatori delle prestazioni
perf_counter_en	Il valore di qualsiasi contatore delle prestazioni di Windows in inglese.	
perf_instance.discovery	L'elenco delle istanze degli oggetti dei contatori delle prestazioni di Windows.	
perf_instance_en.discovery	L'elenco delle istanze degli oggetti dei contatori delle prestazioni di Windows, rilevate utilizzando i nomi degli oggetti in inglese.	
proc_info	Varie informazioni su processi specifici.	Processi
registry.data	Restituisce i dati per il nome valore specificato nella chiave del Registro di sistema di Windows.	Registro di sistema
registry.get	L'elenco dei valori o delle chiavi del Registro di sistema di Windows presenti nella chiave specificata.	
service.discovery	L'elenco dei servizi di Windows.	Servizi
service.info	Informazioni su un servizio.	
services	L'elenco dei servizi.	
vm.vmemory.size	La dimensione della memoria virtuale in byte o in percentuale del totale.	Memoria virtuale
wmi.get	Esegue una query WMI e restituisce il primo oggetto selezionato.	WMI
wmi.getall	Esegue una query WMI e restituisce l'intera risposta.	

Dettagli della chiave item

I parametri senza parentesi angolari sono obbligatori. I parametri contrassegnati con parentesi angolari **< >** sono facoltativi.

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 Il monitoraggio del registro eventi.
 Valore restituito: *Log*.

Parametri:

- **name** - il nome del canale del registro eventi (*Log Name* nell'interfaccia grafica di Event Viewer);
- **regexp** - una espressione regolare che descrive il pattern richiesto (sensibile alle maiuscole/minuscole);
- **severity** - un'espressione regolare che descrive la severità (non sensibile alle maiuscole/minuscole). Questo parametro accetta un'espressione regolare basata sui seguenti valori: "Information", "Warning", "Error", "Critical", "Verbose" (in esecuzione su Windows Vista o versioni successive).
- **source** - un'espressione regolare che descrive l'identificatore della sorgente (non sensibile alle maiuscole/minuscole);
- **eventid** - un'espressione regolare che descrive l'identificatore o gli identificatori dell'evento (sensibile alle maiuscole/minuscole);
- **maxlines** - il numero massimo di nuove righe al secondo che l'agent invierà a Zabbix server o proxy. Questo parametro sostituisce il valore di `MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - valori possibili:
  - *all* (predefinito);
  - *skip* - salta l'elaborazione dei dati più vecchi (influisce solo sugli item creati di recente).

Commenti:

- L'item deve essere configurato come controllo attivo;
- L'agent non è in grado di inviare eventi dal registro "Forwarded events";
- È supportato Windows Eventing 6.0;
- La selezione di un tipo di informazione diverso da Log per questo item comporterà la perdita del timestamp locale, oltre alle informazioni sulla severità e sulla sorgente del log;
- Vedere anche ulteriori informazioni sul monitoraggio dei log.

Esempi:

```
eventlog[Application]
eventlog[Microsoft-Windows-Application-Experience/Program-Compatibility-Assistant]
eventlog[Security,,"Failure Audit",,^(529|680)$]
eventlog[System,,"Warning|Error"]
eventlog[System,,,,^1$]
eventlog[Windows PowerShell,,,,,,skip]
eventlog[System,,,,@TWOSHORT] #here a custom regular expression named `TWOSHORT` is referenced (defined as
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Il numero di righe nel registro eventi di Windows.
 Valore restituito: *Integer*.

Parametri:

- **name** - il nome del canale del registro eventi (*Log Name* nell'interfaccia grafica di Event Viewer);
- **regexp** - una espressione regolare che descrive il pattern richiesto (sensibile alle maiuscole/minuscole);
- **severity** - un'espressione regolare che descrive la severità (non sensibile alle maiuscole/minuscole). Questo parametro accetta un'espressione regolare basata sui seguenti valori: "Information", "Warning", "Error", "Critical", "Verbose" (in esecuzione su Windows Vista o versioni successive).
- **source** - un'espressione regolare che descrive l'identificatore della sorgente (non sensibile alle maiuscole/minuscole);
- **eventid** - un'espressione regolare che descrive l'identificatore o gli identificatori dell'evento (sensibile alle maiuscole/minuscole);
- **maxproclines** - il numero massimo di nuove righe al secondo che l'agent analizzerà (non può superare 10000). Il valore predefinito è 10*`MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - valori possibili:
  - *all* (predefinito);
  - *skip* - salta l'elaborazione dei dati più vecchi (influisce solo sugli item creati di recente).

Commenti:

- L'item deve essere configurato come active check;
- L'agent non è in grado di inviare eventi dal registro "Forwarded events";
- È supportato Windows Eventing 6.0;
- La selezione di un tipo di informazione non Log per questo item comporterà la perdita del timestamp locale, nonché delle informazioni sulla severità del log e sulla sorgente;
- Vedere anche ulteriori informazioni sul monitoraggio dei log.

Esempi:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 L'elenco delle interfacce di rete (include tipo di interfaccia, stato, indirizzo IPv4, descrizione).
 Valore restituito: *Testo*.

Commenti:

- Sono supportati i nomi delle interfacce multi-byte;
- Le interfacce disabilitate non sono elencate;
- L'abilitazione/disabilitazione di alcuni componenti può modificarne l'ordine nel nome dell'interfaccia di Windows;
- Alcune versioni di Windows (ad esempio, Server 2008) potrebbero richiedere l'installazione degli aggiornamenti più recenti per supportare i caratteri non ASCII nei nomi delle interfacce.

perf_counter[counter,<interval>]

 Il valore di qualsiasi contatore di prestazioni di Windows.
 Valore restituito: *Integer*, *float*, *string* o *text* (a seconda della richiesta).

Parametri:

- **counter** - il percorso del contatore;

- **interval** - gli ultimi N secondi per memorizzare il valore medio. Il `interval` deve essere compreso tra 1 e 900 secondi (inclusi) e il valore predefinito è 1.

Commenti:

- Questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza;
- `interval` viene usato per i contatori che richiedono più di un campione (come l'utilizzo della CPU), quindi il controllo restituisce ogni volta un valore medio degli ultimi "interval" secondi;
- Performance Monitor può essere usato per ottenere l'elenco dei contatori disponibili.
- Vedi anche: Contatori di prestazioni di Windows.

perf_counter_en[counter,<interval>]

 Il valore di qualsiasi contatore di prestazioni di Windows in inglese.
 Valore restituito: *Integer*, *float*, *string* o *text* (a seconda della richiesta).

Parametri:

- **counter** - il percorso del contatore in inglese;
- **interval** - gli ultimi N secondi per memorizzare il valore medio. Il `interval` deve essere compreso tra 1 e 900 secondi (inclusi) e il valore predefinito è 1.

Commenti:

- Questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza;
- `interval` viene usato per i contatori che richiedono più di un campione (come l'utilizzo della CPU), quindi il controllo restituisce ogni volta un valore medio degli ultimi "interval" secondi;
- Questo item è supportato solo su **Windows Server 2008/Vista** e versioni successive;
- È possibile trovare l'elenco delle stringhe in inglese visualizzando la seguente chiave del registro: `HKEY_LOCAL_MACHINE\SOFTWARE\M NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[object]

 L'elenco delle istanze degli oggetti dei contatori delle prestazioni di Windows. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametro:

- **object** - il nome dell'oggetto (localizzato).

perf_instance_en.discovery[object]

 L'elenco delle istanze degli oggetti dei contatori delle prestazioni di Windows, individuate utilizzando i nomi degli oggetti in inglese. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

Parametro:

- **object** - il nome dell'oggetto (in inglese).

proc_info[process,<attribute>,<type>]

 Varie informazioni su uno o più processi specifici.
 Valore restituito: *Float*.

Parametri:

- **process** - il nome del processo;
- **attribute** - l'attributo del processo richiesto;
- **type** - il tipo di rappresentazione (significativo quando esiste più di un processo con lo stesso nome)

Commenti:

- Sono supportati i seguenti `attributes`:
  - *vmsize* (predefinito) - dimensione della memoria virtuale del processo in Kbyte;
  - *wkset* - dimensione del working set del processo (quantità di memoria fisica usata dal processo) in Kbyte;
  - *pf* - numero di page fault;
  - *ktime* - tempo kernel del processo in millisecondi;
  - *utime* - tempo utente del processo in millisecondi;
  - *io_read_b* - numero di byte letti dal processo durante le operazioni di I/O;
  - *io_read_op* - numero di operazioni di lettura eseguite dal processo;
  - *io_write_b* - numero di byte scritti dal processo durante le operazioni di I/O;
  - *io_write_op* - numero di operazioni di scrittura eseguite dal processo;
  - *io_other_b* - numero di byte trasferiti dal processo durante operazioni diverse da quelle di lettura e scrittura;
  - *io_other_op* - numero di operazioni di I/O eseguite dal processo, diverse da quelle di lettura e scrittura;
  - *gdiobj* - numero di oggetti GDI usati dal processo;

    - *userobj* - numero di oggetti USER usati dal processo.
- I types validi sono:
    - *avg* (predefinito) - valore medio per tutti i processi denominati <process>;
    - *min* - valore minimo tra tutti i processi denominati <process>;
    - *max* - valore massimo tra tutti i processi denominati <process>;
    - *sum* - somma dei valori per tutti i processi denominati <process>.
- Su un sistema a 64 bit, per il corretto funzionamento di questo item è necessario un agent Zabbix a 64 bit.

Esempi:

```
proc_info[iexplore.exe,wkset,sum] #recupera la quantità di memoria fisica utilizzata da tutti i processi d
proc_info[iexplore.exe,pf,avg] #recupera il numero medio di page fault per i processi di Internet Explorer
```

registry.data[key,<value name>]

 Restituisce i dati per il nome valore specificato nella chiave del Registro di Windows.
 Valore restituito: *Integer*, *string* o *text* (a seconda del tipo di valore)

Parametri:

- **key** - la chiave del Registro, inclusa la chiave radice; sono consentite le abbreviazioni della chiave radice (ad esempio HKLM);
- **value name** - il nome del valore del Registro nella chiave (stringa vuota "" per impostazione predefinita). Il valore predefinito viene restituito se il nome del valore non viene specificato.

Commenti:

- Abbreviazioni della chiave radice supportate:
    - 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
- Le chiavi con spazi devono essere racchiuse tra virgolette doppie.

Esempi:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #restituisce i dati
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #restituisce i dati d
```

registry.get[key,<mode>,<name regexp>]

 L'elenco dei valori o delle chiavi del Registro di Windows situati nella chiave specificata.
 Valore restituito: *oggetto JSON*.

Parametri:

- **key** - la chiave del Registro, inclusa la chiave radice; sono consentite le abbreviazioni della chiave radice (ad esempio HKLM) (vedere i commenti di `registry.data[]` per l'elenco completo delle abbreviazioni);
- **mode** - valori possibili:
values (predefinito) oppure *keys*;
- **name regexp** - scopre solo i valori i cui nomi corrispondono alla regexp (predefinito - scopre tutti i valori). Consentito solo con *values* come *mode*.

Le chiavi con spazi devono essere racchiuse tra virgolette doppie.

Esempi:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
Il JSON includerà i dettagli della chiave, dell'ultima sottochiave, del nome del valore, del tipo di valor
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #restituisce i dati di tutti
Il JSON includerà i dettagli della chiave, dell'ultima sottochiave, del nome del valore, del tipo di valor
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #restituisce tutte le sottochi
Il JSON includerà i dettagli della chiave e dell'ultima sottochiave.
```

service.discovery

 L'elenco dei servizi Windows. Utilizzato per low-level discovery.
 Valore restituito: *oggetto JSON*.

service.info[service,<param>]

 Informazioni su un servizio.
 Valore restituito: - *Integer* - con `param` come *state*, *startup*;

 In particolare per *state*: - 0 - in esecuzione - 1 - in pausa - 2 - avvio in sospeso - 3 - pausa in sospeso - 4 - ripresa in sospeso - 5 - arresto in sospeso - 6 - arrestato - 7 - sconosciuto - 255 - servizio inesistente

 In particolare per *startup*: - 0 - automatico - 1 - automatico ritardato - 2 - manuale - 3 - disabilitato - 4 - sconosciuto - 5 - avvio automatico con trigger - 6 - avvio automatico ritardato con trigger - 7 - avvio manuale con trigger - *String* - con `param` come *displayname*, *path*, *user*; - *Text* - con `param` come *description*.

Parametri:

- **service** - un nome di servizio reale o il relativo nome visualizzato come appare nello snap-in Servizi di MMC;
- **param** - *state* (predefinito), *displayname*, *path*, *user*, *startup* o *description*.

Commenti:

- Elementi come `service.info[service,state]` e `service.info[service]` restituiranno le stesse informazioni;
- Solo con `param` impostato su *state* questo item restituisce un valore per i servizi inesistenti (255).

Esempi:

```
service.info[SNMPTRAP] - stato del servizio SNMPTRAP;
service.info[SNMP Trap] - stato dello stesso servizio, ma con il nome visualizzato specificato;
service.info[EventLog,startup] - tipo di avvio del servizio EventLog
```

services[<type>,<state>,<exclude>]

 L'elenco dei servizi.
 Valore restituito: *0* - se vuoto; *Text* - l'elenco dei servizi separati da una nuova riga.

Parametri:

- **type** - *all* (predefinito), *automatic*, *manual* o *disabled*;
- **state** - *all* (predefinito), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* o *paused*;
- **exclude** - i servizi da escludere dal risultato. I servizi esclusi devono essere elencati tra virgolette doppie, separati da virgole, senza spazi.

Esempi:

```
services[,started] #restituisce l'elenco dei servizi avviati;
services[automatic, stopped] #restituisce l'elenco dei servizi arrestati che dovrebbero essere in esecuzio
services[automatic, stopped, "service1,service2,service3"] #restituisce l'elenco dei servizi arrestati che
```

vm.vmemory.size[<type>]

 La dimensione della memoria virtuale in byte o in percentuale del totale.
 Valore restituito: *Integer* - per i byte; *float* - per la percentuale.

Parametro:

- **type** - valori possibili: *available* (memoria virtuale disponibile), *pavailable* (memoria virtuale disponibile, in percentuale), *pused* (memoria virtuale utilizzata, in percentuale), *total* (memoria virtuale totale, predefinito) oppure *used* (memoria virtuale utilizzata)

Commenti:

- Il monitoraggio delle statistiche della memoria virtuale si basa su:
  - La memoria virtuale totale in Windows (memoria fisica totale + dimensione del file di paging);
  - La quantità massima di memoria che Zabbix agent può impegnare;
  - Il limite corrente di memoria impegnata per il sistema o per Zabbix agent, a seconda di quale sia inferiore.

Esempio:

```
vm.vmemory.size[pavailable] #restituisce la memoria virtuale disponibile, in percentuale
```

wmi.get[<namespace>,<query>]

 Esegue una query WMI e restituisce il primo oggetto selezionato.
 Valore restituito: *Integer*, *float*, *string* o *text* (a seconda della richiesta).

Parametri:

- **namespace** - lo spazio dei nomi WMI;
- **query** - la query WMI che restituisce un singolo oggetto.

Le query WMI vengono eseguite con WQL.

Esempio:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #returns the sta
```

wmi.getall[<namespace>,<query>]

 Esegue una query WMI e restituisce l'intera risposta. Può essere utilizzato per la low-level discovery.
 Valore restituito: *oggetto JSON*

Parametri:

- **namespace** - lo spazio dei nomi WMI;
- **query** - la query WMI.

Commenti:

- Le query WMI vengono eseguite con WQL.
- Il preprocessing JSONPath può essere utilizzato per puntare a valori più specifici nel JSON restituito.

Esempio:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #returns status inf
```

Monitoraggio dei servizi Windows

Questo tutorial fornisce istruzioni passo passo per configurare il monitoraggio dei servizi Windows. Si presuppone che Zabbix server e agent siano configurati e operativi.

Passo 1

Ottieni il nome del servizio.

Puoi ottenere il nome del servizio aprendo lo snap-in Servizi di MMC e visualizzando le proprietà del servizio. Nella scheda *Generale* dovresti vedere un campo chiamato *Nome del servizio*. Il valore che segue è il nome che utilizzerai quando configuri un item per il monitoraggio. Ad esempio, se volessi monitorare il servizio "workstation", il tuo servizio potrebbe essere: **lanmanworkstation**.

Passaggio 2

Configurare un item per monitorare il servizio.

L'item `service.info[service,<param>]` recupera informazioni su un determinato servizio. A seconda delle informazioni necessarie, specificare l'opzione `param`, che accetta i seguenti valori: *displayname*, *state*, *path*, *user*, *startup* oppure *description*. Il valore predefinito è *state* se `param` non è specificato (`service.info[service]`).

Il tipo di valore restituito dipende dal `param` scelto: integer per *state* e *startup*; character string per *displayname*, *path* e *user*; text per *description*.

Esempio:

- *Chiave:* `service.info[lanmanworkstation]`
- *Tipo di informazione:* Numerico (senza segno)

L'item `service.info[lanmanworkstation]` recupererà informazioni sullo stato del servizio come valore numerico. Per mappare un valore numerico a una rappresentazione testuale nel frontend ("0" come "Running", "1" come "Paused", ecc.), è possibile configurare una mappatura dei valori sull'host su cui è configurato l'item. Per farlo, collegare all'host il template *Windows services by Zabbix agent* oppure *Windows services by Zabbix agent active*, oppure configurare sull'host una nuova mappa dei valori basata sulla mappa dei valori *Windows service state* configurata nei template menzionati.

Si noti che entrambi i template menzionati hanno una regola di discovery configurata che rileverà automaticamente i servizi. Se non lo si desidera, è possibile disabilitare la regola di discovery a livello di host una volta che il template è stato collegato all'host.

Individuazione dei servizi Windows

La low-level discovery offre un modo per creare automaticamente item, trigger e grafici per diverse entità su un computer. Zabbix può iniziare automaticamente a monitorare i servizi Windows sulla tua macchina, senza la necessità di conoscere il nome esatto di un servizio o di creare manualmente item per ciascun servizio. È possibile utilizzare un filtro per generare item, trigger e grafici reali solo per i servizi di interesse.

Monitoraggio dei file di log

Panoramica

Zabbix può essere utilizzato per il monitoraggio centralizzato e l'analisi dei file di log con o senza supporto per la rotazione dei log.

Le notifiche possono essere utilizzate per avvisare gli utenti quando un file di log contiene determinate stringhe o pattern di stringhe.

Per monitorare un file di log è necessario avere:

- Zabbix agent in esecuzione sull'host.
- item di monitoraggio dei log configurato.

Attention:
La dimensione massima di un file di log monitorato dipende dal supporto per file di grandi dimensioni.

Configurazione

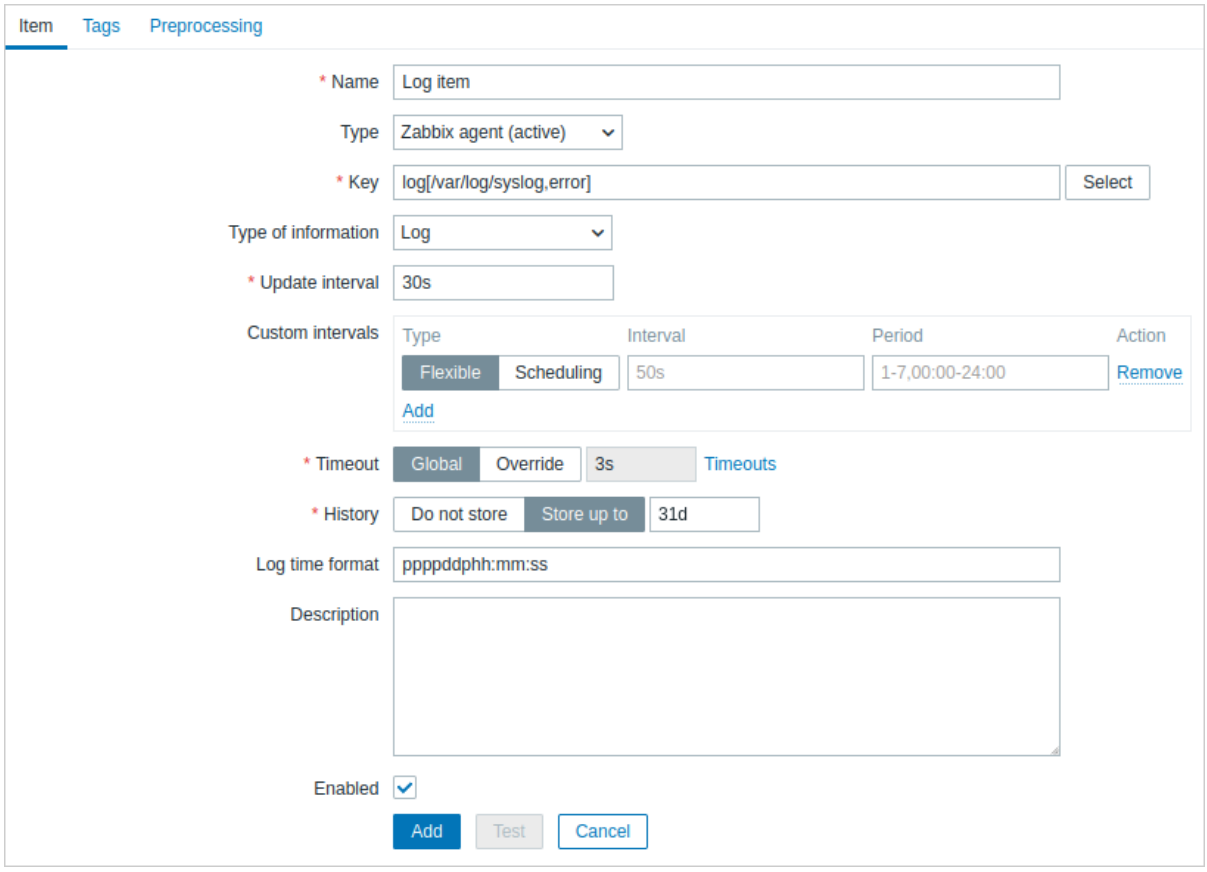
Verificare i parametri dell'agent

Assicurarsi che nel file di configurazione dell'agent:

- Il parametro `Hostname` corrisponda al nome dell'host nel frontend.
- I server nel parametro `ServerActive` siano specificati per l'elaborazione dei controlli attivi.

Configurazione dell'item

Configura un item per il monitoraggio dei log.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

In particolare, per gli item di monitoraggio dei log è necessario specificare:

Tipo	Selezionare qui **Zabbix agent (active)**.
Chiave	Utilizzare una delle seguenti chiavi item: `log[]` o `logrt[]`: Queste due chiavi item consentono di monitorare i log e filtrare le voci di log in base all'espressione regolare del contenuto, se presente. Ad esempio: `log[/var/log/syslog,error]`. Assicurarsi che il file disponga dei permessi di lettura per l'utente 'zabbix'; in caso contrario, lo stato dell'item sarà impostato su 'unsupported'. **log.count[]** o **logrt.count[]**: Queste due chiavi item consentono di restituire solo il numero di righe corrispondenti. Per informazioni dettagliate sull'utilizzo di queste chiavi item e dei relativi parametri, consultare la sezione sulle chiavi item dell'agent Zabbix supportate.

Tipo di informazioni	Precompilato automaticamente: Per gli item `log[]` o `logrt[]` - *Log*; Per gli item `log.count[]` o `logrt.count[]` - *Numeric (unsigned)*. Se si utilizza facoltativamente il parametro `output`, è possibile selezionare manualmente un tipo di informazioni appropriato diverso da *Log*. Tenere presente che la scelta di un tipo di informazioni diverso da Log comporterà la perdita del timestamp locale.
Intervallo di aggiornamento (in sec)	Il parametro definisce la frequenza con cui Zabbix agent verificherà la presenza di modifiche nel file di log. Impostandolo su 1 secondo, si assicura di ricevere i nuovi record il prima possibile.
Formato dell'ora del log	In questo campo è possibile specificare facoltativamente il modello per l'analisi del timestamp della riga di log. Segnaposto supportati: * **y**: *Anno (1970-2038)* * **M**: *Mese (01-12)* * **d**: *Giorno (01-31)* * **h**: *Ora (00-23)* * **m**: *Minuto (00-59)* * **s**: *Secondo (00-59)* Se lasciato vuoto, il timestamp sarà impostato su 0 nel formato Unix, che rappresenta `January 1, 1970`. Ad esempio, si consideri la seguente riga del file di log di Zabbix agent: `23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).` Inizia con sei posizioni di caratteri per il PID, seguite da data, ora e dal resto del messaggio. Il formato dell'ora del log per questa riga sarebbe "pppppp:yyyyMMdd:hhmmss". Tenere presente che i caratteri "p" e ":" sono segnaposto e possono essere qualsiasi carattere ad eccezione di "yMdhms".

Note importanti

- Il server e l'agent mantengono la traccia della dimensione di un log monitorato e dell'ultima ora di modifica (per `logrt`) in due contatori. Inoltre:
  - L'agent utilizza internamente anche i numeri inode (su UNIX/GNU/Linux), gli indici di file (su Microsoft Windows) e i valori MD5 dei primi 512 byte del file di log per migliorare le decisioni quando i file di log vengono troncati e ruotati.
  - Su sistemi UNIX/GNU/Linux si presume che i file system in cui sono archiviati i file di log riportino numeri inode, che possono essere usati per tracciare i file.
  - Su Microsoft Windows Zabbix agent determina il tipo di file system su cui risiedono i file di log e usa:
    * Su file system NTFS indici di file a 64 bit.
    * Su file system ReFS (solo da Microsoft Windows Server 2012) ID file a 128 bit.
    * Su file system in cui gli indici di file cambiano (ad esempio FAT32, exFAT) viene usato un algoritmo di fallback per adottare un approccio sensato in condizioni incerte quando la rotazione dei file di log produce più file di log con la stessa ora di ultima modifica.
  - I numeri inode, gli indici di file e i valori MD5 vengono raccolti internamente da Zabbix agent. Non vengono trasmessi a Zabbix server e vengono persi quando Zabbix agent viene arrestato.
  - Non modificare l'ultima ora di modifica di un file di log (ad esempio con `touch`) e non sostituire un file di log monitorato copiando un file nel suo nome originale (questo crea un nuovo inode). In entrambi i casi Zabbix può trattare il file come un file diverso e rileggerlo dall'inizio, con il rischio di generare avvisi duplicati.
  - Se esistono più file di log corrispondenti per l'item `logrt[]` e Zabbix agent sta seguendo il più recente di essi e questo file di log più recente viene eliminato, viene registrato un messaggio di avviso `"there are no files matching "<regexp mask>" in "<directory>"`. Zabbix agent ignora i file di log con un'ora di modifica inferiore all'ora di modifica più recente vista dall'agent per l'item `logrt[]` in verifica.
- L'agent inizia a leggere il file di log dal punto in cui si era fermato la volta precedente.
- Il numero di byte già analizzati (il contatore della dimensione) e l'ultima ora di modifica (il contatore del tempo) vengono memorizzati nel database di Zabbix e inviati all'agent per assicurare che l'agent inizi a leggere il file di log da questo punto nei casi in cui l'agent è appena stato avviato o ha ricevuto item che in precedenza erano disabilitati o non supportati. Tuttavia, se l'agent ha ricevuto un contatore della dimensione diverso da zero dal server, ma l'item `logrt[]` o `logrt.count[]` non riesce a trovare file corrispondenti, il contatore della dimensione viene reimpostato a 0 per analizzare dall'inizio se i file compaiono in seguito.
- Ogni volta che il file di log diventa più piccolo del contatore della dimensione del log noto all'agent, il contatore viene reimpostato a zero e l'agent inizia a leggere il file di log dall'inizio tenendo conto del contatore del tempo.
- Se nella directory sono presenti più file corrispondenti con la stessa ora di ultima modifica, l'agent cerca di analizzare correttamente tutti i file di log con la stessa ora di modifica ed evitare di saltare dati o analizzare gli stessi dati due volte, anche se ciò non può essere garantito in tutte le situazioni. L'agent non presume alcuno schema particolare di rotazione

dei file di log né ne determina uno. Quando vengono presentati più file di log con la stessa ora di ultima modifica, l'agent li elaborerà in ordine lessicografico decrescente. Pertanto, per alcuni schemi di rotazione i file di log verranno analizzati e segnalati nel loro ordine originale. Per altri schemi di rotazione l'ordine originale dei file di log non verrà rispettato, il che può portare a segnalare i record dei file di log corrispondenti in un ordine alterato (il problema non si verifica se i file di log hanno ore di ultima modifica diverse).

- Zabbix agent elabora i nuovi record di un file di log una volta ogni *Update interval* secondi.
- Zabbix agent non invia più di `maxlines` di un file di log al secondo. Il limite impedisce il sovraccarico delle risorse di rete e CPU e sostituisce il valore predefinito fornito dal parametro `MaxLinesPerSecond` nel file di configurazione dell'agent.
- Per trovare la stringa richiesta Zabbix elaborerà 10 volte più nuove righe rispetto a quanto impostato in `MaxLinesPerSecond`. Quindi, ad esempio, se un item `log[]` o `logrt[]` ha *Update interval* di 1 secondo, per impostazione predefinita l'agent analizzerà non più di 200 record del file di log e invierà non più di 20 record corrispondenti a Zabbix server in un singolo controllo. Aumentando `MaxLinesPerSecond` nel file di configurazione dell'agent o impostando il parametro `maxlines` nella chiave dell'item, il limite può essere aumentato fino a 10000 record del file di log analizzati e 1000 record corrispondenti inviati a Zabbix server in un singolo controllo. Se *Update interval* è impostato a 2 secondi, i limiti per un singolo controllo saranno impostati a un valore 2 volte superiore rispetto a *Update interval* di 1 secondo.
- Inoltre, i valori `log` e `log.count` sono sempre limitati al 50% della dimensione del buffer di invio dell'agent, anche se non vi sono valori non di log al suo interno. Quindi, affinché i valori `maxlines` vengano inviati in una singola connessione (e non in più connessioni), il parametro `BufferSize` di Zabbix agent deve essere almeno `maxlines` x 2. Zabbix agent può caricare i dati durante la raccolta dei log e quindi liberare il buffer, mentre Zabbix agent 2 interromperà la raccolta dei log fino a quando i dati non saranno caricati e il buffer non sarà liberato, operazione eseguita in modo asincrono.
- In assenza di item di log, l'intera dimensione del buffer dell'agent viene usata per i valori non di log. Quando arrivano valori di log, essi sostituiscono i valori non di log più vecchi secondo necessità, fino al 50% designato.
- Per record di file di log più lunghi di 256kB, solo i primi 256kB vengono confrontati con l'espressione regolare e il resto del record viene ignorato. Tuttavia, se Zabbix agent viene arrestato mentre sta gestendo un record lungo, lo stato interno dell'agent viene perso e il record lungo può essere analizzato di nuovo e in modo diverso dopo che l'agent viene riavviato.
- Nota speciale per i separatori di percorso `\`: se `file_format` è `file\.log`, allora non dovrebbe esserci una directory `file`, poiché non è possibile definire in modo univoco se `.` sia escapato oppure sia il primo simbolo del nome del file.
- Le espressioni regolari per `logrt` sono supportate solo nel nome del file, la corrispondenza con espressioni regolari per la directory non è supportata.
- Su piattaforme UNIX un item `logrt[]` diventa NOTSUPPORTED se non esiste una directory in cui ci si aspetta di trovare i file di log.
- Su Microsoft Windows, se una directory non esiste l'item non diventerà NOTSUPPORTED (ad esempio, se la directory è scritta in modo errato nella chiave dell'item).
- L'assenza di file di log per l'item `logrt[]` non lo rende NOTSUPPORTED. Gli errori di lettura dei file di log per l'item `logrt[]` vengono registrati come avvisi nel file di log di Zabbix agent ma non rendono l'item NOTSUPPORTED.
- Il file di log di Zabbix agent può essere utile per capire perché un item `log[]` o `logrt[]` è diventato NOTSUPPORTED. Zabbix può monitorare il file di log del proprio agent, tranne quando `DebugLevel=4` o `DebugLevel=5`.
- La ricerca di un punto interrogativo tramite un'espressione regolare, ad esempio `\?`, può produrre falsi positivi se il file di testo contiene simboli NUL, poiché questi vengono sostituiti con `?` da Zabbix per continuare l'elaborazione della riga fino al carattere di nuova riga.

Estrazione della parte corrispondente di un'espressione regolare

A volte può essere necessario estrarre solo il valore di interesse da un file di destinazione, invece di restituire l'intera riga quando viene trovata una corrispondenza con un'espressione regolare.

Gli item di log consentono di estrarre i valori desiderati dalle righe corrispondenti. Questa operazione viene eseguita tramite il parametro aggiuntivo **output** negli item `log` e `logrt`.

L'utilizzo del parametro 'output' consente di indicare il "gruppo di cattura" della corrispondenza a cui siamo interessati.

Ad esempio:

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

dovrebbe consentire di restituire il numero di voci rilevato nel contenuto di:

```
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
```

Verrà restituito solo il numero, perché `\1` fa riferimento al primo e unico gruppo di cattura: `([0-9]+)`.

Inoltre, grazie alla possibilità di estrarre e restituire un numero, il valore può essere utilizzato per definire i trigger.

Utilizzo del parametro maxdelay

Il parametro `maxdelay` negli item di log consente di ignorare alcune righe più vecchie dei file di log per analizzare entro i secondi

specificati da maxdelay le righe più recenti.

Warning:
Specificare maxdelay > 0 può portare a **ignorare record importanti del file di log e a perdere avvisi**. Usalo con cautela e a tuo rischio solo quando necessario.

Per impostazione predefinita, gli item per il monitoraggio dei log seguono tutte le nuove righe che compaiono nei file di log. Tuttavia, esistono applicazioni che in alcune situazioni iniziano a scrivere un numero enorme di messaggi nei propri file di log. Ad esempio, se un database o un server DNS non è disponibile, tali applicazioni inondano i file di log con migliaia di messaggi di errore quasi identici finché il normale funzionamento non viene ripristinato. Per impostazione predefinita, tutti questi messaggi verranno diligentemente analizzati e le righe corrispondenti inviate al server come configurato negli item log e logrt.

La protezione integrata contro il sovraccarico consiste in un parametro configurabile maxlines (protegge il server da troppe righe di log corrispondenti in ingresso) e in un limite pari a 10*maxlines (protegge la CPU e l'I/O dell'host dal sovraccarico causato da agent in un singolo controllo). Tuttavia, ci sono ancora 2 problemi con la protezione integrata. Primo, al server viene segnalato un gran numero di messaggi potenzialmente poco informativi, che occupano spazio nel database. Secondo, a causa del numero limitato di righe analizzate al secondo, agent può rimanere indietro rispetto ai record di log più recenti per ore. Molto probabilmente, preferirai essere informato prima sulla situazione attuale nei file di log invece di scorrere per ore vecchi record.

La soluzione a entrambi i problemi è l'uso del parametro maxdelay. Se viene specificato maxdelay > 0, durante ogni controllo vengono misurati il numero di byte elaborati, il numero di byte rimanenti e il tempo di elaborazione. A partire da questi numeri, agent calcola un ritardo stimato: quanti secondi servirebbero per analizzare tutti i record rimanenti in un file di log.

Se il ritardo non supera maxdelay, agent procede con l'analisi del file di log come di consueto.

Se il ritardo è maggiore di maxdelay, agent **ignora un blocco di un file di log "saltandolo"** fino a una nuova posizione stimata, in modo che le righe rimanenti possano essere analizzate entro maxdelay secondi.

Nota che agent non legge nemmeno le righe ignorate nel buffer, ma calcola una posizione approssimativa a cui saltare nel file.

Il fatto di saltare righe del file di log viene registrato nel file di log di agent in questo modo:

```
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
```

Il numero "to byte" è approssimativo perché dopo il "salto" agent adatta la posizione nel file all'inizio di una riga di log, che può trovarsi più avanti nel file oppure prima.

A seconda di come la velocità di crescita si confronta con la velocità di analisi del file di log, potresti non vedere alcun "salto", vedere "salti" rari o frequenti, "salti" grandi o piccoli, oppure persino un piccolo "salto" a ogni controllo. Le fluttuazioni nel carico del sistema e la latenza di rete influenzano anche il calcolo del ritardo e quindi il "salto" in avanti per tenere il passo con il parametro maxdelay.

Impostare maxdelay < update interval non è consigliato (potrebbe causare frequenti piccoli "salti").

Note sulla gestione della rotazione dei file di log 'copytruncate'

logrt con l'opzione copytruncate presuppone che file di log diversi abbiano record diversi (almeno i loro timestamp sono diversi), pertanto le somme MD5 dei blocchi iniziali (fino ai primi 512 byte) saranno diverse. Due file con le stesse somme MD5 dei blocchi iniziali significano che uno di essi è l'originale e l'altro è una copia.

logrt con l'opzione copytruncate si impegna a elaborare correttamente le copie dei file di log senza segnalare duplicati. Tuttavia, non sono consigliate situazioni come la creazione di più copie dello stesso file di log con lo stesso timestamp, la rotazione del file di log più frequente dell'intervallo di aggiornamento dell'item logrt[], o riavvii frequenti di agent. L'agent cerca di gestire tutte queste situazioni in modo ragionevole, ma non è possibile garantire risultati ottimali in tutte le circostanze.

Note sui file persistenti per gli item log*[]

Scopo dei file persistenti

Quando Zabbix agent viene avviato, riceve un elenco di controlli attivi da Zabbix server o proxy. Per le metriche log*[] riceve la dimensione del log elaborato e l'ora di modifica per determinare da dove iniziare il monitoraggio del file di log. A seconda della dimensione effettiva del file di log e dell'ora di modifica segnalate dal file system, l'agent decide se continuare il monitoraggio del file di log dalla dimensione del log elaborato oppure riesaminare il file di log dall'inizio.

Un agent in esecuzione mantiene un insieme più ampio di attributi per tenere traccia di tutti i file di log monitorati tra un controllo e l'altro. Questo stato in memoria viene perso quando l'agent viene arrestato.

Il nuovo parametro opzionale `persistent_dir` specifica una directory in cui archiviare questo stato dell'item `log[]`, `log.count[]`, `logrt[]` o `logrt.count[]` in un file. Lo stato dell'item di log viene ripristinato dal file persistente dopo il riavvio di Zabbix agent.

Il caso d'uso principale è il monitoraggio di un file di log situato su un file system mirrorato. Fino a un certo momento il file di log viene scritto su entrambi i mirror. Poi i mirror vengono separati. Sulla copia attiva il file di log continua a crescere, ricevendo nuovi record. Zabbix agent lo analizza e invia a server la dimensione del log elaborato e l'ora di modifica. Sulla copia passiva il file di log rimane invariato, molto indietro rispetto alla copia attiva. In seguito il sistema operativo e Zabbix agent vengono riavviati dalla copia passiva. La dimensione del log elaborato e l'ora di modifica che Zabbix agent riceve da server potrebbero non essere validi per la situazione sulla copia passiva. Per continuare il monitoraggio del file di log dal punto in cui l'agent si era fermato al momento della separazione del mirror del file system, l'agent ripristina il proprio stato dal file persistente.

Operazione dell'agent con file persistente

All'avvio, Zabbix agent non sa nulla dei file persistenti. Solo dopo aver ricevuto un elenco di controlli attivi da Zabbix server (proxy), l'agent vede che alcuni item di log devono essere supportati da file persistenti nelle directory specificate.

Durante il funzionamento dell'agent, i file persistenti vengono aperti in scrittura (con `fopen(filename, "w")`) e sovrascritti con i dati più recenti. La probabilità di perdere i dati del file persistente se la sovrascrittura e la divisione del mirror del file system avvengono contemporaneamente è molto bassa; non è prevista alcuna gestione speciale per questo caso. La scrittura nel file persistente NON è seguita da una sincronizzazione forzata con il supporto di memorizzazione (`fsync()` non viene chiamata).

La sovrascrittura con i dati più recenti viene eseguita dopo la segnalazione riuscita a Zabbix server del record del file di log corrispondente o dei metadati (dimensione del log elaborata e ora di modifica). Ciò può accadere anche a ogni controllo dell'item se il file di log continua a cambiare.

Nessuna azione speciale durante l'arresto dell'agent.

Dopo aver ricevuto un elenco di controlli attivi, l'agent contrassegna i file persistenti obsoleti per la rimozione. Un file persistente diventa obsoleto se:

1. Il corrispondente item di log non è più monitorato.
2. Un item di log viene riconfigurato con una posizione `persistent_dir` diversa rispetto a prima.

La rimozione viene eseguita con un ritardo di 24 ore perché i file di log nello stato NOTSUPPORTED non sono inclusi nell'elenco dei controlli attivi, ma potrebbero diventare SUPPORTED in seguito e i loro file persistenti saranno utili.

Se l'agent viene arrestato prima che siano trascorse 24 ore, i file obsoleti non verranno eliminati, poiché Zabbix agent non riceve più informazioni sulla loro posizione da Zabbix server.

Warning:
Riconfigurare `persistent_dir` di un item di log tornando alla vecchia posizione `persistent_dir` mentre l'agent è arrestato, senza eliminare il vecchio file persistente da parte dell'utente, causerà il ripristino dello stato dell'agent dal vecchio file persistente, con conseguenti messaggi mancati o falsi allarmi.

Denominazione e posizione dei file persistenti

Zabbix agent distingue i controlli attivi in base alle relative chiavi. Ad esempio, `logrt[/home/zabbix/test.log]` e `logrt[/home/zabbix/test.log,]` sono item diversi. La modifica dell'item `logrt[/home/zabbix/test.log,,,10]` nel frontend in `logrt[/home/zabbix/test.log,,,20]` comporterà l'eliminazione dell'item `logrt[/home/zabbix/test.log,,,10]` dall'elenco dei controlli attivi dell'agent e la creazione dell'item `logrt[/home/zabbix/test.log,,,20]` (alcuni attributi vengono mantenuti durante la modifica nel frontend/server, ma non nell'agent).

Il nome del file è composto dalla somma MD5 della chiave dell'item, a cui viene aggiunta la lunghezza della chiave dell'item per ridurre la possibilità di collisioni. Ad esempio, lo stato dell'item `logrt[/home/zabbix50/test.log,,,,,,,,/home/zabbix50/agent_pr` verrà conservato nel file persistente c963ade4008054813bbc0a650bb8e09266.

Più item di tipo log possono utilizzare lo stesso valore di `persistent_dir`.

`persistent_dir` deve essere specificato tenendo conto della struttura specifica del file system, dei punti di montaggio e delle opzioni di montaggio, nonché della configurazione del mirroring dello storage: il file persistente deve trovarsi nello stesso file system sottoposto a mirroring del file di log monitorato.

Se la directory `persistent_dir` non può essere creata o non esiste, oppure se Zabbix agent non dispone di autorizzazioni sufficienti per creare, scrivere, leggere o eliminare file, l'item di tipo log diventa `NOTSUPPORTED`.

Se i diritti di accesso ai file di archiviazione persistente vengono rimossi durante il funzionamento dell'agent o si verificano altri errori (ad esempio, il disco è pieno), gli errori vengono registrati nel file di log dell'agent, ma l'item di log non diventa `NOTSUPPORTED`.

Carico su I/O

Il file persistente dell'item viene aggiornato dopo l'invio riuscito di ogni batch di dati (contenente i dati dell'item) al server. Ad esempio, il valore predefinito di `BufferSize` è 100. Se un item di log ha trovato 70 record corrispondenti, i primi 50 record verranno inviati in un unico batch, il file persistente verrà aggiornato, quindi i restanti 20 record verranno inviati (forse con un certo ritardo quando vengono accumulati più dati) nel secondo batch, e il file persistente verrà aggiornato di nuovo.

Azioni se la comunicazione fallisce tra agent e server

Ogni riga corrispondente di un item `log[]` e `logrt[]` e il risultato di ogni controllo di un item `log.count[]` e `logrt.count[]` richiedono uno slot libero nell'area dedicata al 50% nel buffer di invio dell'agent. Gli elementi del buffer vengono inviati regolarmente al server (o proxy) e gli slot del buffer tornano liberi.

Finché ci sono slot liberi nell'area log dedicata nel buffer di invio dell'agent e la comunicazione tra agent e server (o proxy) fallisce, i risultati del monitoraggio dei log vengono accumulati nel buffer di invio. Questo aiuta a mitigare brevi interruzioni di comunicazione.

Durante interruzioni di comunicazione più lunghe tutti gli slot log vengono occupati e vengono eseguite le azioni seguenti:

- I controlli degli item `log[]` e `logrt[]` vengono interrotti. Quando la comunicazione viene ripristinata e sono disponibili slot liberi nel buffer, i controlli riprendono dal punto precedente. Nessuna riga corrispondente viene persa, viene solo segnalata più tardi.
- I controlli `log.count[]` e `logrt.count[]` vengono interrotti se `maxdelay = 0` (predefinito). Il comportamento è simile a quello degli item `log[]` e `logrt[]` descritto sopra. Si noti che questo può influire sui risultati di `log.count[]` e `logrt.count[]`: ad esempio, un controllo conta 100 righe corrispondenti in un file di log, ma poiché non ci sono slot liberi nel buffer il controllo viene interrotto. Quando la comunicazione viene ripristinata, l'agent conta le stesse 100 righe corrispondenti e anche 70 nuove righe corrispondenti. L'agent ora invia count = 170 come se fossero state trovate in un unico controllo.
- I controlli `log.count[]` e `logrt.count[]` con `maxdelay > 0`: se durante il controllo non si è verificato alcun "salto", il comportamento è simile a quello descritto sopra. Se si è verificato un "salto" oltre le righe del file di log, viene mantenuta la posizione dopo il "salto" e il risultato conteggiato viene scartato. In questo modo, l'agent cerca di tenere il passo con un file di log in crescita anche in caso di guasto della comunicazione.

Gestione degli errori di compilazione e di esecuzione delle espressioni regolari

Se un'espressione regolare utilizzata in un item `log[]`, `logrt[]`, `log.count[]` o `logrt.count[]` non può essere compilata dalla libreria PCRE o PCRE2, l'item passa allo stato NON SUPPORTATO con un messaggio di errore. Per continuare a monitorare l'item di log, è necessario correggere l'espressione regolare.

Se l'espressione regolare viene compilata correttamente, ma non riesce durante l'esecuzione (su alcuni o su tutti i record di log), l'item di log rimane supportato e il monitoraggio continua. L'errore di esecuzione viene registrato nel file di log dell'agent Zabbix (senza il record del file di log).

La frequenza di registrazione è limitata a un errore di esecuzione per controllo, per consentire all'agent Zabbix di monitorare il proprio file di log. Ad esempio, se vengono analizzati 10 record e 3 di questi generano un errore di esecuzione dell'espressione regolare, nel log dell'agent viene prodotto un solo record.

Eccezione: se `MaxLinesPerSecond=1` e `update interval=1` (consentendo di analizzare un solo record per controllo), gli errori di esecuzione delle espressioni regolari non vengono registrati.

zabbix_agentd registra la chiave dell'item in caso di errore di esecuzione, mentre zabbix_agent2 registra l'ID dell'item per facilitare l'identificazione dell'item di log che presenta errori di esecuzione. In caso di errori di esecuzione, si consiglia di riprogettare l'espressione regolare.

Controllo semplice

Panoramica

I controlli semplici vengono normalmente utilizzati per controlli remoti senza agent dei servizi.

Si noti che Zabbix agent non è necessario per i controlli semplici. Zabbix server/proxy è responsabile dell'elaborazione dei controlli semplici (effettuando connessioni esterne, ecc.).

Esempi di utilizzo dei controlli semplici:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
I campi *User name* e *Password* (limitati a 255 caratteri) nella configurazione degli item di controllo semplice sono utilizzati per gli item di monitoraggio VMware; altrimenti vengono ignorati.

Controlli supportati

Le chiavi degli item sono elencate senza parametri opzionali e informazioni aggiuntive. Fare clic sulla chiave dell'item per visualizzare i dettagli completi.

Vedi anche chiavi degli item per il monitoraggio VMware.

Item key	Description
icmpping	L'accessibilità dell'host tramite ping ICMP.
icmppingloss	La percentuale di pacchetti persi.
icmppingretry	L'accessibilità dell'host tramite ping ICMP con tentativi di ritrasmissione.
icmppingsec	Il tempo di risposta del ping ICMP.
net.tcp.service	Verifica se un servizio è in esecuzione e accetta connessioni TCP.
net.tcp.service.perf	Verifica le prestazioni di un servizio TCP.
net.udp.service	Verifica se un servizio è in esecuzione e risponde alle richieste UDP.
net.udp.service.perf	Verifica le prestazioni di un servizio UDP.

Dettagli della chiave item

I parametri senza parentesi angolari sono obbligatori. I parametri contrassegnati con parentesi angolari **< >** sono facoltativi.

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 La raggiungibilità dell'host tramite ping ICMP.
 Valore restituito: *0* - il ping ICMP non riesce; *1* - il ping ICMP riesce.

Parametri:

- **target** - l'indirizzo IP o il nome DNS dell'host;
- **packets** - il numero di pacchetti;
- **interval** - il tempo tra pacchetti successivi in millisecondi;
- **size** - la dimensione del pacchetto in byte;
- **timeout** - il timeout in millisecondi;
- **options** - usato per consentire il redirect: se vuoto (valore predefinito), le risposte reindirizzate sono considerate come host di destinazione non raggiungibile; se impostato su *allow_redirect*, le risposte reindirizzate sono considerate come host di destinazione raggiungibile.

Vedere anche la tabella dei valori predefiniti.

Commento:

- Se viene restituito almeno uno dei pacchetti definiti dal parametro `packets`, l'item restituisce 1. Se nessuno dei pacchetti viene restituito, l'item restituisce 0.

Esempio:

```
icmpping[,4] #Se viene restituito almeno uno dei quattro pacchetti, l'item restituisce 1.
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 La percentuale di pacchetti persi.
 Valore restituito: *Float*.

Parametri:

- **target** - l'IP del host o il nome DNS;
- **packets** - il numero di pacchetti;
- **interval** - il tempo tra pacchetti successivi in millisecondi;
- **size** - la dimensione del pacchetto in byte;
- **timeout** - il timeout in millisecondi;
- **options** - utilizzato per consentire il reindirizzamento: se vuoto (valore predefinito), le risposte reindirizzate vengono trattate come host di destinazione non disponibile; se impostato su *allow_redirect*, le risposte reindirizzate vengono trattate come host di destinazione disponibile.

Vedere anche la tabella dei valori predefiniti.

icmppingretry[<target>,<retries>,<backoff>,<size>,<timeout>,<options>]

 L'accessibilità del host tramite ping ICMP con tentativi ripetuti. Se il primo pacchetto ha esito positivo, l'operazione si interrompe; se il pacchetto fallisce, il tentativo viene ripetuto fino al raggiungimento del numero massimo di tentativi definito nel parametro `retries`. Questo item è utile per ridurre il numero di pacchetti inviati sulla rete.
 Valore restituito: *0* - ping ICMP non riuscito; *1* - ping ICMP riuscito.

Parametri:

- **target** - l'IP del host o il nome DNS;
- **retries** - il numero di volte in cui verrà effettuato un tentativo di ping verso una destinazione, escluso il primo tentativo (0 o maggiore; predefinito 1);
- **backoff** - il numero per cui il tempo di attesa viene moltiplicato a ogni richiesta successiva (intervallo 1.0-5.0; predefinito 1.0);
- **size** - la dimensione del pacchetto in byte;
- **timeout** - il timeout in millisecondi;
- **options** - usato per consentire il reindirizzamento: se vuoto (valore predefinito), le risposte reindirizzate vengono trattate come host di destinazione non disponibile; se impostato su *allow_redirect*, le risposte reindirizzate vengono trattate come host di destinazione disponibile.

Vedere anche la tabella dei valori predefiniti.

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 Il tempo di risposta del ping ICMP (in secondi).
 Valore restituito: *Float*.

Parametri:

- **target** - l'IP del host o il nome DNS;
- **packets** - il numero di pacchetti;
- **interval** - il tempo tra pacchetti successivi in millisecondi;
- **size** - la dimensione del pacchetto in byte;
- **timeout** - il timeout in millisecondi;
- **mode** - valori possibili: *min*, *max* oppure *avg* (predefinito);
- **options** - usato per consentire il reindirizzamento: se vuoto (valore predefinito), le risposte reindirizzate sono trattate come host di destinazione non disponibile; se impostato su *allow_redirect*, le risposte reindirizzate sono trattate come host di destinazione disponibile.

Commenti:

- I pacchetti persi o scaduti per timeout non vengono utilizzati nel calcolo;
- Se il host non è disponibile (timeout raggiunto), l'item restituirà 0;
- Se il valore restituito è inferiore a 0.0001 secondi, il valore verrà impostato a 0.0001 secondi;
- Vedere anche la tabella dei valori predefiniti.

net.tcp.service[service,<ip>,<port>]

 Verifica se un servizio è in esecuzione e accetta connessioni TCP.
 Valore restituito: *0* - il servizio non è disponibile; *1* - il servizio è in esecuzione.

Parametri:

- **service** - valori possibili: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (vedere i dettagli);
- **ip** - l'indirizzo IP o il nome DNS (per impostazione predefinita viene utilizzato l'IP/DNS del host);
- **port** - il numero di porta (per impostazione predefinita viene utilizzato il numero di porta standard del servizio).

Commenti:

- Si noti che con il servizio *tcp* è obbligatorio indicare la porta;
- Questi controlli possono generare messaggi aggiuntivi nei file di log dei demoni di sistema (di solito vengono registrate le sessioni SMTP e SSH);
- Il controllo dei protocolli cifrati (come IMAP sulla porta 993 o POP sulla porta 995) non è attualmente supportato. Come soluzione alternativa, utilizzare `net.tcp.service[tcp,<ip>,port]` per controlli di questo tipo.

Esempio:

```
net.tcp.service[ftp,,45] #Questo item può essere utilizzato per testare la disponibilità del server FTP su
```

Attention:
Se SELinux è in esecuzione in modalità enforced, i controlli semplici TCP/UDP personalizzati potrebbero essere bloccati dalla policy. Per verificare e consentire la nuova connessione in uscita, esaminare i rifiuti di audit: con `grep denied /var/log/audit/audit.log`

net.tcp.service.perf[service,<ip>,<port>]

 Controlla le prestazioni di un servizio TCP.
 Valore restituito: *Float*: *0.000000* - il servizio non è disponibile; *secondi* - il numero di secondi impiegati per connettersi al servizio.

Parametri:

- **service** - valori possibili: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (vedere i dettagli);
- **ip** - l'indirizzo IP o il nome DNS (per impostazione predefinita viene utilizzato l'IP/DNS del host);
- **port** - il numero di porta (per impostazione predefinita viene utilizzato il numero di porta standard del servizio).

Commenti:

- Si noti che con il servizio *tcp* è obbligatorio indicare la porta;
- Il controllo dei protocolli cifrati (come IMAP sulla porta 993 o POP sulla porta 995) al momento non è supportato. Come soluzione alternativa, utilizzare `net.tcp.service[tcp,<ip>,port]` per controlli di questo tipo.

Esempio:

```
net.tcp.service.perf[ssh] #Questo item può essere utilizzato per testare la velocità della risposta inizia
```

net.udp.service[service,<ip>,<port>]

 Controlla se un servizio è in esecuzione e risponde alle richieste UDP.
 Valore restituito: *0* - il servizio non è disponibile; *1* - il servizio è in esecuzione.

Parametri:

- **service** - valori possibili: *ntp* (vedere i dettagli);
- **ip** - l'indirizzo IP o il nome DNS (per impostazione predefinita viene utilizzato l'IP/DNS dell'host);
- **port** - il numero di porta (per impostazione predefinita viene utilizzato il numero di porta standard del servizio).

Esempio:

```
net.udp.service[ntp,,45] #Questo item può essere utilizzato per testare la disponibilità del servizio NTP
```

net.udp.service.perf[service,<ip>,<port>]

 Controlla le prestazioni di un servizio UDP.
 Valore restituito: *Float*: *0.000000* - il servizio non è disponibile; *secondi* - il numero di secondi trascorsi in attesa della risposta dal servizio.

Parametri:

- **service** - valori possibili: *ntp* (vedere i dettagli);
- **ip** - l'indirizzo IP o il nome DNS (per impostazione predefinita viene utilizzato l'IP/DNS del host);
- **port** - il numero di porta (per impostazione predefinita viene utilizzato il numero di porta standard del servizio).

Esempio:

```
net.udp.service.perf[ntp] #Questo item può essere utilizzato per testare il tempo di risposta del servizio
```

Attention:
Per il supporto di SourceIP nei controlli LDAP semplici (ad esempio, `net.tcp.service[ldap]`), è richiesta la versione 2.6.1 o successiva di OpenLDAP.

Elaborazione del timeout

I timeout flessibili degli item, sebbene siano supportati per i controlli semplici, non si applicano agli item `icmpping*` e VMware. Vedere Supporto per timeout flessibili.

Ping ICMP

Zabbix utilizza un'utility esterna, **fping**, per elaborare i ping ICMP (**icmpping**, **icmppingloss**, **icmppingretry**, **icmppingsec**).

Installazione

fping non è incluso con Zabbix e deve essere installato separatamente:

- Diverse piattaforme basate su Unix includono il pacchetto fping nei rispettivi repository predefiniti, ma non è preinstallato. In questo caso, è possibile utilizzare il gestore di pacchetti per installare fping.
- Zabbix fornisce pacchetti fping per RHEL e le sue derivate. Si noti che questi pacchetti sono forniti senza supporto ufficiale.
- fping può anche essere compilato dal codice sorgente.

Configurazione

Specifica la posizione di fping nel parametro *FpingLocation* del file di configurazione di Zabbix server/proxy (o nel parametro *Fping6Location* per l'uso di indirizzi IPv6).

fping deve essere eseguibile dall'utente con cui viene eseguito Zabbix server/proxy e questo utente deve avere diritti sufficienti.

Vedi anche: Problemi noti per l'elaborazione dei simple check con versioni di fping inferiori alla 3.10.

Valori predefiniti

Valori predefiniti, limiti e descrizione dei parametri dei controlli ICMP:

Parametro	Unità	Descrizione	Flag di fping	Valori predefiniti impostati da		Limiti consentiti da Zabbix	
				fping	**Zabbix**	**min**	**max**
packets	numero	Numero di pacchetti di richiesta inviati a una destinazione	-C		3	1	10000
interval	millisecondi	Tempo di attesa tra pacchetti successivi inviati a una singola destinazione	-p	1000		20	illimitato
size	byte	Dimensione del pacchetto in byte 56 byte su x86, 68 byte su x86_64	-b	56 o 68		24	65507
timeout	millisecondi	**fping v3.x** - timeout di attesa dopo l'invio dell'ultimo pacchetto, influenzato dal flag `-C` **fping v4.x** - timeout individuale per ogni pacchetto	-t	**fping v3.x** - 500 **fping v4.x** e versioni successive - ereditato dal flag `-p`, ma non superiore a 2000		50	illimitato
retries	numero	Numero di volte in cui verrà effettuato un tentativo di eseguire il ping di una destinazione, escluso il primo tentativo	-r	3	1	0	illimitato
backoff factor	numero	Numero per il quale il tempo di attesa viene moltiplicato a ogni richiesta successiva	-B	1.5	1.0	1.0	5.0

I valori predefiniti possono differire leggermente a seconda della piattaforma e della versione.

Inoltre, Zabbix utilizza le opzioni di fping `-i interval ms` (da non confondere con il parametro `interval` dell'item menzionato nella tabella precedente, che corrisponde all'opzione `-p` di fping) e `-S source IP address` (oppure `-I` nelle versioni precedenti di fping). Queste opzioni vengono rilevate automaticamente eseguendo controlli con diverse combinazioni di opzioni. Zabbix tenta di rilevare il valore minimo in millisecondi consentito da fping per l'opzione `-i`, provando tre valori: `0`, `1` e `10`. Il primo valore che ha esito positivo viene quindi utilizzato per i successivi controlli ICMP. Questo processo viene eseguito individualmente da ogni processo ICMP pinger.

Le opzioni di fping rilevate automaticamente vengono invalidate ogni ora e rilevate nuovamente al tentativo successivo di eseguire un controllo ICMP. Impostare DebugLevel>=4 per visualizzare i dettagli di questo processo nel file di log del server o del proxy.

Zabbix scrive in un file temporaneo gli indirizzi IP da controllare con una qualsiasi delle chiavi `icmpping*`, quindi passa il file a fping. Se gli item hanno parametri di chiave diversi, in un singolo file vengono scritti solo quelli con parametri di chiave identici. Tutti gli indirizzi IP scritti nel singolo file vengono controllati da fping in parallelo, pertanto il processo ICMP pinger di Zabbix impiegherà una quantità di tempo fissa, indipendentemente dal numero di indirizzi IP presenti nel file.

Chiavi degli item per il monitoraggio VMware

L'elenco del monitoraggio VMware elementi chiave è stato spostato nella sezione monitoraggio VMware.

agent SNMP

Panoramica

Potresti voler utilizzare il monitoraggio SNMP su dispositivi come stampanti, switch di rete, router o UPS, che di solito sono abilitati per SNMP e sui quali sarebbe impraticabile tentare di configurare sistemi operativi completi e agent Zabbix.

Per poter recuperare i dati forniti dagli agent SNMP su questi dispositivi, il server Zabbix deve essere configurato inizialmente con il supporto SNMP specificando il flag `--with-net-snmp`. Si consiglia inoltre di installare i file MIB per garantire che i valori degli item vengano visualizzati nel formato corretto. Senza i file MIB, possono verificarsi problemi di formattazione, ad esempio la visualizzazione dei valori in HEX invece che in UTF-8 o viceversa.

I controlli SNMP vengono eseguiti solo tramite il protocollo UDP.

I daemon di server Zabbix e proxy registrano righe simili alla seguente se ricevono una risposta SNMP non corretta:

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Anche se non coprono tutti i casi problematici, sono utili per identificare i singoli dispositivi SNMP per i quali le richieste combinate dovrebbero essere disabilitate.

Il server/proxy Zabbix ritenterà fino a 5 volte per gli item SNMP `walk` e `get`. Il meccanismo di retry non si applica ai fallimenti di risoluzione DNS.

Per i controlli SNMP legacy (singolo numero OID o stringa), il server/proxy Zabbix ritenterà almeno una volta dopo un tentativo di query non riuscito: tramite il meccanismo di retry della libreria SNMP oppure tramite il meccanismo interno di elaborazione combinata.

Warning:
Se si monitornano dispositivi SNMPv3, assicurarsi che msgAuthoritativeEngineID (noto anche come snmpEngineID o "Engine ID") non sia mai condiviso da due dispositivi. Secondo RFC 2571 (sezione 3.1.1.1) deve essere univoco per ciascun dispositivo.

Warning:
RFC3414 richiede che i dispositivi SNMPv3 mantengano persistenti i propri engineBoots. Alcuni dispositivi non lo fanno, e ciò comporta che i loro messaggi SNMP vengano scartati come obsoleti dopo il riavvio. In tale situazione, la cache SNMP deve essere svuotata manualmente su un server/proxy (utilizzando -R snmp_cache_reload) oppure il server/proxy deve essere riavviato.

Note:
Zabbix memorizza nella cache le mappature SNMPv3 EngineID → IP e riutilizza gli EngineID memorizzati per i controlli successivi invece di inviare ogni volta un probe, riducendo il traffico di rete. Se un EngineID non può essere riutilizzato, verrà eseguito un retry con un probe per individuare il nuovo EngineID.

Configurazione del monitoraggio SNMP

Per iniziare a monitorare un dispositivo tramite SNMP, è necessario eseguire i seguenti passaggi:

Passo 1

Individua la stringa SNMP (o l'OID) dell'item che vuoi monitorare.

Per ottenere un elenco di stringhe SNMP, usa il comando `snmpwalk` (parte del software net-snmp, che dovresti aver installato come parte dell'installazione di Zabbix) oppure uno strumento equivalente:

```
snmpwalk -v 2c -c public <host IP> .
```

Poiché 2c qui indica la versione SNMP, puoi anche sostituirlo con 1, per indicare la versione SNMP 1 sul dispositivo.

Questo dovrebbe restituire un elenco di stringhe SNMP e del loro ultimo valore. Se non accade, è possibile che la "community" SNMP sia diversa da quella standard `public`; in tal caso dovrai scoprirla.

Puoi quindi scorrere l'elenco finché non trovi la stringa che vuoi monitorare, ad esempio: se volessi monitorare i byte in ingresso sul tuo switch sulla porta 3, useresti la stringa `IF-MIB::ifHCInOctets.3` da questa riga:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Ora puoi usare il comando `snmpget` per trovare l'OID numerico di `IF-MIB::ifHCInOctets.3`:

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Nota che l'ultimo numero nella stringa è il numero della porta che vuoi monitorare. Vedi anche: Indici dinamici.

Questo dovrebbe restituirti qualcosa di simile al seguente:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

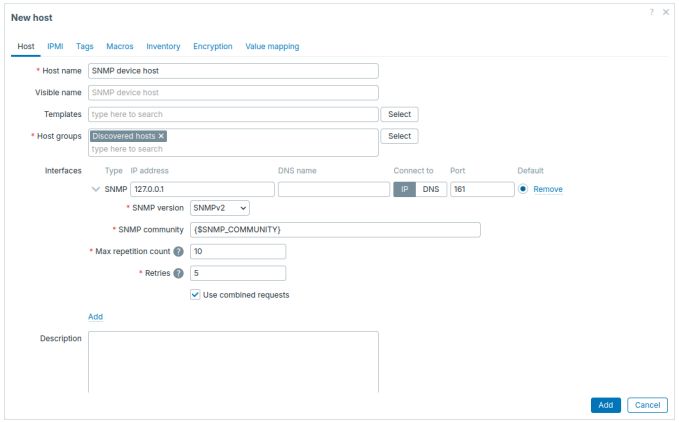
Anche in questo caso, l'ultimo numero nell'OID è il numero della porta.

Note:
Alcuni degli OID SNMP più utilizzati vengono tradotti automaticamente in una rappresentazione numerica da Zabbix.

Nell'ultimo esempio sopra, il tipo di valore è "Counter64", che internamente corrisponde al tipo ASN_COUNTER64. L'elenco completo dei tipi supportati è 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 e ASN_OBJECT_ID. Questi tipi corrispondono approssimativamente a "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" nell'output di **snmpget**, ma possono anche essere visualizzati come "STRING", "Hex-STRING", "OID" e altri, a seconda della presenza di un display hint.

Passo 2

Creare un host corrispondente a un dispositivo.



Aggiungere un'interfaccia SNMP per l'host:

1. Immettere l'indirizzo IP/nome DNS e il numero di porta.
2. Selezionare la *versione SNMP* dal menu a discesa.
3. Aggiungere le credenziali dell'interfaccia in base alla versione SNMP selezionata:
   - SNMPv1, v2 richiedono solo la community (di solito 'public').
   - SNMPv3 richiede opzioni più specifiche; vedere SNMPv3.
4. Specificare il valore massimo di ripetizione (predefinito: 10) per le richieste bulk SNMP native (GetBulkRequest-PDU); solo per gli item `discovery[]` e `walk[]` in SNMPv2 e v3. Si noti che impostare questo valore troppo alto può causare il timeout del controllo dell'agent SNMP.
5. Selezionare la casella *Use combined requests* per consentire l'elaborazione combinata delle richieste SNMP (non correlata alle richieste bulk SNMP native "walk" e "get").

È possibile usare uno dei template SNMP forniti, che aggiungerà automaticamente un insieme di item. Prima di usare un template, verificare che sia compatibile con l'host.

Fare clic su *Add* per salvare l'host.

SNMPv3

I seguenti parametri sono richiesti per SNMPv3:

- *Context name* - Immettere il nome del contesto per identificare l'item nella subnet SNMP.
I macro utente vengono risolti in questo campo.
- *Security name* - Immettere il nome di sicurezza.
I macro utente vengono risolti in questo campo.
- *Security level* - Selezionare il livello di sicurezza:
  - *noAuthNoPriv* - non vengono utilizzati protocolli di autenticazione né di privacy;
  - *AuthNoPriv* - viene utilizzato il protocollo di autenticazione, non quello di privacy;

    - *AuthPriv* - vengono utilizzati entrambi i protocolli di autenticazione e di privacy.
- *Authentication protocol* - Selezionare il protocollo di autenticazione: *MD5*, *SHA1*; con net-snmp 5.8 e versioni successive *SHA224*, *SHA256*, *SHA384* o *SHA512*.
- *Authentication passphrase* - Immettere la passphrase di autenticazione.
I macro utente vengono risolti in questo campo.
- *Privacy protocol* - Selezionare il protocollo di privacy: *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) o *AES256C* (Cisco).
Vedere le note sul supporto del protocollo di privacy.
- *Privacy passphrase* - Immettere la passphrase di privacy.
I macro utente vengono risolti in questo campo.

In caso di credenziali SNMPv3 errate (security name, authentication protocol/passphrase, privacy protocol):

- Zabbix riceve un errore ERROR da net-snmp, tranne nel caso di *Privacy passphrase* errata, in cui Zabbix riceve un errore TIMEOUT da net-snmp.
- La disponibilità dell'interfaccia SNMP passerà al rosso (non disponibile).

Note:
Le modifiche a *Authentication protocol*, *Authentication passphrase*, *Privacy protocol* o *Privacy passphrase*, effettuate senza modificare il *Security name*, vengono normalmente applicate automaticamente quando la corrispondente interfaccia SNMPv3 viene aggiornata in Zabbix. Nei casi in cui venga modificato anche il *Security name*, tutti i parametri verranno aggiornati immediatamente.

Supporto del protocollo di privacy

A seconda del sistema operativo e della configurazione di net-snmp, alcuni protocolli di privacy potrebbero non essere disponibili:

- Su alcuni sistemi operativi più recenti (ad esempio, RHEL9) il supporto per DES è stato rimosso dal pacchetto `net-snmp`.
- I protocolli di crittografia AES192 e più forti non sono supportati nativamente sui sistemi operativi più vecchi di RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04, openSUSE Leap 15.5.

Per verificare se la libreria net-snmp supporta AES192+, usa una delle seguenti opzioni:

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

Se l'output contiene `--enable-blumenthal-aes`, AES192+ è supportato.

Nota che `net-snmp-config` fa parte del pacchetto di sviluppo per SNMP (`libsnmp-dev` per Debian/Ubuntu, `net-snmp-devel` per CentOS/RHEL/OL/SUSE) e potrebbe non essere installato per impostazione predefinita.

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

Se l'output contiene `Invalid privacy protocol specified after -3x flag: AES-256`, AES192+ non è supportato. Se l'output contiene `No hostname specified.`, AES192+ non è supportato.

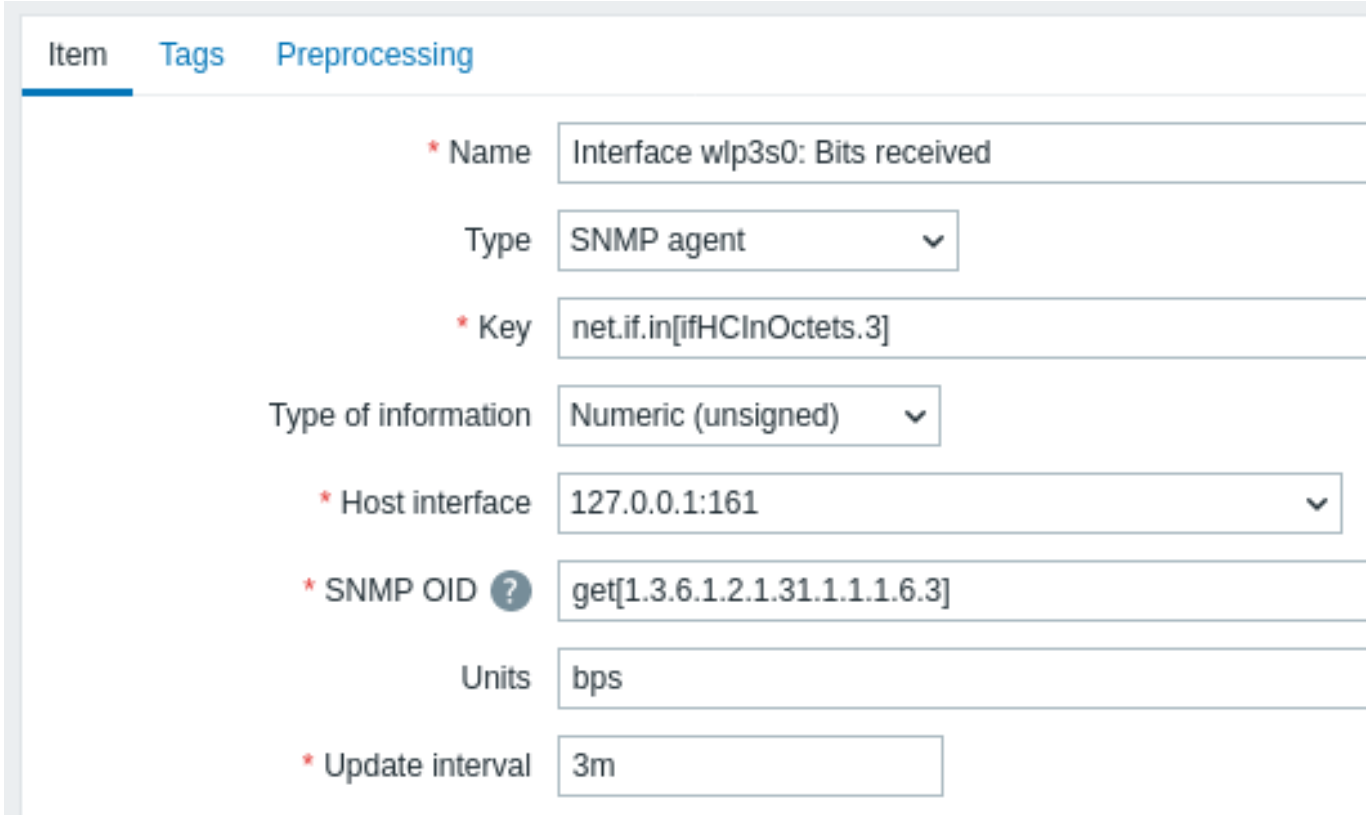
Se la tua libreria net-snmp non supporta i protocolli AES192 e superiori, ricompila net-snmp con l'opzione `--enable-blumenthal-aes`, quindi ricompila il server Zabbix specificando l'opzione `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-config`.

Passaggio 3

Creare un item per il monitoraggio.

Ora torna in Zabbix e fai clic su *Items* per l'host SNMP creato in precedenza. A seconda che tu abbia utilizzato o meno un template durante la creazione dell'host, avrai un elenco di item SNMP associati al tuo host oppure un elenco vuoto. Supponiamo che tu voglia creare personalmente l'item utilizzando le informazioni appena raccolte con snmpwalk e snmpget; fai quindi clic su *Create item*.

Compila i parametri obbligatori nel nuovo modulo dell'item:



Parametro	Descrizione
Name	Inserisci il nome dell'item.
Type	Seleziona qui **SNMP agent**.
Key	Inserisci una chiave significativa.
Host interface	Assicurati di selezionare l'interfaccia SNMP, ad esempio quella del tuo switch/router.

Parametro	Descrizione
SNMP OID	Utilizza uno dei formati supportati per inserire i valori OID: **walk[OID1,OID2,...]** - recupera un sottoalbero di valori. Ad esempio: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Questa opzione utilizza richieste SNMP bulk native (GetBulkRequest-PDU) in modo asincrono. Le impostazioni di timeout per questo item possono essere configurate nel modulo di configurazione dell'item. È consigliabile impostare un valore di timeout basso per evitare lunghi ritardi se il dispositivo non è raggiungibile, poiché verranno effettuati fino a 5 tentativi se quelli precedenti scadono o hanno esito negativo (ad esempio, un timeout di 3 secondi può comportare un'attesa di 15 secondi). Puoi utilizzare questo item come item master, con item dipendenti che estraggono i dati dall'item master tramite la pre-elaborazione. È possibile specificare più OID in una singola operazione snmp walk, ad esempio `walk[OID1,OID2,...]`, per elaborare in modo asincrono un OID alla volta. Se la richiesta bulk non restituisce risultati, viene effettuato un tentativo di recuperare un singolo record senza una richiesta bulk. I nomi MIB sono supportati come parametri; pertanto, `walk[1.3.6.1.2.1.2.2.1.2]` e `walk[ifDescr]` restituiranno lo stesso output. Se vengono specificati più OID/MIB, ad esempio `walk[ifDescr,ifType,ifPhysAddress]`, l'output sarà un elenco concatenato. Le richieste GetBulk vengono utilizzate con le interfacce SNMPv2 e v3, mentre per le interfacce SNMPv1 viene utilizzato GetNext; il numero massimo di ripetizioni per le richieste bulk viene configurato a livello di interfaccia. Il parametro del numero massimo di ripetizioni influisce sulle richieste bulk determinando il numero massimo di OID restituiti in una singola risposta bulk. Un valore più alto produce risposte bulk più grandi, riducendo il numero di trasmissioni necessarie. Tuttavia, alcuni dispositivi potrebbero non supportare valori molto elevati, causando potenziali problemi. Questo item restituisce l'output dell'utility snmpwalk con i parametri -Oe -Ot -On. Puoi utilizzare questo item come item master in SNMP discovery. **get[OID]** - recupera un *singolo* valore in modo asincrono. Ad esempio: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Le impostazioni di timeout per questo item possono essere configurate nel modulo di configurazione dell'item. È consigliabile impostare un valore di timeout basso per evitare lunghi ritardi se il dispositivo non è raggiungibile, poiché verranno effettuati fino a 5 tentativi se quelli precedenti scadono o hanno esito negativo (ad esempio, un timeout di 3 secondi può comportare un'attesa di 15 secondi). **OID** - (legacy) inserisce un singolo OID testuale o numerico per recuperare un singolo valore in modo sincrono, facoltativamente combinato con altri valori. Ad esempio: `1.3.6.1.2.1.31.1.1.1.6.3`. Per questa opzione, il timeout del controllo dell'item sarà uguale al valore impostato nel file di configurazione del server. È **consigliato** utilizzare gli item `walk[OID]` e `get[OID]` per ottenere prestazioni migliori. Tutti gli item `walk[OID]` e `get[OID]` vengono eseguiti in modo asincrono: non è necessario ricevere la risposta a una richiesta prima di avviare altri controlli. Anche la risoluzione DNS è asincrona. La concorrenza massima dei controlli asincroni è 1000 (definita da MaxConcurrentChecksPerPoller). Il numero di poller SNMP asincroni è definito dal parametro StartSNMPPollers. Nota che, per le statistiche sul traffico di rete restituite da uno qualsiasi dei metodi, è necessario aggiungere un passaggio *Change per second* nella scheda *Preprocessing*; in caso contrario, dal dispositivo SNMP otterrai il valore cumulativo invece della variazione più recente.

Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Ora salva l'item e vai a *Monitoring* > *Latest data* per visualizzare i dati SNMP.

Esempio 1

Esempio generale:

Parametro	Descrizione
OID	1.2.3.45.6.7.8.0 (oppure .1.2.3.45.6.7.8.0)
Key	<Stringa univoca da utilizzare come riferimento per i trigger> Per esempio, ”my_param”.

Si noti che l'OID può essere fornito sia in forma numerica che come stringa. Tuttavia, in alcuni casi, l'OID in formato stringa deve essere convertito nella rappresentazione numerica. L'utilità snmpget può essere utilizzata a questo scopo:

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Esempio 2

Monitoraggio dell'uptime:

Parameter	Descrizione
OID	MIB::sysUpTime.0
Key	router.uptime
Value type	Float
Units	uptime
Preprocessing step: Custom multiplier	0.01

Richieste bulk SNMP native

L'item **walk[OID1,OID2,...]** consente di utilizzare la funzionalità SNMP nativa per le richieste bulk (PDU GetBulkRequest), disponibile nelle versioni 2/3 di SNMP.

Una richiesta GetBulk in SNMP esegue più richieste GetNext e restituisce il risultato in un'unica risposta. Può essere utilizzata sia per gli item SNMP regolari sia per la discovery SNMP, al fine di ridurre i round trip di rete.

L'item SNMP **walk[OID1,OID2,...]** può essere utilizzato come master item che raccoglie i dati con un'unica richiesta, insieme a item dipendenti che analizzano la risposta secondo necessità tramite la pre-elaborazione.

Si noti che l'utilizzo delle richieste bulk SNMP native non è correlato all'opzione di combinazione delle richieste SNMP, che rappresenta il metodo proprietario di Zabbix per combinare più richieste SNMP (vedere la sezione successiva).

Per gli item SNMP bulk verranno effettuati fino a cinque tentativi per evitare errori nel caso in cui uno dei pacchetti venga perso. Il timeout per gli item SNMP con `get` e `walk` (impostato nel modulo configurazione dell'item) viene applicato all'intera sessione. Il timeout viene applicato indipendentemente dal fatto che i dati vengano recuperati completamente; se i dati vengono ricevuti solo parzialmente (ad esempio, se i dati vengono raccolti correttamente per un solo OID tra più OID), l'item diventa non supportato con il messaggio ”Only partial data received”. Se viene raggiunto il timeout, verrà effettuato un nuovo tentativo, il timeout verrà reimpostato e l'ultima richiesta verrà inviata nuovamente, consentendo alla sessione di continuare dall'ultima richiesta nel caso in cui un singolo pacchetto venga perso o arrivi in ritardo. Si consiglia di impostare un valore di timeout basso per evitare lunghi ritardi se il dispositivo non è raggiungibile, poiché verranno effettuati fino a 5 tentativi se quelli precedenti scadono o non riescono (ad esempio, un timeout di 3 secondi può comportare un'attesa di 15 secondi).

Funzionamento interno dell'elaborazione combinata

Il server e il proxy Zabbix possono interrogare i dispositivi SNMP per più valori in una singola richiesta. Questo riguarda diversi tipi di item SNMP:

- item SNMP regolari
- item SNMP con indici dinamici
- regole di discovery di basso livello SNMP

Tutti gli item SNMP su una singola interfaccia con parametri identici vengono pianificati per essere interrogati contemporaneamente. I primi due tipi di item vengono acquisiti dai poller in batch di massimo 128 item, mentre le regole di discovery di basso livello vengono elaborate singolarmente, come in precedenza.

A un livello inferiore, per l'interrogazione dei valori vengono eseguiti due tipi di operazioni: ottenere più oggetti specificati e attraversare un albero OID.

Per l'operazione di ”ottenimento” viene utilizzato un GetRequest-PDU con un massimo di 128 associazioni di variabili. Per l'operazione di ”attraversamento”, viene utilizzato un GetNextRequest-PDU per SNMPv1 e un GetBulkRequest con il campo ”max-repetitions” impostato a un massimo di 128 per SNMPv2 e SNMPv3.

Pertanto, i vantaggi dell'elaborazione combinata per ciascun tipo di item SNMP sono descritti di seguito:

- gli item SNMP regolari beneficiano dei miglioramenti dell'operazione di "ottenimento";
- gli item SNMP con indici dinamici beneficiano dei miglioramenti sia dell'operazione di "ottenimento" sia di quella di "attraversamento": l'operazione di "ottenimento" viene utilizzata per la verifica dell'indice, mentre quella di "attraversamento" viene utilizzata per la creazione della cache;
- le regole di discovery di basso livello SNMP beneficiano dei miglioramenti dell'operazione di "attraversamento".

Tuttavia, esiste un problema tecnico: non tutti i dispositivi sono in grado di restituire 128 valori per richiesta. Alcuni restituiscono sempre una risposta corretta, mentre altri rispondono con un errore "tooBig(1)" oppure non rispondono affatto quando la risposta potenziale supera un determinato limite.

Per trovare il numero ottimale di oggetti da interrogare per un determinato dispositivo, Zabbix utilizza la strategia seguente. Inizia con cautela interrogando 1 valore per richiesta. Se l'operazione ha esito positivo, interroga 2 valori per richiesta. Se anche questa operazione ha esito positivo, interroga 3 valori per richiesta e continua in modo analogo moltiplicando per 1,5 il numero di oggetti interrogati, ottenendo la seguente sequenza di dimensioni delle richieste: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Tuttavia, quando un dispositivo rifiuta di fornire una risposta corretta (ad esempio, per 42 variabili), Zabbix esegue due operazioni.

Innanzitutto, per il batch di item corrente dimezza il numero di oggetti in una singola richiesta e interroga 21 variabili. Se il dispositivo è attivo, la richiesta dovrebbe funzionare nella stragrande maggioranza dei casi, perché è noto che 28 variabili funzionano e 21 sono significativamente inferiori. Tuttavia, se anche questa operazione non riesce, Zabbix torna a interrogare i valori uno alla volta. Se anche a questo punto l'operazione non riesce, il dispositivo non sta sicuramente rispondendo e la dimensione della richiesta non è il problema.

La seconda operazione che Zabbix esegue per i batch di item successivi consiste nell'iniziare con l'ultimo numero di variabili riuscito (28 nel nostro esempio) e continuare ad aumentare la dimensione delle richieste di 1 fino a raggiungere il limite. Ad esempio, supponendo che la dimensione massima della risposta sia di 32 variabili, le richieste successive avranno dimensioni 29, 30, 31, 32 e 33. L'ultima richiesta non riuscirà e Zabbix non emetterà mai più una richiesta di dimensione 33. Da quel momento in poi, Zabbix interrogherà al massimo 32 variabili per questo dispositivo.

Se le richieste di grandi dimensioni non riescono con questo numero di variabili, ciò può significare una di due cose. Non è possibile conoscere i criteri esatti utilizzati da un dispositivo per limitare la dimensione della risposta, ma cerchiamo di approssimarli utilizzando il numero di variabili. La prima possibilità è che questo numero di variabili sia vicino al limite effettivo della dimensione della risposta del dispositivo nel caso generale: a volte la risposta è inferiore al limite, a volte è superiore. La seconda possibilità è che un pacchetto UDP in una delle due direzioni sia semplicemente andato perso. Per questi motivi, se Zabbix riceve una risposta non riuscita, riduce il numero massimo di variabili da provare per rientrare maggiormente nell'intervallo gestibile dal dispositivo, ma solo fino a due volte.

Nell'esempio precedente, se una richiesta con 32 variabili non riesce, Zabbix riduce il conteggio a 31. Se anche questa richiesta non riesce, Zabbix riduce il conteggio a 30. Tuttavia, Zabbix non ridurrà il conteggio al di sotto di 30, perché presumerà che gli ulteriori errori siano dovuti alla perdita di pacchetti UDP, anziché al limite del dispositivo.

Se, tuttavia, un dispositivo non è in grado di gestire correttamente le richieste combinate per altri motivi e l'euristica descritta sopra non funziona, per ogni interfaccia è disponibile l'impostazione "Use combined requests", che consente di disabilitare le richieste combinate per quel dispositivo.

Se le richieste combinate causano risposte parziali o malformate che producono calcoli errati al secondo (delta) (ad esempio, picchi apparenti nei contatori delle interfacce), disabilitare *Use combined requests* per l'interfaccia interessata per forzare richieste separate per ogni item; ciò spesso impedisce i falsi picchi. In alternativa, valutare l'utilizzo di item asincroni `get[]` o `walk[]`, che vengono eseguiti in modo asincrono e non sono soggetti al batching *Use combined requests* per interfaccia: possono essere utilizzati al posto dei controlli OID sincroni legacy per evitare problemi correlati alle richieste combinate. Cercare nei log del server/proxy voci simili a quella mostrata nella sezione Panoramica per identificare i dispositivi interessati.

Inoltre, se l'interfaccia diventa frequentemente non disponibile, potrebbe essere necessario aumentare il parametro `UnavailableDelay` nei file di configurazione del server Zabbix o del proxy Zabbix per ridurre la frequenza delle richieste. Gli item potrebbero diventare non supportati se durante la discovery o l'attraversamento degli OID vengono ricevuti dati parziali.

Indici dinamici

Panoramica

Sebbene sia possibile trovare il numero di indice richiesto (ad esempio, di un'interfaccia di rete) tra gli OID SNMP, a volte non è possibile fare completo affidamento sul fatto che il numero di indice rimanga sempre lo stesso.

I numeri di indice possono essere dinamici: possono cambiare nel tempo e, di conseguenza, il tuo item potrebbe smettere di funzionare.

Per evitare questo scenario, è possibile definire un OID che tenga conto della possibilità che un numero di indice cambi.

Ad esempio, se devi recuperare il valore dell'indice da aggiungere a **ifInOctets** che corrisponde all'interfaccia **GigabitEthernet0/1** su un dispositivo Cisco, utilizza il seguente OID:

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

La sintassi

Viene utilizzata una sintassi speciale per l'OID:

<OID dei dati>["index","<OID di base dell'indice>","<stringa da cercare>"]

Parametro	Descrizione
OID dei dati	OID principale da utilizzare per il recupero dei dati nell'item.
index	Metodo di elaborazione. Attualmente è supportato un solo metodo: **index** - cerca l'indice e lo aggiunge all'OID dei dati
OID di base dell'indice	Questo OID verrà cercato per ottenere il valore dell'indice corrispondente alla stringa.
stringa da cercare	La stringa da utilizzare per una corrispondenza esatta con un valore durante la ricerca. Sensibile alle maiuscole e minuscole.

Esempio

Ottenere l'utilizzo della memoria del processo *apache*.

Se si utilizza questa sintassi OID:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

il numero di indice verrà cercato qui:

```
...
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"
...
```

Ora abbiamo l'indice, 5388. L'indice verrà aggiunto all'OID dei dati per ricevere il valore che ci interessa:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

Cache della ricerca degli indici

Quando viene richiesto un item con indice dinamico, Zabbix recupera e memorizza nella cache l'intera tabella SNMP sotto l'OID di base per l'indice, anche se una corrispondenza verrebbe trovata prima. Questo viene fatto nel caso in cui un altro item faccia riferimento allo stesso OID di base in seguito: Zabbix cercherà l'indice nella cache, invece di interrogare nuovamente il host monitorato. Si noti che ogni processo poller utilizza una cache separata.

In tutte le successive operazioni di recupero dei valori viene verificato solo l'indice trovato. Se non è cambiato, il valore viene richiesto. Se è cambiato, la cache viene ricostruita: ogni poller che rileva un indice modificato esegue nuovamente la scansione della tabella SNMP dell'indice.

OID speciali

Alcuni degli OID SNMP più utilizzati vengono tradotti automaticamente da Zabbix in una rappresentazione numerica. Ad esempio, **ifIndex** viene tradotto in **1.3.6.1.2.1.2.2.1.1**, mentre **ifIndex.0** viene tradotto in **1.3.6.1.2.1.2.2.1.1.0**.

La tabella contiene l'elenco degli OID speciali.

Special OID	Identifier	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	Un valore univoco per ogni interfaccia.
ifDescr	1.3.6.1.2.1.2.2.1.2	Una stringa di testo contenente informazioni sull'interfaccia. Questa stringa dovrebbe includere il nome del produttore, il nome del prodotto e la versione dell'interfaccia hardware.
ifType	1.3.6.1.2.1.2.2.1.3	Il tipo di interfaccia, distinto in base al protocollo o ai protocolli fisici/di collegamento immediatamente 'sotto' il livello di rete nello stack dei protocolli.

Special OID	Identifier	Description
ifMtu	1.3.6.1.2.1.2.2.1.4	La dimensione del datagramma più grande che può essere inviato/ricevuto sull'interfaccia, specificata in ottetti.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Una stima della larghezza di banda attuale dell'interfaccia in bit al secondo.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	L'indirizzo dell'interfaccia al livello di protocollo immediatamente 'sotto' il livello di rete nello stack dei protocolli.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Lo stato amministrativo corrente dell'interfaccia.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Lo stato operativo corrente dell'interfaccia.
ifInOctets	1.3.6.1.2.1.2.2.1.10	Il numero totale di ottetti ricevuti sull'interfaccia, inclusi i caratteri di framing.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Il numero di pacchetti unicast di sottorete consegnati a un protocollo di livello superiore.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Il numero di pacchetti non unicast (cioè broadcast o multicast di sottorete) consegnati a un protocollo di livello superiore.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Il numero di pacchetti in ingresso che sono stati scelti per essere scartati anche se non erano stati rilevati errori, per impedirne la consegna a un protocollo di livello superiore. Un possibile motivo per scartare un pacchetto di questo tipo potrebbe essere liberare spazio nel buffer.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Il numero di pacchetti in ingresso che contenevano errori che ne impedivano la consegna a un protocollo di livello superiore.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Il numero di pacchetti ricevuti tramite l'interfaccia che sono stati scartati a causa di un protocollo sconosciuto o non supportato.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Il numero totale di ottetti trasmessi dall'interfaccia, inclusi i caratteri di framing.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Il numero totale di pacchetti che i protocolli di livello superiore hanno richiesto di trasmettere e che non erano indirizzati a un indirizzo multicast o broadcast a questo sottolivello, inclusi quelli che sono stati scartati o non inviati.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Il numero totale di pacchetti che i protocolli di livello superiore hanno richiesto di trasmettere e che erano indirizzati a un indirizzo multicast o broadcast a questo sottolivello, inclusi quelli che sono stati scartati o non inviati.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Il numero di pacchetti in uscita che sono stati scelti per essere scartati anche se non erano stati rilevati errori, per impedirne la trasmissione. Un possibile motivo per scartare un pacchetto di questo tipo potrebbe essere liberare spazio nel buffer.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Il numero di pacchetti in uscita che non hanno potuto essere trasmessi a causa di errori.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	La lunghezza della coda dei pacchetti in uscita (in pacchetti).

File MIB

Introduzione

MIB è l'acronimo di Management Information Base. I file MIB consentono di utilizzare rappresentazioni testuali degli OID (Object Identifier). È possibile utilizzare OID grezzi durante il monitoraggio dei dispositivi SNMP con Zabbix, ma se si preferisce utilizzare una rappresentazione testuale, è necessario installare i file MIB.

Ad esempio,

`ifHCOutOctets`

è la rappresentazione testuale dell'OID

`1.3.6.1.2.1.31.1.1.1.10`

Installazione dei file MIB

Sui sistemi basati su Debian:

```
apt install snmp-mibs-downloader
download-mibs
```

Sui sistemi basati su RedHat:

```
dnf install net-snmp-libs
```

Abilitazione dei file MIB

Sui sistemi basati su RedHat, i file MIB dovrebbero essere abilitati per impostazione predefinita. Sui sistemi basati su Debian, è necessario modificare il file `/etc/snmp/snmp.conf` e commentare la riga che contiene `mibs :`

```
# Poiché i pacchetti snmp vengono forniti senza file MIB per motivi di licenza, il caricamento
# dei MIB è disabilitato per impostazione predefinita. Se hai aggiunto i MIB, puoi riabilitare
# il loro caricamento commentando la riga seguente.
mibs :
```

Test dei file MIB

Il test degli MIB SNMP può essere eseguito utilizzando l'utility `snmpwalk`. Se non è installata, usa le seguenti istruzioni.

Sui sistemi basati su Debian:

```
apt install snmp
```

Sui sistemi basati su RedHat:

```
dnf install net-snmp-utils
```

Dopo di che, il seguente comando non deve restituire errori quando interroghi un dispositivo di rete:

```
$ 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
[...]
```

Utilizzo dei MIB in Zabbix

La cosa più importante da tenere presente è che i processi di Zabbix non vengono informati delle modifiche apportate ai file MIB. Quindi, dopo ogni modifica, è necessario riavviare Zabbix server o proxy, ad esempio:

```
systemctl restart zabbix-server
```

Dopodiché, le modifiche apportate ai file MIB avranno effetto.

Utilizzo di file MIB personalizzati

Esistono file MIB standard inclusi in ogni distribuzione GNU/Linux. Tuttavia, alcuni produttori di dispositivi forniscono i propri.

Supponiamo che tu voglia utilizzare il file MIB CISCO-SMI. Le seguenti istruzioni lo scaricheranno e installeranno:

```
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
```

Ora dovresti essere in grado di utilizzarlo. Prova a tradurre il nome dell'oggetto *ciscoProducts* dal file MIB in OID:

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

Se ricevi errori invece dell'OID, assicurati che tutti i comandi precedenti non abbiano restituito errori.

La traduzione del nome dell'oggetto ha funzionato, sei pronto a usare il file MIB personalizzato. Nota il prefisso del nome MIB (*CISCO-SMI::*) utilizzato nella query. Ne avrai bisogno quando utilizzi gli strumenti da riga di comando così come Zabbix.

Non dimenticare di riavviare Zabbix server/proxy prima di utilizzare questo file MIB in Zabbix.

> **Attention:**
> Tieni presente che i file MIB possono avere dipendenze. Vale a dire, un MIB può richiederne un altro. Per soddisfare queste dipendenze devi installare tutti i file MIB coinvolti.

SNMP trap

Panoramica

La ricezione dei trap SNMP è l'opposto dell'interrogazione di dispositivi abilitati SNMP.

In questo caso, le informazioni vengono inviate da un dispositivo abilitato SNMP a `snmptrapd` e vengono raccolte o "trappate" da Zabbix server o Zabbix proxy da un file.

Di solito, i trap vengono inviati in seguito a una modifica di qualche condizione e l'agent si connette al server sulla porta 162 (a differenza della porta 161 sul lato dell'agent, che viene usata per le interrogazioni). L'uso dei trap può rilevare alcuni problemi brevi che si verificano tra un intervallo di interrogazione e l'altro e che potrebbero non essere rilevati dai dati interrogati.

La ricezione dei trap SNMP in Zabbix è progettata per funzionare con `snmptrapd` e con uno dei meccanismi per inoltrare i trap a Zabbix: uno script Bash o Perl oppure SNMPTT.

> **Note:**
> Il modo più semplice per configurare il monitoraggio dei trap dopo aver configurato Zabbix è usare la soluzione con script Bash, perché Perl e SNMPTT spesso mancano nelle distribuzioni moderne e richiedono una configurazione più complessa. Tuttavia, questa soluzione usa uno script configurato come `traphandle`. Per prestazioni migliori nei sistemi di produzione, usa la soluzione Perl incorporata (uno script con l'opzione do `perl` oppure SNMPTT).

Il flusso di lavoro della ricezione di un trap:

1. `snmptrapd` riceve un trap.
2. `snmptrapd` passa il trap allo script ricevitore (Bash, Perl) o a SNMPTT.
3. Il ricevitore analizza, formatta e scrive il trap in un file.
4. Lo SNMP trapper di Zabbix legge e analizza il file dei trap.
5. Per ogni trap, Zabbix trova tutti gli item dello SNMP trapper con interfacce host corrispondenti all'indirizzo del trap ricevuto. Nota che durante la corrispondenza viene usato solo l'*IP* o il *DNS* selezionato nell'interfaccia host.
6. Per ogni item trovato, il trap viene confrontato con la regexp in `snmptrap[regexp]`. Il trap viene impostato come valore di **tutti** gli item corrispondenti. Se non viene trovato alcun item corrispondente e c'è un item `snmptrap.fallback`, il trap viene impostato come suo valore.
7. Se il trap non è stato impostato come valore di alcun item, Zabbix per impostazione predefinita registra il trap non corrispondente. (Questa opzione è configurata tramite *Registra trap SNMP non corrispondenti* in *Amministrazione* > *Generale* > *Altro*.)

Note sul failover HA

Durante il passaggio del nodo ad alta disponibilità (HA), Zabbix continuerà l'elaborazione a partire dall'ultimo record entro l'ultimo timestamp ISO 8601; se lo stesso record non viene trovato, allora verrà utilizzato solo il timestamp per identificare l'ultima posizione.

Configurazione dei trap SNMP

Questo tipo di item richiede la seguente configurazione del frontend:

Passaggio 1: creare un'interfaccia SNMP per l'host

1. In *Data collection* > *Hosts*, crea un host oppure modifica quello esistente.
2. Nel campo *Interfaces*, seleziona il tipo di interfaccia *SNMP*.
3. Inserisci l'indirizzo IP/nome DNS e il numero di porta.
4. Seleziona la *versione SNMP* dal menu a discesa: (*SNMPv1*, *SNMPv2*, oppure *SNMPv3*) e aggiungi le credenziali dell'interfaccia in base alla versione SNMP selezionata.

L'indirizzo di ciascun trap ricevuto verrà confrontato con gli indirizzi IP e DNS di tutte le interfacce SNMP per trovare gli host corrispondenti.

Passaggio 2: configurare l'item

1. Per l'host, crea un item oppure modifica quello esistente.
2. Nel campo *Key*, usa una delle chiavi SNMP trap:
   - `snmptrap[regexp]` - intercetta tutti i trap SNMP che corrispondono all'espressione regolare specificata nel parametro `regexp`; se `regexp` non è specificato, intercetta qualsiasi trap.
 Nel parametro sono supportate le macro utente e le espressioni regolari globali.
 Valore restituito: trap SNMP.
 Questo item può essere impostato solo per interfacce SNMP.

- `snmptrap.fallback` - intercetta tutti i trap SNMP che non sono stati intercettati da nessuno degli item `snmptrap[]` per quella interfaccia.
 Valore restituito: trap SNMP.
 Questo item può essere impostato solo per interfacce SNMP.

Note:
Al momento non è supportata la corrispondenza di espressioni regolari multilinea.

3. Imposta *Tipo di informazione* su *Log* per consentire l'analisi dei timestamp. Sono accettabili anche altri formati, come *Numerico*, ma potrebbero richiedere un gestore trap personalizzato.

Configurazione del monitoraggio delle trap SNMP

Configurazione di Zabbix server/proxy

Per leggere i trap, Zabbix server o proxy deve essere configurato per avviare il processo SNMP trapper e puntare al file dei trap che viene scritto da SNMPTT o da un ricevitore di trap Bash/Perl. Per farlo, modifica il file di configurazione (`zabbix_server.conf` o `zabbix_proxy.conf`):

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
Se viene utilizzato il parametro systemd `PrivateTmp`, è improbabile che questo file funzioni in `/tmp`.

Configurazione del ricevitore di trap Bash

Requisiti: solo snmptrapd.

Uno script ricevitore di trap Bash può essere usato per inoltrare i trap a Zabbix server da snmptrapd usando il file trapper. Per configurarlo, aggiungi l'opzione `traphandle` al file di configurazione di snmptrapd (`snmptrapd.conf`), vedi l'esempio.

Note:
Potrebbe essere necessario riavviare snmptrapd per applicare le modifiche alla sua configurazione.

Configurazione del ricevitore trap Perl

Requisiti: Perl, Net-SNMP compilato con `--enable-embedded-perl` (attivato per impostazione predefinita da Net-SNMP 5.4)

Un ricevitore trap Perl (vedere `misc/snmptrap/zabbix_trap_receiver.pl`) può essere usato per inoltrare i trap a Zabbix server direttamente da snmptrapd. Per configurarlo:

- Aggiungere lo script Perl al file di configurazione di snmptrapd (`snmptrapd.conf`), ad esempio:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- Configurare il ricevitore, ad esempio:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
Potrebbe essere necessario riavviare snmptrapd per applicare le modifiche alla sua configurazione.

Note:
Se il nome dello script non è tra virgolette, snmptrapd rifiuterà di avviarsi con messaggi simili ai seguenti:


```
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
```

Configurazione di SNMPTT

Per prima cosa, `snmptrapd` deve essere configurato per usare SNMPTT.

Note:
Per ottenere le migliori prestazioni, SNMPTT dovrebbe essere configurato come daemon usando `snmptthandler-embedded` per inoltrargli i trap. Vedere le istruzioni per la configurazione di SNMPTT.

Quando SNMPTT è configurato per ricevere i trap, configurare `snmptt.ini`:

1. Abilitare l'uso del modulo Perl del pacchetto `NET-SNMP`:

```
net_snmp_perl_enable = 1
```

2. Registrare i trap nel file dei trap che verrà letto da Zabbix:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. Impostare il formato data-ora:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
Il pacchetto `net-snmp-perl` è stato rimosso in RHEL 8.0-8.2; è stato reintrodotto in RHEL 8.3. Per ulteriori informazioni, vedere i problemi noti.

Ora formattare i trap in modo che Zabbix li riconosca (modificare `snmptt.conf`):

1. Ogni istruzione `FORMAT` deve iniziare con `ZBXTRAP [address]`, dove `[address]` verrà confrontato con gli indirizzi IP e DNS delle interfacce SNMP su Zabbix. Ad esempio:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Vedere ulteriori informazioni sul formato dei trap SNMP qui sotto.

Attention:
Non usare trap sconosciuti - Zabbix non sarà in grado di riconoscerli. I trap sconosciuti possono essere gestiti definendo un evento generale in `snmptt.conf`:


```
EVENT general .* "General event" Normal
```

Formato del trap SNMP

Tutti i ricevitori trap Perl personalizzati e la configurazione trap SNMPTT devono formattare il trap nel modo seguente:

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

dove

- `[timestamp]` - il timestamp nel formato `%Y-%m-%dT%H:%M:%S%z`.
- `ZBXTRAP` - intestazione che indica che in questa riga inizia un nuovo trap.
- `[address]` - indirizzo IP usato per trovare l'host per questo trap.

Nota che `ZBXTRAP` e `[address]` verranno rimossi dal messaggio durante l'elaborazione. Se il trap è formattato in modo diverso, Zabbix potrebbe analizzare i trap in modo imprevisto.

Esempio di 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
```

Questo produrrà il seguente trap per l'interfaccia SNMP con 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
```

Requisiti di sistema

Note:
Si consiglia di installare i file MIB per garantire che i valori degli item siano visualizzati nel formato corretto. Senza i file MIB, potrebbero verificarsi problemi di formattazione, come la visualizzazione dei valori in HEX invece che in UTF-8 o viceversa.

Supporto per file di grandi dimensioni

Zabbix supporta file di grandi dimensioni per i file SNMP trapper. La dimensione massima del file che Zabbix può leggere è 2^63 (8 EiB). Si noti che il filesystem può imporre un limite inferiore alla dimensione del file.

Rotazione dei log

Zabbix non fornisce alcun sistema di rotazione dei log: questa operazione deve essere gestita dall'utente. La rotazione dei log dovrebbe prima rinominare il vecchio file e solo successivamente eliminarlo, in modo che non si perdano trap:

1. Zabbix apre il file trap nell'ultima posizione nota e passa al punto 3.
2. Zabbix verifica se il file attualmente aperto è stato ruotato confrontando il numero inode con il numero inode del file trap definito. Se non esiste alcun file aperto, Zabbix reimposta l'ultima posizione e torna al punto 1.
3. Zabbix legge i dati dal file attualmente aperto e imposta la nuova posizione.
4. I nuovi dati vengono analizzati. Se si trattava del file ruotato, il file viene chiuso e si torna al punto 2.
5. Se non ci sono nuovi dati, Zabbix attende 1 secondo e torna al punto 2.

File system

A causa dell'implementazione del file trap, Zabbix necessita che il file system supporti gli inode per distinguere i file (le informazioni vengono acquisite tramite una chiamata `stat()`).

Esempi di configurazione utilizzando diverse versioni del protocollo SNMP

Questo esempio utilizza snmptrapd e uno script Bash ricevitore per inoltrare i trap al server Zabbix.

Configurazione:

1. Configurare Zabbix per avviare lo SNMP trapper e impostare il file dei trap. Aggiungere a `zabbix_server.conf`:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Scaricare lo script Bash in `/usr/sbin/zabbix_trap_handler.sh`:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/temp
```

Se necessario, modificare la variabile `ZABBIX_TRAPS_FILE` nello script. Per utilizzare il valore predefinito, creare prima la directory principale:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Aggiungere quanto segue a `snmtrapd.conf` (fare riferimento all'esempio funzionante)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Potrebbe essere necessario riavviare snmptrapd affinché acquisisca le modifiche alla configurazione.

4. Creare un item SNMP TEST (tenere presenti i requisiti di configurazione iniziali):

   Tipo: *SNMP trap*
 Tipo di informazioni: *Log*
 Interfaccia host: *SNMP 127.0.0.1*
 Chiave: `snmptrap["linkUp"]`
 Formato dell'ora del log: `yyyyMMdd.hhmmss`

Si noti che viene utilizzato il formato di data e ora ISO 8601.

5. Ora configureremo `snmptrapd` per la versione del protocollo SNMP scelta e invieremo trap di test utilizzando l'utility `snmptrap`.

SNMPv1, SNMPv2

I protocolli SNMPv1 e SNMPv2 si basano sull'autenticazione tramite "community string". Nell'esempio seguente utilizzeremo "secret" come community string. Deve essere impostata sullo stesso valore nei mittenti dei trap SNMP.

Si noti che, sebbene sia ancora ampiamente utilizzato negli ambienti di produzione, SNMPv2 non offre alcuna crittografia né una reale autenticazione del mittente. I dati vengono inviati in testo semplice e pertanto queste versioni del protocollo devono essere utilizzate solo in ambienti sicuri, come una rete privata, e non devono mai essere utilizzate su reti pubbliche o di terze parti.

La versione 1 di SNMP non viene praticamente più utilizzata, poiché non supporta i contatori a 64 bit ed è considerata un protocollo legacy.

Per abilitare l'accettazione dei trap SNMPv1 o SNMPv2, aggiungere la riga seguente a `snmptrapd.conf`. Sostituire `secret` con la community string SNMP configurata sui mittenti dei trap SNMP:

```
authCommunity log,execute,net secret
```

Ora possiamo inviare un trap di test utilizzando `snmptrap`. In questo esempio utilizzeremo l'OID comune "link up":

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 risolve i problemi di sicurezza di SNMPv1/v2 e fornisce autenticazione e crittografia. È possibile utilizzare i metodi di autenticazione MD5 o diversi metodi SHA, nonché DES o diversi cifrari AES.

Configurare `snmptrapd` per accettare trap SNMPv3 aggiungendo le righe seguenti a `snmptrapd.conf`:

```
##### Creare un utente USM SNMPv3 con autenticazione e privacy
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
##### Consentire all'utente di registrare i trap ed eseguire gli script del gestore dei trap
authUser log,execute <securityName>
```

Ad esempio:

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
Si noti la parola chiave "execute", che consente di eseguire script per questo modello di sicurezza utente.

Per testare la configurazione dei trap SNMPv3, inviare un trap di test utilizzando il comando seguente:

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x8000000
```

Warning:
Se si desidera utilizzare metodi di crittografia avanzati come AES192 o AES256, utilizzare net-snmp a partire dalla versione 5.8. Potrebbe essere necessario ricompilarlo con l'opzione `configure: --enable-blumenthal-aes`. Le versioni precedenti di net-snmp non supportano AES192/AES256. Vedere anche: Strong Authentication or Encryption.

Verifica

In entrambi gli esempi verranno visualizzate righe simili nel file `/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
```

Il valore dell'item in Zabbix sarà:

```
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
```

Esempio con 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
```

Vedi anche

- Articolo del blog di Zabbix sulle trap SNMP
- Configurazione di snmptrapd (documentazione ufficiale di net-snmp)
- Configurazione di snmptrapd per ricevere notifiche SNMPv3 (documentazione ufficiale di net-snmp)

Zabbix interno

Panoramica

I controlli interni consentono di monitorare i processi interni di Zabbix. In altre parole, è possibile monitorare ciò che accade con Zabbix server o Zabbix proxy.

I controlli interni vengono calcolati:

- su Zabbix server - se il host è monitorato dal server
- su Zabbix proxy - se il host è monitorato dal proxy

I controlli interni vengono elaborati dal server o dal proxy indipendentemente dallo stato di manutenzione del host.

Per utilizzare questo item, selezionare il tipo di item **Zabbix internal**.

Note:
I controlli interni vengono elaborati dai poller di Zabbix.

Prestazioni

L'utilizzo di alcuni item interni può influire negativamente sulle prestazioni. Questi item sono:

- `zabbix[host,,items]`
- `zabbix[host,,items_unsupported]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[queue,,]`
- `zabbix[requiredperformance]`
- `zabbix[stats,,,queue,,]`
- `zabbix[triggers]`

Anche le sezioni del frontend Informazioni di sistema e Coda sono interessate.

Controlli supportati

Le chiavi degli item sono elencate senza parametri personalizzabili e informazioni aggiuntive.
Fare clic sulla chiave dell'item per visualizzare tutti i dettagli.

Chiave dell'item	Descrizione
zabbix[apm,]	Le statistiche dei dati di telemetria ricevuti dal collector APM.
zabbix[boottime]	L'ora di avvio del processo Zabbix server o Zabbix proxy, in secondi.
zabbix[cluster,discovery,nodes]	Rileva i nodi del cluster ad alta disponibilità.
zabbix[connector_queue]	Il numero di valori accodati nella coda del connector.
zabbix[discovery_queue]	Il numero di controlli di rete accodati nella coda di discovery.
zabbix[host,,items]	Il numero di item abilitati (supportati e non supportati) sul host.
zabbix[host,,items_unsupported]	Il numero di item non supportati abilitati sul host.
zabbix[host,maintenance]	Lo stato di manutenzione corrente del host.
zabbix[host,active_agent,available]	La disponibilità dei controlli dell'agent attivo sul host.
zabbix[host,discovery,interfaces]	I dettagli di tutte le interfacce configurate del host nel frontend Zabbix.
zabbix[host,,available]	La disponibilità dell'interfaccia principale per un determinato tipo di controlli sul host.
zabbix[hosts]	Il numero di host monitorati.
zabbix[items]	Il numero di item abilitati (supportati e non supportati).
zabbix[items_unsupported]	Il numero di item non supportati.
zabbix[java,]	Le informazioni sul Java gateway Zabbix.
zabbix[lld_queue]	Il numero di valori accodati nella coda di elaborazione della discovery di basso livello.
zabbix[preprocessing]	Le statistiche dei valori ricevuti dal gestore della pre-elaborazione.
zabbix[preprocessing_queue]	Il numero di valori accodati nella coda di pre-elaborazione.
zabbix[process,,]	La percentuale di tempo che un determinato processo Zabbix o un gruppo di processi (identificati da <type> e <mode>) ha trascorso in <state>.
zabbix[proxy,]	Le informazioni sul proxy Zabbix.
zabbix[proxy,discovery]	L'elenco dei proxy Zabbix.
zabbix[proxy group,available]	Il numero di proxy online in un gruppo di proxy.
zabbix[proxy group,pavailable]	La percentuale di proxy online in un gruppo di proxy.

Chiave dell'item	Descrizione
zabbix[proxy group„proxies]	L'elenco dei proxy Zabbix in un gruppo di proxy.
zabbix[proxy group„state]	Lo stato di un gruppo di proxy.
zabbix[proxy group,discovery]	Restituisce un elenco di gruppi di proxy con dati di configurazione e dati in tempo reale.
zabbix[proxy_buffer,buffer,]	Restituisce le statistiche sull'utilizzo del buffer di memoria del proxy.
zabbix[proxy_buffer,state,changes]	Restituisce il numero di modifiche dello stato tra le modalità di buffer su disco e in memoria dall'avvio.
zabbix[proxy_buffer,state,current]	Restituisce lo stato operativo corrente in cui vengono archiviati i nuovi dati.
zabbix[proxy_history]	Il numero di valori nella tabella della cronologia del proxy in attesa di essere inviati al server.
zabbix[queue„]	Il numero di item monitorati nella coda che sono ritardati di almeno <from> secondi, ma meno di <to> secondi.
zabbix[rcache„]	Le statistiche sulla disponibilità della cache di configurazione Zabbix.
zabbix[requiredperformance]	Le prestazioni richieste del server Zabbix o del proxy Zabbix, espresse come nuovi valori al secondo previsti.
zabbix[stats„]	Restituisce le metriche interne di un server Zabbix o di un proxy. Se sono specificati <ip> e <port>, le metriche vengono recuperate dall'istanza remota; in caso contrario, dall'istanza locale.
zabbix[stats„,queue„]	Restituisce le metriche interne della coda di un server Zabbix o di un proxy. Se sono specificati <ip> e <port>, le metriche vengono recuperate dall'istanza remota; in caso contrario, dall'istanza locale.
zabbix[tcache„]	Le statistiche sull'efficacia della cache delle funzioni di trend Zabbix.
zabbix[triggers]	Il numero di trigger abilitati nel database Zabbix, con tutti gli item abilitati sugli host abilitati.
zabbix[uptime]	Il tempo di attività del processo Zabbix server o proxy, in secondi.
zabbix[vcache,buffer,]	Le statistiche sulla disponibilità della cache dei valori Zabbix.
zabbix[vcache,cache,]	Le statistiche sull'efficacia della cache dei valori Zabbix.
zabbix[version]	La versione del server Zabbix o del proxy.
zabbix[vmware,buffer,]	Le statistiche sulla disponibilità della cache VMware di Zabbix.
zabbix[vps,written]	Il numero totale di valori della cronologia scritti nel database.
zabbix[wcache„]	Le statistiche e la disponibilità della cache di scrittura Zabbix.

Dettagli della chiave item

- I parametri senza parentesi angolari sono obbligatori e devono essere utilizzati *così come sono* (ad esempio, "host" e "available" in `zabbix[host,<type>,available]`).
- I parametri con parentesi angolari **< >** devono essere sostituiti con un valore valido. Se un parametro ha un valore predefinito, può essere omesso.
- I valori per gli item e i parametri degli item contrassegnati come "not supported on proxy" possono essere recuperati solo se l'host è monitorato dal server. Viceversa, i valori "not supported on server" possono essere recuperati solo se l'host è monitorato dal proxy.

zabbix[apm,<type>]

 Le statistiche dei dati di telemetria ricevuti dall'APM collector:

- *requests* - il numero di richieste ricevute dall'APM collector (contatore):
  - *accepted* - il numero di richieste accettate per l'elaborazione;
  - *dropped* - il numero di richieste eliminate;
- *rows* - il numero di righe di dati di telemetria scritte nell'origine dati, per tipo di dati (contatore).

Valore restituito: *oggetto JSON*.

Parametri:

- **type** - *traces*, *metrics* o *logs*. Se non specificato, vengono restituite le statistiche per tutti i tipi.

Commenti:

- Questo item non è supportato sul server Zabbix.

Esempio di valori restituiti per `zabbix[apm]`:

```
{
  "requests": {
    "accepted": 15230,
```

```
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
    "metrics_gauge": 31250,
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```


zabbix[boottime]

 L'ora di avvio del processo di Zabbix server o Zabbix proxy in secondi.
 Valore restituito: *Integer*.

zabbix[cep]

 Complex event processing statistics.
 Return value: *JSON*.

zabbix[cluster,discovery,nodes]

 Rileva i nodi del cluster ad alta disponibilità.
 Valore restituito: *oggetto JSON*.

Commenti:

- Questo item può essere utilizzato nel rilevamento di basso livello.
- I campi restituiti corrispondono principalmente alle proprietà dell'oggetto nodo ad alta disponibilità.
- Campi aggiuntivi:
  - `db_timestamp` - ora corrente del server (timestamp Unix) inclusa nella risposta
  - `lastaccess_age` - secondi trascorsi dall'ultimo heartbeat del nodo (db_timestamp - lastaccess)
  - `status` - stato del nodo:
    - 0 - standby
    - 1 - arrestato manualmente
    - 2 - non disponibile
    - 3 - attivo

Esempio di valori restituiti:


```
[
  {
    "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]

 Il numero di valori accodati nella coda del connector.
 Valore restituito: *Integer*.

zabbix[discovery_queue]

 Il numero di controlli di rete accodati nella coda di discovery.
 Valore restituito: *Integer*.

zabbix[host,„items]

 Il numero di item abilitati (supportati e non supportati) sull'host.
 Valore restituito: *Intero*.

zabbix[host,„items_unsupported]

 Il numero di item non supportati abilitati sull'host.
 Valore restituito: *Integer*.

zabbix[host,„maintenance]

 Lo stato di manutenzione corrente del host.
 Valori restituiti: *0* - stato normale; *1* - manutenzione con raccolta dati; *2* - manutenzione senza raccolta dati.

Commenti:

- Questo item viene sempre elaborato da Zabbix server indipendentemente dalla posizione del host (sul server o sul proxy). Il proxy non riceverà questo item con i dati di configurazione.
- Il secondo parametro deve essere vuoto ed è riservato per uso futuro.

zabbix[host,active_agent,available]

 La disponibilità dei controlli attivi dell'agent sull'host.
 Valori restituiti: *0* - sconosciuto; *1* - disponibile; *2* - non disponibile.

zabbix[host,discovery,interfaces]

 The details of all configured interfaces of the host in Zabbix frontend.
 Return value: *JSON object*.

Comments:

- This item can be used in low-level discovery;
- This item is not supported on Zabbix proxy.

zabbix[host,<type>,available]

 La disponibilità dell'interfaccia principale di un particolare tipo di controlli sul host.
 Valori restituiti: *0* - non disponibile; *1* - disponibile; *2* - sconosciuto.

Parametri:

- **type** - *agent*, *snmp*, *ipmi* o *jmx*.

Commenti:

- Il valore dell'item viene calcolato in base ai parametri di configurazione relativi a irraggiungibilità/indisponibilità del host.

zabbix[hosts]

 Il numero di host monitorati.
 Valore restituito: *intero*.

zabbix[items]

 Il numero di item abilitati (supportati e non supportati).
 Valore restituito: *Integer*.

zabbix[items_unsupported]

 Il numero di item non supportati.
 Valore restituito: *Integer*.

zabbix[java,„<param>]

 Informazioni su Zabbix Java gateway.
 Valori restituiti: *1* - se `<param>` è *ping*; *versione di Java gateway* - se `<param>` è *version* (ad esempio: "8.0.0").

Parametri:

- **param** - *ping* o *version*.

Commenti:

- Questo item può essere usato per verificare la disponibilità di Java gateway usando la funzione di trigger `nodata()`.
- Il secondo parametro deve essere vuoto ed è riservato per un uso futuro.

zabbix[lld_queue]

 Il numero di valori accodati nella coda di elaborazione del low-level discovery.
 Valore restituito: *Integer*.

Commenti:

- Questo item può essere utilizzato per monitorare la lunghezza della coda di elaborazione del low-level discovery.

zabbixpreprocessing

 Statistiche dei valori ricevuti dal gestore di preprocessing:

- *queued* - il numero e la dimensione dei valori in coda che richiedono preprocessing (contatore)
- *direct* - il numero e la dimensione dei valori in coda che non richiedono preprocessing (contatore)
- *queue* - il conteggio dei valori accodati nella coda di preprocessing (uguale a `zabbix[preprocessing_queue]`)

Valore restituito: *JSON*.

Esempio di valori restituiti:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 Il numero di valori accodati nella coda di preprocessing.
 Valore restituito: *Integer*.

Commenti:

- Questo item può essere utilizzato per monitorare la lunghezza della coda di preprocessing.

zabbix[process,<type>,<mode>,<state>]

 La percentuale di tempo che un determinato processo Zabbix o un gruppo di processi (identificati da `<type>` e `<mode>`) ha trascorso nello `<state>`. Viene calcolata solo per l'ultimo minuto.
 Valore restituito: *Float*.

Parametri:

- **type**:
  - per i processi del server: *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*;
  - per i processi del proxy: *agent poller*, *apm manager*, *apm worker*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;
- **mode**:
  - *avg* - valore medio per tutti i processi di un determinato tipo (predefinito);
  - *count* - restituisce il numero di fork per un determinato tipo di processo; `<state>` non deve essere specificato;
  - *max* - valore massimo;
  - *min* - valore minimo;

    - *<process number>* - numero del processo (compreso tra 1 e il numero di istanze pre-fork; ad esempio, se sono in esecuzione 4 trapper, il valore è compreso tra 1 e 4);
- **state**:
    - *busy* - il processo è nello stato occupato, ad esempio durante l'elaborazione di una richiesta (predefinito);
    - *idle* - il processo è nello stato inattivo e non esegue alcuna attività.

Commenti:

- Se <mode> è il numero di un processo Zabbix non in esecuzione (ad esempio, con 5 poller in esecuzione, se <mode> è specificato come 6), tale item diventerà non supportato.
- Minimo e massimo si riferiscono alla percentuale di utilizzo di un singolo processo. Pertanto, se in un gruppo di 3 poller le percentuali di utilizzo per processo sono 2, 18 e 66, min restituirà 2 e max restituirà 66.
- I processi comunicano nella memoria condivisa ciò che stanno facendo e il processo di self-monitoring riepiloga questi dati ogni secondo. I cambiamenti di stato (busy/idle) vengono registrati al momento della modifica; pertanto, un processo che diventa busy viene registrato come tale e non modifica né aggiorna lo stato finché non diventa idle. Questo garantisce che anche i processi completamente bloccati vengano registrati correttamente come occupati al 100%.
- Attualmente, "busy" significa "non in sospensione", ma in futuro potrebbero essere introdotti ulteriori stati, ad esempio in attesa di lock, durante l'esecuzione di query al database e così via. Si noti che i poller asincroni sono considerati busy se hanno raggiunto il limite impostato dal parametro di configurazione MaxConcurrentChecksPerPoller del server/proxy.
- Su Linux e sulla maggior parte degli altri sistemi, la risoluzione è di 1/100 di secondo.

Esempi:

```
zabbix[process,poller,avg,busy] #tempo medio trascorso dai processi poller durante l'esecuzione di attivit
zabbix[process,"icmp pinger",max,busy] #tempo massimo trascorso nell'esecuzione di attività da qualsiasi p
zabbix[process,"history syncer",2,busy] #tempo trascorso nell'esecuzione di attività dal processo history
zabbix[process,trapper,count] #quantità di processi trapper attualmente in esecuzione
```

zabbix[proxy,<name>,<param>]

 Informazioni sul proxy Zabbix.
 Valore restituito: *Integer*.

Parametri:

- **name** - il nome del proxy;
- **param**:
    - *lastaccess* - il timestamp dell'ultimo messaggio heartbeat ricevuto dal proxy;
    - *delay* - da quanto tempo i valori raccolti non vengono inviati; calcolato come "ritardo del proxy" + ("ora corrente del server" - "lastaccess del proxy"), dove il "ritardo del proxy" è la differenza tra l'ora corrente del proxy e il timestamp del valore non inviato più vecchio sul proxy.

Commenti:

- Questo item non è supportato sugli host monitorati da un proxy o da un gruppo di proxy.
- La funzione `fuzzytime()` può essere utilizzata per verificare la disponibilità del proxy.

Esempio:

```
zabbix[proxy,"Germany",lastaccess] #il timestamp dell'ultimo messaggio heartbeat ricevuto dal proxy "Germa
```

zabbix[proxy,discovery]

 L'elenco dei proxy Zabbix con nome, modalità, crittografia, compressione, versione, ultima visualizzazione, numero di host, numero di item, valori richiesti al secondo (vps), stato della versione (corrente/obsoleta/non supportata), timeout per tipo di item, nome del gruppo di proxy (se il proxy appartiene a un gruppo), stato (sconosciuto/offline/online).
 Valore restituito: *oggetto JSON*.

Commento:

- Questo item non è supportato sugli host monitorati da un proxy o da un gruppo di proxy.

Esempio di valori restituiti:

```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
```

```
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
  }
]
```


zabbix[proxy group,<name>,available]

 Il numero di proxy online in un gruppo di proxy.
 Valore restituito: *Integer*.

Parametri:

- **name** - il nome del gruppo di proxy.

Commento:

- Questo item non è supportato sugli host monitorati da un proxy o da un gruppo di proxy.

zabbix[proxy group,<name>,pavailable]

 La percentuale di proxy online in un gruppo di proxy.
 Valore restituito: *Float*.

Parametri:

- **name** - il nome del gruppo di proxy.

Commento:

- Questo item non è supportato sugli host monitorati da un proxy o da un gruppo di proxy.

zabbix[proxy group,<name>,proxies]

 L'elenco dei proxy Zabbix in un gruppo di proxy con nome, modalità, crittografia, compressione, versione, ultima visualizzazione, numero di host, numero di item, valori richiesti al secondo (vps), stato della versione (corrente/obsoleta/non supportata), timeout, nome del gruppo di proxy, stato (sconosciuto/non in linea/in linea).
 Valore restituito: *JSON*.

Parametri:

- **name** - il nome del gruppo di proxy.

Commento:

- Questo item non è supportato sugli host monitorati da un proxy o da un gruppo di proxy.

zabbix[proxy group,<name>,state]

 Lo stato di un gruppo di proxy.
 Valore restituito: *0* - sconosciuto; *1* - offline; *2* - in fase di ripristino; *3* - online; *4* - in peggioramento.

Parametri:

- **name** - il nome del gruppo di proxy.

Commento:

- Questo item non è supportato sugli host monitorati da un proxy o da un gruppo di proxy.

zabbix[proxy group,discovery]

 Restituisce un elenco di proxy group con dati di configurazione e dati in tempo reale. I dati di configurazione includono il nome del proxy group, il ritardo di failover e il numero minimo di proxy online richiesti. I dati in tempo reale includono lo stato del proxy group (vedere i commenti per i dettagli), il numero di proxy online e la percentuale di proxy online.
 Valore restituito: *JSON*.

Commenti:

- Questo item non è supportato sugli host monitorati da un proxy o da un proxy group.
- Questo item non restituisce i proxy non appartenenti a un gruppo.
- Se è presente un valore non valido per `failover_delay` o `min_online`, viene restituito il valore speciale −1 per indicarlo. I valori non validi possono verificarsi se vengono utilizzate macro per la configurazione e le macro non possono essere espanse in un valore valido.
- Lo stato del proxy group viene restituito come intero: *0* - sconosciuto; *1* - offline; *2* - in ripristino; *3* - online; *4* - in peggioramento.

Esempio di valori restituiti:

```
{
    "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>]

 Statistiche sull'utilizzo del buffer di memoria del proxy.
 Valori restituiti: *Integer* (per la dimensione); *Float* (per la percentuale).

Parametri:

- **mode**:
  - *total* - dimensione totale del buffer (può essere utilizzato per verificare se il buffer di memoria è abilitato);
  - *free* - dimensione del buffer libero;
  - *pfree* - percentuale del buffer libero;
  - *used* - dimensione del buffer utilizzato;
  - *pused* - percentuale del buffer utilizzato.

Commenti:

- Restituisce l'errore 'Proxy memory buffer is disabled' quando il buffer di memoria è disabilitato;
- Questo item non è supportato sul server Zabbix.

zabbix[proxy_buffer,state,changes]

 Returns the number of state changes between disk/memory buffer modes since start.
 Return values: *Integer*; *0* - the memory buffer is disabled.

Comments:

- Frequent state changes indicate that either the memory buffer size or age must be increased;
- If the memory buffer state is monitored infrequently (for example, once a minute) then the buffer might flip its state without it being registered.

zabbix[proxy_buffer,state,current]

 Restituisce lo stato operativo corrente in cui vengono archiviati i nuovi dati.
 Valori restituiti: *0* - disco; *1* - memoria.

Commento:

- 0 viene restituito anche quando il buffer di memoria è disabilitato.

zabbix[proxy_history]

 Il numero di valori nella tabella della cronologia del proxy in attesa di essere inviati al server.
 Valori restituiti: *Integer*.

Commento:

- Questo item non è supportato sul server Zabbix.

zabbix[queue,<from>,<to>]

 Il numero di item monitorati nella coda che sono in ritardo di almeno <from> secondi, ma di meno di <to> secondi.
 Valore restituito: *Integer*.

Parametri:

- **from** - in ritardo di almeno (il valore predefinito è 6 secondi);
- **to** - in ritardo di al massimo (il valore predefinito è infinito).

Commenti:

- Sono supportati nei parametri i suffissi di tempo (`s,m,h,d,w`).

Attention:
Il proxy di Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calculation of queues during maintenance per i dettagli.

zabbix[rcache,<cache>,<mode>]

 Le statistiche di disponibilità della cache di configurazione di Zabbix.
 Valori restituiti: *Integer* (per la dimensione); *Float* (per la percentuale).

Parametri:

- **cache** - *buffer*;
- **mode**:
  - *total* - la dimensione totale del buffer;
  - *free* - la dimensione del buffer libero;
  - *pfree* - la percentuale di buffer libero;
  - *used* - la dimensione del buffer utilizzato;
  - *pused* - la percentuale di buffer utilizzato.

zabbix[requiredperformance]

 Le prestazioni richieste di Zabbix server o Zabbix proxy, in nuovi valori al secondo attesi.
 Valore restituito: *Float*.

Commenti:

- Corrisponde approssimativamente a "Required server performance, new values per second" in *Reports* > *System information*.

zabbix[stats,<ip>,<port>]

 Restituisce le metriche interne di un server o proxy Zabbix. Se <ip> e <port> sono specificati, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
 Valori restituiti: *oggetto JSON*.

Parametri:

- **ip** - l'elenco di IP/DNS/mask di rete dei server/proxy da interrogare in remoto (il valore predefinito è 127.0.0.1);
- **port** - la porta del server/proxy da interrogare in remoto (il valore predefinito è 10051).

Commenti:

- La richiesta delle statistiche sarà accettata solo dagli indirizzi elencati nel parametro `StatsAllowedIP` server/proxy dell'istanza di destinazione.
- Questo item restituisce un insieme selezionato di metriche interne. Per i dettagli, vedere Monitoraggio remoto delle statistiche di Zabbix.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Restituisce le metriche interne della coda (vedi `zabbix[queue,<from>,<to>]`) di un server Zabbix o proxy. Se <ip> e <port> sono specificati, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
 Valori restituiti: *oggetto JSON*.

Parametri:

- **ip** - l'elenco di indirizzi IP/DNS/network mask dei server/proxy da interrogare in remoto (il valore predefinito è `127.0.0.1`);
- **port** - la porta del server/proxy da interrogare in remoto (il valore predefinito è 10051);
- **from** - ritardato di almeno (il valore predefinito è 6 secondi);
- **to** - ritardato di al massimo (il valore predefinito è infinito).

Commenti:

- La richiesta stats sarà accettata solo dagli indirizzi elencati nel parametro `StatsAllowedIP` server/proxy dell'istanza di destinazione.
- Questo item restituisce un insieme selezionato di metriche interne. Per i dettagli, vedere Monitoraggio remoto delle stats di Zabbix.

> **Attention:**
> Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calcolo delle code durante la manutenzione per i dettagli.

zabbix[tcache,cache,<parameter>]

 Le statistiche di efficacia della cache della funzione trend di Zabbix.
 Valori restituiti: *Integer* (per la dimensione); *Float* (per la percentuale).

Parametri:

- **parameter**:
  - *all* - richieste totali della cache (predefinito);
  - *hits* - hit della cache;
  - *phits* - percentuale di hit della cache;
  - *misses* - miss della cache;
  - *pmisses* - percentuale di miss della cache;
  - *items* - numero di item memorizzati nella cache;
  - *requests* - numero di richieste memorizzate nella cache;
  - *pitems* - percentuale di item memorizzati nella cache rispetto a item memorizzati nella cache + richieste. Una percentuale bassa indica molto probabilmente che la dimensione della cache può essere ridotta.
  - *paccessed* - percentuale di voci accessate nelle ultime 24 ore.

Commenti:

- Questo item non è supportato su Zabbix proxy.

zabbix[triggers]

 Il numero di trigger abilitati nel database di Zabbix, con tutti gli item abilitati su host abilitati.
 Valore restituito: *Integer*.

Commenti:

- Questo item non è supportato su Zabbix proxy.

zabbix[uptime]

 L'uptime del processo del server o del proxy Zabbix in secondi.
 Valore restituito: *Integer*.

zabbix[vcache,buffer,<mode>]

 Le statistiche di disponibilità della cache dei valori di Zabbix.
 Valori restituiti: *Integer* (per la dimensione); *Float* (per la percentuale).

Parametri:

- **mode**:
  - *total* - la dimensione totale del buffer;
  - *free* - la dimensione del buffer libero;
  - *pfree* - la percentuale di buffer libero;
  - *used* - la dimensione del buffer utilizzato;
  - *pused* - la percentuale di buffer utilizzato.

Commenti:

- Questo item non è supportato su Zabbix proxy.

zabbix[vcache,cache,<parameter>]

 Le statistiche di efficacia della cache dei valori di Zabbix.
 Valore restituito: *Integer*. Se <parameter> è impostato su *mode*, il valore restituito è: *0* - modalità normale; *1* - modalità a memoria ridotta.

Parametri:

- **parameter**:
  - *requests* - il numero totale di richieste;
  - *hits* - il numero di hit della cache (valori storici prelevati dalla cache);

  - *misses* - il numero di miss della cache (valori storici prelevati dal database);
  - *mode* - la modalità operativa della cache dei valori.

Commenti:

- Una volta attivata la modalità a memoria ridotta, la cache dei valori rimarrà in questo stato per 24 ore, anche se il problema che ha attivato questa modalità viene risolto prima.
- È possibile usare questa chiave con il passo di preprocessing *Change per second* per ottenere statistiche dei valori al secondo.
- Questo item non è supportato su Zabbix proxy.

zabbixversion

 La versione di Zabbix server o proxy.
 Valore restituito: *String*. Ad esempio: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Le statistiche di disponibilità della cache vmware di Zabbix.
 Valori restituiti: *Integer* (per la dimensione); *Float* (per la percentuale).

Parametri:

- **mode**:
  - *total* - la dimensione totale del buffer;
  - *free* - la dimensione del buffer libero;
  - *pfree* - la percentuale di buffer libero;
  - *used* - la dimensione del buffer utilizzato;
  - *pused* - la percentuale di buffer utilizzato.

zabbix[vps,written]

 Il numero totale di valori dello storico scritti nel database.
 Valore restituito: *Integer*.

zabbix[wcache,<cache>,<mode>]

 Le statistiche e la disponibilità della write cache di Zabbix.
 Valori restituiti: *Integer* (per numero/dimensione); *Float* (per percentuale).

Parametri:

- **cache** - *values*, *history*, *index* o *trend*;
- **mode**:
  - con il parametro di cache *values*:
    - *all* - il numero totale di valori elaborati da server/proxy Zabbix, esclusi gli item non supportati (contatore) (predefinito);
    - *float* - il numero di valori float elaborati (contatore);
    - *uint* - il numero di valori interi senza segno elaborati (contatore);
    - *str* - il numero di valori carattere/stringa elaborati (contatore);
    - *log* - il numero di valori log elaborati (contatore);
    - *text* - il numero di valori text elaborati (contatore);
    - *bin* - il numero di valori binari elaborati (contatore);
    - *json* - il numero di valori JSON elaborati (contatore);
    - *not supported* - il numero di volte in cui l'elaborazione di un item ha portato l'item a diventare non supportato o a mantenere tale stato (contatore);
  - con i parametri di cache *history*, *index*, *trend*:
    - *pfree* - la percentuale di buffer libero (predefinito);
    - *total* - la dimensione totale del buffer;
    - *free* - la dimensione del buffer libero;
    - *used* - la dimensione del buffer utilizzato;
    - *pused* - la percentuale di buffer utilizzato.

Commenti:

- Il parametro di cache `trend` non è supportato con Zabbix proxy.
- La history cache viene utilizzata per memorizzare i valori degli item. Un numero basso indica problemi di prestazioni sul lato del database.
- La history index cache viene utilizzata per indicizzare i valori memorizzati nella history cache.
- Dopo che la history cache è stata riempita e poi svuotata, la history index cache continuerà a mantenere alcuni dati. Questo comportamento è previsto e aiuta il sistema a funzionare in modo più efficiente evitando l'elaborazione aggiuntiva necessaria per ridimensionare continuamente la memoria.
- La trend cache memorizza l'aggregato dell'ora corrente per tutti gli item che ricevono dati.

- È possibile usare la chiave `zabbix[wcache,values]` con il passo di preprocessing *Change per second* per ottenere statistiche dei valori al secondo.

Zabbix trapper

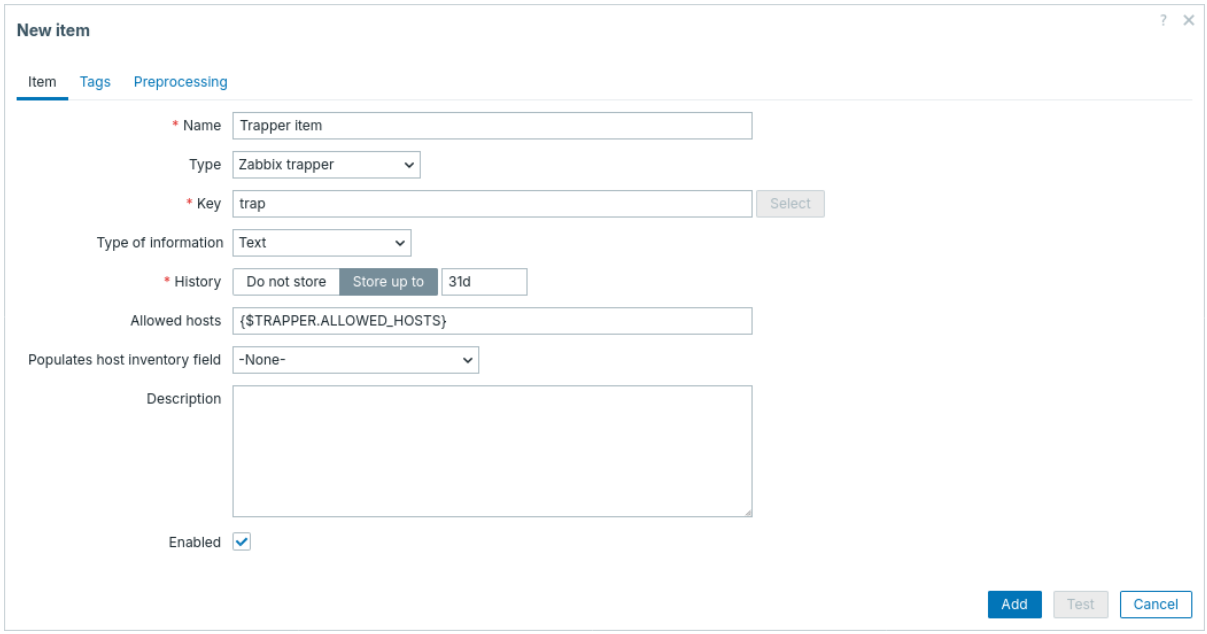
Panoramica

Gli item trapper accettano dati in ingresso invece di interrogarli. Questo è utile per qualsiasi dato che si desideri inviare a Zabbix.

Configurazione

Per configurare un item trapper:

1. Vai su *Data collection* > *Hosts*.
2. Fai clic su *Items* nella riga dell'host.
3. Fai clic su *Create item*.
4. Inserisci i parametri dell'item nel modulo di configurazione.



Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item trapper sono:

Type	Seleziona "Zabbix trapper".
Key	Inserisci una chiave che verrà usata per riconoscere l'item quando si inviano dati a Zabbix server.
Type of information	Seleziona il tipo di informazioni che corrisponderà al formato dei dati che verranno inviati.
Allowed hosts	Elenco di indirizzi IP separati da virgole (facoltativamente in notazione CIDR) o nomi DNS. Il valore predefinito è il user macro `{$TRAPPER.ALLOWED_HOSTS}`. Per impostazione predefinita, `{$TRAPPER.ALLOWED_HOSTS}` è `127.0.0.1,::1`. Puoi inserire gli indirizzi direttamente nel campo oppure modificare il valore del macro. Se lasciato vuoto, le connessioni in ingresso saranno rifiutate da tutti gli host. Se specificato, le connessioni in ingresso saranno accettate solo dagli host elencati qui. Se il supporto IPv6 è abilitato, allora '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' sono trattati allo stesso modo e '::/0' consentirà qualsiasi indirizzo IPv4 o IPv6. '0.0.0.0/0' può essere usato per consentire qualsiasi indirizzo IPv4. Nota che gli "indirizzi IPv6 compatibili con IPv4" (prefisso 0000::/96) sono supportati ma deprecati da RFC4291. Esempio: 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} Sono supportati spazi, user macro e host macro {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Prima di inviare i valori, potrebbe essere necessario attendere fino a 60 secondi dopo aver salvato l'item, finché Zabbix server non applica le modifiche da un aggiornamento della cache di configurazione.

Invio dei dati

L'invio dei dati a Zabbix server o proxy è possibile utilizzando l'utilità Zabbix sender o il protocollo di Zabbix sender. L'invio dei dati a Zabbix server è inoltre possibile utilizzando il metodo API `history.push`.

Zabbix sender

Per inviare dati a Zabbix server o proxy usando l'utility Zabbix sender, puoi eseguire il seguente comando per inviare il valore "test value":

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Per inviare il valore "test value", vengono usate le seguenti opzioni del comando:

- `-z` per specificare l'indirizzo IP di Zabbix server
- `-p` per specificare il numero di porta di Zabbix server (10051 per impostazione predefinita)
- `-s` per specificare l'host (assicurati di usare il nome tecnico invece del nome host visibile)
- `-k` per specificare la chiave dell'item configurato nell'item trapper
- `-o` per specificare il valore da inviare

Attention:
Il processo Zabbix trapper non espande le macro usate nella chiave dell'item per verificare l'esistenza della corrispondente chiave dell'item per l'host di destinazione.

Per ulteriori informazioni sulla comunicazione tra Zabbix sender e Zabbix server o proxy, consulta *Zabbix sender protocol*.

history.push

Per inviare dati a Zabbix server utilizzando il metodo API `history.push`, è possibile effettuare la seguente richiesta HTTP POST contenente alcuni valori di test:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

Se la richiesta è corretta, la risposta restituita dall'API potrebbe apparire come segue:

```
{
    "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
}
```

Gli errori nei dati della risposta indicano che l'invio dei dati per specifici item non ha superato la validazione da parte di Zabbix server.
Ciò può accadere per i seguenti motivi:

- l'utente che invia i dati non dispone del permesso di *lettura* per l'host dell'item;

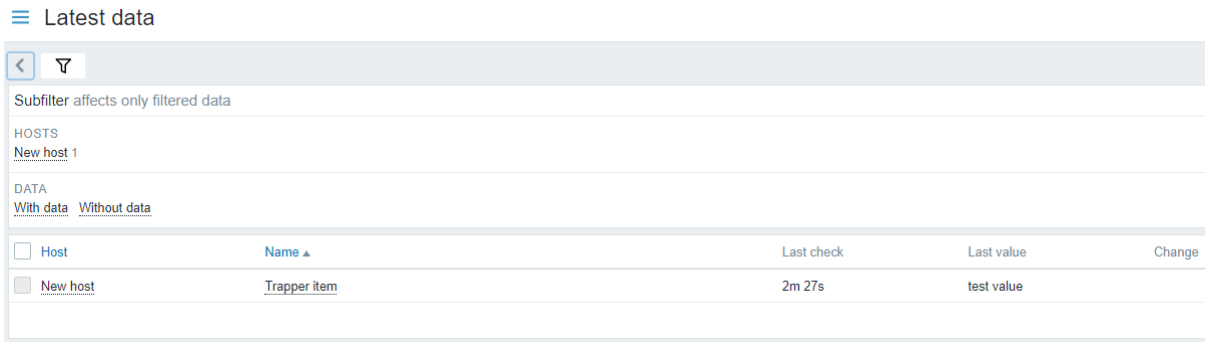
- l'host è disabilitato o in manutenzione senza raccolta dati;
- l'item non esiste o non è ancora incluso nella cache di configurazione del server;
- l'item è disabilitato oppure il suo tipo è diverso da Zabbix trapper o HTTP agent (con trapping abilitato);
- l'IP o il DNS dell'utente non è impostato nell'elenco *Allowed hosts* dell'item;
- un altro item ha un valore con un timestamp duplicato a livello di nanosecondi.

L'assenza di errori indica che i valori inviati sono stati accettati per l'elaborazione, che include il preprocessing (se presente), l'elaborazione dei trigger e il salvataggio nel database.
Si noti che anche l'elaborazione di un valore accettato può non riuscire (ad esempio, durante il preprocessing), con conseguente scarto del valore.

Per ulteriori informazioni su come lavorare con Zabbix API, vedere API.

Visualizzazione dei dati

Una volta inviati i dati, puoi andare su *Monitoring* > *Latest data* per vedere il risultato:



Note:
Se viene inviato un singolo valore numerico, il graph dei dati mostrerà una linea orizzontale a sinistra e a destra del punto temporale del valore.

Controllo esterno

Panoramica

Un external check è un controllo eseguito da Zabbix server tramite l'esecuzione di uno script shell o di un binario. Tuttavia, quando gli host sono monitorati da un Zabbix proxy, gli external check vengono eseguiti dal proxy.

Gli external check non richiedono alcun agent in esecuzione sull'host monitorato.

La sintassi della chiave dell'item è:

`script[<parameter1>,<parameter2>,...]`

Dove:

ARGUMENT	DEFINITION
script	Nome di uno script shell o di un binario.
parameter(s)	Parametri opzionali della riga di comando.

Se non si desidera passare alcun parametro allo script, è possibile usare:

```
script[] or
script
```

Zabbix server o proxy cercherà la directory specificata per gli script esterni ed eseguirà il comando (vedere il parametro `ExternalScripts` nel file di configurazione di Zabbix server/proxy). Il comando verrà eseguito con lo stesso utente di Zabbix server/proxy, quindi eventuali permessi di accesso o variabili di ambiente devono essere gestiti in uno script wrapper, se necessario. Anche i permessi sul comando devono consentire a quell'utente di eseguirlo. Sono disponibili per l'esecuzione solo i comandi presenti nella directory specificata.

Warning:
Non abusare degli external check, poiché ogni script richiede l'avvio di un processo fork da parte di Zabbix server/proxy, e l'esecuzione di molti script può ridurre significativamente le prestazioni di Zabbix.

Esempio di utilizzo

Esecuzione dello script `check_oracle.sh` con il primo parametro `'-h'`. Il secondo parametro verrà sostituito con l'indirizzo IP o il nome DNS, a seconda della selezione nelle proprietà dell'host.

`check_oracle.sh["-h","{HOST.CONN}"]`

Supponendo che l'host sia configurato per usare l'indirizzo IP, il server/proxy Zabbix eseguirà:

`check_oracle.sh '-h' '192.168.1.4'`

Risultato del controllo esterno

Il valore restituito da un controllo esterno è l'output standard insieme all'errore standard prodotto dal controllo.

> **Attention:**
> Un item che restituisce testo (informazioni di tipo carattere, log o testo) non diventerà unsupported in caso di output sull'errore standard.

Il valore restituito è limitato a 16 MB (inclusi gli spazi finali che vengono troncati); si applicano anche i limiti del database.

Se lo script richiesto non viene trovato oppure Zabbix server/proxy non dispone delle autorizzazioni per eseguirlo, l'item diventerà unsupported e verrà visualizzato un messaggio di errore corrispondente.

In caso di timeout, l'item diventerà unsupported, verrà visualizzato un messaggio di errore corrispondente e il processo creato tramite fork per lo script verrà terminato.

Monitor del database

Panoramica

Il tipo di item *Database monitor* nel frontend di Zabbix viene utilizzato per il monitoraggio ODBC (controlli ODBC).

ODBC è un'API middleware del linguaggio di programmazione C per l'accesso ai sistemi di gestione di database (DBMS). Il concetto ODBC è stato sviluppato da Microsoft e successivamente portato su altre piattaforme.

Zabbix può interrogare qualsiasi database supportato da ODBC. Per farlo, Zabbix non si connette direttamente ai database, ma utilizza l'interfaccia ODBC e i driver configurati in ODBC. Questo consente un monitoraggio più efficiente di diversi database per molteplici scopi (ad esempio, il controllo di code specifiche del database, statistiche di utilizzo, ecc.).

Zabbix supporta unixODBC, che è una delle implementazioni open source dell'API ODBC più comunemente utilizzate.

> **Attention:**
> Vedi anche: problemi noti per i controlli ODBC.

Installazione di unixODBC

Il modo consigliato per installare unixODBC è utilizzare i repository di pacchetti predefiniti del sistema operativo Linux. Nelle distribuzioni Linux più diffuse, unixODBC è incluso per impostazione predefinita nel repository dei pacchetti. Se i pacchetti non sono disponibili, i file sorgente possono essere ottenuti dalla homepage di unixODBC: http://www.unixodbc.org/download.html.

Per installare unixODBC, usa il gestore di pacchetti del sistema che preferisci:

```
##### Per sistemi Ubuntu/Debian:
apt install unixodbc unixodbc-dev

##### Per sistemi basati su RedHat/Fedora:
dnf install unixODBC unixODBC-devel

##### Per sistemi basati su SUSE:
zypper in unixODBC-devel
```

> **Attention:**
> Il pacchetto `unixodbc-dev` o `unixODBC-devel` è necessario per compilare Zabbix con il supporto unixODBC. Per abilitare il supporto ODBC, Zabbix deve essere compilato con la seguente opzione di configurazione:

> `--with-unixodbc[=ARG] # Usa il driver ODBC tramite il pacchetto unixODBC.`

Installazione dei driver unixODBC

Il driver di database unixODBC deve essere installato per il database che verrà monitorato. Per un elenco dei database e dei driver supportati, vedere la homepage di unixODBC: http://www.unixodbc.org/drivers.html.

> **Note:**
> In alcune distribuzioni Linux, i driver di database sono inclusi nei repository dei pacchetti.

MySQL

Per installare il driver di database MySQL unixODBC, utilizzare il gestore di pacchetti del sistema scelto:

```
##### Per sistemi Ubuntu/Debian:
apt install odbc-mariadb

##### Per sistemi basati su RedHat/Fedora:
dnf install mariadb-connector-odbc

##### Per sistemi basati su SUSE:
zypper install mariadb-connector-odbc
```

Per installare il driver di database senza un gestore di pacchetti, fare riferimento alla documentazione MySQL per `mysql-connector-odbc`, oppure alla documentazione MariaDB per `mariadb-connector-odbc`.

PostgreSQL

Per installare il driver di database PostgreSQL unixODBC, utilizzare il gestore di pacchetti del sistema scelto:

```
##### Per sistemi Ubuntu/Debian:
apt install odbc-postgresql

##### Per sistemi basati su RedHat/Fedora:
dnf install postgresql-odbc

##### Per sistemi basati su SUSE:
zypper install psqlODBC
```

Per installare il driver di database senza un gestore di pacchetti, fare riferimento alla documentazione di PostgreSQL.

Oracle

Per installare il driver di database unixODBC, fare riferimento alla documentazione Oracle.

MSSQL

Per installare il driver di database MSSQL unixODBC, utilizzare il gestore di pacchetti del sistema scelto:

```
##### Per sistemi Ubuntu/Debian:
apt install tdsodbc

##### Per sistemi basati su RedHat/Fedora (pacchetti EPEL: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### Per sistemi basati su SUSE:
zypper install libtdsodbc0
```

Per installare il driver di database senza un gestore di pacchetti, fare riferimento alla guida utente di FreeTDS.

Configurazione di unixODBC

Per configurare unixODBC, è necessario modificare i file `odbcinst.ini` e `odbc.ini`. È possibile verificare il percorso di questi file eseguendo il seguente comando:

```
odbcinst -j
```

Il risultato del comando dovrebbe contenere informazioni simili alle seguenti:

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

odbcinst.ini

Il file `odbcinst.ini` elenca i driver di database ODBC installati. Se `odbcinst.ini` manca, è necessario crearlo manualmente.

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parameter	Description
TEST_MYSQL	Nome del driver di database.
Description	Descrizione del driver di database.
Driver	Percorso della libreria del driver di database.
FileUsage	Determina se il driver di database supporta la connessione a un server di database senza il supporto per l'accesso ai file locali (0); supporta la lettura dei dati dai file (1); supporta la scrittura dei dati nei file (2).
Threading	Livello di serializzazione dei thread. Supportato per PostgreSQL. Dalla versione 1.6, se il gestore dei driver è compilato con il supporto ai thread, è possibile aggiungere un'altra voce del driver.

odbc.ini

Il file `odbc.ini` viene utilizzato per configurare le origini dati. Si noti che l'elenco dei parametri supportati dipende dal driver del database (ad esempio, i database Oracle possono utilizzare ServerName invece di Server, ecc.).

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Parameter	Description
TEST_MYSQL	Nome dell'origine dati (DSN).
Description	Descrizione dell'origine dati.
Driver	Nome del driver del database (come specificato in `odbcinst.ini`).
Server	IP/DNS del server del database.
User	Utente del database per la connessione.
Password	Password dell'utente del database.
Port	Porta di connessione al database.
Socket	Socket di connessione al database.
Database	Nome del database.

Per altre possibili opzioni dei parametri di configurazione, vedere la documentazione MySQL.

Il file `odbc.ini` per PostgreSQL può contenere parametri aggiuntivi:

```
[TEST_PSQL]
Description=Database di test PostgreSQL
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=8.0+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
```

```
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

Parametro	Descrizione
ReadOnly	Specifica se la connessione al database consente solo operazioni di lettura (query `SELECT`) e limita le modifiche (istruzioni `INSERT`, `UPDATE` e `DELETE`); utile negli scenari in cui i dati devono rimanere invariati.
Protocol	Versione del protocollo backend di PostgreSQL (ignorata quando si utilizzano connessioni SSL).
ShowOidColumn	Specifica se includere l'Object ID (OID) in SQLColumns.
FakeOidIndex	Specifica se creare un indice univoco fittizio su OID.
RowVersioning	Specifica se consentire alle applicazioni di rilevare se i dati sono stati modificati da altri utenti mentre si sta tentando di aggiornare una riga. Si noti che questo parametro può velocizzare il processo di aggiornamento, poiché, per aggiornare una riga, non è necessario specificare ogni singola colonna nella clausola `WHERE`.
ShowSystemTables	Specifica se il driver del database deve trattare le tabelle di sistema come tabelle normali in SQLTables; utile per l'accessibilità, in quanto consente la visibilità delle tabelle di sistema.
Fetch	Specifica se il driver deve utilizzare automaticamente declare cursor/fetch per gestire le istruzioni `SELECT` e mantenere una cache di 100 righe.
BoolsAsChar	Controlla la mappatura dei tipi Boolean. Se impostato su "Yes", i valori Boolean vengono mappati a `SQL_CHAR`; altrimenti, vengono mappati a `SQL_BIT`.
SSLmode	Specifica la modalità SSL per la connessione.
ConnSettings	Impostazioni aggiuntive inviate al backend al momento della connessione.

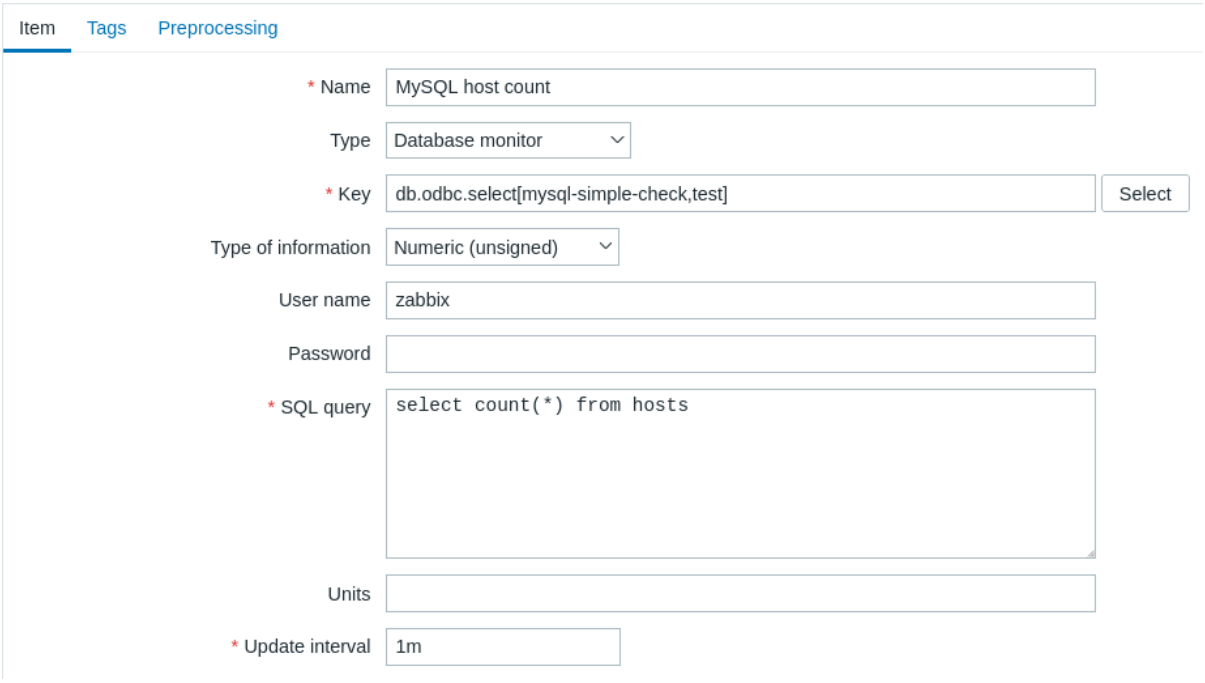
Test della connessione ODBC

Per verificare se la connessione ODBC funziona correttamente, puoi utilizzare l'utilità `isql` (inclusa nel pacchetto `unixODBC`):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Configurazione dell'item nel frontend di Zabbix

Configura un item di **monitoraggio database**.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Per gli item di monitoraggio database, devi specificare:

Type	Seleziona qui "Database monitor".
Key	Inserisci una delle chiavi item supportate: **db.odbc.select[]** - questo item restituisce un solo valore (la prima colonna della prima riga del risultato della query SQL); **db.odbc.get[]** - questo item restituisce più righe/colonne in formato JSON; **db.odbc.discovery[]** - questo item restituisce dati di low-level discovery.
User name	Inserisci il nome utente del database (fino a 255 caratteri). Questo parametro è facoltativo se il nome utente del database è specificato nel file `odbc.ini`. Se viene usata una stringa di connessione e il campo *User name* non è vuoto, viene aggiunto alla stringa di connessione come `UID=<user>`.
Password	Inserisci la password dell'utente del database (fino a 255 caratteri). Questo parametro è facoltativo se la password è specificata nel file `odbc.ini`. Se viene usata una stringa di connessione e il campo *Password* non è vuoto, viene aggiunto alla stringa di connessione come `PWD=<password>`. In questo campo sono supportati caratteri speciali. Per tutti i database tranne Oracle, le password contenenti `}` devono essere racchiuse tra parentesi graffe `{}`. La password verrà aggiunta alla stringa di connessione dopo il nome utente, ad esempio `UID=<username>;PWD=P?;)*word`. Per testare la stringa risultante, puoi eseguire il seguente comando: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Inserisci la query SQL. Nota che con `db.odbc.select[]`, la query deve restituire un solo valore.
Type of information	Seleziona qui il tipo di informazioni che verrà restituito dalla query. Se il tipo di informazioni viene selezionato in modo errato, l'item diventerà non supportato.

Note importanti

- Gli item di monitoraggio del database diventeranno non supportati se non vengono avviati processi *odbc poller* nella configurazione del server o del proxy. Per attivare gli ODBC poller, impostare il parametro `StartODBCPollers` nel file di configurazione di Zabbix server oppure, per i controlli eseguiti dal proxy, nel file di configurazione di Zabbix proxy.
- Il valore del parametro *Timeout* nel modulo di configurazione dell'item viene usato come timeout di accesso ODBC e come timeout di esecuzione della query. Si noti che queste impostazioni di timeout potrebbero essere ignorate se il driver ODBC installato non le supporta.
- Il comando SQL deve restituire un result set, come qualsiasi query che utilizza l'istruzione `select`. La sintassi della query dipenderà dall'RDBMS che la elaborerà. La sintassi della richiesta a una procedura di archiviazione deve iniziare con la parola chiave `call`.

Dettagli della chiave item

I parametri senza parentesi angolari sono obbligatori. I parametri contrassegnati dalle parentesi angolari **< >** sono facoltativi.

db.odbc.select[<breve descrizione univoca>,<dsn>,<stringa di connessione>]

 Restituisce un valore, cioè la prima colonna della prima riga del risultato della query SQL.
 Valore restituito: dipende dalla query SQL.

Parametri:

- **breve descrizione univoca** - una breve descrizione univoca per identificare l'item (da utilizzare nei trigger, ecc.);
- **dsn** - il nome dell'origine dati (come specificato in `odbc.ini`);
- **stringa di connessione** - la stringa di connessione (può contenere argomenti specifici del driver).

Commenti:

- Sebbene `dsn` e `stringa di connessione` siano parametri facoltativi, è richiesto almeno uno dei due; se entrambi sono definiti, `dsn` verrà ignorato.
- Se una query restituisce più di una colonna, viene letta solo la prima colonna. Se una query restituisce più di una riga, viene letta solo la prima riga.

db.odbc.get[<unique short description>,<dsn>,<connection string>]

 Trasforma il risultato della query SQL in un array JSON.
 Valore restituito: *oggetto JSON*.

Parametri:

- **unique short description** - una breve descrizione univoca per identificare l'item (da usare nei trigger, ecc.);
- **dsn** - il nome dell'origine dati (come specificato in `odbc.ini`);
- **connection string** - la stringa di connessione (può contenere argomenti specifici del driver).

Commenti:

- Sebbene `dsn` e `connection string` siano parametri facoltativi, è richiesto almeno uno dei due; se entrambi sono definiti, `dsn` verrà ignorato.
- È possibile restituire più righe/colonne in formato JSON. Questo item può essere usato come master item che raccoglie tutti i dati in una singola chiamata di sistema, mentre il preprocessing JSONPath può essere usato negli item dipendenti per estrarre valori individuali. Per ulteriori informazioni, vedere un esempio del formato restituito, usato nella low-level discovery.

Esempio:

```
# Connessione per il driver ODBC MySQL 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<breve descrizione univoca>,<dsn>,<stringa di connessione>]

 Trasforma il risultato della query SQL in un array JSON, utilizzato per il low-level discovery. I nomi delle colonne del risultato della query vengono convertiti in nomi di macro di low-level discovery associati ai valori dei campi rilevati. Queste macro possono essere utilizzate per creare prototipi di item, trigger, ecc.
 Valore restituito: *oggetto JSON*.

Parametri:

- **breve descrizione univoca** - una breve descrizione univoca per identificare l'item (da utilizzare nei trigger, ecc.);
- **dsn** - il nome dell'origine dati (come specificato in `odbc.ini`);
- **stringa di connessione** - la stringa di connessione (può contenere argomenti specifici del driver).

Commenti:

- Sebbene `dsn` e `stringa di connessione` siano parametri facoltativi, è necessario specificarne almeno uno; se entrambi sono definiti, `dsn` verrà ignorato.

Messaggi di errore

I messaggi di errore ODBC sono strutturati in campi per fornire informazioni dettagliate. Ad esempio, un messaggio di errore potrebbe avere questo aspetto:

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "`Cannot execute ODBC query`" - messaggio di Zabbix
- "`[SQL_ERROR]`" - codice di ritorno ODBC
- "`[42601]`" - SQLState
- "`[7]`" - codice di errore nativo
- "`[ERROR: syntax error at or near ";"; Error while executing the query]`" - messaggio di errore nativo

Si noti che la lunghezza del messaggio di errore è limitata a 2048 byte, quindi il messaggio può essere troncato. Se è presente più di un record diagnostico ODBC, Zabbix tenta di concatenarli (separati da `|`) per quanto consentito dal limite di lunghezza.

agent HTTP

Panoramica

Questo tipo di item consente il polling dei dati tramite il protocollo HTTP/HTTPS. È anche possibile il trapping usando l'utilità Zabbix sender o il protocollo di Zabbix sender (per inviare dati a Zabbix server o proxy), oppure usando il metodo API `history.push` (per inviare dati a Zabbix server).

I controlli degli item HTTP vengono eseguiti da Zabbix server. Tuttavia, quando gli host sono monitorati da un proxy Zabbix, i controlli degli item HTTP vengono eseguiti dal proxy.

I controlli degli item HTTP non richiedono alcun agent in esecuzione sull'host monitorato.

HTTP agent supporta sia HTTP che HTTPS. Zabbix seguirà facoltativamente i reindirizzamenti (vedere l'opzione *Segui reindirizzamenti* più avanti). Il numero massimo di reindirizzamenti è impostato in modo fisso a 10 (usando l'opzione cURL `CURLOPT_MAXREDIRS`).

> **Attention:**
> Zabbix server/proxy deve essere inizialmente configurato con il supporto cURL (libcurl).

I controlli HTTP vengono eseguiti in modo asincrono: non è necessario ricevere la risposta a una richiesta prima che vengano avviati altri controlli. Anche la risoluzione DNS è asincrona.

La concorrenza massima dei controlli asincroni è 1000 (definita da MaxConcurrentChecksPerPoller).

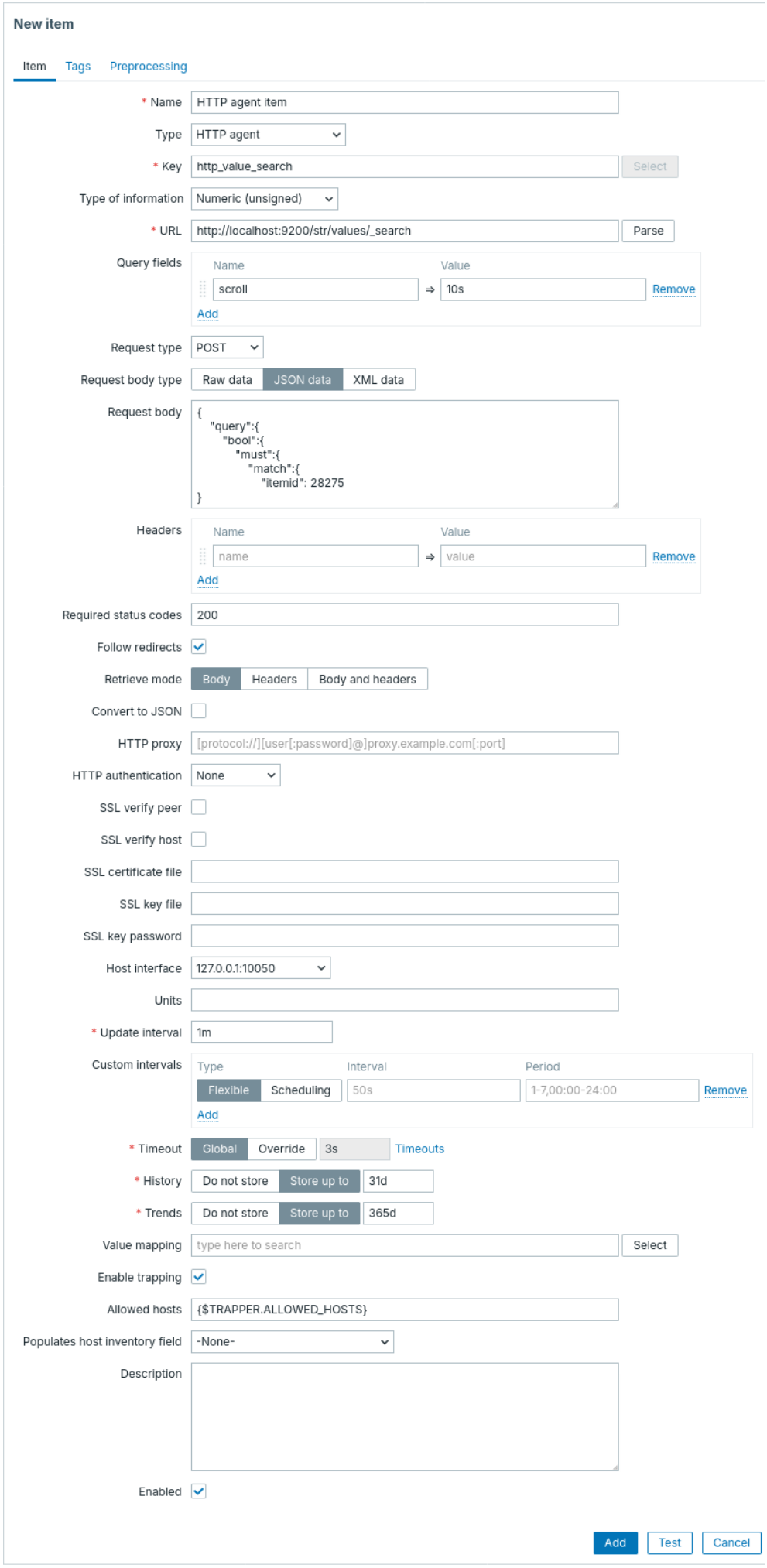
Il numero di poller asincroni di HTTP agent è definito dal parametro StartHTTPAgentPollers.

La funzionalità delle connessioni persistenti di cURL è stata aggiunta ai controlli HTTP agent a partire da Zabbix 7.0.

Configurazione

Per configurare un item HTTP:

1. Vai a *Raccolta dati* > *Host*.
2. Fai clic su *Item* nella riga dell'host.
3. Fai clic su *Crea item*.
4. Inserisci i parametri dell'item nel modulo.


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

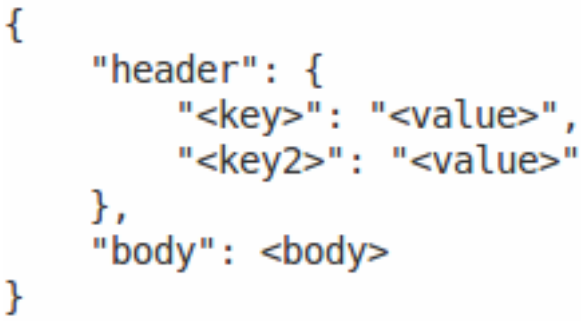
Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item HTTP sono:

Parametro	Descrizione
Tipo	Seleziona qui **HTTP agent**.
Chiave	Inserisci una chiave univoca per l'item.

URL	URL a cui connettersi e da cui recuperare i dati. Ad esempio:
	`https://www.example.com`
	`http://www.example.com/download`
	I nomi di dominio possono essere specificati utilizzando caratteri Unicode. Vengono automaticamente convertiti in punycode ASCII durante l'esecuzione del controllo HTTP.
	Il pulsante *Analizza* può essere utilizzato per separare i campi di query facoltativi (come `?key=value&limit=10`) dall'URL, spostando attributi e valori in *Campi di query* per la codifica automatica dell'URL.
	Per motivi di sicurezza, non includere nomi utente o password reali negli URL. Utilizza invece le macro utente per evitare di esporre le credenziali.
	Limitato a 2048 caratteri.
	Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello.
	Imposta l'opzione cURL `CURLOPT_URL`.

Campi di query	Variabili per l'URL (vedi sopra). Specificati come coppie attributo-valore. I valori vengono codificati automaticamente nell'URL. I valori delle macro vengono risolti e quindi codificati automaticamente nell'URL. Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello. Imposta l'opzione cURL `CURLOPT_URL`.
Tipo di richiesta	Seleziona il tipo di metodo della richiesta: *GET*, *POST*, *PUT* o *HEAD*.
Tipo di corpo della richiesta	Seleziona il tipo di corpo della richiesta: **Dati grezzi** - corpo personalizzato della richiesta HTTP; le macro vengono sostituite, ma non viene eseguita alcuna codifica. **Dati JSON** - corpo della richiesta HTTP in formato JSON. Le macro possono essere utilizzate come stringa, numero, true e false; le macro utilizzate come stringhe devono essere racchiuse tra virgolette doppie. I valori delle macro vengono risolti e quindi sottoposti automaticamente a escape. Se negli header non è specificato "Content-Type", il valore predefinito sarà "Content-Type: application/json". **Dati XML** - corpo della richiesta HTTP in formato XML. Le macro possono essere utilizzate come nodo di testo, attributo o sezione CDATA. I valori delle macro vengono risolti e quindi sottoposti automaticamente a escape in un nodo di testo e in un attributo. Se negli header non è specificato "Content-Type", il valore predefinito sarà "Content-Type: application/xml".
Corpo della richiesta	Inserisci il corpo della richiesta. Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello.
Header	Header HTTP personalizzati da inviare durante l'esecuzione di una richiesta. Specificati come coppie attributo-valore. Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello. Imposta l'opzione cURL `CURLOPT_HTTPHEADER`.
Codici di stato richiesti	Elenco dei codici di stato HTTP previsti. Se Zabbix riceve un codice non presente nell'elenco, l'item diventerà non supportato. Se vuoto, non viene eseguito alcun controllo. Ad esempio: 200,201,210-299. Macro supportate nell'elenco: macro utente, macro di discovery di basso livello. Utilizza l'opzione cURL `CURLINFO_RESPONSE_CODE`.
Segui reindirizzamenti	Seleziona la casella per seguire i reindirizzamenti HTTP. Imposta l'opzione cURL `CURLOPT_FOLLOWLOCATION`.

Parametro	Descrizione
Modalità di recupero	Seleziona la parte della risposta da recuperare: **Corpo** - solo il corpo **Header** - solo gli header **Corpo e header** - corpo e header
Converti in JSON	Gli header vengono salvati come coppie attributo-valore sotto la chiave "header" se *Modalità di recupero* non è impostata su *Corpo*. Se viene rilevato 'Content-Type: application/json', il corpo viene salvato come oggetto; altrimenti viene memorizzato come stringa, ad esempio: 
Proxy HTTP	Puoi specificare un proxy HTTP da utilizzare, nel formato `[protocol://][username[:password]@]proxy.example.com[:port]`. Il prefisso facoltativo `protocol://` può essere utilizzato per specificare protocolli proxy alternativi (ad esempio `https`, `socks4`, `socks5`; consulta la documentazione; il supporto per il prefisso del protocollo è stato aggiunto in cURL 7.21.7). Se non viene specificato alcun protocollo, il proxy verrà trattato come proxy HTTP. Se specifichi il protocollo errato, la connessione avrà esito negativo e l'item diventerà non supportato. Per impostazione predefinita viene utilizzata la porta 1080. Se specificato, il proxy sovrascriverà le variabili d'ambiente relative al proxy, come `http_proxy` e `HTTPS_PROXY`. Se non specificato, il proxy non sovrascriverà le variabili d'ambiente relative al proxy. Il valore inserito viene passato "così com'è", senza eseguire controlli di validità. Nota che con il proxy HTTP è supportata solo l'autenticazione semplice. Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello. Imposta l'opzione cURL `CURLOPT_PROXY`.
Autenticazione HTTP	Seleziona l'opzione di autenticazione: **Nessuna** - non viene utilizzata alcuna autenticazione; **Basic** - viene utilizzata l'autenticazione di base; **NTLM** - viene utilizzata l'autenticazione NTLM (Windows NT LAN Manager); **Kerberos** - viene utilizzata l'autenticazione Kerberos (vedi anche: Configurazione di Kerberos con Zabbix); **Digest** - viene utilizzata l'autenticazione Digest. Imposta l'opzione cURL `CURLOPT_HTTPAUTH`.
Nome utente	Inserisci il nome utente (fino a 255 caratteri). Questo campo è disponibile se *Autenticazione HTTP* è impostata su *Basic*, *NTLM*, *Kerberos* o *Digest*. Sono supportate le macro utente e le macro di discovery di basso livello.
Password	Inserisci la password dell'utente (fino a 255 caratteri). Questo campo è disponibile se *Autenticazione HTTP* è impostata su *Basic*, *NTLM*, *Kerberos* o *Digest*. Sono supportate le macro utente e le macro di discovery di basso livello.
Verifica peer SSL	Seleziona la casella per verificare il certificato SSL del web server. Il certificato del server viene verificato rispetto ai certificati delle autorità di certificazione (CA) attendibili presenti nella posizione CA a livello di sistema. Puoi sovrascrivere la posizione dei file CA utilizzando il parametro di configurazione SSLCALocation del server o del proxy Zabbix. Imposta l'opzione cURL `CURLOPT_SSL_VERIFYPEER`.
Verifica host SSL	Seleziona la casella per verificare che il campo *Common Name* o il campo *Subject Alternate Name* del certificato del web server corrisponda. Imposta l'opzione cURL `CURLOPT_SSL_VERIFYHOST`.

Parametro	Descrizione
File del certificato SSL	Nome del file del certificato SSL utilizzato per l'autenticazione del client. Il file del certificato deve essere in formato PEM[1]. Se il file del certificato contiene anche la chiave privata, lascia vuoto il campo *File della chiave SSL*. Se la chiave è crittografata, specifica la password nel campo *Password della chiave SSL*. La directory contenente questo file è specificata dal parametro di configurazione `SSLCertLocation` del server o del proxy Zabbix. Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello. Imposta l'opzione cURL `CURLOPT_SSLCERT`.
File della chiave SSL	Nome del file della chiave privata SSL utilizzato per l'autenticazione del client. Il file della chiave privata deve essere in formato PEM[1]. La directory contenente questo file è specificata dal parametro di configurazione `SSLKeyLocation` del server o del proxy Zabbix. Macro supportate: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macro utente, macro di discovery di basso livello. Imposta l'opzione cURL `CURLOPT_SSLKEY`.
Password della chiave SSL	Password del file della chiave privata SSL. Macro supportate: macro utente, macro di discovery di basso livello. Imposta l'opzione cURL `CURLOPT_KEYPASSWD`.
Timeout	Zabbix non dedicherà più del tempo impostato all'elaborazione dell'URL (1-600 secondi). In realtà, questo parametro definisce il tempo massimo per stabilire una connessione all'URL e il tempo massimo per eseguire una richiesta HTTP. Pertanto, Zabbix non dedicherà più di 2 x *Timeout* secondi a un singolo controllo. Imposta l'opzione cURL `CURLOPT_TIMEOUT`. Per ulteriori informazioni sul parametro *Timeout*, consulta gli attributi generali degli item.
Abilita trapping	Selezionando questa casella, l'item funzionerà anche come item trapper e accetterà i dati inviati al server o al proxy Zabbix utilizzando l'utility Zabbix sender o il protocollo Zabbix sender, oppure accetterà i dati inviati al server Zabbix utilizzando il metodo API `history.push`. Per ulteriori informazioni sull'invio dei dati, consulta: Item trapper.
Host consentiti	Visibile solo se la casella *Abilita trapping* è selezionata. Elenco di indirizzi IP separati da virgole (facoltativamente in notazione CIDR) o nomi DNS. Il valore predefinito è la macro utente `{$TRAPPER.ALLOWED_HOSTS}`. Per impostazione predefinita, `{$TRAPPER.ALLOWED_HOSTS}` è `127.0.0.1,::1`. Puoi inserire direttamente gli indirizzi nel campo oppure modificare il valore della macro. Se lasciato vuoto, le connessioni in ingresso verranno rifiutate da tutti gli host. Se specificato, le connessioni in ingresso verranno accettate solo dagli host elencati qui. Se il supporto IPv6 è abilitato, `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` vengono trattati allo stesso modo e `::/0` consentirà qualsiasi indirizzo IPv4 o IPv6. `0.0.0.0/0` può essere utilizzato per consentire qualsiasi indirizzo IPv4. Nota che gli "indirizzi IPv6 compatibili con IPv4" (prefisso `0000::/96`) sono supportati, ma deprecati da RFC4291. Esempio: `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}` In questo campo sono consentiti spazi e macro utente. Macro host: `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.IP}`, `{HOST.DNS}`, `{HOST.CONN}` sono consentite in questo campo.

Note:
Se il campo *Proxy HTTP* viene lasciato vuoto, un altro modo per utilizzare un proxy HTTP consiste nell'impostare le variabili d'ambiente relative al proxy.
Per HTTP - imposta la variabile d'ambiente `http_proxy` per l'utente del server Zabbix. Ad esempio:
 `http_proxy=http://proxy_ip:proxy_port`.
Per HTTPS - imposta la variabile d'ambiente `HTTPS_PROXY`. Ad esempio:
 `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Ulteriori dettagli sono disponibili eseguendo un comando shell: *# man curl*.

Attention:
[1] Zabbix supporta solo file di certificato e di chiave privata in formato PEM. Se i dati del certificato e della chiave privata si trovano in un file in formato PKCS #12 (solitamente con estensione `*.p12` o `*.pfx`), puoi generare il file PEM utilizzando i seguenti comandi:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Esempi

Esempio 1

Invia semplici richieste GET per recuperare dati da servizi come Elasticsearch:

- Crea un item GET con URL: `localhost:9200/?pretty`
- Osserva la risposta:

```
{
  "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"
}
```

- Ora estrai il numero di versione usando un passaggio di preprocessing JSONPath: `$.version.number`

Esempio 2

Invia semplici richieste POST per recuperare dati da servizi come Elasticsearch:

- Crea un item POST con URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Configura il seguente corpo POST per ottenere il carico del processore (media di 1 min per core)

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- Ricevuto:

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
```

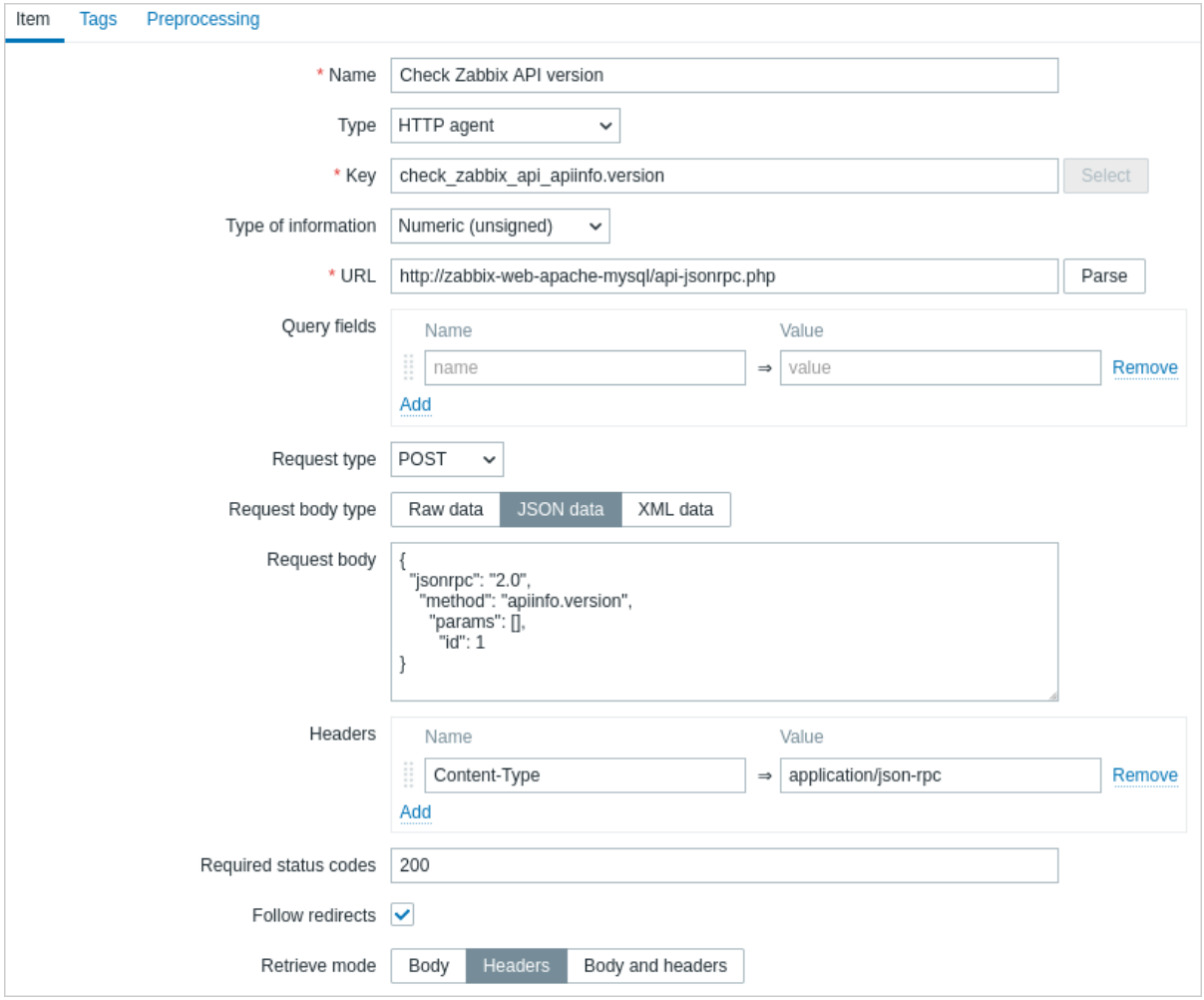
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_id": "dqX9VWEBV6sEKSMyk6sw",
                "_score": 1.0,
                "_source": {
                    "itemid": 28275,
                    "value": "0.138750",
                    "clock": 1517566136,
                    "ns": 25388713,
                    "ttl": 604800
                }
            }]
        }
    }
```


- Ora usa un passaggio di preprocessing JSONPath per ottenere il valore dell'item: `$.hits.hits[0]._source.value`

Esempio 3

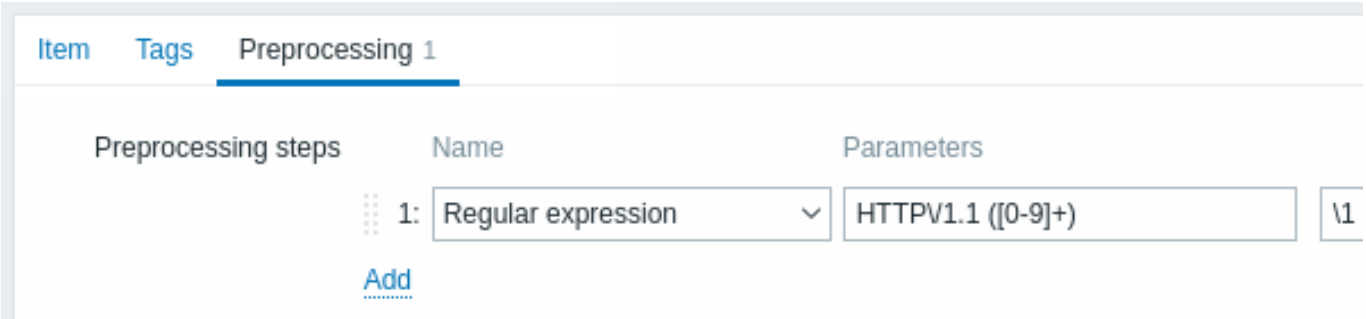
Verifica se Zabbix API è attiva, usando apiinfo.version.

- Configurazione dell'item:

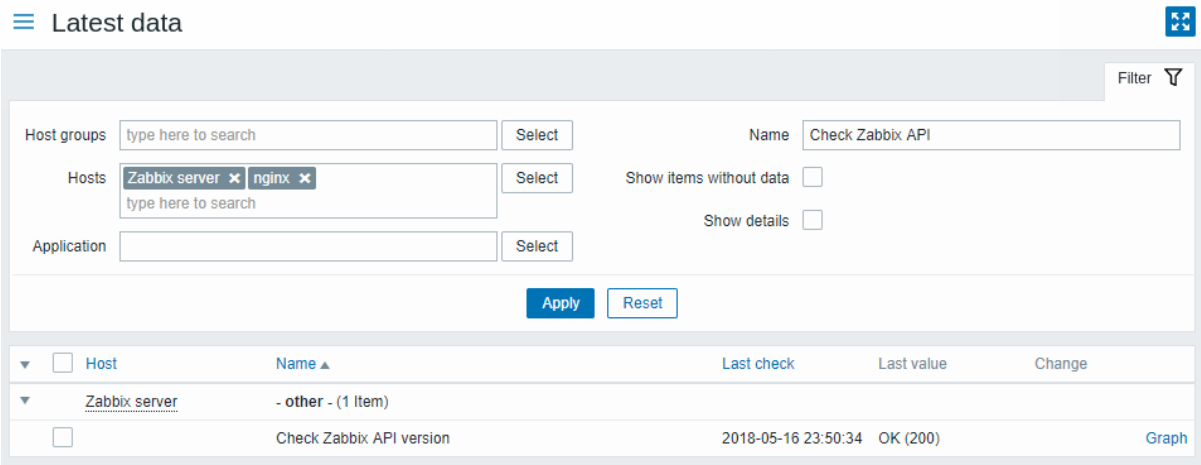


Si noti l'uso del metodo POST con dati JSON, l'impostazione delle intestazioni della richiesta e la richiesta di restituire solo le intestazioni:

- Preprocessing del valore dell'item con espressione regolare per ottenere il codice HTTP:



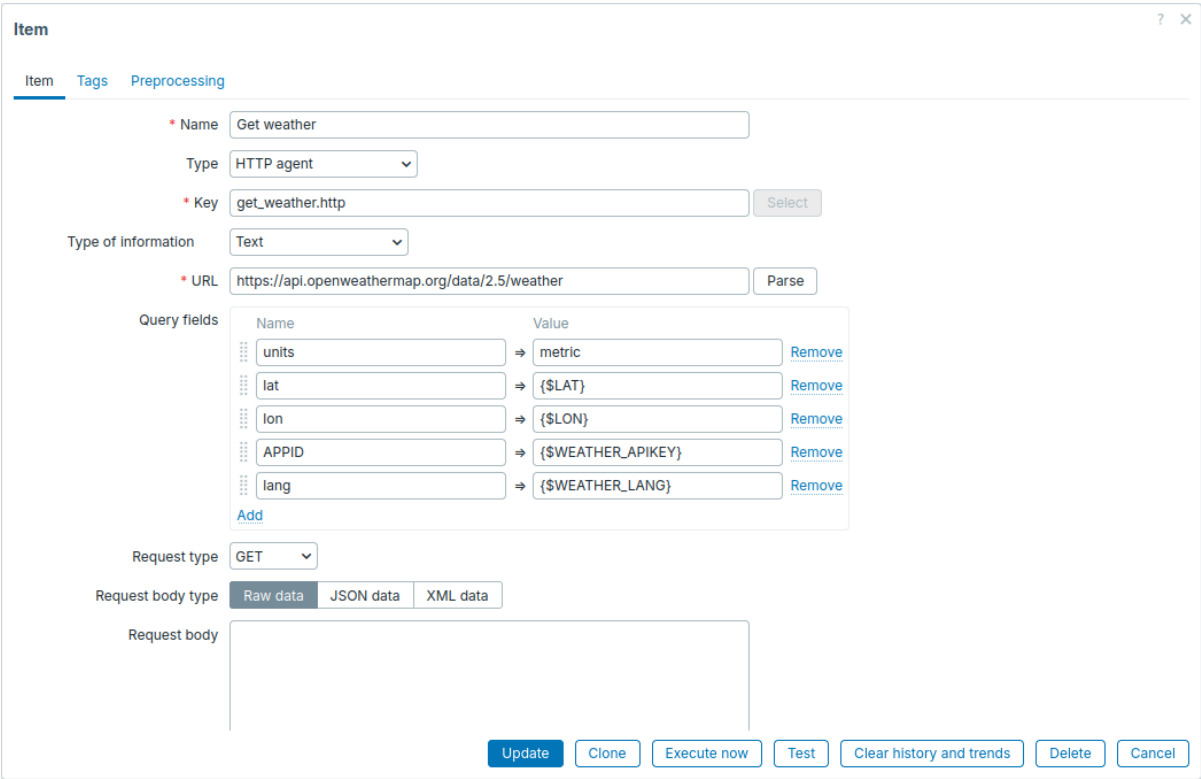
- Verifica del risultato in *Latest data*:



Esempio 4

Recupero delle informazioni meteo collegandosi al servizio pubblico Openweathermap.

- Configura un item master per la raccolta di dati in blocco in un singolo JSON:



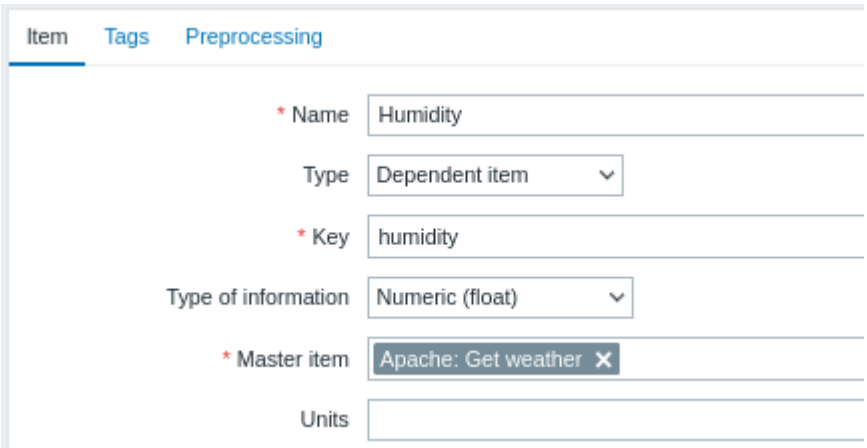
Nota l'uso delle macro nei campi di query. Consulta le Openweathermap API per sapere come compilarli.

Esempio di JSON restituito in risposta all'HTTP agent:

```
{
    "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
    }
}
```

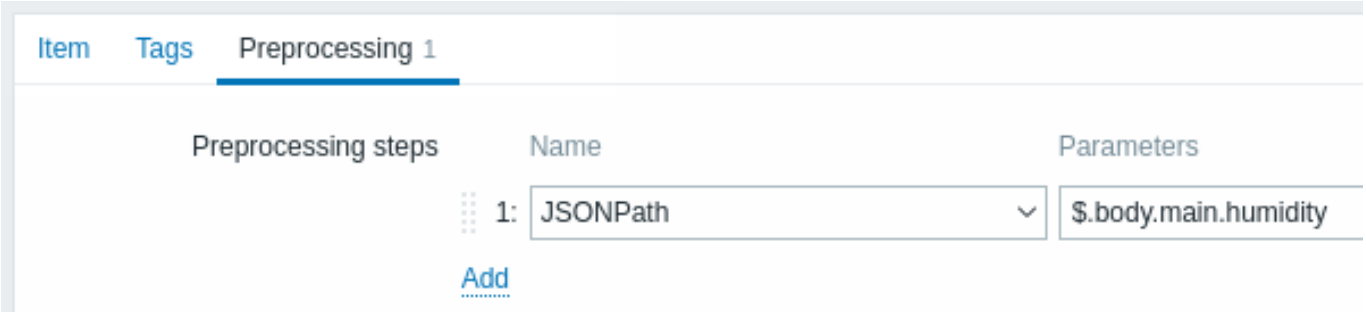

Il passo successivo è configurare gli item dipendenti che estraggono i dati dal JSON.

- Configura un item dipendente di esempio per l'umidità:

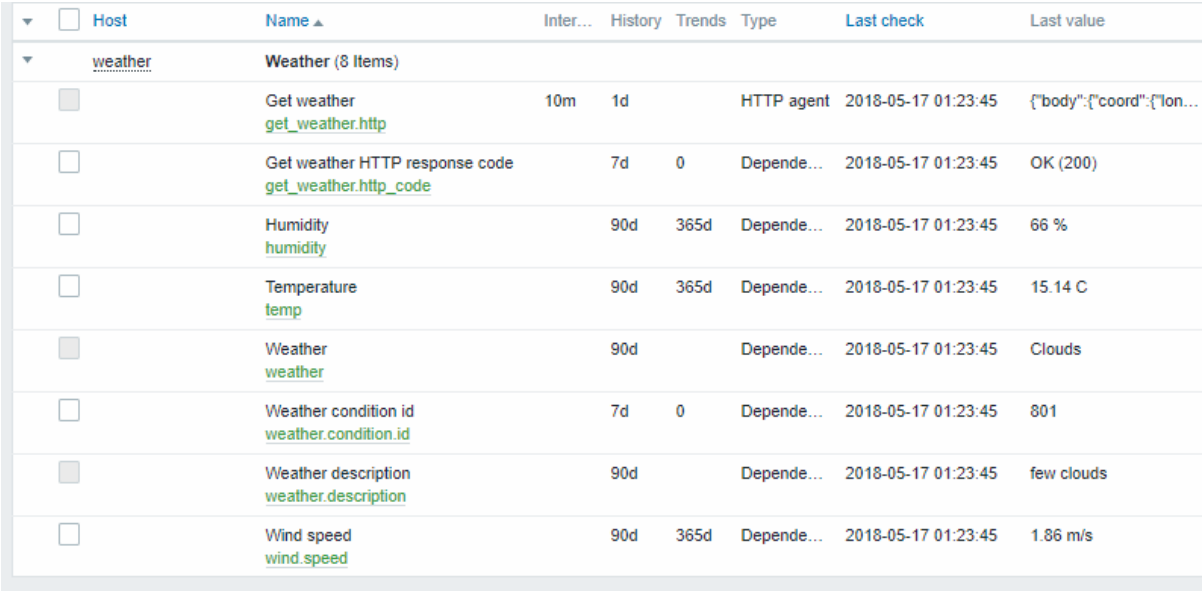


Altre metriche meteo, come 'Temperature', vengono aggiunte nello stesso modo.

- Esempio di preprocessing del valore di un item dipendente con JSONPath:



- Controlla il risultato dei dati meteo in *Latest data*:

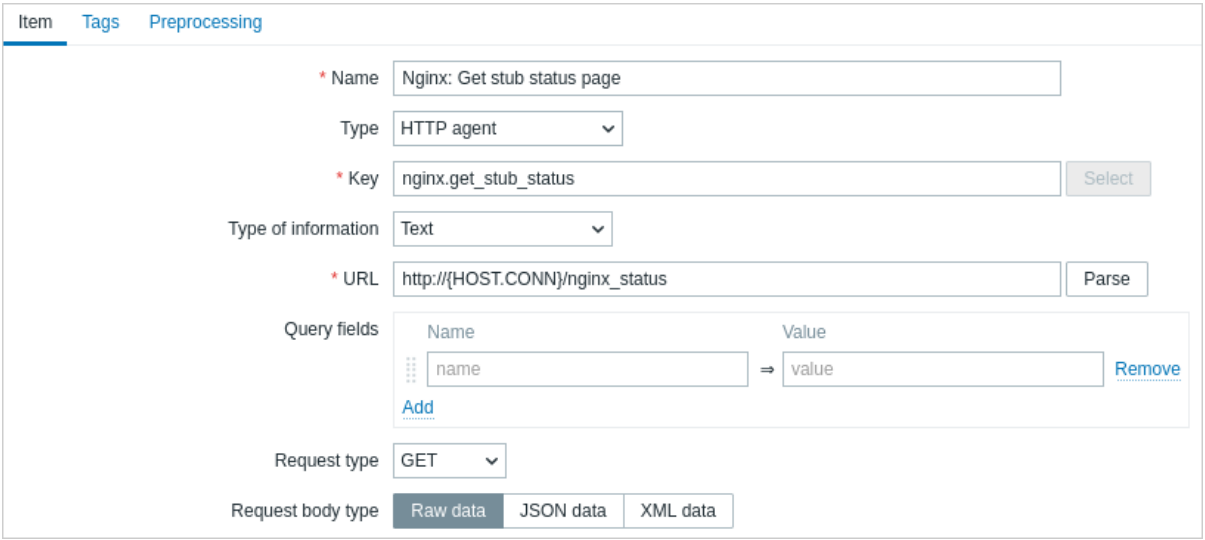


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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

Esempio 5

Connessione alla pagina di stato di NGINX e recupero delle relative metriche in blocco.

- Configurare NGINX seguendo la guida ufficiale.
- Configurare un item master per la raccolta dei dati in blocco:

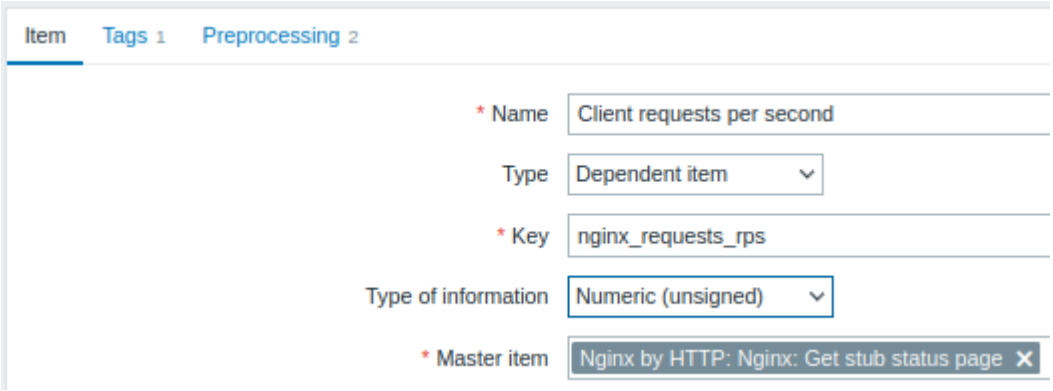


Esempio di output dello stub di stato di NGINX:

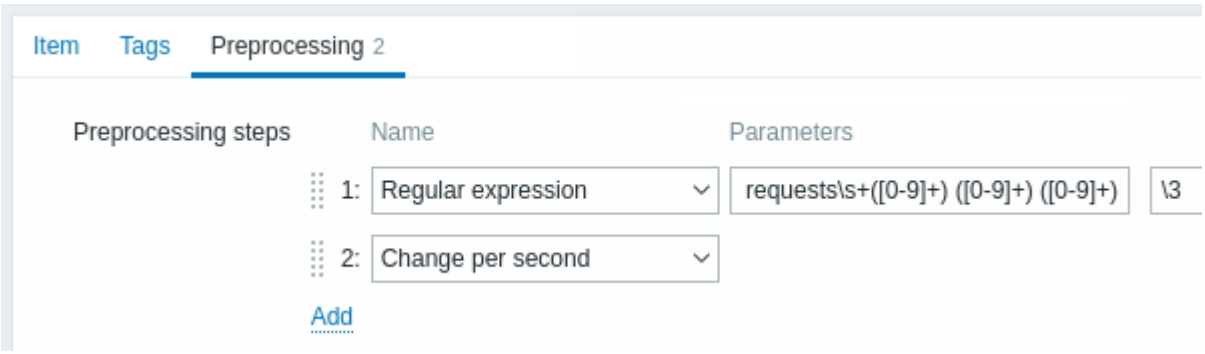
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Il passaggio successivo consiste nel configurare gli item dipendenti che estraggono i dati.

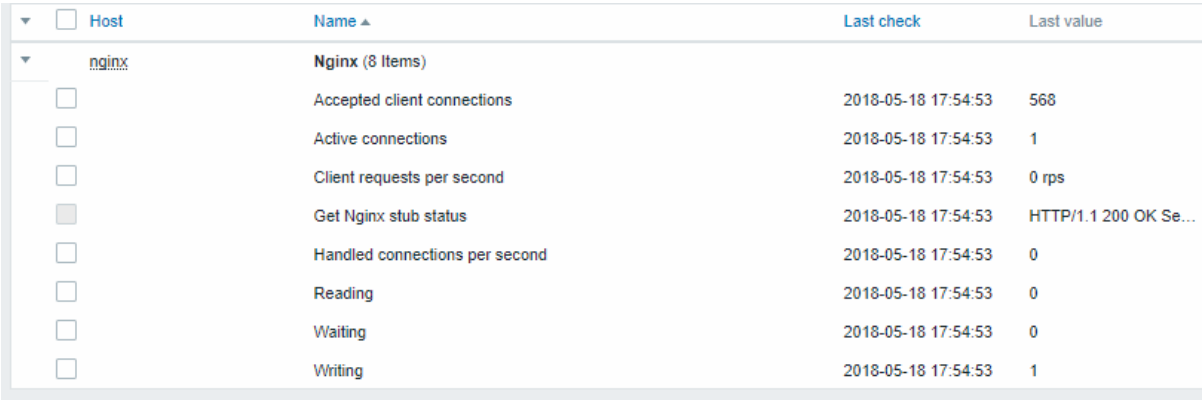
- Configurare un item dipendente di esempio per le richieste al secondo:



- Esempio di preelaborazione del valore di un item dipendente con l'espressione regolare `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Verificare il risultato completo del modulo stub in *Latest data*:



Controllo Prometheus

Panoramica

Zabbix può interrogare le metriche esposte nel formato lineare di Prometheus.

Per iniziare a raccogliere i dati di Prometheus sono necessari due passaggi:

- un HTTP master item che punti all'endpoint dati appropriato, ad esempio `https://<prometheus host>/metrics`
- item dipendenti che utilizzano un'opzione di preprocessing di Prometheus per interrogare i dati richiesti dalle metriche raccolte dal master item

Esistono due opzioni di preprocessing dei dati di Prometheus:

- *Prometheus pattern* - usato negli item normali per interrogare i dati di Prometheus
- *Prometheus to JSON* - usato negli item normali e per la discovery di basso livello. In questo caso, i dati di Prometheus interrogati vengono restituiti in formato JSON.

Elaborazione in blocco

L'elaborazione in blocco è supportata per gli item dipendenti. Per abilitare la memorizzazione nella cache e l'indicizzazione, la pre-elaborazione *Prometheus pattern* deve essere il **primo** passaggio di pre-elaborazione. Quando *Prometheus pattern* è il primo passaggio di pre-elaborazione, i dati Prometheus analizzati vengono memorizzati nella cache e indicizzati dalla prima condizione

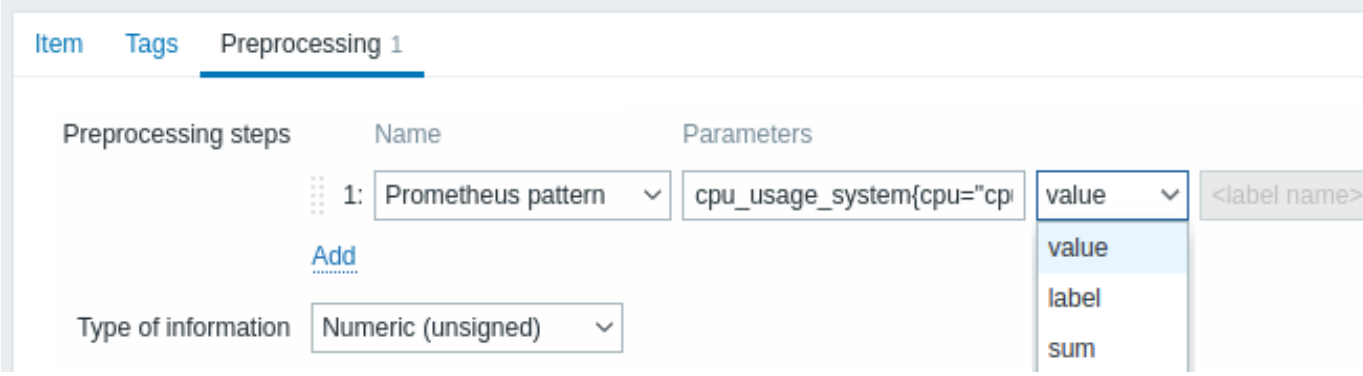
`<label>==<value>` nel passaggio di pre-elaborazione *Prometheus pattern*. Questa cache viene riutilizzata durante l'elaborazione di altri item dipendenti in questo batch. Per prestazioni ottimali, la prima label dovrebbe essere quella con il maggior numero di valori diversi.

Se è necessario eseguire altra pre-elaborazione prima del primo passaggio, questa dovrebbe essere spostata nell'item master oppure in un nuovo item dipendente che verrà usato come item master per gli item dipendenti.

Configurazione

A condizione di avere configurato l'item master HTTP, è necessario creare un item dipendente che utilizzi un passaggio di pre-elaborazione Prometheus:

1. Inserire i parametri generali dell'item dipendente nel modulo di configurazione.
2. Accedere alla scheda *Preprocessing*.
3. Selezionare un'opzione di pre-elaborazione Prometheus (*Prometheus pattern* o *Prometheus to JSON*).



I seguenti parametri sono specifici dell'opzione di pre-elaborazione *Prometheus pattern*:

Parametro	Descrizione	Esempi
Pattern	Per definire il pattern di dati richiesto, è possibile utilizzare un linguaggio di query simile al linguaggio di query Prometheus (vedere la tabella di confronto), ad esempio: <metric name> - selezione in base al nome della metrica {__name__="<metric name>"} - selezione in base al nome della metrica {__name__=~"<regex>"} - selezione in base al nome della metrica che corrisponde a un'espressione regolare {<label name>="<label value>",...} - selezione in base al nome dell'etichetta {<label name>=~"<regex>",...} - selezione in base al nome dell'etichetta che corrisponde a un'espressione regolare {__name__=~".*"}==<value> - selezione in base al valore della metrica Oppure una combinazione degli elementi precedenti: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> Il valore dell'etichetta può essere costituito da qualsiasi sequenza di caratteri UTF-8, ma i caratteri barra rovesciata, virgolette doppie e avanzamento riga devono essere sottoposti rispettivamente a escape come \\, \" e \n; gli altri caratteri non devono essere sottoposti a escape.	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu="cpu-total"}* *cpu_usage_system{cpu=".*"}* *wmi_service_state{name="dhcp"}==1* *wmi_os_timezone{timezone=~".*"}==1*

Parametro	Descrizione	Esempi
Result processing	Specificare se restituire il valore, l'etichetta o applicare la funzione appropriata (se il pattern corrisponde a più righe e il risultato deve essere aggregato): **value** - restituisce il valore della metrica (errore se corrispondono più righe) **label** - restituisce il valore dell'etichetta specificata nel campo *Label* (errore se corrispondono più metriche) **sum** - restituisce la somma dei valori **min** - restituisce il valore minimo **max** - restituisce il valore massimo **avg** - restituisce il valore medio **count** - restituisce il numero di valori Questo campo è disponibile solo per l'opzione *Prometheus pattern*.	Vedere anche gli esempi di utilizzo dei parametri riportati di seguito.
Output	Definire il nome dell'etichetta (facoltativo). In questo caso viene restituito il valore corrispondente al nome dell'etichetta. Questo campo è disponibile solo per l'opzione *Prometheus pattern*, se nel campo *Result processing* è selezionato 'Label'.	

Esempi di utilizzo dei parametri

1. Il caso d'uso più comune consiste nel restituire il **value**. Per restituire il valore di `/var/db` da:

   `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`

utilizzare i seguenti parametri:

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - selezionare 'value'

2. Potrebbe essere utile anche il valore **medio** di tutti i parametri `node_disk_usage_bytes`:

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - selezionare 'avg'

3. Sebbene Prometheus supporti solo dati numerici, viene comunemente utilizzata una soluzione alternativa per restituire anche la descrizione testuale pertinente. È possibile ottenere questo risultato con un filtro e specificando l'etichetta. Pertanto, per restituire il valore dell'etichetta 'color' da:

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

utilizzare i seguenti parametri:

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - selezionare 'label'
- *Label* - specificare 'color'

Il filtro (basato sul valore numerico '1') corrisponderà alla riga appropriata, mentre l'etichetta restituirà la descrizione dello stato di salute (attualmente 'green', ma potenzialmente anche 'red' o 'yellow').

Prometheus in JSON

I dati di Prometheus possono essere utilizzati per la discovery a basso livello. In questo caso sono necessari dati in formato JSON e l'opzione di preprocessing *Prometheus to JSON* restituirà esattamente questo.

Per ulteriori dettagli, vedere Discovery using Prometheus data.

Confronto del linguaggio di query

La tabella seguente elenca le differenze e le somiglianze tra PromQL e il linguaggio di query di preprocessing Prometheus di Zabbix.

Selettore di vettore istantaneo PromQL		Preprocessing Prometheus di Zabbix
Differenze		
Target della query	server Prometheus	Testo in formato di esposizione Prometheus
Restituisce	Vettore istantaneo	Valore di metrica o etichetta (pattern Prometheus) Array di metriche per un singolo valore in JSON (Prometheus to JSON)
Operatori di corrispondenza delle etichette	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Espressione regolare usata nella corrispondenza del nome di etichetta o metrica	RE2	PCRE
Operatori di confronto	Vedi elenco	È supportato solo **==** (uguale) per il filtraggio dei valori
Somiglianze		
Selezione per nome di metrica uguale a una stringa	<nome metrica> o {__name__="<nome metrica>"}	<nome metrica> o {__name__="<nome metrica>"}
Selezione per nome di metrica che corrisponde a un'espressione regolare	{__name__=~"<regex>"}	{__name__=~"<regex>"}
Selezione per valore di <nome etichetta> uguale a una stringa	{<nome etichetta>="<valore etichetta>",...}	{<nome etichetta>="<valore etichetta>",...}
Selezione per valore di <nome etichetta> che corrisponde a un'espressione regolare	{<nome etichetta>=~"<regex>",...}	{<nome etichetta>=~"<regex>",...}
Selezione per valore uguale a una stringa	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

agent IPMI

Panoramica

È possibile monitorare lo stato e la disponibilità dei dispositivi Intelligent Platform Management Interface (IPMI) in Zabbix. Per eseguire i controlli IPMI, il server Zabbix deve essere inizialmente configurato con il supporto IPMI.

IPMI è un'interfaccia standardizzata per la gestione remota "lights-out" o "out-of-band" dei sistemi informatici. Consente di monitorare direttamente lo stato dell'hardware dalle cosiddette schede di gestione "out-of-band", indipendentemente dal sistema operativo o dal fatto che la macchina sia accesa.

Il monitoraggio IPMI di Zabbix funziona solo per i dispositivi che supportano IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP, ecc.).

Un processo di gestione IPMI pianifica i controlli IPMI tramite i poller IPMI. Un host viene sempre interrogato da un solo poller IPMI alla volta, riducendo il numero di connessioni aperte ai controller BMC. È quindi sicuro aumentare il numero di poller IPMI senza preoccuparsi di sovraccaricare il controller BMC. Il processo di gestione IPMI viene avviato automaticamente quando viene avviato almeno un poller IPMI.

Vedere anche i problemi noti relativi ai controlli IPMI.

Configurazione

Configurazione del host

Un host deve essere configurato per elaborare i controlli IPMI. Deve essere aggiunta un'interfaccia IPMI, con i rispettivi indirizzo IP e numeri di porta, e devono essere definiti i parametri di autenticazione IPMI.

Per maggiori dettagli, vedere la configurazione dei host.

Configurazione del server

Per impostazione predefinita, il server Zabbix non è configurato per avviare alcun poller IPMI, quindi eventuali item IPMI aggiunti non funzioneranno. Per modificare questo comportamento, aprire il file di configurazione del server Zabbix (`zabbix_server.conf`) come root e cercare la seguente riga:

```
# StartIPMIPollers=0
```

Rimuovere il commento e impostare il numero di poller, ad esempio a 3, in modo che diventi:

```
StartIPMIPollers=3
```

Salvare il file e riavviare zabbix_server in seguito.

Configurazione dell'item

Quando si configura un item a livello di host:

- Selezionare 'IPMI agent' come *Tipo*
- Inserire una chiave dell'item univoca all'interno dell'host (ad esempio, ipmi.fan.rpm)
- Per *Interfaccia host* selezionare l'interfaccia IPMI pertinente (IP e porta). Si noti che sull'host deve esistere un'interfaccia IPMI.
- Specificare il *sensore IPMI* (ad esempio 'FAN MOD 1A RPM' su Dell Poweredge) da cui recuperare la metrica. Per impostazione predefinita, deve essere specificato l'ID del sensore. È anche possibile usare dei prefissi prima del valore:
  - `id:` - per specificare l'ID del sensore;
  - `name:` - per specificare il nome completo del sensore. Questo può essere utile nelle situazioni in cui i sensori possono essere distinti solo specificando il nome completo.
- Selezionare il rispettivo tipo di informazione ('Numerico (float)' in questo caso; per i sensori discreti - 'Numerico (unsigned)'), le unità (molto probabilmente 'rpm') e qualsiasi altro attributo dell'item richiesto

Controlli supportati

L'agent IPMI supporta l'item integrato **ipmi.get**, che restituisce informazioni relative ai sensori IPMI e può essere utilizzato per la discovery dei sensori IPMI.
 Valore restituito: *oggetto JSON*

Timeout e terminazione della sessione

I timeout dei messaggi IPMI e il numero di tentativi sono definiti nella libreria OpenIPMI. A causa dell'attuale progettazione di OpenIPMI, non è possibile rendere questi valori configurabili in Zabbix, né a livello di interfaccia né di item.

Il timeout di inattività della sessione IPMI per LAN è di 60 +/-3 secondi. Attualmente non è possibile implementare l'invio periodico del comando Activate Session con OpenIPMI. Se non ci sono controlli degli item IPMI da Zabbix verso uno specifico BMC per un tempo superiore al timeout di sessione configurato nel BMC, allora il successivo controllo IPMI dopo la scadenza del timeout andrà in timeout a causa dei timeout dei singoli messaggi, dei tentativi o di un errore di ricezione. Successivamente verrà aperta una nuova sessione e verrà avviata una nuova scansione completa del BMC. Potrebbe essere aperta una nuova porta UDP per gestire la nuova sessione. L'inattività è definita dall'assenza sia di richieste in uscita sia di risposte in ingresso. Se si desidera evitare nuove scansioni non necessarie del BMC, si consiglia di impostare l'intervallo di polling degli item IPMI al di sotto del timeout di inattività della sessione IPMI configurato nel BMC.

Note sui sensori discreti IPMI

Per trovare i sensori su un host, avviare Zabbix server con **DebugLevel=4** abilitato. Attendere alcuni minuti e cercare i record di individuazione dei sensori nel file di log di Zabbix server:

```
$ 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
```

Per decodificare i tipi e gli stati dei sensori IPMI, è disponibile una copia delle specifiche IPMI 2.0 (si noti che non sono previsti ulteriori aggiornamenti delle specifiche IPMI).

Il primo parametro da cui iniziare è "reading_type". Utilizzare "Table 42-1, Event/Reading Type Code Ranges" delle specifiche per decodificare il codice "reading_type". La maggior parte dei sensori nel nostro esempio ha "reading_type:0x1", che significa sensore "threshold". "Table 42-3, Sensor Type Codes" mostra che "type:0x1" significa sensore di temperatura, "type:0x2" - sensore

di tensione, "type:0x4" - ventola, ecc. I sensori threshold a volte sono chiamati sensori "analogici" perché misurano parametri continui come temperatura, tensione, giri al minuto.

Un altro esempio: un sensore con "reading_type:0x3". "Table 42-1, Event/Reading Type Code Ranges" indica che i codici del tipo di lettura da 02h a 0Ch significano sensore "Generic Discrete". I sensori discreti hanno fino a 15 possibili stati (in altre parole, fino a 15 bit significativi). Per esempio, per il sensore 'CATERR' con "type:0x7", la "Table 42-3, Sensor Type Codes" mostra che questo tipo significa "Processor" e il significato dei singoli bit è: 00h (il bit meno significativo) - IERR, 01h - Thermal Trip, ecc.

Nel nostro esempio ci sono alcuni sensori con "reading_type:0x6f". Per questi sensori, la "Table 42-1, Event/Reading Type Code Ranges" consiglia di utilizzare "Table 42-3, Sensor Type Codes" per decodificare il significato dei bit. Per esempio, il sensore 'Power Unit Stat' ha tipo "type:0x9", che significa "Power Unit". L'offset 00h significa "PowerOff/Power Down". In altre parole, se il bit meno significativo è 1, allora il server è spento. Per testare questo bit, è possibile utilizzare la funzione `bitand` con la maschera '1'. L'espressione del trigger potrebbe essere

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

per avvisare dello spegnimento di un server.

Note sui nomi dei sensori discreti in OpenIPMI-2.0.16, 2.0.17, 2.0.18 e 2.0.19

I nomi dei sensori discreti in OpenIPMI-2.0.16, 2.0.17 e 2.0.18 spesso hanno uno "0" aggiuntivo (o un'altra cifra o lettera) aggiunto alla fine. Ad esempio, mentre `ipmitool` e OpenIPMI-2.0.19 visualizzano i nomi dei sensori come "`PhysicalSecurity`" o "`CATERR`", in OpenIPMI-2.0.16, 2.0.17 e 2.0.18 i nomi sono rispettivamente "`PhysicalSecurity0`" o "`CATERR0`".

Quando si configura un item IPMI con Zabbix server che utilizza OpenIPMI-2.0.16, 2.0.17 e 2.0.18, usare questi nomi che terminano con "0" nel campo *IPMI sensor* degli item agent IPMI. Quando il proprio Zabbix server viene aggiornato a una nuova distribuzione Linux, che utilizza OpenIPMI-2.0.19 (o successiva), gli item con questi sensori discreti IPMI diventeranno "NOT SUPPORTED". È necessario modificare i loro nomi *IPMI sensor* (rimuovendo lo '0' finale) e attendere un po' di tempo prima che tornino di nuovo "Enabled".

Note sulla disponibilità simultanea di sensori di soglia e discreti

Alcuni agent IPMI forniscono sia un sensore di soglia sia un sensore discreto con lo stesso nome. La preferenza viene sempre data al sensore di soglia.

Note sulla terminazione della connessione

Se i controlli IPMI non vengono eseguiti (per qualsiasi motivo: tutti gli item IPMI del host disabilitati/non supportati, host disabilitato/eliminato, host in manutenzione, ecc.), la connessione IPMI verrà terminata dal server o dal proxy Zabbix entro 3 o 4 ore, a seconda del momento in cui il server/proxy Zabbix è stato avviato.

SSH agent

Panoramica

I controlli SSH vengono eseguiti come monitoraggio senza agent. Zabbix agent non è necessario per i controlli SSH.

Per eseguire i controlli SSH, il server Zabbix deve essere inizialmente configurato con supporto SSH2 (libssh o libssh2). Vedi anche: Requisiti.

> **Attention:**
> A partire da RHEL 8, è supportato solo libssh. Per altre distribuzioni, si consiglia libssh rispetto a libssh2.

Configurazione

Autenticazione con passphrase

I controlli SSH forniscono due metodi di autenticazione: una coppia utente/password e uno basato su file di chiave.

Se non si intende utilizzare le chiavi, non è richiesta alcuna configurazione aggiuntiva, oltre al collegamento di libssh o libssh2 a Zabbix, se si esegue la compilazione dal codice sorgente.

Autenticazione con file di chiave

Per utilizzare l'autenticazione basata su chiave per gli item SSH, sono necessarie alcune modifiche alla configurazione del server.

Aprire il file di configurazione del server Zabbix (`zabbix_server.conf`) come `root` e cercare la seguente riga:

```
##### SSHKeyLocation=
```

Decommentarla e impostare il percorso completo della cartella in cui si troveranno le chiavi pubbliche e private:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Salvare il file e riavviare successivamente il server Zabbix.

Il percorso `/home/zabbix` qui è la directory home dell'account utente *zabbix*, e `.ssh` è una directory in cui, per impostazione predefinita, le chiavi pubbliche e private verranno generate da un comando ssh-keygen all'interno della directory home.

Di solito i pacchetti di installazione del server Zabbix delle diverse distribuzioni OS creano l'account utente *zabbix* con una directory home in un'altra posizione, ad esempio `/var/lib/zabbix` (come per gli account di sistema).

Prima di generare le chiavi, è possibile riallocare la directory home in `/home/zabbix`, in modo che corrisponda al parametro di configurazione `SSHKeyLocation` del server Zabbix menzionato sopra.

Note:
I seguenti passaggi possono essere saltati se l'account *zabbix* è stato aggiunto manualmente secondo la sezione di installazione. In tal caso, la directory home dell'account *zabbix* è molto probabilmente già `/home/zabbix`.

Per modificare la directory home dell'account utente *zabbix*, tutti i processi in esecuzione che la utilizzano devono essere arrestati:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Per modificare la posizione della directory home tentando di spostarla (se esiste), eseguire il seguente comando:

```
usermod -m -d /home/zabbix zabbix
```

È anche possibile che una directory home non esistesse nella vecchia posizione, quindi dovrebbe essere creata nella nuova posizione. Un tentativo sicuro per farlo è:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Per essere certi che tutto sia protetto, è possibile eseguire comandi aggiuntivi per impostare i permessi della directory home:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

I processi precedentemente arrestati possono ora essere avviati di nuovo:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Ora è possibile eseguire i passaggi per generare le chiavi pubbliche e private con i seguenti comandi (per una migliore leggibilità, i prompt dei comandi sono commentati):

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |        .        |
##### |       o         |
##### | .      o        |
##### |+      . S       |
##### |.+    o =        |
##### |E .    * =       |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
Le chiavi pubbliche e private (`id_rsa.pub` e `id_rsa`) sono state generate per impostazione predefinita nella directory `/home/zabbix/.ssh`, che corrisponde al parametro di configurazione `SSHKeyLocation` del server Zabbix.

Attention:
Tipi di chiave diversi da "rsa" possono essere supportati dallo strumento ssh-keygen e dai server SSH, ma potrebbero non essere supportati da libssh2 utilizzato da Zabbix.

Modulo di configurazione della shell

Questo passaggio deve essere eseguito una sola volta per ogni host che verrà monitorato tramite controlli SSH.

Utilizzando i seguenti comandi, il file della chiave **pubblica** può essere installato su un host remoto *10.10.10.10*, in modo che i controlli SSH possano essere eseguiti con un account *root* (per una migliore leggibilità, i prompt dei comandi sono commentati):

```
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.
```

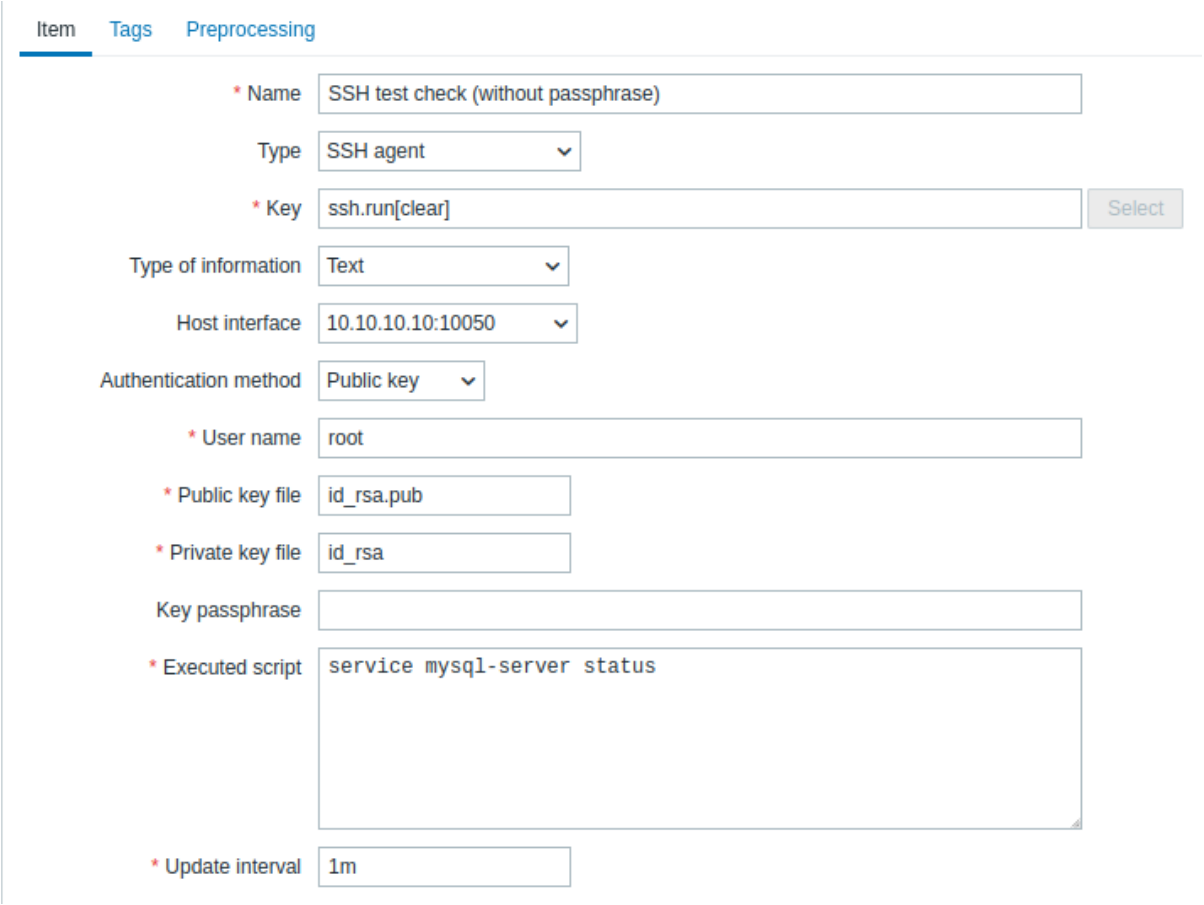
Ora è possibile verificare l'accesso SSH utilizzando la chiave privata predefinita (`/home/zabbix/.ssh/id_rsa`) per l'account utente *zabbix*:

```
sudo -u zabbix ssh root@10.10.10.10
```

Se l'accesso ha esito positivo, la parte di configurazione nella shell è terminata e la sessione SSH remota può essere chiusa.

Configurazione dell'item

I comandi effettivi da eseguire devono essere inseriti nel campo *Script eseguito* nella configurazione dell'item. È possibile eseguire più comandi uno dopo l'altro inserendoli su una nuova riga. In questo caso, anche i valori restituiti saranno formattati su più righe.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item SSH sono:

Parametro	Descrizione	Commenti
Tipo	Selezionare qui **SSH agent**.	

Parametro	Descrizione	Commenti

Chiave	Chiave dell'item univoca (per host) nel formato `ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]`	**unique short description** è obbligatoria e deve essere univoca per ogni item SSH dell'host.
		
		La porta predefinita è 22, non la porta specificata nell'interfaccia a cui è assegnato questo item.
		
		ssh options consente di passare opzioni SSH aggiuntive nel formato `key1=value1;key2=value2,value3`. È possibile passare più valori per una chiave separandoli con una virgola (in questo caso, il parametro deve essere racchiuso tra virgolette); più chiavi di opzione possono essere passate separandole con un punto e virgola.
		
		Sono supportate le seguenti chiavi di opzione: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Il supporto per la chiave e il valore dell'opzione dipende dalla libreria SSH (ad esempio, `PubkeyAcceptedKeyTypes` è supportata solo con libssh); se un'opzione non è supportata, verrà restituito un errore e l'item diventerà non supportato.
		
		Il segno "+" per aggiungere impostazioni di cifratura e "!" per disabilitare impostazioni di cifratura specifiche (come in GnuTLS e OpenSSL) non sono supportati.
		
		Esempi:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128-`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-he`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey`
		
		subsystem consente di passare un sottosistema SSH, limitando la connessione SSH a operazioni specifiche consentite dal sottosistema (ad esempio, trasferimenti di file tramite SFTP o gestione di dispositivi di rete tramite NETCONF). L'uso di un sottosistema può richiedere anche l'uso di una sintassi specifica dello script nel parametro *Script eseguito*.
		
		Esempi:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Metodo di autenticazione	Uno tra "Password" o "Chiave pubblica".	
Nome utente	Nome utente (fino a 255 caratteri) per l'autenticazione sull'host remoto. Obbligatorio.	
File della chiave pubblica	Nome del file della chiave pubblica se il *Metodo di autenticazione* è "Chiave pubblica". Obbligatorio.	Esempio: *id_rsa.pub* - nome predefinito del file della chiave pubblica generato dal comando ssh-keygen.
File della chiave privata	Nome del file della chiave privata se il *Metodo di autenticazione* è "Chiave pubblica". Obbligatorio.	Esempio: *id_rsa* - nome predefinito del file della chiave privata.

Parametro	Descrizione	Commenti
Password oppure *Passphrase della chiave*	Password (fino a 255 caratteri) per l'autenticazione oppure Passphrase **se** è stata utilizzata per la chiave privata.	Lasciare vuoto il campo *Passphrase della chiave* se non è stata utilizzata alcuna passphrase. Vedere anche i problemi noti relativi all'uso delle passphrase.

Script eseguito	Comando o comandi shell eseguiti utilizzando la sessione SSH remota.	Il valore restituito dai comandi shell eseguiti è limitato a 16 MB (inclusi gli spazi bianchi finali, che vengono troncati); si applicano anche i limiti del database.
		
		La libreria libssh2 potrebbe troncare gli script eseguibili a circa 32 kB.
		
		Esempi:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		Esempio (per il sottosistema NETCONF):
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

Telnet agent

Panoramica

I controlli Telnet vengono eseguiti come monitoraggio senza agente. L'agente Zabbix non è necessario per i controlli Telnet.

Campi configurabili

I comandi effettivi da eseguire devono essere inseriti nel campo **Executed script** nella configurazione dell'item.
 È possibile eseguire più comandi uno dopo l'altro inserendoli su una nuova riga. In questo caso anche il valore restituito sarà formattato su più righe.

Caratteri supportati con cui il prompt della shell può terminare:

- $
- #
- >
- %

Note:
Una riga del prompt telnet che termina con uno di questi caratteri verrà rimossa dal valore restituito, ma solo per il primo comando nell'elenco dei comandi, cioè solo all'inizio della sessione telnet.

Key	Description
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	Esegue un comando su un dispositivo remoto utilizzando una connessione telnet

Attention:
Se un controllo telnet restituisce un valore con caratteri non ASCII e con codifica non UTF8, allora il parametro *<encoding>* della key deve essere specificato correttamente. Per maggiori dettagli, vedere la pagina encoding of returned values.

agent JMX

Panoramica

Il monitoraggio JMX può essere utilizzato per monitorare i contatori JMX di un'applicazione Java.

Il monitoraggio JMX dispone del supporto nativo in Zabbix sotto forma di un demone Zabbix chiamato "Zabbix Java gateway".

Per recuperare il valore di un particolare contatore JMX su un host, il server Zabbix interroga lo **Java gateway** di Zabbix, che a sua volta utilizza la API di gestione JMX per interrogare da remoto l'applicazione di interesse.

Per maggiori dettagli e per la configurazione, vedere la sezione Zabbix Java gateway.

> **Warning:**
> La comunicazione tra Java gateway e l'applicazione JMX monitorata non deve essere bloccata da firewall.

Abilitazione del monitoraggio JMX remoto per un'applicazione Java

Un'applicazione Java non richiede l'installazione di software aggiuntivo, ma deve essere avviata con le opzioni da riga di comando specificate di seguito per supportare il monitoraggio JMX remoto.

Come requisito minimo, se desideri semplicemente iniziare monitorando una semplice applicazione Java su un host locale senza alcuna sicurezza abilitata, avviala con queste opzioni:

```
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
```

In questo modo Java si mette in ascolto delle connessioni JMX in ingresso sulla porta 12345, solo dall'host locale, e viene configurato per non richiedere autenticazione né SSL.

Se vuoi consentire connessioni su un'altra interfaccia, imposta il parametro -Djava.rmi.server.hostname sull'IP di tale interfaccia.

Se desideri adottare impostazioni di sicurezza più rigorose, sono disponibili molte altre opzioni Java. Ad esempio, l'esempio seguente avvia l'applicazione con un insieme di opzioni più versatile e la apre a una rete più ampia, non solo all'host locale.

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /path/to/your/application.jar
```

La maggior parte (se non tutte) di queste impostazioni può essere specificata in `$JRE/lib/management/management.properties` (o ovunque si trovi quel file nel tuo sistema).

Nota che, se desideri utilizzare SSL, devi modificare lo script startup.sh aggiungendo le opzioni `-Djavax.net.ssl.*` a Java gateway, in modo che sappia dove trovare i key store e i trust store.

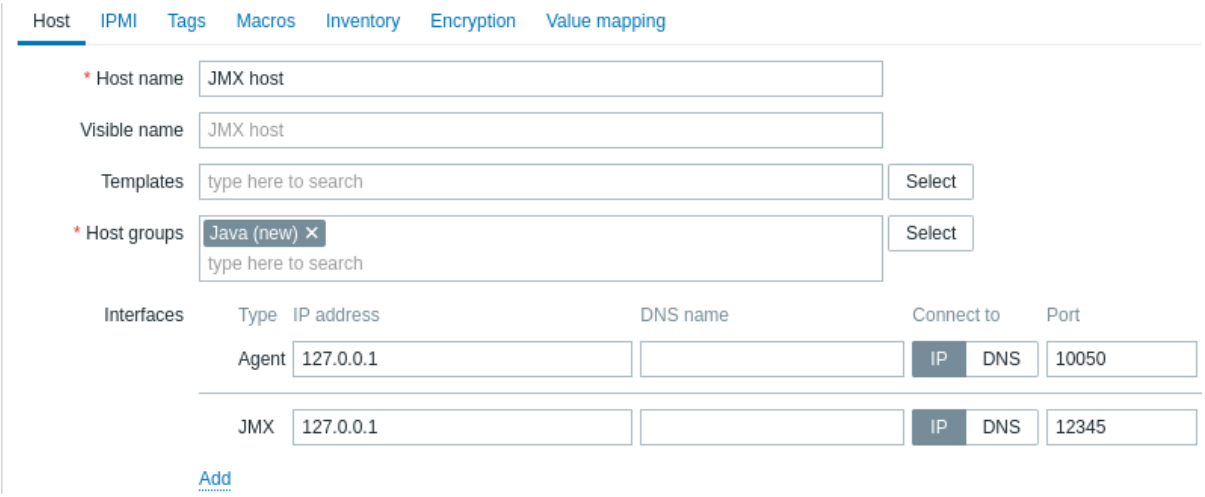
Per una descrizione dettagliata, vedi Monitoring and Management Using JMX.

Configurazione delle interfacce JMX e degli item nel frontend di Zabbix

Con il Java gateway in esecuzione, il server che sa dove trovarlo e un'applicazione Java avviata con il supporto per il monitoraggio JMX remoto, è il momento di configurare le interfacce e gli item nella GUI di Zabbix.

Configurazione dell'interfaccia JMX

Si inizia creando un'interfaccia di tipo JMX sul host di interesse.

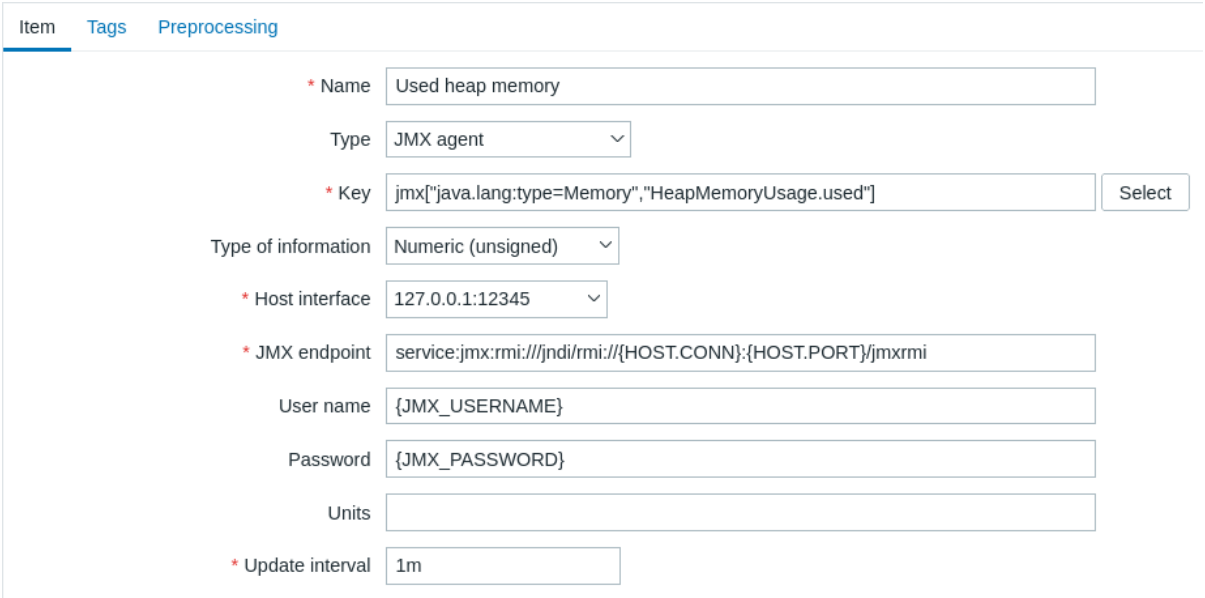


Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

Aggiunta di un item agent JMX

Per ogni contatore JMX di cui si è interessati, si aggiunge un item **JMX agent** associato a quella interfaccia.

La chiave nello screenshot seguente è `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item JMX sono:

Type	Impostare qui **JMX agent**.
Key	La chiave dell'item `jmx[]` contiene tre parametri: **object name** - il nome oggetto di un MBean **attribute name** - il nome di un attributo MBean con eventuali nomi di campi di dati compositi separati da punti **unique short description** - una descrizione univoca che consente più item JMX con lo stesso object name e attribute name sull'host (facoltativo) Vedere più avanti per maggiori dettagli sulle chiavi degli item JMX. È possibile individuare MBean e attributi MBean usando un item `jmx.discovery[]` di low-level discovery.
JMX endpoint	È possibile specificare un endpoint JMX personalizzato. Assicurarsi che i parametri di connessione dell'endpoint JMX corrispondano all'interfaccia JMX. Questo può essere ottenuto usando macro `{HOST.*}` come nel JMX endpoint predefinito. Le macro `{HOST.*}` macros e le macro utente sono supportate.

User name	Specificare il nome utente (fino a 255 caratteri), se è stata configurata l'autenticazione sulla propria applicazione Java. Sono supportate le macro utente.
Password	Specificare la password (fino a 255 caratteri), se è stata configurata l'autenticazione sulla propria applicazione Java. Sono supportate le macro utente.

Se si desidera monitorare un contatore Boolean che è "true" oppure "false", allora si specifica il tipo di informazione come "Numeric (unsigned)" e si seleziona il passaggio di preprocessing "Boolean to decimal" nella scheda Preprocessing. Il server memorizzerà i valori Boolean come 1 o 0, rispettivamente.

Elementi chiave JMX in maggiore dettaglio

Attributi semplici

Il nome oggetto di un MBean non è altro che una stringa che si definisce nella propria applicazione Java. Il nome di un attributo, invece, può essere più complesso. Nel caso in cui un attributo restituisca un tipo di dato primitivo (un intero, una stringa ecc.) non c'è nulla di cui preoccuparsi: la chiave avrà questo aspetto:

`jmx[com.example:Type=Hello,weight]`

In questo esempio, il nome oggetto è "com.example:Type=Hello", il nome dell'attributo è "weight" e il tipo di valore restituito dovrebbe probabilmente essere "Numerico (float)".

Attributi che restituiscono dati compositi

La situazione diventa più complicata quando il tuo attributo restituisce dati compositi. Ad esempio: il nome del tuo attributo è "apple" e restituisce una hash che rappresenta i suoi parametri, come "weight", "color", ecc. La tua chiave può apparire così:

`jmx[com.example:Type=Hello,apple.weight]`

Ecco come vengono separati il nome di un attributo e una chiave hash, utilizzando un punto. Allo stesso modo, se un attributo restituisce dati compositi annidati, le parti vengono separate da un punto:

`jmx[com.example:Type=Hello,fruits.apple.weight]`

Attributi che restituiscono dati tabellari

Gli attributi di dati tabellari sono costituiti da uno o più attributi compositi. Se un attributo di questo tipo è specificato nel parametro del nome dell'attributo, allora il valore di questo item restituirà la struttura completa dell'attributo in formato JSON. I valori dei singoli elementi all'interno dell'attributo di dati tabellari possono essere recuperati utilizzando il preprocessing.

Esempio di attributo di dati tabellari:

`jmx[com.example:type=Hello,foodinfo]`

Valore dell'item:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

Problema con i punti

Fin qui tutto bene. Ma cosa succede se un nome di attributo o una chiave hash contiene il simbolo del punto? Ecco un esempio:

`jmx[com.example:Type=Hello,all.fruits.apple.weight]`

Questo è un problema. Come si fa a indicare a Zabbix che il nome dell'attributo è "all.fruits" e non solo "all"? Come distinguere un punto che fa parte del nome dal punto che separa il nome dell'attributo e le chiavi hash?

È possibile: tutto ciò che devi fare è eseguire l'escape dei punti che fanno parte del nome con una barra rovesciata:

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

Allo stesso modo, se la tua chiave hash contiene un punto, devi eseguirne l'escape:

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

Altri problemi

Un carattere barra rovesciata in un nome di attributo deve essere preceduto da un carattere di escape:

```
jmx[com.example:type=Hello,c:\\documents]
```

Per la gestione di qualsiasi altro carattere speciale nella chiave item JMX, vedere la sezione sul formato della chiave item section.

Questo è davvero tutto qui. Buon monitoraggio JMX!

Tipi di dati non primitivi

È possibile lavorare con MBean personalizzati che restituiscono tipi di dati non primitivi, i quali sovrascrivono il metodo **toString()**.

Utilizzo di endpoint personalizzati con JBoss EAP 6.4

Gli endpoint personalizzati consentono di lavorare con protocolli di trasporto diversi dal RMI predefinito.

Per illustrare questa possibilità, proviamo a configurare il monitoraggio di JBoss EAP 6.4 come esempio. Per prima cosa, facciamo alcune ipotesi:

- Hai già installato Zabbix Java gateway. In caso contrario, puoi farlo seguendo la documentazione.
- server Zabbix e Java gateway sono installati con il prefisso `/usr/local/`
- JBoss è già installato in `/opt/jboss-eap-6.4/` ed è in esecuzione in modalità standalone
- Supporremo che tutti questi componenti funzionino sullo stesso host
- Firewall e SELinux sono disabilitati (o configurati di conseguenza)

Impostiamo alcune semplici configurazioni in `zabbix_server.conf`:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

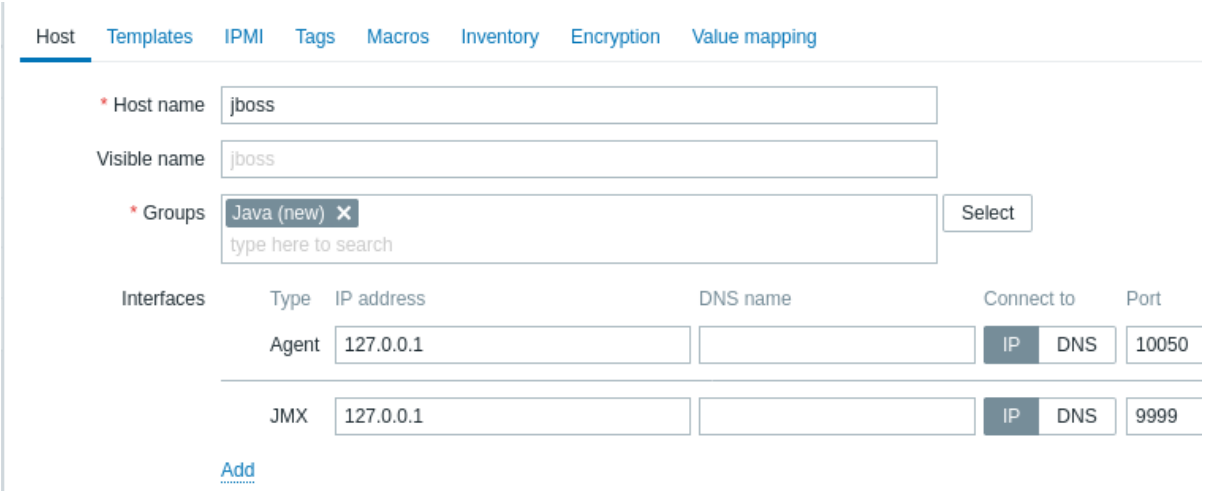
E nel file di configurazione `zabbix_java/settings.sh` (o `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Verifica che JBoss ascolti sulla sua porta di gestione standard:

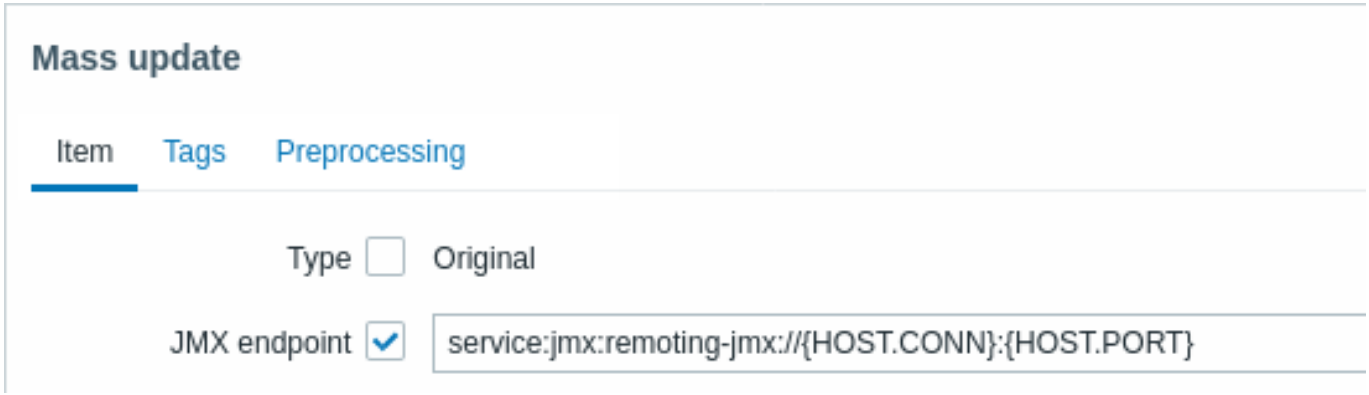
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Ora creiamo un host con interfaccia JMX 127.0.0.1:9999 in Zabbix.



Poiché sappiamo che questa versione di JBoss utilizza il protocollo JBoss Remoting anziché RMI, possiamo aggiornare in blocco il parametro dell'endpoint JMX per gli item del nostro template JMX di conseguenza:

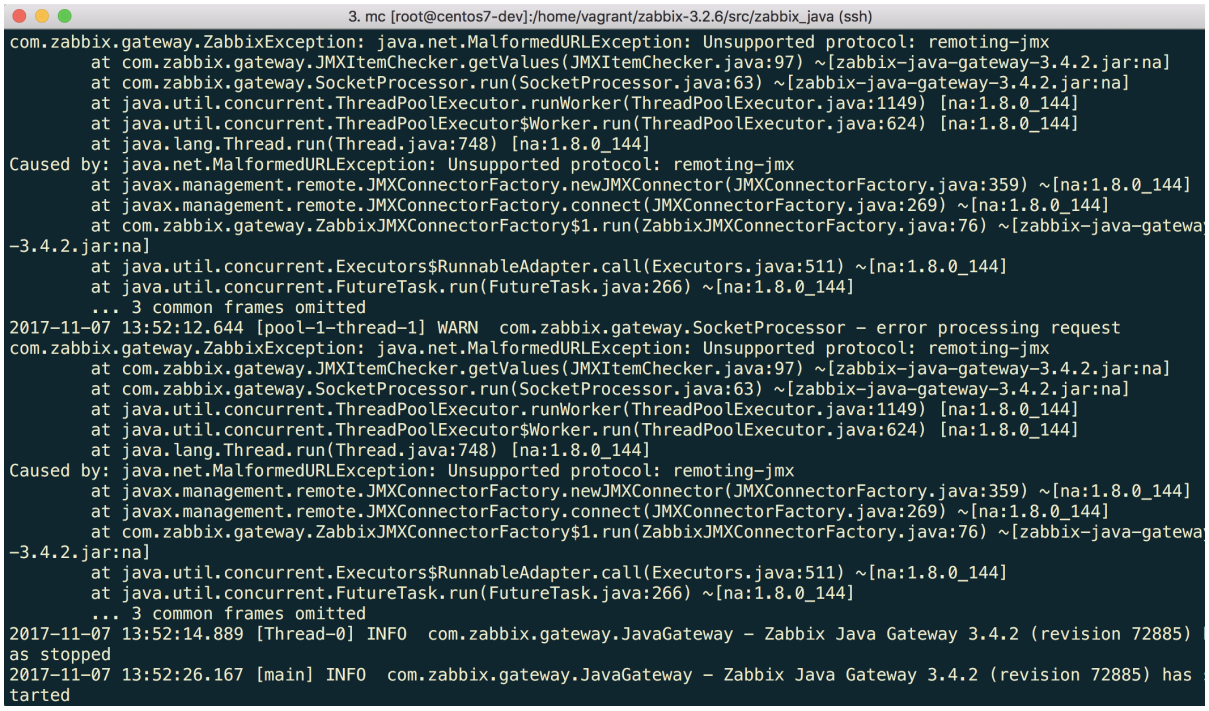
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Aggiorniamo la cache di configurazione:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Nota che potresti incontrare inizialmente un errore.



"Unsupported protocol: remoting-jmx" significa che Java gateway non sa come lavorare con il protocollo specificato. Questo può essere risolto creando un file `~/needed_modules.txt` con il seguente contenuto:

```
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
```

e quindi eseguendo il comando:

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

In questo modo, Java gateway avrà tutti i moduli necessari per lavorare con jmx-remoting. Non resta che riavviare Java gateway, attendere un po' e, se hai fatto tutto correttamente, vedrai che i dati di monitoraggio JMX inizieranno ad arrivare in Zabbix (vedi anche: Latest data).

Item calcolato

Panoramica

Questo tipo di item consente di eseguire calcoli sulle metriche già esistenti. Un item calcolato non raccoglie dati autonomamente.

Ad esempio, è possibile:

- Contare due valori.
- Calcolare la media dei valori di diversi item.
- Calcolare il valore totale di un gruppo di item.

Il valore risultante viene memorizzato nel database di Zabbix come per qualsiasi altro item; vengono memorizzati sia i valori della cronologia sia quelli dei trend e possono essere generati grafici.

Il calcolo si basa su una formula. La sintassi della formula è condivisa con le espressioni dei trigger.

È infatti possibile mantenere semplici i trigger inserendo la logica complessa di elaborazione dei dati in un item calcolato, quindi fare riferimento all'item calcolato in un trigger utilizzando la chiave dell'item.

Esistono due possibilità per specificare l'origine dei dati per i calcoli:

1. Specificare gli item nella formula.
2. Eseguire l'aggregazione dei dati in base a un filtro che seleziona gli item.

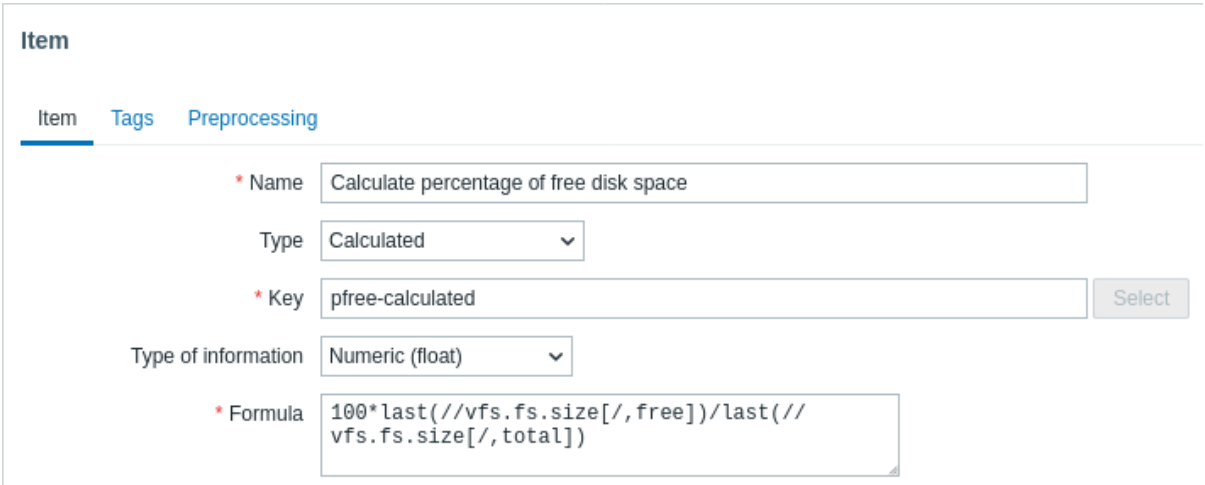
La seconda opzione consente di calcolare medie di gruppo, totali e così via.

Tutti i calcoli vengono eseguiti dal server Zabbix e gestiti dai processi `history poller`; proxy e agent non sono coinvolti in alcun modo. Si noti che un uso esteso degli item calcolati con periodi di tempo lunghi può influire sulle prestazioni del server.

Si noti che gli item calcolati possono accedere alla cronologia di altri item, inclusi gli item a cui l'utente che crea l'item calcolato non ha accesso. Se questo comportamento non è desiderato, è possibile disabilitare gli item calcolati impostando il parametro di configurazione `StartHistoryPollers` su 0.

Campi configurabili

Per utilizzare gli item calcolati, selezionare il tipo di item **Calculated**.



La **key** è un identificatore univoco dell'item (per host). È possibile creare qualsiasi nome di key utilizzando i simboli supportati. La key consente di fare riferimento all'item calcolato nei trigger.

La definizione del calcolo deve essere inserita nel campo **Formula**.

Non esiste alcuna relazione tra la formula e la key. I parametri della key non vengono utilizzati in alcun modo nella formula.

A differenza delle espressioni dei trigger, Zabbix elabora gli item calcolati in base all'intervallo di aggiornamento dell'item, non quando riceve un nuovo valore.

Se si seleziona *Numeric (unsigned)* come tipo di informazione e il risultato del calcolo è un valore a virgola mobile, questo viene troncato a un numero intero.

Formula di calcolo semplice

La sintassi di una formula semplice è:

`function(/host/key,<parameter1>,<parameter2>,...)`

dove:

Elemento	Descrizione
`function`	Una delle funzioni supportate: last, min, max, avg, count, ecc.
`host`	Host dell'item utilizzato per il calcolo. L'host corrente può essere omesso (cioè come in `function(//key,parameter,...)`).
`key`	Chiave dell'item utilizzato per il calcolo. Gli item che restituiscono valori con tipo di dato binario o JSON data type non sono supportati. Se si modifica la chiave item di un item referenziato, la chiave deve essere aggiornata manualmente nella formula. Tutti gli item referenziati dalle funzioni cronologia devono esistere e raccogliere dati.
`parameter(s)`	Parametri della funzione, se richiesti. Sono supportati i suffissi di tempo e i suffissi di dimensione della memoria.

Una formula più complessa può utilizzare una combinazione di funzioni, operatori e parentesi.

Vedere esempi di formule di item calcolati:

- (1) Traffico totale sull'interfaccia
- (2) Percentuale di spazio libero su disco
- (3) Percentuale di traffico in ingresso
- (4) Numero medio di valori elaborati

Alcune variabili possono essere passate alla formula con macro utente. Si noti che le macro utente:

- verranno risolte - se utilizzate per fare riferimento a un parametro di funzione, a un parametro di filtro item o a una costante.
- **non** verranno risolte - se fanno riferimento a una funzione, al nome host, alla chiave item, a un parametro della chiave item o a un operatore.

È consentito il confronto con stringhe.

Se nella cache non sono presenti dati recenti e nella funzione non è definito alcun periodo di interrogazione, Zabbix per impostazione predefinita risalirà fino a una settimana nel passato per interrogare il database alla ricerca di valori storici.

Un item calcolato può diventare non supportato in diversi casi:

- item referenziato
  - non viene trovato
  - è disabilitato oppure appartiene a un host disabilitato
  - non è supportato (tranne con la funzione nodata() e gli operatori con valori sconosciuti)
- nessun dato per calcolare una funzione
- divisione per zero
- sintassi non corretta utilizzata

Aggregazioni dei dati

Le aggregazioni dei dati si basano su un filtro che seleziona più item. Anche le aggregazioni dei dati sono un tipo di item calcolato, tuttavia nella formula si utilizza una sintassi leggermente diversa.

Per recuperare le aggregazioni utilizzare una delle funzioni di aggregazione supportate: `avg`, `max`, `min`, `sum`, ecc. Quindi aggiungere la funzione **foreach** come unico parametro e il relativo filtro item per selezionare gli item richiesti:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Una funzione **foreach** (ad esempio *avg_foreach*, *count_foreach*, ecc.) restituisce un valore aggregato per ciascun item selezionato. Gli item vengono selezionati utilizzando il filtro item (`/host/key?[group="host group"]`), dalla cronologia degli item. Per maggiori dettagli, vedere funzioni foreach.

Se alcuni item non hanno dati per il periodo richiesto, vengono ignorati nel calcolo. Se nessun item ha dati, la funzione restituirà un errore.

In alternativa, è possibile elencare più item come parametri per l'aggregazione:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Si noti che `function` qui deve essere una funzione di cronologia/trend.

Vedere esempi di calcoli aggregati:

- (5) Traffico totale in ingresso per host
- (6) Spazio totale su disco per gruppo
- (7) Carico medio del processore per gruppo
- (8) Carico medio del processore per gruppo e tag
- (9) Numero medio di valori elaborati
- (10) Numero medio di query per gruppo
- (11) Numero totale di controlli DNS riusciti
- (12) Numero totale di item non supportati per gruppo
- (13) Statistiche dell'interfaccia per gruppo

Le macro utente e le macro di low-level discovery sono supportate in:

- parametri della chiave item
- parametri della funzione
- condizioni del filtro (nome del gruppo host e nome del tag)
- costanti dell'espressione

Un calcolo aggregato può diventare non supportato se:

- non viene trovato nessuno degli item referenziati (ciò può accadere se la chiave item non è corretta, nessuno degli item esiste oppure tutti i gruppi inclusi non sono corretti)
- non ci sono dati per calcolare una funzione

Esempi di item calcolati

(1) Traffico totale sull'interfaccia

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Calcola la larghezza di banda totale su eth0.

(2) Percentuale di spazio libero su disco

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix prenderà gli ultimi valori dello spazio libero e dello spazio totale su disco (su `/`) e calcolerà la percentuale in base alla formula indicata.

(3) Percentuale del traffico in ingresso

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Calcola la percentuale del traffico in ingresso rispetto al traffico totale.

(4) Numero medio di valori elaborati

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Calcola una media su 10 minuti del numero di valori elaborati da Zabbix.

(5) Traffico totale in ingresso per il host

```
sum(last_foreach(/host/net.if.in[*]))
```

Calcola il totale di tutti gli item che corrispondono a `net.if.in[*]` sul host.

(6) Spazio totale su disco per il gruppo

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Calcola lo spazio totale su disco per tutti i `MySQL Servers`.

(7) Carico medio del processore per il gruppo

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Calcola il carico medio del processore per tutti i `MySQL Servers`.

(8) Carico medio del processore per gruppo e tag

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Calcola il carico medio della CPU su tutti gli host in più gruppi di host che hanno tag specifici.

(9) Numero medio di query per gruppo

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Calcola la media a 5 minuti del numero di query al secondo per `MySQL Servers`.

(10) Numero totale di controlli DNS riusciti

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Calcola il numero totale di controlli DNS riusciti su tutti gli host. Il modulo item mostrato è `net.dns[192.0.2.1,example.com,A]` come esempio di una possibile chiave.

Nota che i caratteri jolly devono corrispondere al numero di parametri nella chiave (qui `net.dns` ha tre parametri: ip, name, type).

(11) Numero totale di item non supportati per gruppo

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Calcola il numero totale di item non supportati per `Zabbix servers`.

(12) Statistiche dell'interfaccia per il gruppo

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Calcola le statistiche dell'interfaccia per il gruppo `video`.

Esempi di sintassi corretta/errata

Le espressioni (incluse le chiamate di funzione) non possono essere utilizzate come parametri history, trend o foreach delle funzioni. Tuttavia, tali funzioni possono essere utilizzate nei parametri di altre funzioni (non storiche).

Espressione	Esempio
Valida	`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)))`
Non valida	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

Si noti che, in un'espressione come:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

(dove `resptime[*]` è un esempio di pattern di chiave item che corrisponde alle chiavi `resptime[1]`, `resptime[2]` ecc., utilizzato per acquisire il tempo di risposta da diversi nodi attivi del cluster), non è possibile garantire che entrambe le parti dell'equazione abbiano sempre lo stesso insieme di valori. Durante la valutazione di una parte dell'espressione, potrebbe arrivare un nuovo valore per il periodo richiesto; di conseguenza, l'altra parte dell'espressione avrà un insieme di valori diverso.

Item dipendente

Panoramica

Ci sono situazioni in cui un item raccoglie più metriche contemporaneamente oppure ha ancora più senso raccogliere simultaneamente metriche correlate, per esempio:

- Utilizzo della CPU dei singoli core
- Traffico di rete in ingresso/in uscita/totale

Per consentire la raccolta massiva delle metriche e l'uso simultaneo in più item correlati, Zabbix supporta gli item dipendenti. Gli item dipendenti dipendono dal master item che raccoglie i loro dati simultaneamente, con una sola query. Un nuovo valore per il master item popola automaticamente i valori degli item dipendenti. Gli item dipendenti non possono avere un intervallo di aggiornamento diverso da quello del master item.

Le opzioni di preprocessing di Zabbix possono essere utilizzate per estrarre la parte necessaria per l'item dipendente dai dati del master item.

Il preprocessing è gestito da un processo `preprocessing manager`, insieme a thread worker che eseguono i passaggi di preprocessing. Tutti i valori con preprocessing, ricevuti da diversi raccoglitori di dati, passano attraverso il preprocessing manager prima di essere aggiunti alla cache della cronologia. Viene utilizzata una comunicazione IPC basata su socket tra i raccoglitori di dati (poller, trapper, ecc.) e il processo di preprocessing.

Zabbix server o Zabbix proxy (se l'host è monitorato dal proxy) eseguono i passaggi di preprocessing ed elaborano gli item dipendenti.

Gli item di qualsiasi tipo, anche gli item dipendenti, possono essere impostati come master item. Livelli aggiuntivi di item dipendenti possono essere utilizzati per estrarre parti più piccole dal valore di un item dipendente esistente.

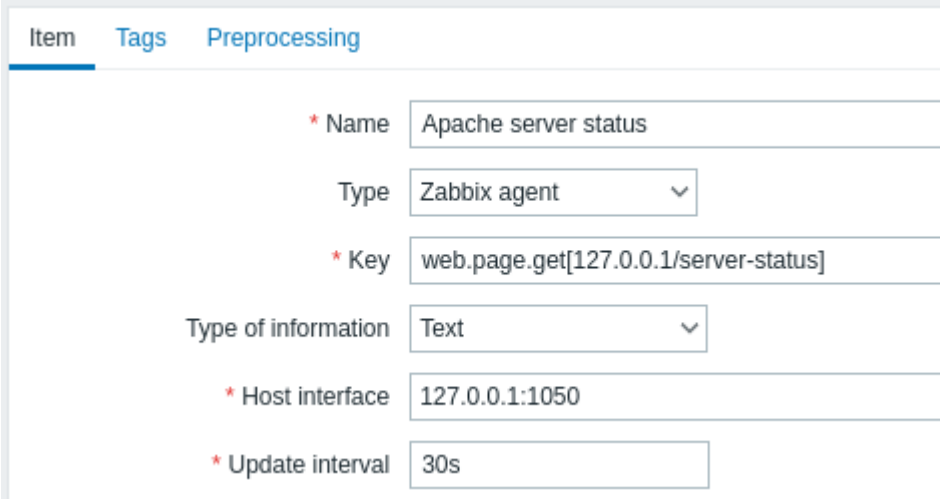
Limitazioni

- Sono consentite solo dipendenze sullo stesso host (template)
- Un prototipo di item può dipendere da un altro prototipo di item o da un normale item dello stesso host
- Un item dipendente su un host con item master da template non verrà esportato in XML

Configurazione dell'item

Un item dipendente dipende dal suo item master per i dati. Per questo motivo il **master item** deve essere configurato (o esistere) prima:

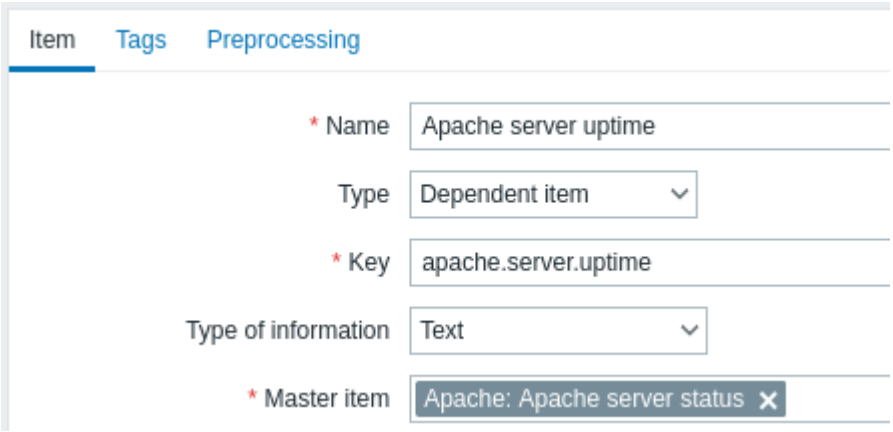
1. Vai a *Raccolta dati* > *Host*.
2. Fai clic su *Item* nella riga dell'host.
3. Fai clic su *Crea item*.
4. Inserisci i parametri dell'item nel modulo.



Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

Fai clic su *Aggiungi* per salvare il master item.

Quindi puoi configurare un **item dipendente**.



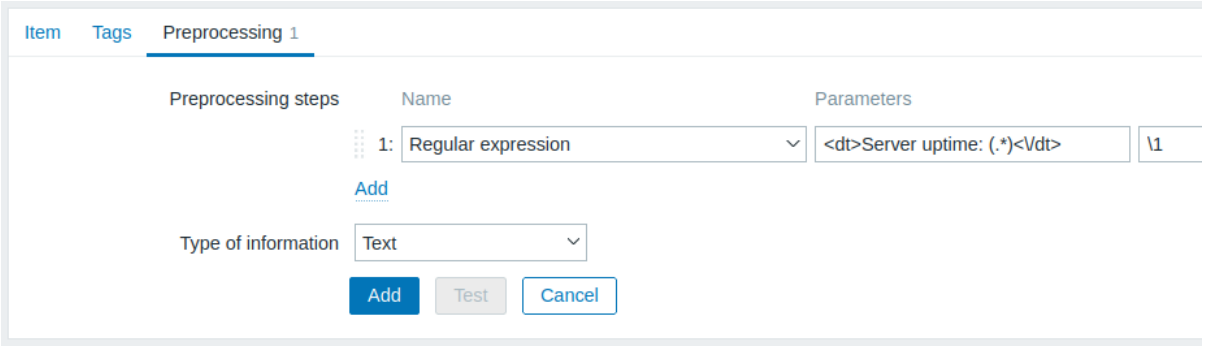
Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item dipendenti sono:

Type	Seleziona **Dependent item** qui.

Key	Inserisci una chiave che verrà usata per riconoscere l'item.
Master item	Seleziona il master item. Il valore del master item verrà usato per popolare il valore dell'item dipendente.
Type of information	Seleziona il tipo di informazione che corrisponderà al formato dei dati che verranno memorizzati.

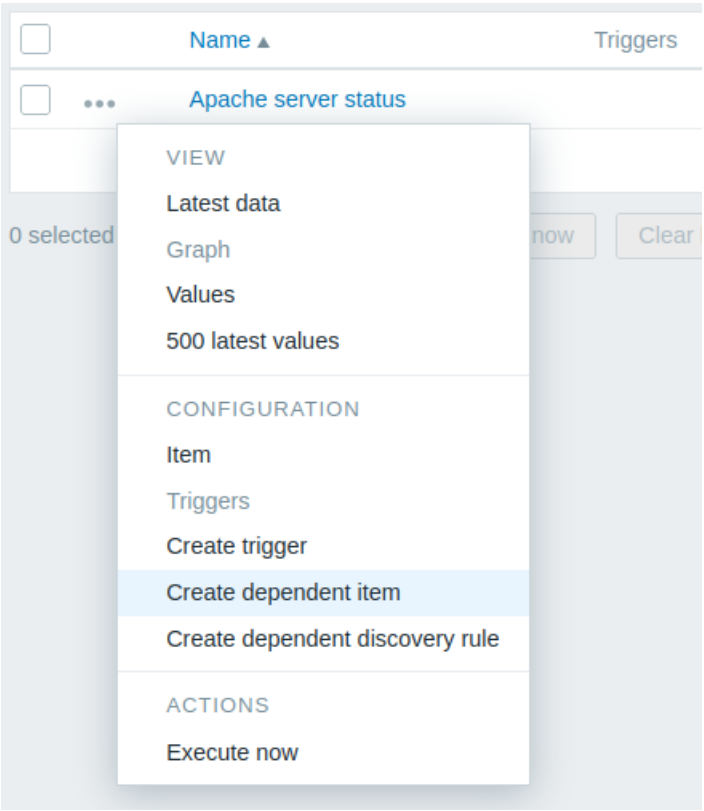
Puoi usare il preprocessing del valore dell'item per estrarre la parte richiesta del valore del master item.



Senza preprocessing, il valore dell'item dipendente sarà esattamente lo stesso del valore del master item.

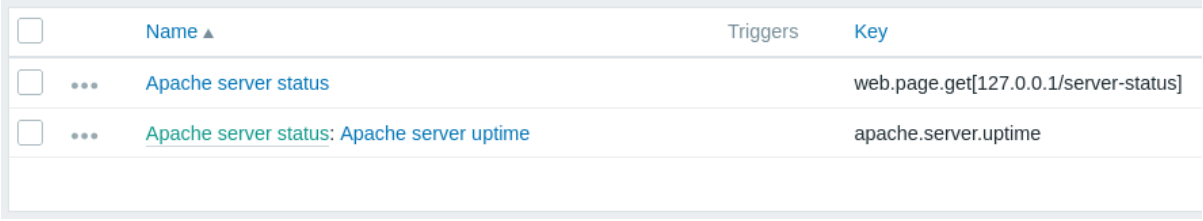
Fai clic su *Aggiungi* per salvare l'item dipendente.

Un collegamento per creare più rapidamente un item dipendente è accessibile facendo clic sul pulsante ••• nell'elenco degli item e selezionando *Create dependent item*.



Visualizzazione

Nell'elenco degli item, gli item dipendenti vengono visualizzati con il nome del loro item master come prefisso.



Se un item master viene eliminato, vengono eliminati anche tutti i suoi item dipendenti.

Item script

Panoramica

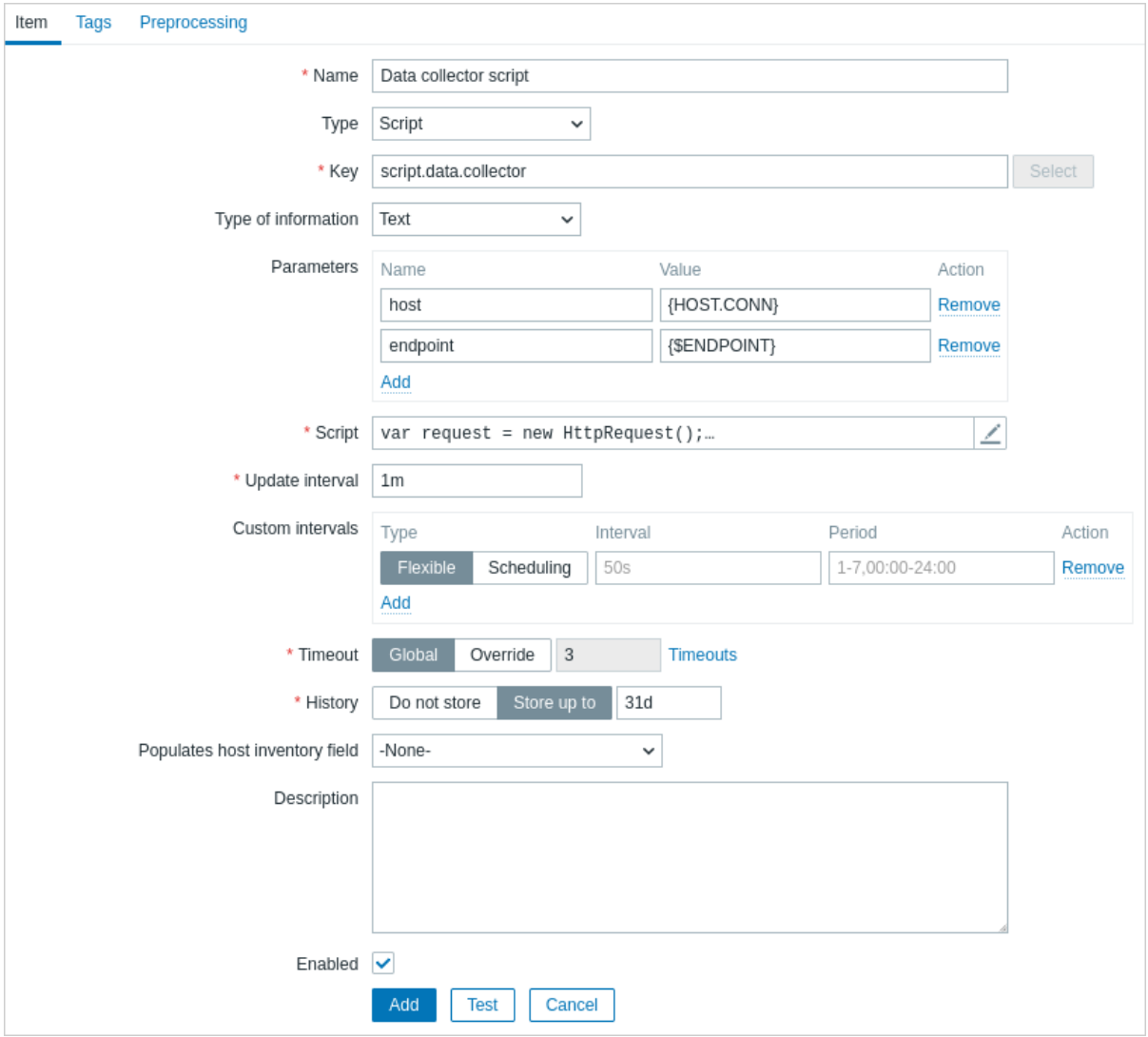
Gli item script possono essere utilizzati per raccogliere dati eseguendo codice JavaScript definito dall'utente, con la possibilità di recuperare dati tramite HTTP/HTTPS. Oltre allo script, è possibile specificare un elenco facoltativo di parametri (coppie di nome e valore) e un timeout.

Questo tipo di item può essere utile in scenari di raccolta dati che richiedono più passaggi o una logica complessa. Ad esempio, un item script può essere configurato per effettuare una chiamata HTTP, quindi elaborare in qualche modo i dati ricevuti nel primo passaggio e passare il valore trasformato alla seconda chiamata HTTP.

Gli item script vengono elaborati dai poller di Zabbix server o proxy.

Configurazione

Nel campo *Type* del modulo di configurazione dell'item selezionare Script, quindi compilare i campi richiesti.



Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item Script sono:

Field	Description
Key	Inserire una chiave univoca che verrà utilizzata per identificare l'item.
Parameters	Specificare le variabili da passare allo script come coppie attributo-valore. Sono supportate le macro utente. Per vedere quali macro integrate sono supportate, cercare "Script-type item" nella tabella delle macro supportate.

Field	Description
Script	Inserire il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona della matita accanto ad esso. Questo codice deve fornire la logica per restituire il valore della metrica. Il codice ha accesso a tutti i parametri e a tutti gli oggetti JavaScript aggiuntivi aggiunti da Zabbix. Vedere anche: Guida JavaScript.
Timeout	Timeout di esecuzione di JavaScript (1-600s; se superato, verrà restituito un errore). Si noti che, a seconda dello script, potrebbe essere necessario più tempo prima che il timeout venga attivato. Per ulteriori informazioni sul parametro *Timeout*, vedere attributi generali dell'item.

Esempi

Raccolta semplice dei dati

Raccogliere il contenuto di *https://www.example.com/release_notes*:

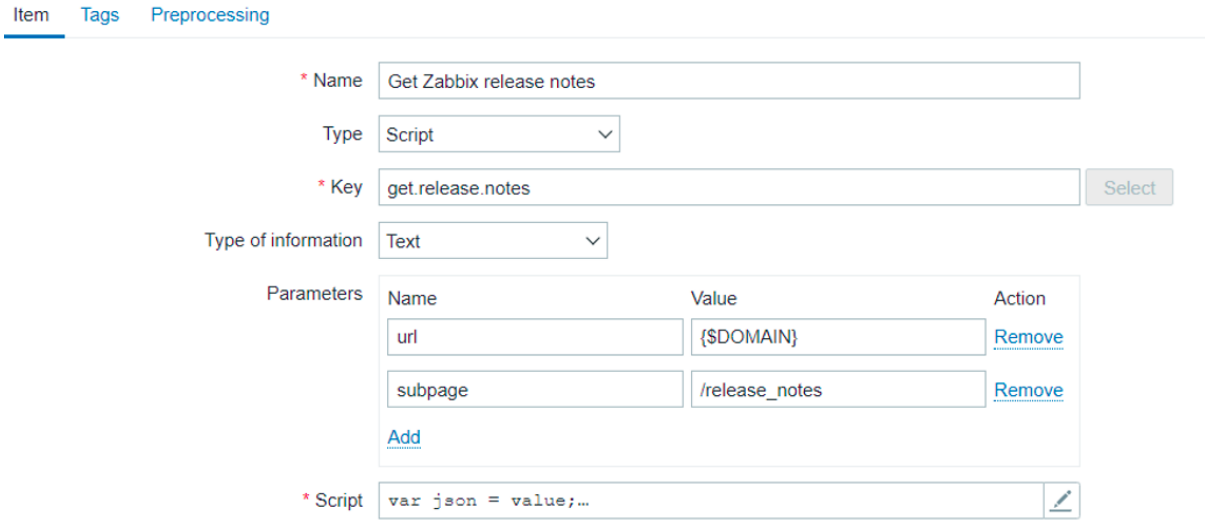
- Creare un item di tipo "Script".
- Nel campo *Script*, inserire:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Raccolta dei dati con parametri

Raccogliere il contenuto di una pagina specifica e utilizzare i parametri:

- Creare un item di tipo "Script" e due parametri:
  - **url : {$DOMAIN}** (la macro utente {$DOMAIN} deve essere definita, preferibilmente a livello di host)
  - **subpage : /release_notes**



- Nel campo *Script*, inserire:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

Richieste HTTP multiple

Raccogliere il contenuto sia di *https://www.example.com* sia di *https://www.example.com/release_notes*:

- Creare un item di tipo "Script".
- Nel campo *Script*, inserire:

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

Logging

Aggiungere la voce ”Log test” al log del server Zabbix e ricevere in risposta il valore dell’item ”1”:

- Creare un item di tipo ”Script”.
- Nel campo *Script*, inserire:

```
Zabbix.log(3, 'Log test');
return 1;
```

Item del browser

Panoramica

Gli item Browser consentono di monitorare siti web e applicazioni web complesse usando un browser.

> **Attention:**
> Il supporto degli item Browser è attualmente sperimentale.

Gli item Browser raccolgono dati eseguendo codice JavaScript definito dall’utente e recuperando i dati tramite HTTP/HTTPS. Questo item può simulare azioni correlate al browser, come fare clic, inserire testo, navigare tra le pagine web e altre interazioni dell’utente con siti web o applicazioni web.

Oltre allo script, è possibile specificare un elenco opzionale di parametri (coppie nome e valore) e un timeout.

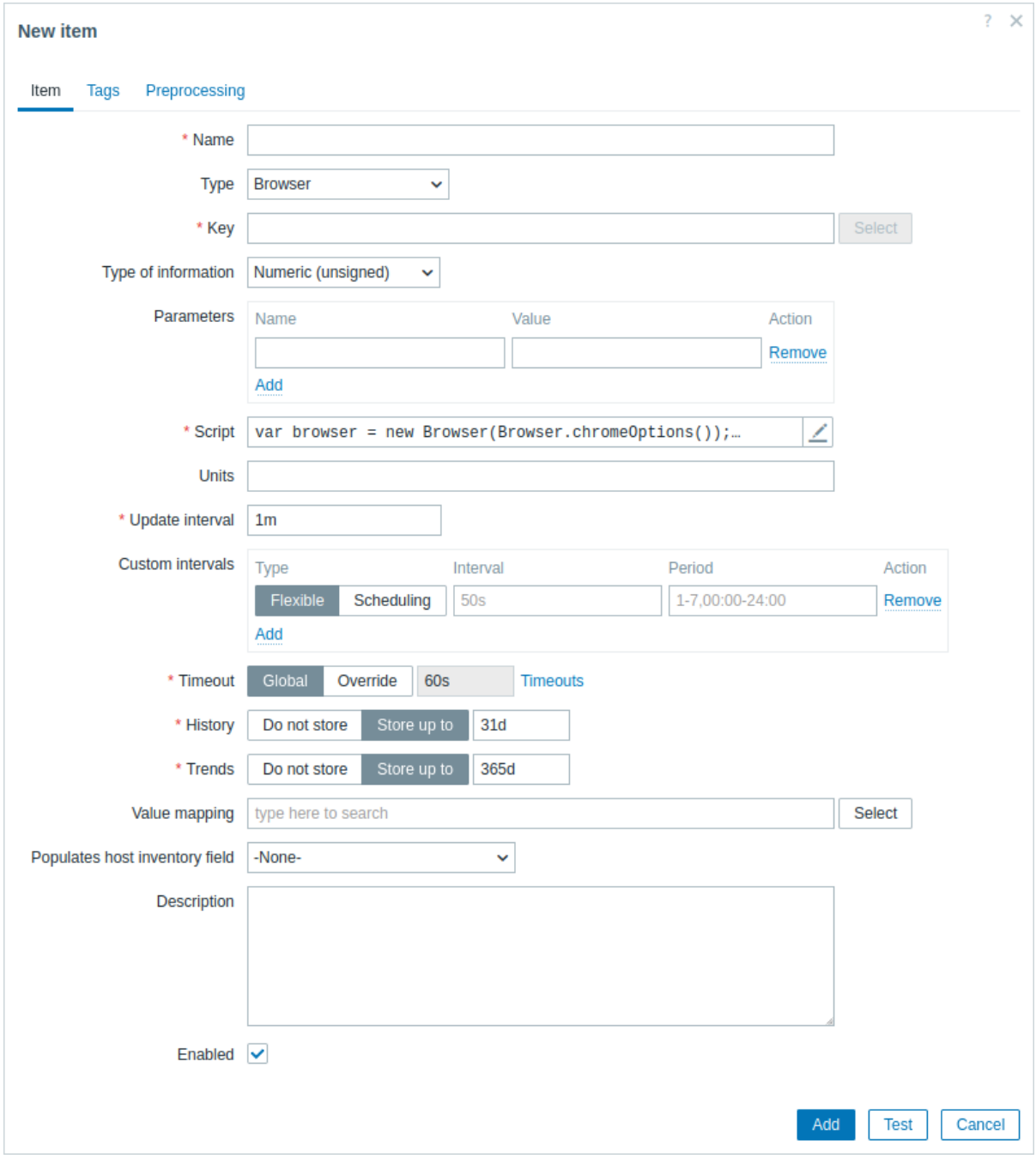
> **Attention:**
> L’item implementa parzialmente lo standard W3C WebDriver con Selenium Server oppure con un WebDriver semplice (ad esempio, ChromeDriver) come endpoint di test web. Per far funzionare l’item, impostare il parametro `WebDriverURL` nel file di configurazione di Zabbix server/proxy (se si usa ChromeDriver, vedere Security Considerations). Per prestazioni migliori, prendere in considerazione l’uso di un server dedicato per l’ambiente di test web.

I controlli degli item Browser vengono eseguiti ed elaborati dai browser pollers di Zabbix server o proxy. Se necessario, è possibile regolare il numero di istanze pre-fork dei browser pollers nel parametro `StartBrowserPollers` del file di configurazione di Zabbix server/proxy.

Per il monitoraggio di siti web e applicazioni web complesse, è disponibile il template Website by Browser come template pronto all’uso.

Configurazione

Nel campo *Tipo* del modulo di configurazione dell’item, selezionare Browser e quindi compilare i campi richiesti.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item Browser sono:

Campo	Descrizione
Key	Inserire una chiave univoca che verrà utilizzata per identificare l'item.
Parameters	Specificare le variabili da passare allo script come coppie attributo-valore. Sono supportate le macro utente. Per vedere quali macro integrate sono supportate, cercare "Browser-type item" nella tabella delle macro supportate.
Script	Inserire il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona della matita accanto ad esso. Questo codice deve fornire la logica per restituire il valore della metrica. Il codice ha accesso a tutti i parametri, a tutti gli oggetti JavaScript aggiuntivi e agli oggetti JavaScript dell'item Browser aggiunti da Zabbix. Vedere anche: Guida JavaScript.
Timeout	Timeout di esecuzione di JavaScript (1-600s; se superato, verrà restituito un errore). Si noti che, a seconda dello script, potrebbe essere necessario più tempo prima che il timeout venga attivato. Per ulteriori informazioni sul parametro *Timeout*, vedere attributi generali dell'item.

Esempi

Per un esempio di come configurare Zabbix per il monitoraggio dei siti web usando il template *Website by Browser*, vedere Monitorare i siti web con gli item Browser.

Script predefinito

Lo script seguente:

1. Inizializza una sessione del browser.
2. Passa a un URL specificato.
3. Raccoglie le voci delle prestazioni e le statistiche della sessione e le restituisce come stringa JSON.

Nel campo *Script*, inserire:

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

Inizializzare il browser con capacità personalizzate

Lo script seguente:

1. Inizializza una sessione del browser per il browser disponibile in base al primo browser corrispondente nell'ordine specificato nello script.
2. Definisce le capacità del browser, inclusa la strategia di caricamento della pagina e le opzioni specifiche per ciascun browser, come la modalità headless per i browser Chrome, Firefox e Microsoft Edge.

Nel campo *Script*, inserire:

```
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"
            }
        ]
    }
});
```

Inizializzare il browser con GUI

Per impostazione predefinita, le sessioni del browser (Safari escluso) vengono inizializzate in modalità headless, il che significa che l'interfaccia grafica utente (GUI) del browser non viene visualizzata.

Lo script seguente inizializza una sessione del browser con la GUI abilitata.

Si noti che, se WebDriver non riesce a individuare il file binario del browser, è possibile specificarne manualmente il percorso.

```
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:
Se i test sono in esecuzione su un server remoto o in un container, è possibile utilizzare un client Virtual Network Computing (VNC) per connettersi al server VNC della macchina. In questo modo è possibile visualizzare e interagire da remoto con la GUI del browser.

Acquisire screenshot

Il seguente script:

1. Inizializza una sessione del browser.
2. Imposta la dimensione della viewport del browser per determinare la dimensione dello screenshot (specificata come parametri, vedi sotto).
3. Apre un URL (specificato come parametro, vedi sotto).
4. Raccoglie le statistiche della sessione, acquisisce uno screenshot e lo aggiunge alle statistiche raccolte.
5. Gestisce gli errori acquisendo i messaggi di errore e uno screenshot.
6. Restituisce i risultati raccolti come stringa JSON.

Lo script utilizza anche i parametri del modulo di configurazione dell'item:

- webURL - http://example.com
- width - 1920
- height - 1080

Nel campo *Script*, inserire:

```
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);
}
```

Verifica dell'accesso a Zabbix

Lo script seguente:

1. Inizializza una sessione del browser.
2. Apre una pagina (specificata come parametro, vedi sotto).
3. Inserisce il nome utente e la password (specificati come parametri, vedi sotto).
4. Trova e fa clic sul pulsante di accesso.
5. Trova e fa clic sul pulsante di disconnessione.
6. Raccoglie i dati sulle prestazioni prima e dopo l'accesso, nonché dopo la disconnessione.
7. Gestisce gli errori acquisendo i messaggi di errore e una schermata.
8. Restituisce i risultati raccolti come stringa JSON.

Lo script utilizza anche i parametri del modulo di configurazione dell'item:

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

Nel campo *Script*, inserire:

```
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);
}
```

Trovare i link

Il seguente script:

1. Inizializza una sessione del browser.
2. Definisce una funzione per rimuovere gli elementi duplicati da un array (vedere il passaggio 5).
3. Accede a una pagina (specificata come parametri, vedere sotto).
4. Trova i link nella pagina.
5. Rimuove i link duplicati per garantire che siano univoci.
6. Estrae solo i link che iniziano con ”http”.
7. Formatta i link estratti secondo una struttura specifica.
8. Gestisce gli errori acquisendo i messaggi di errore e uno screenshot.
9. Restituisce i risultati raccolti come stringa JSON.

Lo script utilizza anche i parametri del modulo di configurazione dell’item:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

Nel campo *Script*, inserire:

```
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);
}
```

Query di telemetria

Questo tipo di item consente di recuperare o aggregare dati di telemetria. I dati di telemetria vengono utilizzati per il monitoraggio delle prestazioni delle applicazioni (APM), che consente di vedere come le richieste attraversano le applicazioni e i servizi, misurarne le prestazioni e individuare la causa delle operazioni lente o non riuscite.

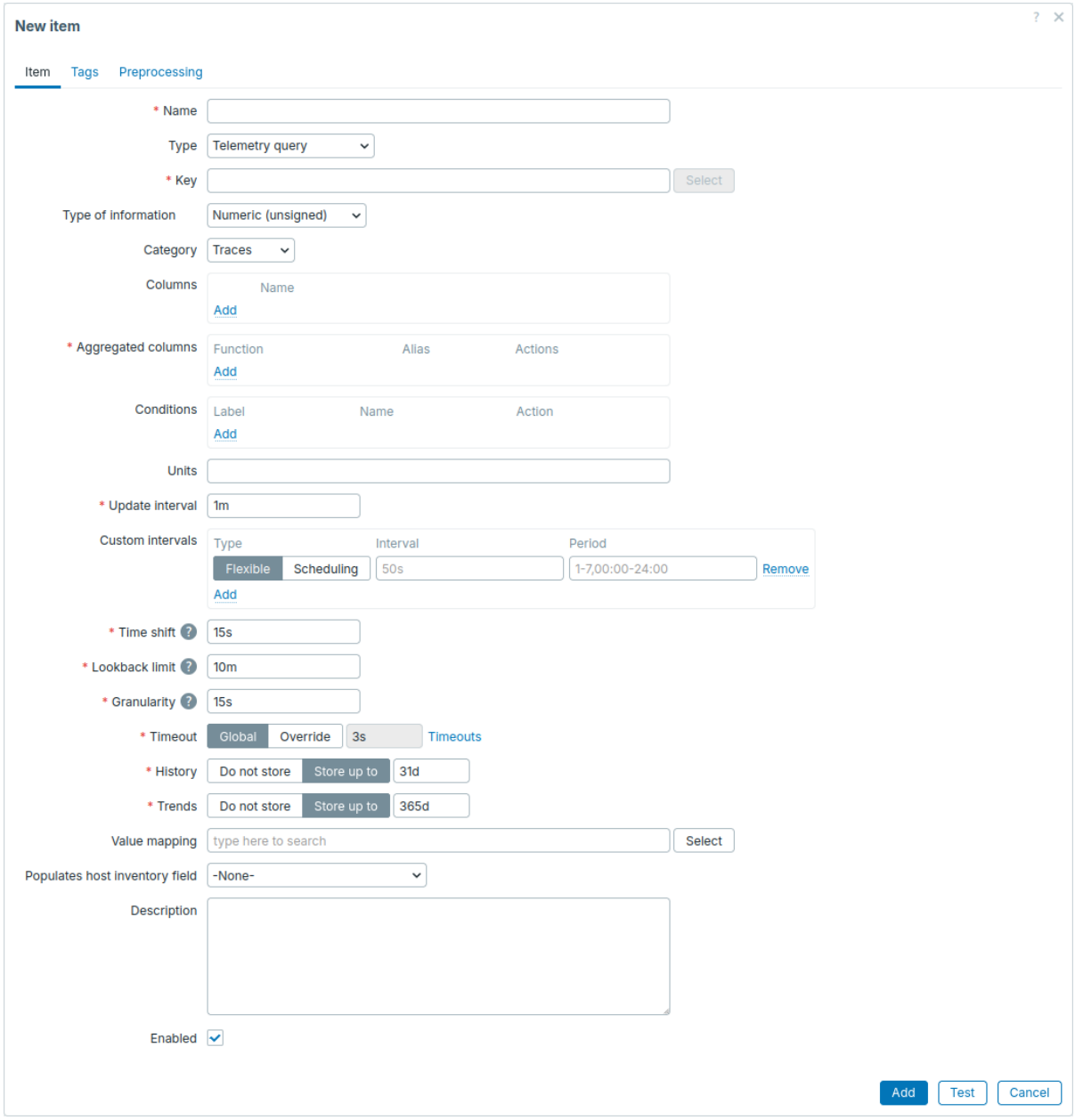
I dati di telemetria sono disponibili in tre tipi: tracce (il percorso seguito da una richiesta attraverso le applicazioni e i servizi), metriche (misurazioni, come il tempo di risposta) e log (registrazioni testuali degli eventi). Utilizzando l'item di query di telemetria, è possibile, ad esempio, trovare tutte le tracce in cui un servizio ha restituito un errore, ottenere il tempo di risposta medio dalle metriche o cercare nei log un messaggio di errore specifico.

Quando si configura un item di query di telemetria, si definisce un insieme di istruzioni (una query) che indica a Zabbix quali dati di telemetria richiedere, come filtrarli e come aggregarli. Quando viene eseguito il controllo dell'item, Zabbix invia questa query alla fonte dei dati di telemetria e recupera il risultato in base alla query.

Per utilizzare questo item, specificare la fonte dei dati di telemetria e impostare il parametro `StartTelemetryQueryPollers` nella configurazione del server Zabbix o del proxy Zabbix, a seconda di quale dei due monitora l'host dell'item.

Configurazione

Quando crei un item, imposta il relativo tipo su *Telemetry query*:



Parametro	Descrizione
Nome	Vedi i parametri comuni a tutti gli item.
Tipo	Seleziona *Telemetry query*.
Chiave	Vedi i parametri comuni a tutti gli item.
Tipo di informazione	
Categoria	Seleziona il tipo di segnale da interrogare: *Traces*, *Metrics* o *Logs*.
Punti metrica	Seleziona il tipo di punto metrica da interrogare: *Sum*, *Gauge*, *Histogram* o *Exponential histogram*. Questo parametro è disponibile se *Categoria* è impostata su *Metrics*.
Colonne	Aggiungi le colonne dei dati di telemetria da interrogare. Per le colonne che contengono coppie chiave-valore, inserisci anche il nome della chiave. I record di queste colonne saranno inclusi nel valore dell'item così come sono.

Parametro	Descrizione
Colonne aggregate	Aggiungi le colonne dei dati di telemetria da interrogare e specifica come calcolare un valore dai relativi record: **Funzione** - seleziona la funzione da applicare ai record della colonna: *min*, *max*, *avg*, *count*, *sum* o *percentile*. **Percentuale** - inserisci il percentile da calcolare dai record della colonna (se *Funzione* è impostata su *percentile*). **Colonna** - seleziona la colonna da cui effettuare il calcolo (se *Funzione* non è impostata su *count*). **Alias** - inserisci un nome univoco per il valore calcolato, utilizzato per identificarlo nel valore dell'item.
Tipo di calcolo	Seleziona come valutare le condizioni impostate nel parametro *Condizioni* di seguito: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso *Tipo* e la stessa *Chiave attributo* vengono combinate con *Or*. **And** - tutte le condizioni devono essere soddisfatte. **Or** - deve essere soddisfatta almeno una condizione. **Espressione personalizzata** - le condizioni devono essere soddisfatte come specificato nella formula, ad esempio `A and (B or C)`.
Condizioni	Questo parametro è disponibile se sono state impostate due o più condizioni. Aggiungi condizioni per filtrare i record delle colonne dei dati di telemetria: **Tipo** - seleziona la colonna dei dati di telemetria a cui applicare la condizione. **Chiave attributo** - inserisci la chiave a cui applicare la condizione (se *Tipo* è impostato su una colonna con coppie chiave-valore). **Operatore** - seleziona il confronto da utilizzare (vedi sotto) tra il valore del record e *Valore attributo*. **Valore attributo** - inserisci il valore con cui confrontare il valore del record (se *Operatore* non è impostato su *Exists*). Se *Tipo* è impostato su una colonna con un singolo valore, la condizione verrà applicata al record della colonna; se la colonna contiene un array di valori, la condizione corrisponderà al record se almeno un valore nell'array soddisfa la condizione. Se *Tipo* è impostato su una colonna che contiene coppie chiave-valore, la condizione verrà applicata al valore della chiave specificata in *Chiave attributo*; se la colonna contiene un array di coppie chiave-valore, la condizione corrisponderà al record se almeno una coppia con la *Chiave attributo* specificata soddisfa la condizione. Per le colonne con un singolo valore, puoi utilizzare questi operatori: **Equals** - corrisponde ai record uguali a *Valore attributo* (con distinzione tra maiuscole e minuscole). **Does not equal** - corrisponde ai record diversi da *Valore attributo* (con distinzione tra maiuscole e minuscole). **Contains** - corrisponde ai record che contengono *Valore attributo* (con distinzione tra maiuscole e minuscole). **Does not contain** - corrisponde ai record che non contengono *Valore attributo* (con distinzione tra maiuscole e minuscole). Per le colonne che contengono coppie chiave-valore, puoi utilizzare questi operatori: **Equals** - corrisponde ai record in cui *Chiave attributo* esiste e il relativo valore è uguale a *Valore attributo* (con distinzione tra maiuscole e minuscole). **Does not equal** - corrisponde ai record in cui *Chiave attributo* esiste e il relativo valore è diverso da *Valore attributo* (con distinzione tra maiuscole e minuscole). **Exists** - corrisponde ai record in cui *Chiave attributo* esiste (con distinzione tra maiuscole e minuscole).
Unità *Intervallo di aggiornamento* *Intervalli personalizzati*	Vedi i parametri comuni a tutti gli item.

Parametro	Descrizione
Spostamento temporale	Inserisci quanto indietro nel tempo, rispetto all'ora corrente, deve terminare l'intervallo temporale interrogato dall'item. Sono supportati un numero di secondi (ad esempio 1, 60, 300), un valore con un suffisso temporale (ad esempio 30s, 1m, 2h, 1d) o macro utente. Ogni query richiede dati per un intervallo temporale. L'intervallo in genere inizia dal punto in cui è terminato l'intervallo temporale della query precedente e termina nel punto impostato da *Spostamento temporale*. Se l'intervallo temporale della query precedente è terminato più tardi dell'ora corrente meno *Granularità*, l'intervallo temporale della query corrente inizia all'ora corrente meno *Granularità*, sovrapponendosi quindi all'intervallo temporale della query precedente. Se non esiste una query precedente da cui continuare, l'intervallo inizia dal punto impostato da *Limite di ricerca retrospettiva*. Se il monitoraggio dell'host di questo item passa a un'entità diversa (ad esempio, un proxy diverso di un gruppo di proxy), l'item potrebbe non sapere più quando è terminata la query precedente. Ciò può causare valori duplicati.
Limite di ricerca retrospettiva	Inserisci quanto indietro nel tempo, rispetto all'ora corrente, deve cercare dati l'item quando non esiste una query precedente da cui continuare (ad esempio, alla prima esecuzione dell'item) o quando la query precedente è più vecchia dell'*Intervallo di aggiornamento* (ad esempio, perché Zabbix server era inattivo). Sono supportati un numero di secondi (ad esempio 1, 60, 300), un valore con un suffisso temporale (ad esempio 30s, 1m, 2h, 1d) o macro utente. Se la fonte dei dati di telemetria contiene una grande quantità di dati (ad esempio, oltre un milione di record all'ora), un *Limite di ricerca retrospettiva* elevato può ridurre significativamente le prestazioni quando l'item lo utilizza.
Granularità	Inserisci la durata di una porzione in cui viene suddiviso l'intero intervallo temporale della query. Sono supportati un numero di secondi (ad esempio 1, 60, 300), un valore con un suffisso temporale (ad esempio 30s, 1m, 2h, 1d) o macro utente.
Timeout *Cronologia* *Trend* *Mappatura dei valori* *Compila il campo dell'inventario dell'host* *Descrizione* *Abilitato*	Vedi i parametri comuni a tutti gli item.

Colonne di dati di telemetria supportate

Questa sezione elenca le colonne supportate nei seguenti parametri dell'item: *Columns*, *Aggregated columns* e *Conditions*. Queste sono le colonne utilizzate da ClickHouse per archiviare i dati di telemetria, in base allo schema dell'esportatore ClickHouse di OpenTelemetry.

Selezionare la *Category*:

- Traces
- Metrics: Sum
- Metrics: Gauge
- Metrics: Histogram
- Metrics: Exponential histogram
- Logs

Tracce

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
Durata	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Timestamp	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

Metriche: Somma

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Metriche: Gauge

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
Attributi	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Metriche: Istogramma

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

Metriche: istogramma esponenziale

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

Log

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
Body	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓

Colonna	*Colonne*	*Colonne aggregate*	*Condizioni*
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

Cronologia e tendenze

Panoramica

La cronologia e i trend sono i due modi di memorizzare i dati raccolti in Zabbix.

Mentre la cronologia conserva ogni valore raccolto, i trend mantengono informazioni medie su base oraria e pertanto richiedono meno risorse.

Conservazione della cronologia

Puoi impostare per quanti giorni la cronologia verrà conservata:

- nelle proprietà dell'item form
- durante l'aggiornamento massivo degli item
- durante la configurazione delle attività di housekeeper

Tutti i dati più vecchi verranno rimossi dall'housekeeper.

Il consiglio generale è di conservare la cronologia per il minor numero di giorni possibile, in modo da non sovraccaricare il database con molti valori storici.

Invece di conservare una lunga cronologia, puoi conservare più a lungo i dati di trend. Ad esempio, potresti conservare la cronologia per 14 giorni e i trend per 5 anni.

Puoi farti un'idea abbastanza precisa di quanto spazio sia richiesto dai dati di cronologia rispetto ai dati di trend facendo riferimento alla pagina sul dimensionamento del database.

Anche mantenendo una cronologia più breve, potrai comunque consultare i dati meno recenti nei grafici, poiché i grafici utilizzeranno i valori di trend per visualizzare i dati più vecchi.

Attention:
Se la cronologia è impostata su '0', l'item aggiornerà solo gli item dipendenti e l'inventario. Nessuna funzione trigger verrà valutata, perché la valutazione dei trigger si basa esclusivamente sui dati della cronologia.

Note:
Come metodo alternativo per preservare la cronologia, valuta l'utilizzo della funzionalità di esportazione della cronologia dei moduli caricabili.

Conservazione dei trend

I trend sono un meccanismo integrato di riduzione dei dati storici che memorizza i valori minimo, massimo, medio e il numero totale di valori per ogni ora per i tipi di dati numerici.

Puoi impostare per quanti giorni i trend verranno conservati:

- nelle proprietà dell'item form
- durante l'aggiornamento massivo degli item
- durante la configurazione delle attività di Housekeeper

Di solito i trend possono essere conservati molto più a lungo della cronologia. Qualsiasi dato più vecchio verrà rimosso dall'housekeeper.

Zabbix server accumula i dati dei trend in fase di esecuzione nella cache dei trend, man mano che i dati arrivano. Il server scrive nel database i trend dell'**ora precedente** di ogni item (dove il frontend può trovarli) nelle seguenti situazioni:

- il server riceve il primo valore dell'ora corrente dell'item
- mancano 5 minuti o meno alla fine dell'ora corrente e non ci sono ancora valori dell'ora corrente dell'item
- il server si arresta

Per vedere i trend in un grafico è necessario attendere almeno fino all'inizio dell'ora successiva (se l'item viene aggiornato frequentemente) e al massimo fino alla fine dell'ora successiva (se l'item viene aggiornato raramente), cioè al massimo 2 ore.

Quando il server scrive la cache dei trend e nel database esistono già trend per questa ora (ad esempio, il server è stato riavviato a metà ora), il server deve usare istruzioni di aggiornamento invece di semplici inserimenti. Pertanto, in un'installazione più grande, se è necessario un riavvio, è preferibile arrestare il server alla fine di un'ora e avviarlo all'inizio dell'ora successiva per evitare la sovrapposizione dei dati dei trend.

Le tabelle della cronologia non partecipano in alcun modo alla generazione dei trend.

Attention:
Se i trend sono impostati a '0', Zabbix server non calcola né memorizza affatto i trend.

Note:
I trend vengono calcolati e memorizzati con lo stesso tipo di dato dei valori originali. Di conseguenza, i calcoli del valore medio dei valori di tipo unsigned vengono arrotondati e quanto più piccolo è l'intervallo dei valori, tanto meno preciso sarà il risultato. Ad esempio, se un item ha i valori 0 e 1, il valore medio sarà 0, non 0.5.
Inoltre, il riavvio del server potrebbe causare una perdita di precisione nei calcoli del valore medio per il tipo di dato unsigned per l'ora corrente.

Parametri utente

Overview

A user parameter lets you define a custom item key and the command used to retrieve its value when no built-in Zabbix agent item provides the data you need. A user parameter is a user-defined command referenced by a user-defined item key and executed by the Zabbix agent when the corresponding item is checked.

Note:
On UNIX systems, the agent executes user parameter commands using `/bin/sh`, under the operating system account running the agent (usually `zabbix`). That account must have sufficient permissions to execute the command and access any files, sockets, or credentials it uses.

There are two types of user parameters:

- **Simple user parameters** do not accept arguments; they define a fixed item key and a command to execute.
- **Flexible user parameters** accept arguments (specified as parameters in the item key) and pass them to the command, allowing a single definition to serve multiple items.

User parameter configuration

A user parameter is defined using the `UserParameter` configuration parameter.

A user parameter has the following syntax:

```
UserParameter=<key>,<shell command>
```

where:

- <key> - a unique item key that identifies the user parameter.
 When configuring an item, specify this key to reference the command defined by the user parameter.
 For a simple user parameter, the key is fixed and does not accept arguments. For a flexible user parameter, append `[*]` to the key (for example, `mysql.status[*]`) to indicate that it accepts arguments. Specify these arguments as parameters in the item key when configuring the item. The command can refer to the parameters using positional references $1–$9.
- <shell command> - a shell command or a path to a custom script that the agent executes to obtain the item value.
 For the flexible user parameter, the command can use positional references $1–$9 to refer to the corresponding parameters in the item key (for example, `UserParameter=mysql.status[*],mysqladmin status --user=$1 --password=$2`). Zabbix parses the parameters enclosed in `[]` in the item key (i.e., `mysql.status[zabbix,{$MYSQL.PASSWORD}]`) and substitutes the positional references in the command accordingly.
 $0 is substituted with the original command before positional references are expanded. Positional references are interpreted regardless of whether they are enclosed in double

(") or single (') quotes. To use positional references without substitution, specify a double dollar sign. For example, `awk '{print $$2}'` passes $2 to the command unchanged.
 Note that positional references prefixed with $ are searched for and replaced by the Zabbix agent only in flexible user parameters. For simple user parameters, positional references are not processed, so escaping $ is unnecessary.

Attention:
Certain characters cannot be passed as arguments by default. For the complete list of characters, see UnsafeUserParameters.

`UserParameter` must be configured on each Zabbix agent that uses it: - Directly in the Zabbix agent configuration file (`zabbix_agentd.conf` or `zabbix_agent2.conf`). - In a separate `*.conf` file, provided that the file is included by the agent configuration.

More than one `UserParameter` can be defined per agent, as multiple files can be included with different sets of parameters. Each user parameter must have a unique item key within the monitored host.

The `UserParameterDir` configuration parameter lets you specify the working directory for executing `UserParameter` commands. If configured, the agent changes its working directory to the specified directory before executing a command. Commands can be specified using a relative path with the `./` prefix, for example, `UserParameter=mykey,./myscript.sh`.

To configure a user parameter:

1. Write a custom script or command to retrieve the required data.
2. Add the command and its unique item key to the `UserParameter` configuration parameter in the Zabbix agent configuration file (`zabbix_agentd.conf`).
3. Reload user parameters from the configuration file by running the runtime control command:

   zabbix_agentd -R userparameter_reload

   Alternatively, restart the agent to apply the configuration changes.
4. Create a new item on the monitored host with the item key defined in `UserParameter`.
 Set the item's type to *Zabbix Agent* or *Zabbix Agent (active)*.
 Make sure that the item's *Type of information* is configured correctly; otherwise, Zabbix will reject the returned value.

See Extending Zabbix agents for an example of configuring and using user parameters.

User parameters are subject to the Zabbix agent check timeout. If the timeout is reached, the process executing the user parameter is terminated. See Command execution for more information.

Note that long-running scripts may consume system resources and delay other checks.

Risultato del comando

Il valore restituito dal comando è l'output standard insieme all'errore standard prodotto dal comando.

Attention:
Un item che restituisce testo (informazioni di tipo character, log o text) non diventerà unsupported in caso di output sull'errore standard.

Il valore restituito è limitato a 16MB (inclusi gli spazi finali che vengono troncati); si applicano anche i limiti del database.

I parametri utente che restituiscono testo (informazioni di tipo character, log o text) possono anche restituire spazi vuoti. In caso di risultato non valido, l'item diventerà unsupported.

Esempi di parametri utente semplici

Un comando semplice:

```
UserParameter=ping,echo 1
```

L'agent restituirà sempre '1' per un item con chiave 'ping'.

Un esempio più complesso:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

L'agent restituirà '1' se il server MySQL è attivo, altrimenti '0'.

Esempio 1

Qualcosa di molto semplice:

```
UserParameter=ping[*],echo $1
```

Possiamo definire un numero illimitato di item per il monitoraggio, tutti con il formato ping[qualcosa].

- ping[0] - restituirà sempre '0'
- ping[aaa] - restituirà sempre 'aaa'

Esempio 2

Aggiungiamo più funzionalità!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

Questo parametro può essere utilizzato per monitorare la disponibilità del database MySQL. Possiamo passare nome utente e password:

```
mysql.ping[zabbix,our_password]
```

Esempio 3

Quante righe corrispondono a un'espressione regolare in un file?

```
UserParameter=wc[*],grep -c "$2" $1
```

Questo parametro può essere utilizzato per calcolare il numero di righe in un file.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Estendere gli agent Zabbix

Questo tutorial fornisce istruzioni passo passo su come estendere le funzionalità dell'agent Zabbix utilizzando un user parameter.

Passaggio 1

Scrivere uno script o una riga di comando per recuperare il parametro richiesto.

Ad esempio, possiamo scrivere il seguente comando per ottenere il numero totale di query eseguite da un server MySQL:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

Quando viene eseguito, il comando restituisce il numero totale di query SQL.

Passaggio 2

Aggiungere il comando al file di configurazione dell'agent di Zabbix (`zabbix_agentd.conf`):

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions è un identificatore univoco. Può essere qualsiasi identificatore di chiave valido, ad esempio *queries*.

Testare questo parametro utilizzando l'agent Zabbix con il flag `-t` (se viene eseguito come root, tenere tuttavia presente che l'agent potrebbe avere autorizzazioni diverse quando viene avviato come daemon):

```
zabbix_agentd -t mysql.questions
```

Passaggio 3

Ricaricare i parametri utente dal file di configurazione eseguendo:

```
zabbix_agentd -R userparameter_reload
```

È anche possibile riavviare l'agent invece di usare il comando di controllo a runtime.

Testare il parametro utilizzando l'utilità `zabbix_get`.

Passaggio 4

Aggiungi un nuovo item con `Key=mysql.questions` all'host monitorato. Il tipo dell'item deve essere *Zabbix Agent* oppure *Zabbix Agent (active)*.

Tieni presente che il tipo dei valori restituiti deve essere impostato correttamente sul server Zabbix. In caso contrario, Zabbix non li accetterà.

Contatori delle prestazioni di Windows

Panoramica

È possibile monitorare efficacemente i contatori delle prestazioni di Windows utilizzando la chiave `perf_counter[]`.

Ad esempio:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

oppure

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Per ulteriori informazioni sull'utilizzo di questa chiave o del suo equivalente disponibile solo in inglese, `perf_counter_en`, vedere Chiavi degli item specifiche di Windows.

Per ottenere un elenco completo dei contatori delle prestazioni disponibili per il monitoraggio, è possibile eseguire:

```
typeperf -qx
```

È inoltre possibile utilizzare il low-level discovery per individuare più istanze di oggetti dei contatori delle prestazioni di Windows e automatizzare la creazione di item `perf_counter` per più oggetti istanza.

Rappresentazione numerica

Windows mantiene rappresentazioni numeriche (indici) per i nomi degli oggetti e dei contatori delle prestazioni. Zabbix supporta queste rappresentazioni numeriche come parametri per le chiavi item `perf_counter`, `perf_counter_en` e nei parametri di configurazione `PerfCounter`, `PerfCounterEn`.

Tuttavia, se ne sconsiglia l'uso a meno che non sia possibile garantire che gli indici numerici corrispondano alle stringhe corrette su host specifici. Se è necessario creare item portabili che funzionino su host diversi con varie versioni localizzate di Windows, è possibile utilizzare la chiave `perf_counter_en` o il parametro di configurazione `PerfCounterEn`, che consentono di usare i nomi in inglese indipendentemente dalle impostazioni locali del sistema.

Per individuare gli equivalenti numerici, eseguire `regedit`, quindi individuare *Counter* in `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wi NT\CurrentVersion\Perflib\009`.

La voce del registro contiene informazioni come queste:

```
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
....
```

Qui è possibile trovare i numeri corrispondenti a ogni parte stringa del contatore delle prestazioni, come in `\System\% Processor Time`:

```
System → 2
% Processor Time → 6
```

È quindi possibile utilizzare questi numeri per rappresentare il percorso in forma numerica:

```
\2\6
```

Parametri dei contatori delle prestazioni

È possibile distribuire alcuni parametri PerfCounter per il monitoraggio dei contatori delle prestazioni di Windows.

Ad esempio, è possibile aggiungere questi parametri al file di configurazione di Zabbix agent:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
oppure
PerfCounter=UserPerfCounter2,"\4\24",30
```

Dopo aver configurato questi parametri, è sufficiente utilizzare `UserPerfCounter1` o `UserPerfCounter2` come chiavi per creare i rispettivi item.

Ricordarsi di riavviare Zabbix agent dopo aver apportato modifiche al file di configurazione.

Aggiornamento di massa

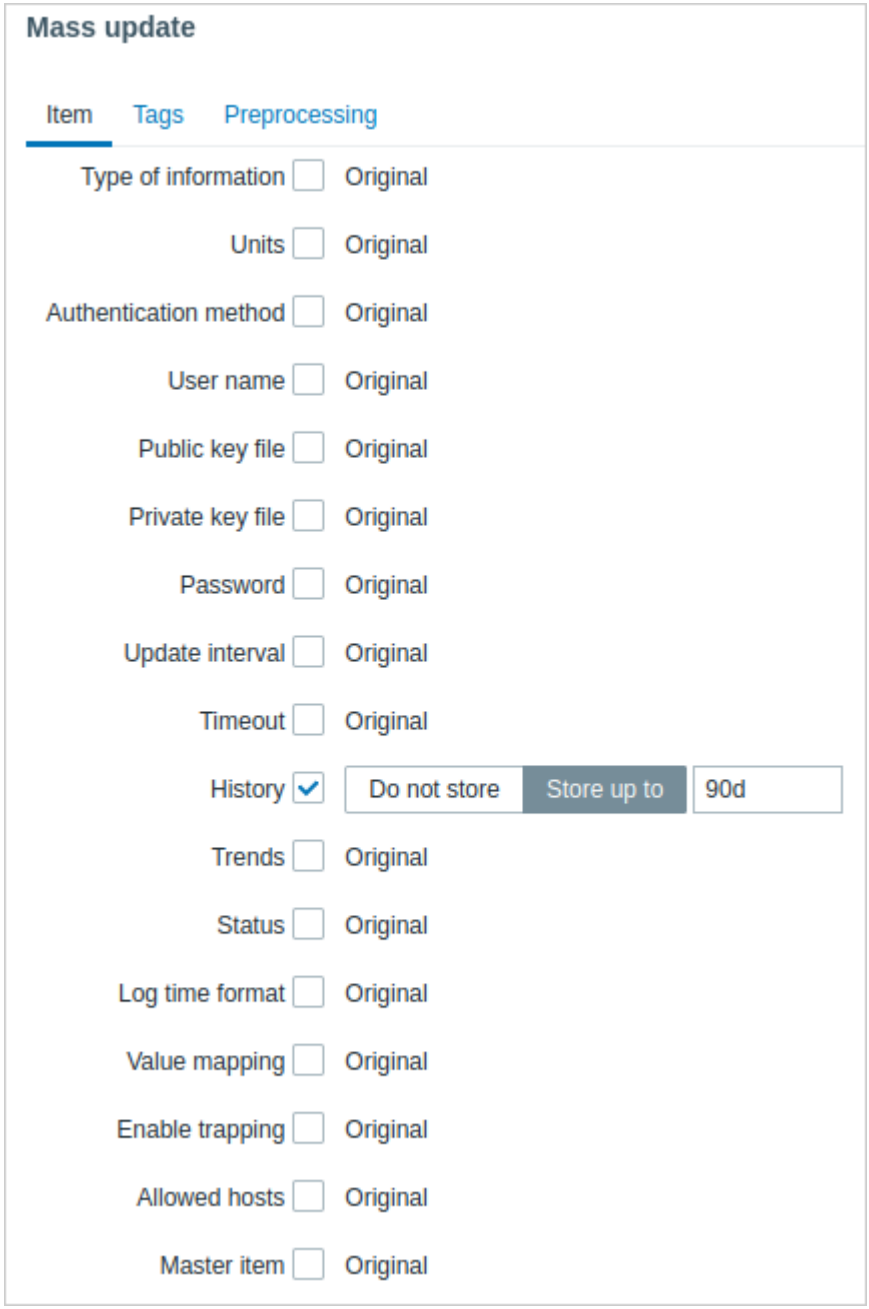
Panoramica

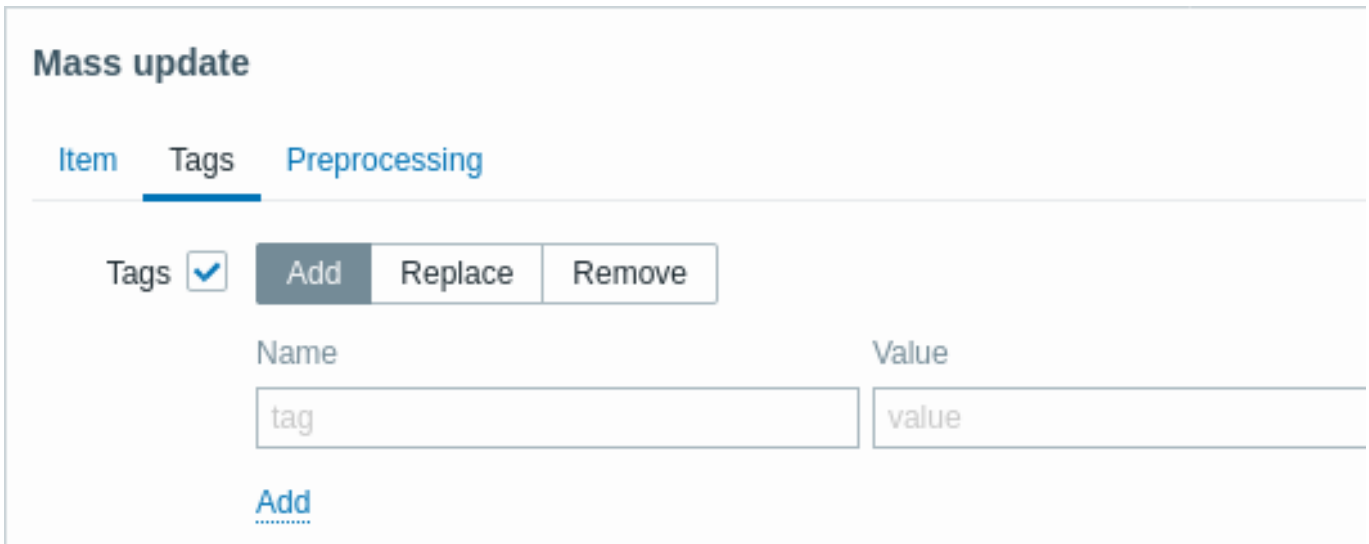
A volte potresti voler modificare alcuni attributi per un numero di oggetti in una singola volta. Invece di aprire ogni singolo oggetto per la modifica, puoi utilizzare la funzione di aggiornamento di massa.

Utilizzo dell'aggiornamento di massa

Per aggiornare più item contemporaneamente, procedere come segue:

- Selezionare le caselle di controllo degli item da aggiornare nell'elenco
- Fare clic su *Aggiornamento di massa* sotto l'elenco
- Passare alla scheda con gli attributi richiesti (*Item*, *Tag* o *Preprocessing*)
- Selezionare le caselle di controllo degli attributi da aggiornare
- Immettere i nuovi valori per gli attributi



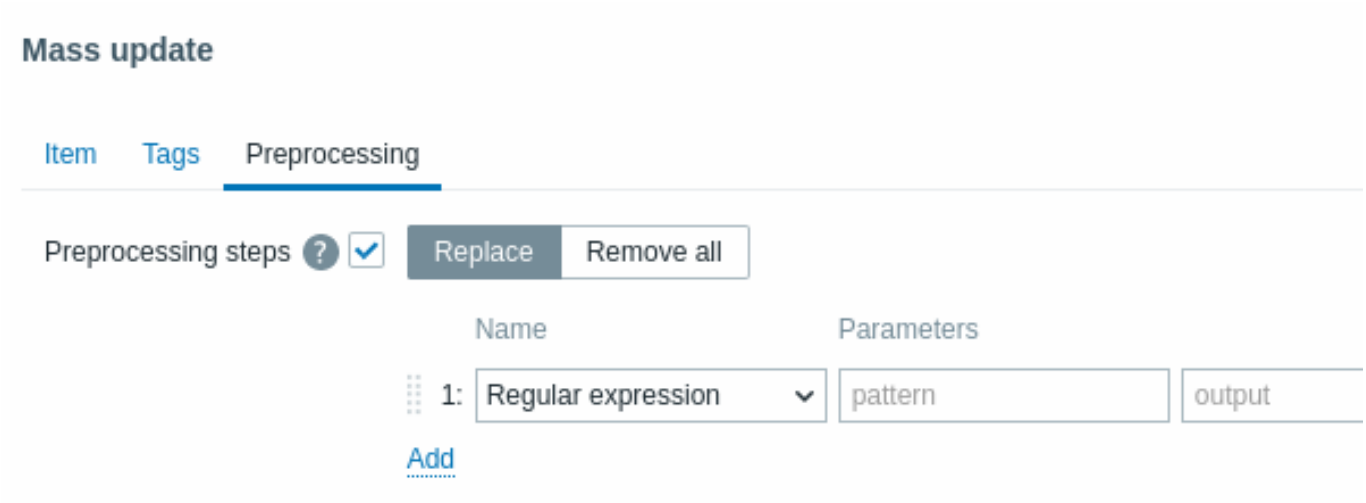


L'opzione *Tag* consente di:

- *Aggiungere* - aggiungere i tag specificati agli item (i tag con lo stesso nome, ma valori diversi non sono considerati "duplicati" e possono essere aggiunti allo stesso item).
- *Sostituire* - rimuovere i tag specificati e aggiungere tag con nuovi valori

- *Rimuovere* - rimuovere i tag specificati dagli item

Le macro utente, le macro `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` e `{HOST.ID}` sono supportate nei tag.



Selezionare la casella di controllo accanto a *Passaggi di preprocessing* per attivare l'aggiornamento di massa dei passaggi di preprocessing.

L'aggiornamento di massa *Preprocessing* consente di:

- *Sostituire* - sostituire i passaggi di preprocessing esistenti sugli item con i passaggi di preprocessing specificati di seguito
- *Rimuovere tutto* - rimuovere tutti i passaggi di preprocessing dagli item

Al termine, fare clic su *Aggiorna*.

Mappatura dei valori

Panoramica

La mappatura dei valori consente di configurare una rappresentazione più intuitiva dei valori ricevuti utilizzando associazioni tra valori numerici/stringa e rappresentazioni testuali.

Ad esempio, quando il valore di un item è "0" o "1", le mappature dei valori possono essere utilizzate per rappresentare questi valori in modo più comprensibile:

- 0 → Non disponibile
- 1 → Disponibile

Le mappature dei valori per i tipi di backup dei dati potrebbero essere configurate come segue:

- F → Completo
- D → Differenziale
- I → Incrementale

Gli intervalli di valori per la tensione potrebbero essere configurati come segue:

- <=209 → Bassa
- 210-230 → OK
- >=231 → Alta

La mappatura dei valori viene utilizzata nel frontend di Zabbix e nelle notifiche inviate dai tipi di supporto.

Attention:
La sostituzione del valore ricevuto con la rappresentazione configurata viene eseguita sia nel frontend di Zabbix sia nel server; tuttavia, il server gestisce la sostituzione solo nei seguenti casi:

- durante il popolamento dell'inventario host;
- durante l'espansione delle macro supportate {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} e {EVENT.CAUSE.OPDATA}.

Le mappature dei valori vengono impostate su template o host. Una volta configurate, sono disponibili per tutti gli item all'interno del rispettivo template o host. Quando si configurano gli item, specificare il nome di una mappatura dei valori configurata in precedenza nel parametro *Value mapping*.

Note:
Non esiste ereditarietà delle mappe dei valori: host e template non ereditano le mappature dei valori dai template collegati. Gli item del template su un host continueranno a utilizzare le mappature dei valori configurate sul template.

Note:
Le mappature dei valori possono essere utilizzate con item che hanno tipi di informazione *Numeric (unsigned)*, *Numeric (float)* e *Character*.

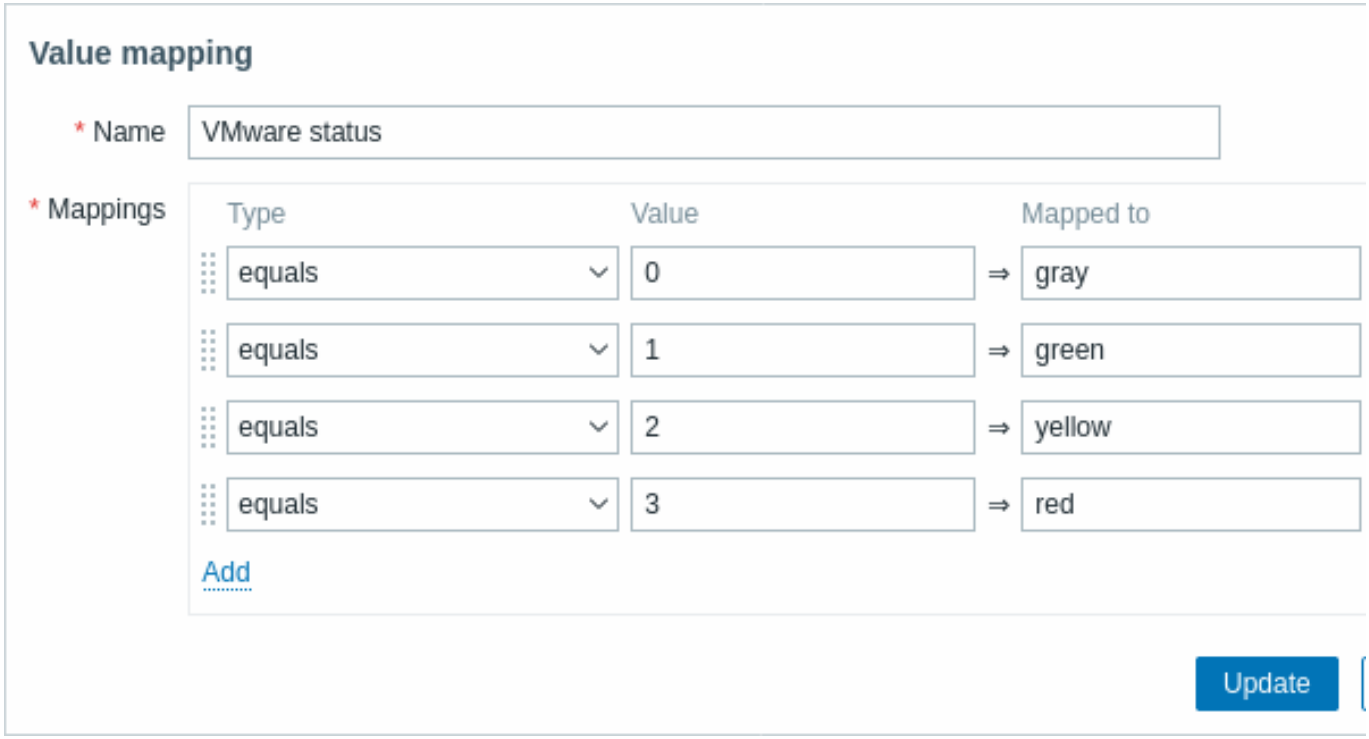
Le mappature dei valori vengono esportate/importate con il rispettivo template o host. Possono anche essere aggiornate in massa utilizzando i moduli di aggiornamento di massa di host e template.

Configurazione

Per configurare una mappatura dei valori, procedere come segue:

1. Aprire il modulo di configurazione dell'host o del template.

2. Nella scheda *Mappatura dei valori*, fare clic su *Aggiungi* per aggiungere una nuova mappatura dei valori oppure fare clic sul nome di una mappatura esistente per modificarla.



Parametri di una mappatura dei valori:

Parametro	Descrizione
Nome	Nome univoco per l'insieme delle mappature dei valori.
Mappature	Regole individuali per la mappatura di valori numerici/stringa a rappresentazioni stringa. La mappatura viene applicata nell'ordine delle regole, che può essere modificato trascinandole.

Parametro	Descrizione
Tipo	Tipo di mappatura: **uguale a** - verranno mappati i valori uguali; **maggiore o uguale a** - verranno mappati i valori uguali o maggiori; **minore o uguale a** - verranno mappati i valori uguali o minori; **nell'intervallo** - verranno mappati i valori nell'intervallo; l'intervallo è espresso come <number1>-<number2> o <number>; sono supportati più intervalli (ad esempio, 1-10,101-110,201); **regexp** - verranno mappati i valori corrispondenti all'espressione regolare (le espressioni regolari globali non sono supportate); **predefinito** - verranno mappati tutti i valori rimanenti, ad eccezione di quelli con mappature specifiche. Per la mappatura degli intervalli sono supportati solo i tipi di valore numerici (*maggiore o uguale a*, *minore o uguale a*, *nell'intervallo*).
Valore	Valore in ingresso (può contenere un intervallo o un'espressione regolare, a seconda del tipo di mappatura).
Mappato a	Rappresentazione stringa (fino a 64 caratteri) per il valore in ingresso.

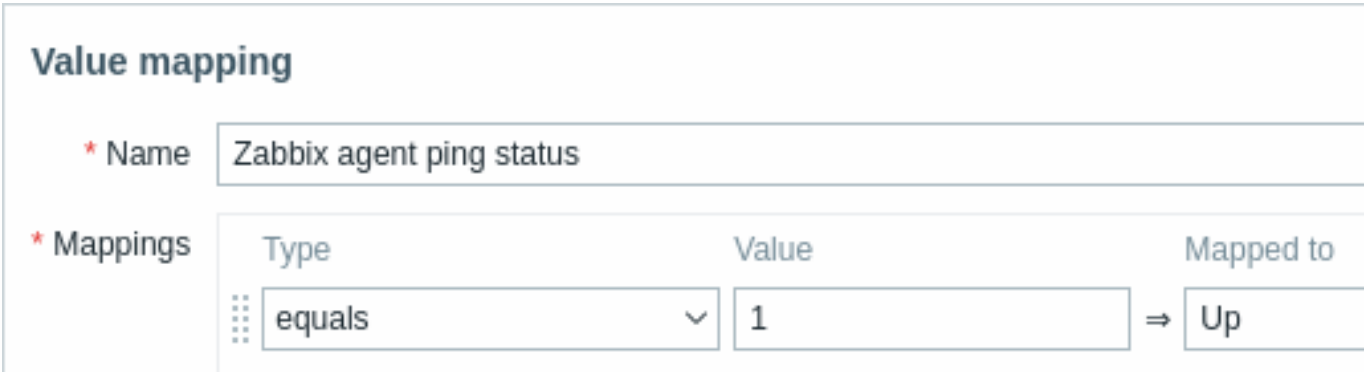
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Quando si visualizza la mappatura dei valori nell'elenco, sono visibili solo le prime tre mappature; tre punti indicano che sono presenti altre mappature.



Esempio di mappatura dei valori

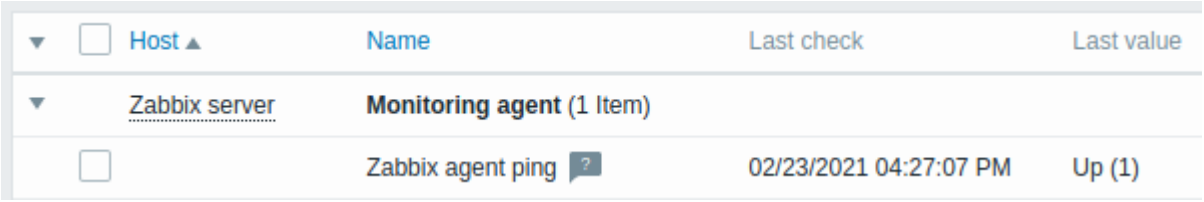
Uno degli item agent predefiniti, *Zabbix agent ping*, utilizza la mappatura dei valori a livello di template "Zabbix agent ping status" per visualizzare i relativi valori.



Nel modulo di configurazione dell'item, è possibile trovare un riferimento a questa mappatura dei valori nel campo *Value mapping*:



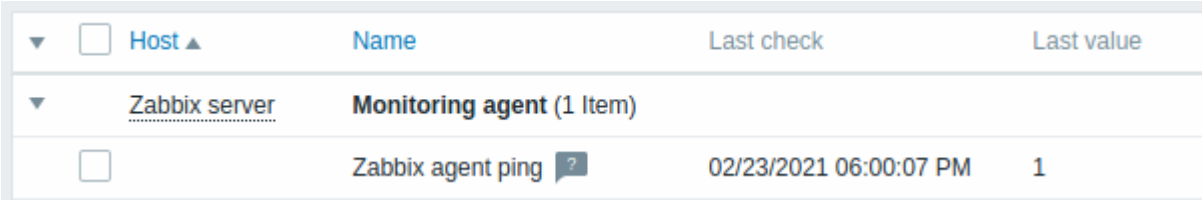
Questa mappatura viene utilizzata nella sezione *Monitoring* > *Latest data* per visualizzare "Up" (con il valore non elaborato tra parentesi).



Host	Name	Last check	Last value
Zabbix server	**Monitoring agent** (1 Item)		
	Zabbix agent ping	02/23/2021 04:27:07 PM	Up (1)

Note:
Nella sezione *Latest data*, i valori visualizzati vengono abbreviati a 20 caratteri. Se viene utilizzata una mappatura dei valori, questa abbreviazione non viene applicata al valore mappato, ma solo al valore non elaborato (visualizzato tra parentesi).

Senza una mappatura dei valori predefinita, verrebbe visualizzato solo ”1”, il cui significato potrebbe essere difficile da comprendere.



Host	Name	Last check	Last value
Zabbix server	**Monitoring agent** (1 Item)		
	Zabbix agent ping	02/23/2021 06:00:07 PM	1

Coda

Panoramica

La coda mostra gli item che sono in attesa di un aggiornamento. La coda è solo una rappresentazione **logica** dei dati. In Zabbix non esiste una coda IPC né alcun altro meccanismo di coda.

Anche gli item monitorati dai proxy sono inclusi nella coda: verranno conteggiati come in coda per il periodo di aggiornamento dei dati di cronologia del proxy.

Nella coda vengono visualizzati solo gli item con tempi di aggiornamento pianificati. Ciò significa che i seguenti tipi di item sono esclusi dalla coda:

- item log, logrt ed event log di Zabbix agent attivi
- item trap SNMP
- item trapper
- item di monitoraggio web
- item dipendenti

Le statistiche mostrate dalla coda sono un buon indicatore delle prestazioni di Zabbix server.

La coda viene recuperata direttamente da Zabbix server utilizzando il protocollo JSON. Le informazioni sono disponibili solo se Zabbix server è in esecuzione.

Attention:
Gli item non vengono conteggiati nella coda se l’interfaccia dell’item diventa non disponibile a causa di problemi di connessione o del mancato corretto funzionamento dell’agent.

Lettura della coda

Per leggere la coda, accedere a *Administration* > *Queue* > *Queue overview*.

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

L'immagine qui riportata è generalmente "ok", quindi possiamo presumere che il server funzioni correttamente.

La coda mostra alcuni item in attesa da un massimo di 30 secondi. Sarebbe utile sapere quali sono questi item.

A questo scopo, selezionare *Queue details* nel menu a discesa del titolo. Ora è possibile visualizzare un elenco degli item in ritardo.

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

Con questi dettagli potrebbe essere possibile scoprire perché questi item sono in ritardo.

Uno o due item in ritardo probabilmente non sono motivo di allarme. Potrebbero essere aggiornati entro un secondo. Tuttavia, se si nota che molti item rimangono in ritardo troppo a lungo, potrebbe esserci un problema più serio.

Vedere anche: Allineamento dei fusi orari quando si utilizzano intervalli di pianificazione.

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

item della coda

È possibile utilizzare uno speciale item interno `zabbix[queue,<from>,<to>]` per monitorare lo stato della coda in Zabbix. Restituirà il numero di item ritardati della quantità di tempo impostata. Per ulteriori informazioni, vedere Item interni.

Cache dei valori

Panoramica

Per rendere molto più veloce il calcolo delle espressioni dei trigger, dei calculated item e di alcune macro, il server Zabbix supporta un'opzione di cache dei valori.

Questa cache in memoria può essere utilizzata per accedere ai dati storici, invece di effettuare chiamate SQL dirette al database. Se i valori storici non sono presenti nella cache, i valori mancanti vengono richiesti al database e la cache viene aggiornata di conseguenza.

I valori degli item rimangono nella cache dei valori fino a quando:

- l'item non viene eliminato (i valori memorizzati nella cache vengono eliminati dopo la successiva sincronizzazione della configurazione);
- il valore dell'item non è al di fuori dell'intervallo di tempo o di conteggio specificato nell'espressione del trigger/item calcolato (il valore memorizzato nella cache viene rimosso quando viene ricevuto un nuovo valore);
- l'intervallo di tempo o di conteggio specificato nell'espressione del trigger/item calcolato viene modificato in modo tale che siano necessari meno dati per il calcolo (i valori memorizzati nella cache non necessari vengono rimossi dopo 24 ore).

Note:
Lo stato della cache dei valori può essere osservato utilizzando l'opzione di runtime control del server `diaginfo` (o `diaginfo=valuecache`) e ispezionando la sezione relativa alle informazioni diagnostiche della cache dei valori. Questo può essere utile per individuare trigger o item calcolati configurati in modo errato.

Per abilitare la funzionalità della cache dei valori, è possibile specificare il parametro facoltativo `ValueCacheSize` nella configurazione del server Zabbix.

Sono supportati due item interni per il monitoraggio della cache dei valori: `zabbix[vcache,buffer,<mode>]` e `zabbix[vcache,cache,<`
Per ulteriori dettagli, vedere gli item interni.

Esegui ora

Panoramica

L'opzione *Esegui ora* consente di eseguire immediatamente i controlli passivi.

La raccolta dei valori degli item in Zabbix segue gli intervalli di aggiornamento configurati. Alcuni item (ad esempio, le regole di low-level discovery) hanno intervalli lunghi e potrebbe essere necessario recuperare immediatamente un nuovo valore, ad esempio per rilevare modifiche nelle risorse individuabili.

L'opzione *Esegui ora* è supportata per i seguenti tipi di item:

- Zabbix agent (passivo)
- Controllo semplice
- Agent SNMP (v1/v2/v3)
- Interno Zabbix
- Controllo esterno
- Monitor database
- Agent HTTP
- Agent IPMI
- Agent SSH
- Agent TELNET
- Agent JMX
- Calcolato
- Dependent item
- Script
- Browser

Se l'item è di tipo "Dependent item", anche il suo item master deve essere uno dei tipi sopra elencati.

Configurazione

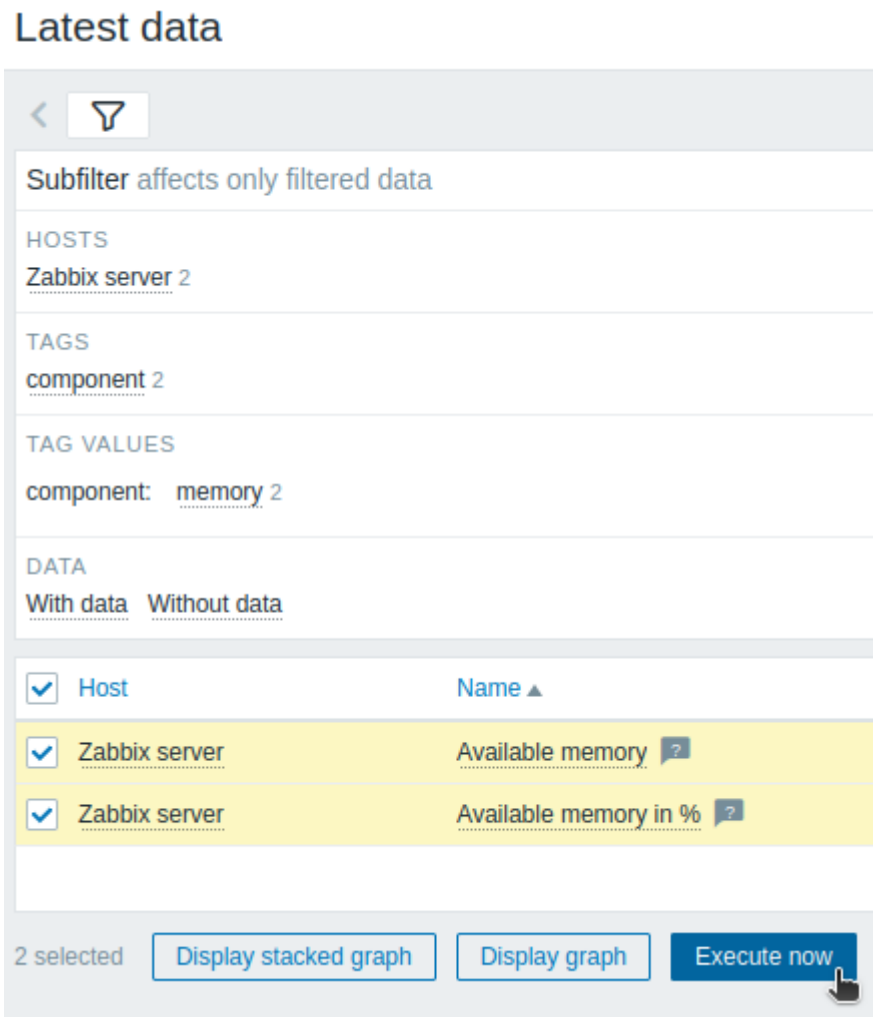
Puoi eseguire i controlli passivi da:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items* oppure *Discovery*
- Il menu item
- Durante la modifica di item o regole di discovery a basso livello (nei rispettivi moduli di configurazione)

Attention:
Il controllo deve esistere nella cache di configurazione per poter essere eseguito; vedi anche CacheUpdateFrequency. La cache **non viene aggiornata** prima dell'esecuzione del controllo, quindi le modifiche recenti alla configurazione di item o delle regole di discovery a basso livello non verranno rilevate. Per testare item o regole LLD che sono in fase di creazione o appena creati, puoi invece usare l'opzione Test.

Ad esempio, per eseguire un controllo passivo in *Monitoring* > *Latest data*:

1. Seleziona gli item; puoi eseguire il controllo per più item contemporaneamente.
2. Fai clic su *Execute now*.



Note:
In *Latest data*, gli utenti possono *Execute now* sugli item degli host per i quali dispongono dell'autorizzazione read-write, oppure dell'autorizzazione *read* con l'azione *Invoke "Execute now" on read-only hosts* action abilitata per il proprio ruolo.

Limitazione dei controlli agent

Panoramica

Puoi controllare quali chiavi item Zabbix agent o agent 2 è autorizzato o negato a usare durante l'esecuzione di controlli item, comandi remoti o script.

Per farlo, usa questi parametri di configurazione dell'agent per definire regole di autorizzazione/negazione:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`
- `AllowKeyRegexp=<pattern>`
- `DenyKeyRegexp=<pattern>`

Il `<pattern>` deve contenere una singola chiave item.
 Per `AllowKey` e `DenyKey`, il `<pattern>` supporta i caratteri jolly (`*`). Il carattere jolly corrisponde a qualsiasi numero di caratteri nella sua posizione e può essere usato per corrispondere a chiavi item

o parametri (ad esempio, vfs.file.*[*]).
 Per AllowKeyRegexp e DenyKeyRegexp, il <pattern> supporta espressioni regolari. Il pattern dell'espressione regolare può essere usato per corrispondere a chiavi item e alle relative stringhe di parametri. Vedi anche esempi di escaping corretto.
 Tieni presente che Zabbix agent e Zabbix agent 2 possono supportare una sintassi di espressioni regolari diversa.

Attention:
Per migliorare la sicurezza, si consiglia di usare chiavi item esatte invece dei caratteri jolly per AllowKey e DenyKey. Si consiglia inoltre di usare i pattern di espressione regolare più specifici possibili per AllowKeyRegexp e DenyKeyRegexp. Per i dettagli, vedi Protezione delle regole di autorizzazione/negazione.

A differenza degli altri parametri di configurazione dell'agent, puoi specificare un numero illimitato di parametri AllowKey, DenyKey, AllowKeyRegexp o DenyKeyRegexp.

Note importanti

- Tutti gli item system.run sono disabilitati per impostazione predefinita (anche quando DenyKey e DenyKeyRegexp sono vuoti), come se fosse impostato DenyKey=system.run[*] oppure DenyKeyRegexp=^system\.run\[.*\]$ come ultima regola. Per questo motivo, è possibile consentire specifici item system.run senza negare esplicitamente gli altri item system.run.
- Se possibile, usa AllowKey/AllowKeyRegexp per consentire solo gli item necessari e negare tutto il resto. Alcune chiavi possono essere sfruttate per leggere file non previsti tramite path traversal (ad esempio, vfs.file.contents["../../../../etc/p
e le nuove versioni di Zabbix agent possono introdurre chiavi non coperte dalle regole DenyKey/DenyKeyRegexp.
- La configurazione AllowKey, DenyKey, AllowKeyRegexp e DenyKeyRegexp non influisce sui parametri agent HostnameItem, HostMetadataItem o HostInterfaceItem.
- Gli item negati diventano non supportati senza alcun suggerimento o messaggio di errore; ad esempio:
  - Il parametro da riga di comando di Zabbix agent --print (-p) non mostrerà le chiavi degli item negati.
  - Il parametro da riga di comando di Zabbix agent --test (-t) restituirà "Unsupported item key." per le chiavi degli item negati.
  - Quando la registrazione è attivata (LogRemoteCommands=1), il file di log di Zabbix agent non registrerà i comandi remoti negati.

Ordine delle regole allow/deny

È possibile specificare un numero illimitato di regole AllowKey, DenyKey, AllowKeyRegexp o DenyKeyRegexp, anche se il loro ordine è importante.

- Le regole vengono valutate una per una, dall'alto verso il basso.
- Quando una chiave di item corrisponde a una regola, viene consentita o negata e la valutazione delle regole si interrompe.

Ad esempio, durante la valutazione di vfs.file.contents[/etc/passwd], le regole vengono elaborate come segue:

```
AllowKey=vfs.file.contents[/tmp/app.log]    # Il pattern della chiave dell'item non corrisponde; l'agent p
AllowKey=vfs.file.contents[/etc/passwd]     # Il pattern della chiave dell'item corrisponde; l'agent conse
DenyKey=vfs.file.*[*]                       # L'agent ignora la regola, poiché la valutazione si è già int
```

La seguente sequenza di regole negherà il controllo dell'item:

```
DenyKey=vfs.file.*[*]                       # Il pattern della chiave dell'item corrisponde; l'agent nega
AllowKey=vfs.file.contents[/etc/passwd]     # L'agent ignora la regola, poiché la valutazione si è già int
AllowKey=vfs.file.contents[/tmp/app.log]    # L'agent ignora la regola, poiché la valutazione si è già int
```

Esempi

I seguenti esempi mostrano i modelli di configurazione comuni per AllowKey, DenyKey, AllowKeyRegexp e DenyKeyRegexp.

Consentire controlli e comandi specifici

Consentire solo due controlli item vfs.file e due comandi system.run:

usando AllowKey e DenyKey:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

usando AllowKeyRegexp e DenyKeyRegexp:

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
Impostare DenyKey=system.run[*] o DenyKeyRegexp=^system\.run\[.*\]$ non è necessario, perché tutti gli altri comandi system.run sono negati per impostazione predefinita.

Consentire gli script

Consenti all'agent Zabbix di eseguire script sugli host tramite tutti i metodi disponibili:

- Script globali che possono essere eseguiti nel frontend o tramite API (questo metodo usa sempre la chiave system.run[myscript.sh]
- Comandi remoti dalle operazioni delle azioni (questo metodo usa sempre la chiave system.run[myscript.sh,nowait])
- Elementi system.run di Zabbix agent con lo script, ad esempio:
  - system.run[myscript.sh]
  - system.run[myscript.sh,wait]
  - system.run[myscript.sh,nowait]

```
AllowKey=system.run[myscript.sh,*]
```

oppure

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

Per controllare il parametro wait/nowait, devi impostare una regola diversa. Ad esempio, puoi consentire solo gli elementi system.run[myscript.sh,wait], escludendo così gli altri metodi:

```
AllowKey=system.run[myscript.sh,wait]
```

oppure

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Protezione delle regole allow/deny

Questi esempi mostrano come proteggere regole AllowKey/AllowKeyRegexp o DenyKey/DenyKeyRegexp troppo permissive.

Considera le seguenti regole:

using AllowKey and DenyKey:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

using AllowKeyRegexp and DenyKeyRegexp:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
Su Windows, devi eseguire l'escape degli spazi nel percorso usando un caret (^) per AllowKey o DenyKey e \s per AllowKeyRegexp o DenyKeyRegexp.

Queste regole contengono un wildcard (*) (o .* per i pattern basati su espressioni regolari), che può essere usato in modo improprio:

- Lo script test.bat può essere eseguito con qualsiasi argomento, inclusi quelli non intenzionali.
- Il pattern vfs.file.* (^vfs\.file\..*$) corrisponde a chiavi item sia con sia senza parametri; tuttavia, tutti gli item vfs.file richiedono parametri.
- Il pattern system.cpu.load[*] (^system\.cpu\.load\[.*\]$) corrisponde solo a chiavi item con parametri; tuttavia gli item system.cpu.load non richiedono parametri.

Per proteggere queste regole, consenti esplicitamente l'esecuzione di test.bat solo con argomenti specifici e nega i pattern corretti delle chiavi item; ad esempio:

using AllowKey and DenyKey:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

using AllowKeyRegexp and DenyKeyRegexp:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^system\.cpu\.load(\[.*\])?$
```

Puoi testare le regole eseguendo i seguenti comandi, che restituiranno ZBX_NOTSUPPORTED.

```
cd "C:\Program Files\Zabbix Agent 2"
zabbix_agent2.exe -t system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat debug"]
zabbix_agent2.exe -t vfs.file.size["C:\ProgramData\MyApp\config.ini"]
zabbix_agent2.exe -t vfs.file.contents["C:\Windows\System32\drivers\etc\hosts"]
zabbix_agent2.exe -t system.cpu.load
zabbix_agent2.exe -t system.cpu.load[all,avg1]
```

Esempi di pattern

Le tabelle seguenti mostrano come vengono confrontati i pattern delle chiavi item:

- Una chiave corrisponde al pattern solo se soddisfa **tutte** le condizioni nella colonna *Matches*.
- I parametri devono essere racchiusi completamente tra parentesi quadre (ad esempio, vfs.file.contents[* e vfs.file.contents*utf8] sono pattern non validi).
- Gli esempi illustrano solo la corrispondenza dei pattern; gli item effettivi richiedono parametri.

Per AllowKey e DenyKey:

Pattern	Matches	Examples
*	Qualsiasi chiave con o senza parametri	
vfs.file.*	La chiave inizia con vfs.file. Con o senza parametri	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	La chiave inizia con vfs. La chiave termina con .contents Nessun parametro	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
vfs.file.*[*]	La chiave inizia con vfs.file. Parametri qualsiasi o vuoti	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
vfs.file.contents	La chiave è vfs.file.contents Nessun parametro	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	La chiave è vfs.file.contents[] Parametri vuoti	Matches: vfs.file.contents[] Does not match: vfs.file.contents

Pattern	Matches	Examples
vfs.file.contents[*]	La chiave è vfs.file.contents Parametri qualsiasi o vuoti	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents
vfs.file.contents[/etc/passwd,*]	La chiave è vfs.file.contents Il primo parametro è /etc/passwd Il secondo parametro è qualsiasi o vuoto	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	La chiave è vfs.file.contents Parametri qualsiasi, almeno uno contenente passwd	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	La chiave è vfs.file.contents Il primo parametro contiene passwd Il secondo parametro è qualsiasi o vuoto	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	La chiave è vfs.file.contents Il primo parametro è /etc/passwd Il secondo parametro è utf8	Matches: vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Per AllowKeyRegexp e DenyKeyRegexp:

Pattern	Matches	Examples
^.*$	Qualsiasi chiave con o senza parametri	
^vfs\.file\..*$	La chiave inizia con vfs.file. Con o senza parametri	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
^vfs\..*\.contents$	La chiave inizia con vfs. La chiave termina con .contents Nessun parametro	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	La chiave inizia con vfs.file. Parametri qualsiasi o vuoti	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
^vfs\.file\.contents$	La chiave è vfs.file.contents Nessun parametro	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]

Pattern	Matches	Examples

`^vfs\.file\.contents\[\]$`	La chiave è `vfs.file.contents[]`
	Parametri vuoti
		Matches:
		`vfs.file.contents[]`
		
		Does not match:
		`vfs.file.contents`

`^vfs\.file\.contents\[.*\]$`	La chiave è `vfs.file.contents`
	Parametri qualsiasi o vuoti
		Matches:
		`vfs.file.contents[/path/to/file]`
		
		Does not match:
		`vfs.file.contents`

`^vfs\.file\.contents\[/etc/passwd,.*\]$`	La chiave è `vfs.file.contents`
	Il primo parametro è `/etc/passwd`
	Il secondo parametro è qualsiasi o vuoto
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`^vfs\.file\.contents\[.*passwd.*\]$`	La chiave è `vfs.file.contents`
	Parametri qualsiasi, almeno uno contenente `passwd`
		Matches:
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`

`^vfs\.file\.contents\[.*passwd.*,.*\]$`	La chiave è `vfs.file.contents`
	Il primo parametro contiene `passwd`
	Il secondo parametro è qualsiasi o vuoto
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/tmp/test]`

`^vfs\.file\.contents\[/etc/passwd,(utf8|windows-1252)\]$`	La chiave è `vfs.file.contents`
	Il primo parametro è `/etc/passwd`
	Il secondo parametro è `utf8` o `windows-1252`
		Matches:
		`vfs.file.contents[/etc/passwd,utf8]`
		`vfs.file.contents[/etc/passwd,windows-1252]`
		
		Does not match:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

Rilevamento dei problemi con i trigger

Il rilevamento dei problemi in Zabbix si basa sui **trigger**.

Un trigger descrive la condizione del problema, consentendo a Zabbix di reagire quando il problema si verifica.

Il flusso completo del rilevamento dei problemi è:

1. Assicurati di avere un item che raccolga i dati.
2. Definisci il trigger per questo item. Senza un trigger, Zabbix si limita a raccogliere dati, ma non reagisce mai a essi.
3. Osserva i problemi rilevati:
    - In *Monitoring* > *Problems*
    - Nel widget *Problems*
4. Ricevi notifiche, esegui comandi remoti (se definiti).

Nota che i template ufficiali di Zabbix contengono trigger predefiniti.
Consulta la guida per applicare il template per monitor Linux.

Che cos'è un "trigger"?

Un trigger descrive la *condizione di problema*, tramite l'espressione del trigger.

La condizione di problema, quando si verifica nella realtà, corrisponde al momento in cui il trigger "scatta" (in Zabbix viene creato un problema).

La condizione di problema può basarsi sull'ultimo valore, sul valore medio, su una stringa rilevata e su molte altre funzioni. I trigger non possono essere creati per item che restituiscono valori con tipo di dato binario o JSON.

Una semplice espressione di trigger calcola la funzione (ad esempio, `max()` per il valore massimo) per i dati specificati tra parentesi (in genere item di dati e finestra temporale), quindi confronta questo calcolo con una soglia.

Per esempio:

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Questo trigger scatta se lo spazio su disco libero, misurato nell'arco di 5 minuti, è rimasto costantemente sotto i 10 GB.

Un trigger può avere i seguenti stati:

Stato del trigger	Descrizione	In Zabbix
OK	L'espressione del trigger è stata valutata come "0" (o FALSE)	Nessun problema creato Il problema è risolto (se esisteva)
Problem	L'espressione del trigger è stata valutata come "1" (o TRUE)	Viene creato un problema

Talvolta la valutazione dell'espressione del trigger non è possibile per qualche motivo. Vedere Stato sconosciuto dell'espressione.

Altre risorse:

- Esempi di trigger
- Configurazione di un trigger
- Best practice

Frequenza di calcolo

Un trigger viene ricalcolato ogni volta che il server Zabbix riceve un **nuovo valore** che fa parte dell'espressione. Quando viene ricevuto un nuovo valore, ogni funzione inclusa nell'espressione viene ricalcolata (non solo quella che ha ricevuto il nuovo valore).

Inoltre, un trigger viene ricalcolato ogni 30 secondi (dal history syncer) se l'espressione contiene uno dei seguenti elementi:

- funzioni di data e ora
- funzione nodata()

I trigger che contengono solo funzioni di trend vengono valutati una volta per il più piccolo intervallo di tempo presente nell'espressione. Mentre molte funzioni dei trigger utilizzano i dati di history dei valori degli item, le funzioni di trend utilizzano i dati di trend.

Esempi di trigger

Questa pagina raccoglie esempi di trigger.

Gli esempi sono ordinati per caso d'uso:

- Host non disponibile
- proxy non disponibile
- Cluster SMTP non disponibile
- Host non raggiungibile
- Riavvio imprevisto
- Modifiche in file importanti
- Modifiche nella query DNS
- Software non corrispondente su host diversi
- Orologi non sincronizzati
- agent obsoleto
- Interfaccia instabile
- Traffico in ingresso elevato
- Spazio su disco insufficiente
- Spazio su disco insufficiente (soglia dinamica)
- Carico CPU elevato
- Confronto del carico CPU
- Confronto del carico CPU a lungo termine
- Stringa contenente caratteri speciali

Rilevare host non disponibile

```
max(/host/zabbix[host,agent,available],5m)=0
```

Questo trigger si attiva se lo Zabbix agent sul host non è stato disponibile per 5 minuti.
 Funzione: max
 Item di dati: zabbix[host,agent,available]

Alternativa:

```
nodata(/host/agent.ping,5m)=1
```

Questo trigger si attiva se non sono stati ricevuti dati dallo Zabbix agent per 5 minuti.
 Funzione: nodata
 Item di dati: agent.ping

Rilevare proxy non disponibile

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

Questo trigger si attiva se i dati del proxy Zabbix sono in ritardo rispetto all'ora del server Zabbix di 1 minuto.
 Funzione: fuzzytime
 Item di dati: zabbix[proxy,{$PROXY_NAME},lastaccess]

Rilevare un cluster SMTP non disponibile

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

Questo trigger si attiva se entrambi i server SMTP non sono disponibili.
 Funzione: last
 Item di dati: net.tcp.service

Rilevare host non raggiungibile

```
count(/host/icmpping,30m,,"0")>5
```

Questo trigger si attiva se l'host non è raggiungibile tramite ping per più di 5 volte negli ultimi 30 minuti.
 Funzione: count
 Item di dati: icmpping

Rilevare un riavvio imprevisto

```
change(/host/system.uptime)<0
```

Questo trigger si attiva se viene rilevata una variazione negativa nel valore del tempo di attività del sistema (indicando un riavvio).
 Funzione: change
 Item di dati: system.uptime

Rilevare modifiche in file importanti

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

Questo trigger si attiva se /etc/passwd è stato modificato. L'espressione è vera quando il checksum precedente di /etc/passwd differisce da quello più recente. Espressioni simili possono essere utili per monitorare le modifiche in file importanti, come /etc/passwd, /etc/inetd.conf, /kernel, ecc.
 Funzione: last
 item di dati: vfs.file.cksum

Rileva modifiche nella query DNS

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Nota le virgolette attorno al secondo operando.

Questo trigger si attiva se il risultato della query non è uguale a quello che restituisce normalmente:

```
example.com             MX       0 mail.example.com
```

Funzione: last
 Data item: net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], con macro definite come segue:

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

Rilevare software non corrispondente su host diversi

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

Questo trigger si attiva se la versione di Ubuntu è diversa su host differenti. Si noti che qui gli operandi sono funzioni che restituiscono stringhe.
 Funzione: last
 Item di dati: vfs.file.contents

Rileva orologi non sincronizzati

```
fuzzytime(/host/system.localtime,10s)=0
```

Il trigger si attiva se l'ora locale del client e l'ora del server Zabbix differiscono di più di 10 secondi.
 Funzione: fuzzytime
 Elemento dati: system.localtime

Nota che system.localtime deve essere configurato come controllo passivo per Zabbix agent; su Zabbix agent 2 può essere configurato come controllo attivo.

Rilevare agent obsoleto

```
find(/host/agent.version,,"like","beta")=1
```

Questo trigger si attiva se Zabbix agent ha una versione beta. Zabbix agent deve essere aggiornato.
 Funzione: find
 Item di dati: agent.version

Rilevare un'interfaccia instabile

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

Questo trigger si attiva se lo stato operativo (up/down/unknown) di eth0 è cambiato più di 5 volte in un'ora.
 Funzione: changecount
 Item di dati: vfs.file.contents

Rileva traffico in ingresso elevato

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

Questo trigger si attiva se il numero di byte ricevuti su eth0 negli ultimi cinque minuti è stato sempre superiore a 100 kilobyte. Probabilmente qualcuno sta scaricando un file di grandi dimensioni.
 Funzione: min
 Item di dati: net.if.in[eth0,bytes]

Rilevare spazio su disco insufficiente

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Il trigger si attiva se lo spazio libero su disco è costantemente (per 5 minuti) inferiore a 10 GB.

È anche possibile definire un'espressione di ripristino:

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

Il problema viene **risolto** solo quando lo spazio libero su disco è costantemente (per 10 minuti) superiore a 40 GB.
 Funzione: min
 Item di dati: vfs.fs.size

Rilevare spazio su disco insufficiente (soglia dinamica)

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

Il trigger si attiva se lo spazio di archiviazione libero (in unità di allocazione) scende al di sotto del 10 percento. Si noti il valore di un altro item utilizzato per ottenere una soglia del trigger adattiva, applicabile allo spazio di archiviazione individuato di varie dimensioni. Funzione: last

Rilevare un carico elevato della CPU

```
last(/host/system.cpu.load[all,avg1])>5
```

Il trigger si attiva quando il carico medio del processore è stato superiore a 5 per un minuto.

Varianti:

```
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
```

Tali trigger analizzano 5 minuti di dati e si attivano solo se il carico della CPU non scende mai sotto 2. Inoltre, questi trigger si attivano:

- solo durante le ore notturne (00:00-06:00)
- in qualsiasi momento tranne che per 2 ore nel cambio di settimana (domenica, 23:00 - lunedì, 01:00)

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

Questo trigger si attiva se il carico del processore è troppo elevato su almeno due dei tre host.
 Funzioni: last, min, dayofweek, time
 Item di dati: system.cpu.load

Confronta i carichi della CPU

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

Il trigger si attiva se il carico medio di oggi supera di oltre due volte il carico medio della stessa ora di ieri (utilizzando lo spostamento temporale `now-1d`).
 Funzione: `avg`
 Item di dati: `system.cpu.load`

Confrontare i carichi CPU a lungo termine

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

Questo trigger si attiva se il carico della CPU sul host è aumentato di oltre il 10% nell'ultimo mese
 Funzione: `trendavg`
 Item di dati: `system.cpu.load`

È inoltre possibile utilizzare il campo Nome evento nella configurazione del trigger per creare un messaggio di avviso significativo, ad esempio per ricevere qualcosa come

```
"Il carico del server Exchange è aumentato del 24% in luglio (0.69) rispetto a giugno (0.56)"
```

il nome dell'evento deve essere definito come:

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

È inoltre utile consentire la chiusura manuale nella configurazione del trigger per questo tipo di problema.

Rilevare una stringa contenente caratteri speciali

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

Il trigger si attiva se il contenuto di `/tmp/hello` è uguale alla stringa definita in `{$HELLO_MACRO}`:

```
{$HELLO_MACRO} = \" //hello ?\"
```

In alternativa, è possibile confrontarlo direttamente con la stringa:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Si noti come i caratteri speciali (\\ e ") vengano sottoposti a escape quando la stringa viene confrontata direttamente.

Funzione: `last`
 Item di dati: `vfs.file.contents`

Configurazione di un trigger

Questa pagina descrive come configurare un trigger nel frontend di Zabbix.

Si noti che i template ufficiali di Zabbix contengono trigger predefiniti.
Vedere la guida per applicare il template a monitor Linux.
È inoltre possibile aggiornare in massa i trigger configurati.

Altre risorse:

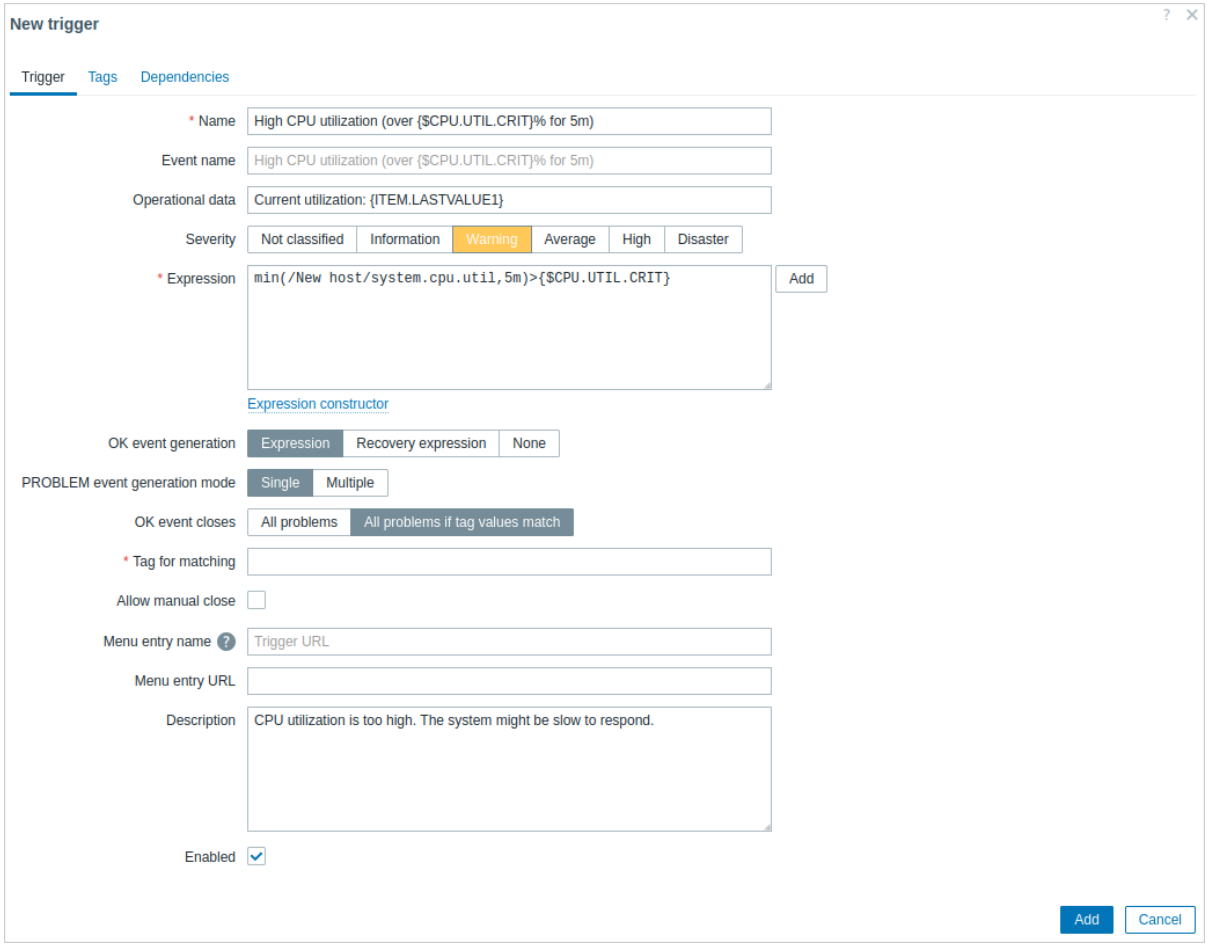
- Rilevamento dei problemi con i trigger (introduzione generale)
- Esempi di trigger (per caso d'uso)
- Espressione e funzioni del trigger (dettagli sulla sintassi)
- Best practice

Configurazione

Per configurare un trigger, procedi come segue:

1. Vai a *Data collection* > *Hosts*.
2. Fai clic su *Triggers* nella riga dell'host.
3. Fai clic su *Create trigger* a destra (oppure sul nome del trigger per modificare un trigger esistente).
4. Inserisci i parametri del trigger nel modulo.

La scheda **Trigger** contiene tutti gli attributi essenziali del trigger.

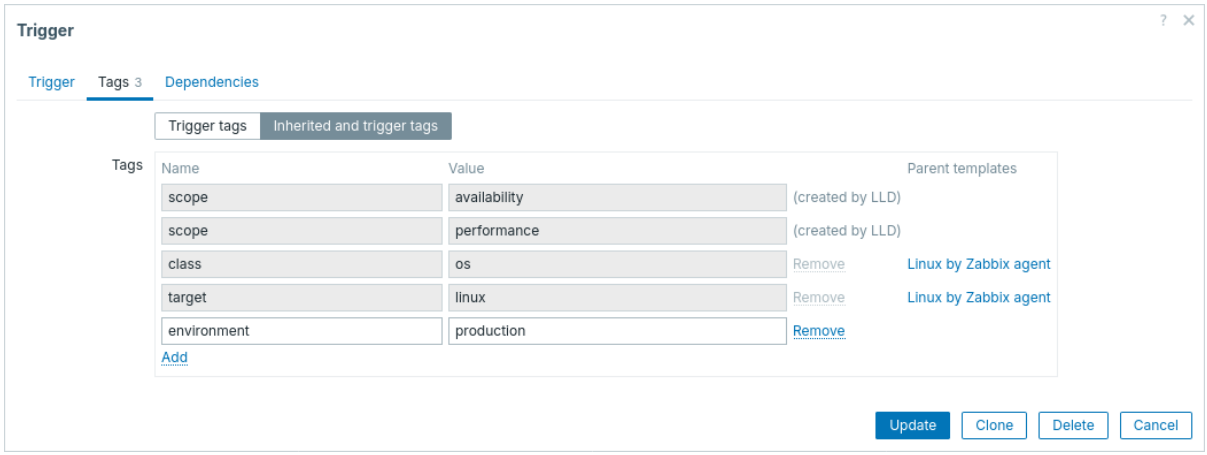


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parameter	Description
Name	Nome del trigger. I macro supportati sono: {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.*} e i macro utente {$MACRO}. I macro **$1, $2...$9** possono essere usati per riferirsi alla prima, seconda...nona costante dell'espressione. *Nota*: i macro $1-$9 verranno risolti correttamente se fanno riferimento a costanti in espressioni relativamente semplici e dirette. Ad esempio, il nome "Processor load above $1 on {HOST.NAME}" cambierà automaticamente in "Processor load above 5 on New host" se l'espressione è last(/New host/system.cpu.load[percpu,avg1])>5.
Event name	Quando definito, questo nome verrà usato per creare il nome dell'evento di problema. Per impostazione predefinita, il nome dell'evento è uguale al nome del trigger. Il nome dell'evento può essere usato per creare avvisi significativi contenenti i dati del problema (vedi esempio). È supportato lo stesso insieme di macro del nome del trigger, più i macro di espressione {TIME}, {TIMESTAMP} e {?EXPRESSION}.

Parameter	Description
Operational data	Inserisci una stringa con macro per visualizzare dati dinamici in tempo reale in *Monitoring* > *Problems*. In alternativa, lascia vuoto questo campo per visualizzare gli ultimi valori di tutti gli item dell'espressione del trigger. È supportato lo stesso insieme di macro del nome del trigger, con la possibilità di risoluzione dinamica. Ad esempio: {ITEM.VALUE<1-9>} si risolve nei valori dell'item nel momento in cui lo stato del trigger cambia (problema creato, risolto, chiuso manualmente o chiuso per correlazione). {ITEM.LASTVALUE<1-9>} si risolve negli ultimi valori dell'item. Nota che la chiusura manuale di un problema non produce un nuovo valore, quindi entrambi i macro continueranno a mostrare il valore del momento del problema. Nota anche che entrambi i macro si risolvono in *UNKNOWN* se l'ultimo valore è più vecchio di *Max history display period* (vedi *Administration > General*).
Severity	Imposta la severity richiesta del trigger facendo clic sui pulsanti.
Expression	Espressione logica usata per definire le condizioni di un problema. Sono supportati i suffissi temporali e i suffissi per la dimensione della memoria. Un problema viene creato dopo che tutte le condizioni incluse nell'espressione sono soddisfatte, cioè quando l'espressione viene valutata come TRUE. Il problema verrà risolto non appena l'espressione viene valutata come FALSE, a meno che non siano specificate condizioni di ripristino aggiuntive in *Recovery expression*.
OK event generation	Opzioni di generazione degli eventi OK: **Expression** - gli eventi OK vengono generati in base alla stessa espressione degli eventi di problema; **Recovery expression** - gli eventi OK vengono generati se l'espressione del problema viene valutata come FALSE e l'espressione di ripristino viene valutata come TRUE; **None** - in questo caso il trigger non tornerà mai autonomamente allo stato OK.
Recovery expression	Espressione logica (facoltativa) che definisce condizioni aggiuntive che devono essere soddisfatte prima che il problema venga risolto, dopo che l'espressione del problema originale è già stata valutata come FALSE. L'espressione di ripristino è utile per l'isteresi del trigger. Non è **possibile** risolvere un problema solo con l'espressione di ripristino se l'espressione del problema è ancora TRUE. Questo campo è disponibile solo se per *OK event generation* è selezionato 'Recovery expression'.
PROBLEM event generation mode	Modalità di generazione degli eventi di problema: **Single** - viene generato un singolo evento quando un trigger entra per la prima volta nello stato 'Problem'; **Multiple** - viene generato un evento a ogni valutazione 'Problem' del trigger.
OK event closes	Seleziona se l'evento OK chiude: **All problems** - tutti i problemi di questo trigger; **All problems if tag values match** - solo i problemi del trigger con valori dei tag evento corrispondenti.
Tag for matching	Inserisci il nome del tag evento da usare per la correlazione degli eventi. Questo campo viene visualizzato se è selezionato 'All problems if tag values match' per la proprietà *OK event closes* ed è obbligatorio in questo caso.
Allow manual close	Seleziona per consentire la chiusura manuale degli eventi di problema generati da questo trigger. La chiusura manuale è possibile quando si riconoscono gli eventi di problema.
Menu entry name	Se non è vuoto, il nome inserito qui (fino a 64 caratteri) viene usato in diverse posizioni del frontend come etichetta per l'URL del trigger specificato nel parametro *Menu entry URL*. Se è vuoto, viene usato il nome predefinito *Trigger URL*. È supportato lo stesso insieme di macro del nome del trigger, più {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}.
Menu entry URL	Se non è vuoto, l'URL inserito qui (fino a 2048 caratteri) è disponibile come collegamento nel menu eventi in diverse posizioni del frontend, ad esempio facendo clic sul nome del problema in *Monitoring* > *Problems* o nel widget dashboard *Problems*. È supportato lo stesso insieme di macro del nome del trigger, più {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}. Nota: i macro utente con valori segreti non verranno risolti nell'URL.
Description	Campo di testo usato per fornire ulteriori informazioni su questo trigger. Può contenere istruzioni per risolvere un problema specifico, i dettagli di contatto del personale responsabile, ecc. È supportato lo stesso insieme di macro del nome del trigger.
Enabled	Deselezionando questa casella, il trigger verrà disabilitato se necessario. I problemi di un trigger disabilitato non vengono più visualizzati nel frontend, ma non vengono eliminati.

La scheda **Tags** consente di definire i tag a livello di trigger. Tutti i problemi di questo trigger saranno contrassegnati con i valori inseriti qui.



Inoltre, l'opzione *Inherited and trigger tags* consente di visualizzare i tag definiti a livello di template se il trigger proviene da quel template. Se sono presenti più template con lo stesso tag, questi tag vengono visualizzati una sola volta e i nomi dei template sono separati da virgole. Un trigger non "eredita" né visualizza i tag a livello di host.

Parameter	Description
Name/Value	Imposta tag personalizzati per contrassegnare gli eventi del trigger. I tag sono una coppia composta da nome e valore del tag. Puoi usare solo il nome oppure associarlo a un valore. Un trigger può avere più tag con lo stesso nome, ma valori diversi. Sono supportati macro utente, macro utente con contesto, macro di discovery a basso livello e funzioni di macro functions con `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` nei tag evento. Le macro di discovery a basso livello possono essere usate all'interno del contesto della macro. La macro {TRIGGER.ID} è supportata nei valori dei tag del trigger. Può essere utile per identificare i trigger creati da prototipi di trigger e, ad esempio, per sopprimere i problemi di questi trigger durante la manutenzione. Se la lunghezza totale del valore espanso supera 255, verrà troncata a 255 caratteri. Vedi tutte le macro supportate per i tag evento. I tag evento possono essere usati per la correlazione degli eventi, nelle condizioni delle azioni e saranno visibili anche in *Monitoring* > *Problems* o nel widget *Problems*.

La scheda **Dependencies** contiene tutte le dipendenze del trigger.

Fai clic su *Add* per aggiungere una nuova dipendenza.

> **Note:**
> Puoi anche configurare un trigger aprendo uno esistente, facendo clic sul pulsante *Clone* e salvando poi con un nome diverso.

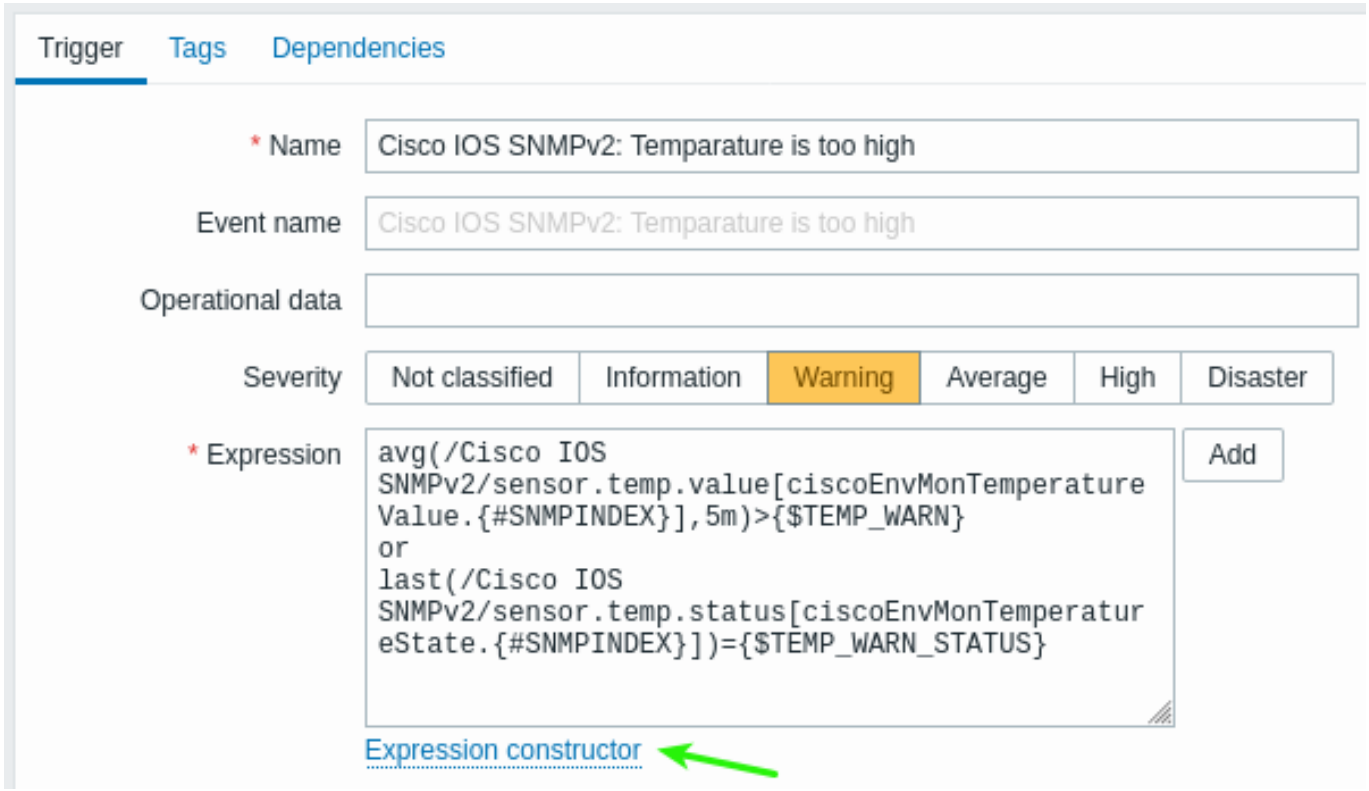
Test dell'espressione del trigger

È possibile testare quale sarebbe il risultato dell'espressione in base al valore ricevuto.

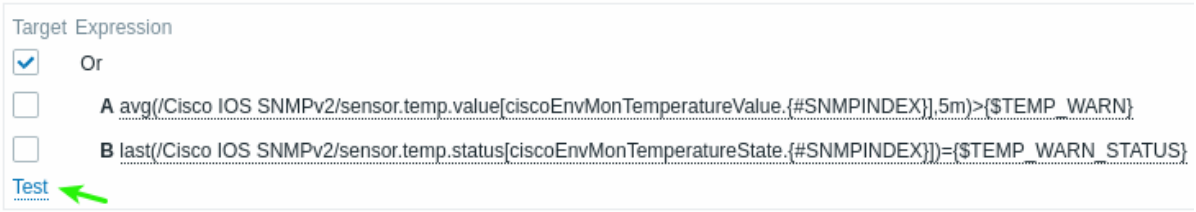
La seguente espressione, tratta da un template ufficiale, viene usata come esempio:

```
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}
```

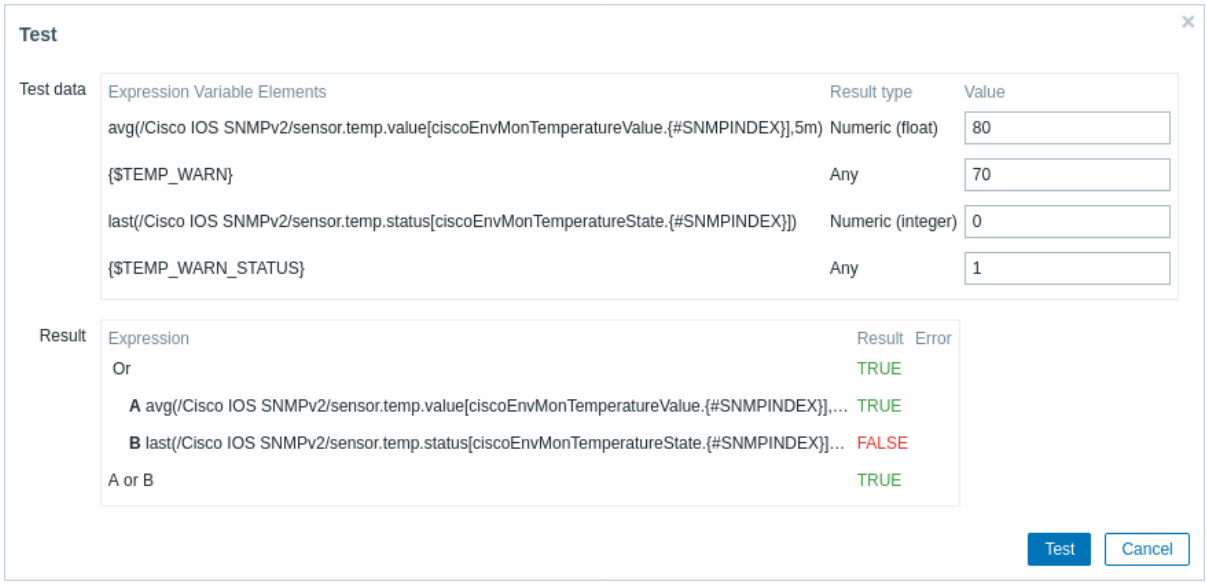
Per testare l'espressione, fare clic su *Expression constructor* sotto il campo dell'espressione.



Nell'Expression constructor, tutte le singole espressioni sono elencate. Per aprire la finestra di test, fare clic su *Test* sotto l'elenco delle espressioni.



Nella finestra di test è possibile inserire valori di esempio ('80', '70', '0', '1' in questo esempio) e quindi vedere il risultato dell'espressione, facendo clic sul pulsante *Test*.



È possibile vedere il risultato delle singole espressioni, così come dell'intera espressione.

"TRUE" significa che l'espressione specificata è corretta. In questo caso A, "80" è maggiore del valore specificato `{$TEMP_WARN}`, "70" in questo esempio. Come previsto, viene visualizzato un risultato "TRUE".

"FALSE" significa che l'espressione specificata non è corretta. In questo caso B, `{$TEMP_WARN_STATUS}` "1" deve essere uguale al valore specificato, "0" in questo esempio. Come previsto, viene visualizzato un risultato "FALSE".

Il tipo di espressione scelto è ”OR”. Se almeno una delle condizioni specificate (A o B in questo caso) è TRUE, anche il risultato complessivo sarà TRUE. Ciò significa che il valore corrente supera il valore di avviso e si è verificato un problema.

Espressione del trigger e funzioni

Panoramica

Questa pagina descrive la sintassi delle espressioni dei trigger e i dettagli della loro valutazione.

La sintassi di un’espressione semplice è:

```
function(/host/item,time_period)<operator><constant>
```

In questa espressione, il primo operando (a sinistra dell’operatore) è una funzione con i relativi parametri tra parentesi (in genere il data item e il periodo di tempo).

La funzione viene utilizzata per analizzare i dati ricevuti durante il periodo di tempo impostato, producendo un valore calcolato.

Questo valore viene quindi confrontato con il secondo operando, utilizzando l’operatore. In questo esempio il secondo operando è una costante, ma può anche essere un’altra funzione.

Per esempio:

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

Questo trigger si attiva se il numero di byte ricevuti su `eth0` durante gli ultimi cinque minuti è stato sempre superiore a 100 kilobyte. In questo caso l’espressione è vera e viene creato un problema.

Le espressioni dei trigger sono estremamente flessibili. Nelle espressioni più complesse, è possibile combinare più funzioni, operatori e costanti.

Vedi anche:

- Esempi di trigger
- Rilevamento dei problemi con i trigger (introduzione generale)

Funzioni

Le funzioni consentono di analizzare i valori raccolti, ad esempio calcolando la media o cercando una stringa specifica.

Fare clic sul gruppo di funzioni desiderato per visualizzare ulteriori dettagli.

Gruppo di funzioni	Funzioni
Funzioni aggregate	avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
Funzioni Foreach	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
Funzioni bitwise	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Funzioni di data e ora	date, dayofmonth, dayofweek, now, time
Funzioni della cronologia	change, changecount, count, countunique, find, first, firstclock, fuzzytime, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath
Funzioni delle tendenze	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum

Gruppo di funzioni	Funzioni
Funzioni matematiche	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
Funzioni degli operatori	between, in
Funzioni predittive	forecast, timeleft
Funzioni stringa	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

Salvo diversa indicazione, queste funzioni sono supportate in:

- Espressioni dei trigger
- item calcolati
- Macro di espressione

Le funzioni Foreach sono supportate solo per i calcoli aggregati.

In genere, le funzioni restituiscono valori numerici per il confronto. Quando restituiscono stringhe, il confronto è possibile con gli operatori **=** e **<>** (vedere l'esempio Rilevare software non corrispondente su host diversi).

Parametri delle funzioni

I parametri delle funzioni consentono di specificare:

- la chiave dell'item (come `/host/key`) per le funzioni che fanno riferimento alla cronologia di un item del host
- il periodo di tempo (e altri parametri specifici della funzione)
- altre espressioni

Chiave item

L'item di riferimento deve trovarsi in uno stato supportato (ad eccezione della funzione **nodata()**, che viene calcolata anche per gli item non supportati).

L'omissione del nome del host nel primo parametro (cioè come in `function(//key,parameter,...)`) è supportata solo in determinati contesti:

- Nella formula degli item calcolati
- Nelle macro di espressione, che possono essere utilizzate in:
  - Il campo *Nome evento*
  - Il nome del grafico
  - L'etichetta degli elementi della mappa "Host" e "Trigger" map elements

In questi contesti, è anche possibile utilizzare la macro `{HOST.HOST}`. `{HOST.HOST<1-9>}` può essere utilizzata nel caso del campo *Nome evento* e dell'elemento della mappa "Trigger" per fare riferimento a uno specifico item nell'espressione del trigger. Quando il nome del host viene omesso o sostituito da `{HOST.HOST}` in questi contesti, il riferimento punta al primo item nell'espressione del trigger o al primo item nel grafico. Al di fuori di questi contesti supportati, l'omissione del nome del host nelle espressioni trigger comporterà un errore. Per un'illustrazione dell'uso della doppia barra nelle macro del Nome evento, vedere l'esempio Compare long-term CPU loads.

Periodo di tempo

I parametri specifici della funzione vengono inseriti dopo la chiave dell'item e sono separati dalla chiave dell'item da una virgola.

La maggior parte delle funzioni numeriche accetta un **periodo di tempo** come parametro. Consente di specificare l'intervallo di interesse. Può essere specificato come periodo di tempo (30s, 10m, 1h) oppure come intervallo di valori (#5 - per gli ultimi cinque valori).

È possibile utilizzare i secondi o i suffissi temporali per indicare il periodo di tempo. Se preceduto da un cancelletto, il parametro assume un significato diverso:

Expression	Description
sum(/host/key,**10m)**	Somma dei valori degli ultimi 10 minuti.
sum(/host/key,**#10)**	Somma degli ultimi dieci valori.

I parametri preceduti da un cancelletto hanno un significato diverso con la funzione **last**: indicano l'N-esimo valore precedente. Pertanto, dati i valori 30, 70, 20, 60, 50 (dal più recente al meno recente):

- `last(/host/key,#2)` restituirebbe '70'
- `last(/host/key,#5)` restituirebbe '50'

Il periodo di tempo viene misurato fino a "now", dove "now" è l'ultimo momento di ricalcolo del trigger (vedere Frequenza di calcolo); "now" non corrisponde all'ora "now" del server.

Il periodo di tempo specifica:

- Tutti i valori compresi tra "now - periodo di tempo" e "now" (oppure, con uno scostamento temporale applicato, tra "now - scostamento temporale - periodo di tempo" e "now").
- Il numero specificato di valori passati, fino a "now".

Se non esistono dati per il periodo di valutazione specificato, il trigger o l'item calcolato che utilizza questa funzione diventa non supportato.

Tenere presente che:

- Se nel trigger viene utilizzata una sola funzione (che fa riferimento alla cronologia dei dati), "now" corrisponde sempre all'ultimo valore ricevuto. Ad esempio, se l'ultimo valore è stato ricevuto un'ora fa, il periodo di tempo verrà considerato fino all'ultimo valore ricevuto un'ora fa.
- Un nuovo trigger viene calcolato non appena viene ricevuto il primo valore (funzioni della cronologia); per le funzioni data e ora e nodata(), verrà calcolato entro 30 secondi. Pertanto, il trigger verrà calcolato anche se il periodo di tempo impostato (ad esempio, un'ora) non è ancora trascorso dalla creazione del trigger. Il trigger verrà inoltre calcolato dopo il primo valore, anche se l'intervallo di tempo è stato impostato, ad esempio, sugli ultimi dieci valori.

Scostamento temporale

È supportato uno scostamento temporale facoltativo con il tempo o il conteggio dei valori come parametro della funzione. Questo parametro consente di fare riferimento ai dati di un periodo passato.

Lo scostamento temporale inizia con `now`, che specifica l'ora corrente, ed è seguito da `+N<unità di tempo>` o `-N<unità di tempo>`, per aggiungere o sottrarre N unità di tempo.

Ad esempio, `avg(/host/key,1h:now-1d)` restituisce il valore medio di un'ora di un giorno fa.

Attention:
Lo scostamento temporale specificato in mesi (M) e anni (y) è supportato solo per le funzioni trend. Le altre funzioni supportano secondi (s), minuti (m), ore (h), giorni (d) e settimane (w).

Scostamento temporale con periodi di tempo assoluti

Nel parametro dello scostamento temporale sono supportati i periodi di tempo assoluti, ad esempio da mezzanotte a mezzanotte per un giorno, da lunedì a domenica per una settimana, dal primo all'ultimo giorno del mese per un mese.

Lo scostamento temporale per i periodi di tempo assoluti inizia con `now`, che specifica l'ora corrente, ed è seguito da un numero qualsiasi di operazioni temporali: `/<unità di tempo>` definisce l'inizio e la fine dell'unità di tempo, ad esempio da mezzanotte a mezzanotte per un giorno; `+N<unità di tempo>` o `-N<unità di tempo>` aggiunge o sottrae N unità di tempo.

Tenere presente che il valore dello scostamento temporale può essere maggiore o uguale a 0, mentre il valore minimo del periodo di tempo è 1.

Parametro	Descrizione
1d:now/d	Ieri
1d:now/d+1d	Oggi
2d:now/d+1d	Gli ultimi 2 giorni
1w:now/w	La settimana scorsa
1w:now/w+1w	Questa settimana

Altre espressioni

I parametri delle funzioni possono contenere altre espressioni, come nella sintassi seguente:

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

Si noti che non è possibile utilizzare altre espressioni se la funzione fa riferimento alla cronologia dell'item. Ad esempio, la sintassi seguente **non** è consentita:

~~`min(/host/key,#5*10)`~~

Sebbene nelle espressioni trigger altre espressioni usate come parametri di funzione siano limitate alle funzioni che non fanno riferimento alla cronologia, questa limitazione non si applica ai calculated items.

Operatori

Per i trigger sono supportati i seguenti operatori **(in ordine decrescente di priorità di esecuzione)**:

Priorità	Operatore	Definizione	**Note per i valori sconosciuti**	Forza il cast dell'operando a float [1]
1	**-**	Meno unario	**-**Unknown → Unknown	Sì
2	**not**	NOT logico	**not** Unknown → Unknown	Sì
3	*	Moltiplicazione	0 * Unknown → Unknown (sì, Unknown, non 0 - per non perdere Unknown nelle operazioni aritmetiche) 1.2 * Unknown → Unknown	Sì
	/	Divisione	Unknown / 0 → errore Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Sì
4	+	Addizione aritmetica	1.2 + Unknown → Unknown	Sì
	-	Sottrazione aritmetica	1.2 - Unknown → Unknown	Sì
5	<	Minore di. L'operatore è definito come: A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	Sì
	<=	Minore o uguale a. L'operatore è definito come: A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	Sì
	>	Maggiore di. L'operatore è definito come: A>B ⇔ (A>B+0.000001)		Sì
	>=	Maggiore o uguale a. L'operatore è definito come: A>=B ⇔ (A≥B-0.000001)		Sì

Priorità	Operatore	Definizione	**Note per i valori sconosciuti**	Forza il cast dell'operando a float [1]
6	**=**	Uguale a. L'operatore è definito come: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		No [1]
	<>	Diverso da. L'operatore è definito come: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		No [1]
7	**and**	AND logico	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	Sì
8	**or**	OR logico	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	Sì

[1] L'operando stringa viene comunque convertito in numerico se:

- l'altro operando è numerico
- su un operando viene usato un operatore diverso da **=** o **<>**

(Se la conversione fallisce, l'operando numerico viene convertito in un operando stringa ed entrambi gli operandi vengono confrontati come stringhe.)

Gli operatori **not**, **and** e **or** sono case-sensitive e devono essere in minuscolo. Devono inoltre essere circondati da spazi o parentesi.

Tutti gli operatori, eccetto **-** unario e **not**, hanno associatività da sinistra. **-** unario e **not** non sono associativi (ciò significa che devono essere usati **-(-1)** e **not (not 1)** invece di **--1** e **not not 1**).

Risultato della valutazione:

- gli operatori **<**, **<=**, **>**, **>=**, **=**, **<>** devono produrre '1' nell'espressione del trigger se la relazione specificata è vera e '0' se è falsa. Se almeno un operando è Unknown, il risultato è Unknown;
- **and** per operandi noti deve produrre '1' se entrambi i suoi operandi risultano diversi da '0'; altrimenti produce '0'; per operandi sconosciuti **and** produce '0' solo se un operando risulta uguale a '0'; altrimenti produce 'Unknown';
- **or** per operandi noti deve produrre '1' se uno qualsiasi dei suoi operandi risulta diverso da '0'; altrimenti produce '0'; per operandi sconosciuti **or** produce '1' solo se un operando risulta diverso da '0'; altrimenti produce 'Unknown';
- il risultato dell'operatore di negazione logica **not** per un operando noto è '0' se il valore del suo operando risulta diverso da '0'; '1' se il valore del suo operando risulta uguale a '0'. Per un operando sconosciuto **not** produce 'Unknown'.

Stato dell'espressione sconosciuto

È possibile che un operando sconosciuto compaia in un'espressione di trigger quando:

- viene usato un item non supportato
- la valutazione di una funzione, per un item supportato, termina con un errore

In questo caso, l'espressione del trigger in genere viene valutata come `Unknown` (poiché non può essere valutata)

È possibile ricevere una notifica sui trigger sconosciuti.

Eccezioni

Nonostante un operando sconosciuto, in alcuni casi le espressioni del trigger possono essere valutate con un risultato noto (Problem/OK):

- La funzione `nodata()` viene valutata indipendentemente dal fatto che l'item referenziato sia supportato o meno.
- Le espressioni con AND/OR possono essere valutate con un risultato noto in due casi:

- **Caso 1**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" viene valutata con un risultato noto ('1' o "Problem"),
- **Caso 2**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" viene valutata con un risultato noto ('0' o "OK").

- Se la valutazione della funzione per un item supportato termina con un errore, il valore della funzione diventa `Unknown` e partecipa come operando sconosciuto nelle valutazioni successive dell'espressione.

Gli operandi sconosciuti possono "scomparire" solo nelle espressioni logiche, come descritto sopra. Nelle espressioni aritmetiche, gli operandi sconosciuti portano sempre a `Unknown` (tranne la divisione per 0).

Attention:
Lo stato dell'espressione sconosciuto non modifica lo stato del trigger (Problem/OK). Quindi, se lo stato del trigger era "Problem" (vedi Caso 1), rimane nello stato di problema anche se la parte nota viene risolta ('1' diventa '0'), perché l'espressione viene ora valutata come `Unknown` e questo non modifica lo stato del trigger.

Se un'espressione del trigger con diversi item non supportati viene valutata come `Unknown`, il messaggio di errore nel frontend fa riferimento all'ultimo item non supportato valutato.

Cache dei valori

I valori richiesti per la valutazione dei trigger vengono memorizzati nella cache dal server Zabbix. Per questo motivo, la valutazione dei trigger provoca un carico più elevato sul database per un certo periodo di tempo dopo il riavvio del server.

La cache dei valori non viene svuotata quando i valori della cronologia degli item vengono rimossi (manualmente o dal processo housekeeper), quindi il server utilizzerà i valori memorizzati nella cache finché non saranno più vecchi dei periodi di tempo definiti nelle funzioni dei trigger o finché il server non verrà riavviato.

Note:
Se nella cache non sono presenti dati recenti e nella funzione non è definito alcun periodo di interrogazione, per impostazione predefinita Zabbix risalirà fino a una settimana nel passato per interrogare il database alla ricerca di valori storici.

Funzioni aggregate

Salvo diversa indicazione, tutte le funzioni elencate qui sono supportate in:

- Espressioni trigger
- Formule degli item calcolati
- Macro di espressione

Le funzioni aggregate possono funzionare con uno dei seguenti elementi:

- la cronologia degli item, ad esempio, `min(/host/key,1h)`
- funzioni foreach come unico parametro, ad esempio, `min(last_foreach(/*/key))` (solo nelle formule degli item calcolati; per usarle nei trigger, fare riferimento all'item calcolato tramite la sua chiave)

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare i dettagli completi.

Function	Description
avg	Il valore medio di un item nell'intervallo di valutazione definito.
bucket_percentile	Calcola il percentile dai bucket di un istogramma.
count	Il conteggio dei valori in un array restituito da una funzione foreach.
histogram_quantile	Calcola il φ-quantile dai bucket di un istogramma.
item_count	Il conteggio degli item esistenti nella configurazione che corrispondono ai criteri del filtro.
kurtosis	La "curtosi" della distribuzione di probabilità nei valori raccolti nell'intervallo di valutazione definito.
mad	La deviazione assoluta mediana nei valori raccolti nell'intervallo di valutazione definito.
max	Il valore più alto di un item nell'intervallo di valutazione definito.
min	Il valore più basso di un item nell'intervallo di valutazione definito.
skewness	L'asimmetria della distribuzione di probabilità nei valori raccolti nell'intervallo di valutazione definito.
stddevpop	La deviazione standard della popolazione nei valori raccolti nell'intervallo di valutazione definito.
stddevsamp	La deviazione standard del campione nei valori raccolti nell'intervallo di valutazione definito.
sum	La somma dei valori raccolti nell'intervallo di valutazione definito.
sumofsquares	La somma dei quadrati nei valori raccolti nell'intervallo di valutazione definito.

Function	Description
varpop	La varianza della popolazione dei valori raccolti nell'intervallo di valutazione definito.
varsamp	La varianza del campione dei valori raccolti nell'intervallo di valutazione definito.

Parametri comuni

- `/host/key` è un primo parametro comune obbligatorio per le funzioni che fanno riferimento alla cronologia dell'item del host.
- `(sec|#num)<:time shift>` è un secondo parametro comune per le funzioni che fanno riferimento alla cronologia dell'item del host, dove:
  - **sec** - periodo massimo di valutazione in secondi (è possibile utilizzare i suffissi temporali).
  - **#num** - intervallo massimo di valutazione nei valori raccolti più recenti (se preceduto da un cancelletto).
  - **time shift** (facoltativo) consente di spostare indietro nel tempo il punto di valutazione. Per ulteriori dettagli sulla specifica del time shift.

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- I parametri opzionali della funzione (o parti di parametro) sono indicati da < >
- I parametri specifici della funzione sono descritti per ciascuna funzione
- I parametri `/host/key` e `(sec|#num)<:time shift>` non devono mai essere racchiusi tra virgolette

avg(/host/key,(sec|#num)<:time shift>)

Il valore medio di un item nel periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzioni foreach supportate: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: vedere i parametri comuni.

Lo spostamento temporale è utile quando è necessario confrontare il valore medio corrente con il valore medio di qualche tempo fa.

Esempi:

```
avg(/host/key,1h) #calcola il valore medio per l'ultima ora fino ad ora
avg(/host/key,#5) #calcola il valore medio degli ultimi cinque valori
avg(/host/key,1h:now-1d) #calcola il valore medio per un'ora, da 25 ore fa a 24 ore fa rispetto ad ora
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) #calcola il carico medio del processo
avg(/host/proc.num,5m)>300 #trigger se il numero medio di processi negli ultimi 5 minuti è stato superiore
```

bucket_percentile(item filter,time period,percentage)

Calcola il percentile dai bucket di un istogramma.

Parametri:

- **item filter** - vedere item filter;
- **time period** - vedere time period;
- **percentage** - percentuale (0-100).

Commenti:

- Supportata solo negli item calcolati;
- Questa funzione è un alias per `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))`.

Esempi:

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #calcola il 95° percentile del tempo d
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #calcola la latenza API al 99° p
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

Il conteggio dei valori in un array restituito da una funzione foreach.
 Funzioni foreach supportate: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri:

- **func_foreach** - funzione foreach per la quale deve essere conteggiato il numero di valori restituiti. Per i dettagli, vedere funzioni foreach. Nota che count_foreach e bucket_rate_foreach supportano parametri aggiuntivi.

- **item filter** - vedere la sintassi del filtro item;
- **time period** - vedere il periodo di tempo;
- **operator** (deve essere tra virgolette doppie). `operatori` supportati:
eq - uguale
ne - diverso
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like - corrisponde se contiene il pattern (sensibile alle maiuscole/minuscole)
bitand - AND bit a bit
regexp - corrispondenza sensibile alle maiuscole/minuscole dell'espressione regolare specificata in `pattern`
iregexp - corrispondenza non sensibile alle maiuscole/minuscole dell'espressione regolare specificata in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra virgolette doppie); supportato se *operator* è specificato nel terzo parametro.

Commenti:

- L'uso di **count()** con una funzione foreach basata sulla history (max_foreach, avg_foreach, ecc.) può avere implicazioni sulle prestazioni, mentre l'uso di **exists_foreach()**, che funziona solo con i dati di configurazione, non avrà questo effetto.
- I parametri opzionali *operator* o *pattern* non possono essere lasciati vuoti dopo una virgola, ma solo omessi completamente.
- Con *bitand* come terzo parametro, il quarto parametro `pattern` può essere specificato come due numeri, separati da '/': **number_to_compare_with/mask**. count() calcola l'"AND bit a bit" tra il valore e la *mask* e confronta il risultato con *number_to_compare_with*. Se il risultato dell'"AND bit a bit" è uguale a *number_to_compare_with*, il valore viene conteggiato.
Se *number_to_compare_with* e *mask* sono uguali, è sufficiente specificare solo la *mask* (senza '/').
- Con *regexp* o *iregexp* come terzo parametro, il quarto parametro `pattern` può essere un'espressione regolare ordinaria o globale (che inizia con '@'). Nel caso delle espressioni regolari globali, la sensibilità alle maiuscole/minuscole viene ereditata dalle impostazioni dell'espressione regolare globale. Ai fini della corrispondenza regexp, i valori float saranno sempre rappresentati con 4 cifre decimali dopo '.'. Si noti inoltre che, per numeri grandi, la differenza tra la rappresentazione decimale (memorizzata nel database) e quella binaria (usata dal server Zabbix) può influire sulla 4a cifra decimale.

Esempi:

```
count(max_foreach(/*/net.if.in[*],1h)) #calcola il numero di item net.if.in che hanno ricevuto dati nell'u
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #calcola il numero di file system con oltre il 95% di
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Calcola il quantile φ a partire dai bucket di un istogramma.
 Supporta la funzione foreach: *bucket_rate_foreach*.

Parametri:

- **quantile** - 0 ≤ φ ≤ 1;
- **bucketN, valueN** - coppie inserite manualmente (>=2) di parametri oppure la risposta di bucket_rate_foreach.

Commenti:

- Supportata solo negli item calcolati;
- Corrisponde funzionalmente a 'histogram_quantile' di PromQL;
- Restituisce -1 se i valori dell'ultimo bucket 'Infinity' ("+*inf*") sono uguali a 0.

Esempi:

```
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)

Il conteggio degli item esistenti nella configurazione che corrispondono ai criteri del filtro.
 Tipo di valore supportato: *Integer*.

Parametro:

- **item filter** - criteri per la selezione degli item; consente il riferimento per gruppo di host, host, chiave item e tag. I caratteri jolly sono supportati. Per maggiori dettagli, vedere item filter.

Commenti:

- Supportato solo negli item calcolati;
- Funziona come alias della funzione *count(exists_foreach(item_filter))*.

Esempi:

```
item_count(/*/agent.ping?[group="Host group 1"]) #calcola il numero di host con l'item agent.ping in "Host
```

kurtosis(/host/key,(sec|#num)<:time shift>)

La "curtosi" della distribuzione di probabilità nei valori raccolti entro il periodo di valutazione definito. Vedi anche: Kurtosis.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #trigger on sharp spikes of CPU utilization over 5 minutes
```

mad(/host/key,(sec|#num)<:time shift>)

La deviazione assoluta mediana dei valori raccolti nel periodo di valutazione definito. Vedi anche: Deviazione assoluta mediana.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

Il valore più alto di un item entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzioni foreach supportate: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
max(/host/key,1h) - min(/host/key,1h) #calcola la differenza tra i valori massimo e minimo nell'ultima ora
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) #restituisce l'utilizzo del disco più e
```

min(/host/key,(sec|#num)<:time shift>)

Il valore più basso di un item nel periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzioni foreach supportate: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
max(/host/key,1h) - min(/host/key,1h) #calcola la differenza tra i valori massimo e minimo nell'ultima ora
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) #restit
```

skewness(/host/key,(sec|#num)<:time shift>)

L'asimmetria della distribuzione di probabilità nei valori raccolti entro il periodo di valutazione definito. Vedere anche: Skewness.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
skewness(/host/vm.memory.size[used],5m)<-20 #trigger on sudden negative drops indicating crashes/restarts
```

stddevpop(/host/key,(sec|#num)<:time shift>)

La deviazione standard della popolazione nei valori raccolti entro il periodo di valutazione definito. Vedi anche: Deviazione standard.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere parametri comuni.

Esempi:

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #trigger if CPU usage fluctuates heavily (population standar
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

La deviazione standard campionaria dei valori raccolti nel periodo di valutazione definito. Vedi anche: Deviazione standard.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere parametri comuni.

Per il funzionamento di questa funzione sono richiesti almeno due valori di dati.

Esempi:

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #trigger if CPU usage fluctuates heavily (sample standard d
```

sum(/host/key,(sec|#num)<:time shift>)

La somma dei valori raccolti entro il periodo di valutazione definito.
 Tipi di valori supportati: *Float*, *Integer*.
 Funzioni foreach supportate: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #calculate the total incoming network traffic f
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) #calculate the total disk space for all
sum(last_foreach(/*/net.dns[*,*,*])) #calculate the total number of successful DNS checks
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

La somma dei quadrati nei valori raccolti entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
sumofsquares(/host/calculated.net.if.rate,1m) #la somma dei quadrati per il tasso di traffico di rete (in
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #la somma dei quadrati degli ultimi va
```

varpop(/host/key,(sec|#num)<:time shift>)

La varianza della popolazione dei valori raccolti entro il periodo di valutazione definito. Vedere anche: Varianza.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach function supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempi:

```
varpop(/host/system.cpu.util[,user],10m)>50 #trigger if CPU usage fluctuates heavily (variance is above 50
```

varsamp(/host/key,(sec|#num)<:time shift>)

La varianza campionaria dei valori raccolti entro il periodo di valutazione definito. Vedere anche: Varianza.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Per il funzionamento di questa funzione sono richiesti almeno due valori di dati.

Esempi:

```
varsamp(/host/system.cpu.util[,user],10m)>50 #trigger if CPU usage fluctuates heavily (sample variance is
```

Vedere tutte le funzioni supportate.

Funzioni foreach

Panoramica

Le funzioni foreach vengono utilizzate nelle calcolazioni aggregate per restituire un valore aggregato per ogni item selezionato dal **filtro item** utilizzato. Viene restituito un array di valori.

Ad esempio, la funzione `avg_foreach` restituirà un array di valori, in cui ogni valore è la *media* del valore storico dell'item selezionato, durante l'intervallo di tempo specificato.

Attention:
Le funzioni foreach sono supportate solo nelle formule di Calculated item. Non possono essere richiamate in semplici chiavi item né utilizzate direttamente per costruire espressioni trigger. Nelle espressioni trigger, un calculated item può essere utilizzato facendo riferimento alla sua chiave.

Il filtro item fa parte della sintassi utilizzata dalle funzioni foreach. L'uso dei caratteri jolly è supportato nel filtro item, quindi gli item richiesti possono essere selezionati in modo molto flessibile.

Funzioni supportate

Function	Description
avg_foreach	Restituisce il valore medio per ciascun item.
bucket_rate_foreach	Restituisce coppie (limite superiore del bucket, valore del rate) adatte all'uso nella funzione `histogram_quantile()`, dove "limite superiore del bucket" è il valore del parametro della chiave dell'item definito dal parametro <parameter number>.

Function	Description
count_foreach	Restituisce il numero di valori per ciascun item.
exists_foreach	Restituisce '1' per ciascun item abilitato.
last_foreach	Restituisce l'ultimo valore per ciascun item.
max_foreach	Restituisce il valore massimo per ciascun item.
min_foreach	Restituisce il valore minimo per ciascun item.
sum_foreach	Restituisce la somma dei valori per ciascun item.

Sintassi della funzione

Le funzioni foreach supportano due parametri comuni: `filtro item` (vedi i dettagli sotto) e `periodo di tempo`:

```
foreach_function(item filter,time period)
```

Per esempio:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

restituirà la media su cinque minuti di ciascun item 'mysql.qps' nel gruppo di server MySQL.

Si noti che alcune funzioni supportano parametri aggiuntivi.

Sintassi del filtro item

Il filtro item:

```
/host/key[parameters]?[conditions]
```

è composto da quattro parti, dove:

- host - nome del host
- key - chiave dell'item (senza parametri)
- parameters - parametri della chiave dell'item
- conditions - condizioni basate sul gruppo host e/o sui tag dell'item (come espressione)

Gli spazi sono consentiti solo all'interno dell'espressione delle condizioni.

Uso dei caratteri jolly

- Il carattere jolly può essere utilizzato per sostituire il nome del host, la chiave dell'item o un singolo parametro della chiave dell'item.
- Il host oppure la chiave dell'item deve essere specificato senza carattere jolly. Quindi `/host/*` e `/*/key` sono filtri validi, ma `/*/*` non è valido.
- Il carattere jolly non può essere utilizzato per una *parte* del nome del host, della chiave dell'item o del parametro della chiave dell'item.
- Il carattere jolly non corrisponde a più di un singolo parametro della chiave dell'item. Pertanto, deve essere specificato un carattere jolly per ciascun parametro separatamente (ad esempio `key[abc,*,*]`).

Espressione delle condizioni

L'espressione delle condizioni supporta:

- operandi:
  - group - gruppo host
  - tag - tag dell'item
  - "<text>" - costante stringa, con il carattere di escape \ per eseguire l'escape di " e \
- operatori di confronto tra stringhe con distinzione tra maiuscole e minuscole: =, <>
- operatori logici: `and`, `or`, `not`
- raggruppamento con parentesi: ( )

Le virgolette per le costanti stringa sono obbligatorie. È supportato solo il confronto completo tra stringhe con distinzione tra maiuscole e minuscole.

Warning:
Quando si specificano i tag nel filtro (ad esempio `tag="tagname:value"`), i due punti ":" vengono utilizzati come delimitatore. Tutto ciò che segue viene considerato il valore del tag. Pertanto, attualmente non è supportato specificare un nome tag che contenga ":".

Esempi

Può essere utilizzato un filtro complesso che fa riferimento alla chiave dell'item, al gruppo host e ai tag, come illustrato negli esempi:

Syntax example	Description
`/host/key[abc,*]`	Corrisponde a item simili su questo host.
`/*/key`	Corrisponde allo stesso item di qualsiasi host.
`/*/key?[group="ABC" and tag="tagname:value"]`	Corrisponde allo stesso item di qualsiasi host del gruppo ABC che abbia tag 'tagname:value'.
`/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]`	Corrisponde a item simili di qualsiasi host del gruppo ABC o DEF con i rispettivi tag.

Tutti gli item referenziati devono esistere e raccogliere dati. Solo gli item abilitati su host abilitati sono inclusi nei calcoli. Gli item nello stato unsupported non sono inclusi.

Attention:
Se la chiave dell'item di un item referenziato viene modificata, il filtro deve essere aggiornato manualmente.

La specifica di un gruppo host padre include il gruppo padre e tutti i gruppi host nidificati con i relativi item.

Periodo di tempo

Il parametro **secondo** consente di specificare il periodo di tempo per l'aggregazione. Il periodo di tempo può essere espresso solo come intervallo temporale; non è supportata la quantità di valori (preceduta da **#**).

In questo parametro è possibile utilizzare i simboli delle unità supportate per comodità, ad esempio `5m` (cinque minuti) invece di `300s` (300 secondi) oppure `1d` (un giorno) invece di `86400` (86400 secondi).

Per la funzione `last_foreach`, il periodo di tempo è un parametro facoltativo (supportato a partire da Zabbix 7.0) e può essere omesso:

```
last_foreach(/*/key?[group="host group"])
```

Il periodo di tempo non è supportato con la funzione `exists_foreach`.

Parametri aggiuntivi

bucket_rate_foreach

La funzione `bucket_rate_foreach` supporta un terzo parametro opzionale:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

dove `<parameter number>` è la posizione del valore "bucket" nella chiave dell'item. Ad esempio, se il valore "bucket" in `myItem[aaa,0.2]` è `0.2`, la sua posizione è 2.

Il valore predefinito di `<parameter number>` è `1`.

count_foreach

La funzione `count_foreach` supporta un terzo e un quarto parametro opzionali:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Dove:

- **operator** è l'operatore condizionale per i valori dell'item (deve essere racchiuso tra virgolette doppie). Operatori supportati:
eq - uguale
ne - diverso
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like - corrisponde se contiene il pattern (sensibile alle maiuscole/minuscole)
bitand - AND bit a bit
regexp - corrispondenza sensibile alle maiuscole/minuscole con l'espressione regolare specificata in `pattern`
iregexp - corrispondenza non sensibile alle maiuscole/minuscole con l'espressione regolare specificata in `pattern`.
- **pattern** è il pattern richiesto (gli argomenti stringa devono essere racchiusi tra virgolette doppie); è supportato se *operator* è specificato nel terzo parametro.

Commenti:

- I parametri opzionali *operator* o *pattern* non possono essere lasciati vuoti dopo una virgola, ma solo omessi completamente.

- Con *bitand* come terzo parametro, il quarto parametro `pattern` può essere specificato come due numeri, separati da '/': **number_to_compare_with/mask**. `count_foreach()` calcola l'"AND bit a bit" tra il valore e la *mask* e confronta il risultato con *number_to_compare_with*. Se il risultato dell'"AND bit a bit" è uguale a *number_to_compare_with*, il valore viene conteggiato.
Se *number_to_compare_with* e *mask* sono uguali, è necessario specificare solo la *mask* (senza '/').
- Con *regexp* o *iregexp* come terzo parametro, il quarto parametro `pattern` può essere un'espressione regolare ordinaria o globale (che inizia con '@'). Nel caso delle espressioni regolari globali, la sensibilità alle maiuscole/minuscole viene ereditata dalle impostazioni dell'espressione regolare globale. Ai fini della corrispondenza regexp, i valori float saranno sempre rappresentati con 4 cifre decimali dopo '.'. Si noti inoltre che, per numeri grandi, la differenza tra la rappresentazione decimale (memorizzata nel database) e quella binaria (usata dal server Zabbix) può influire sulla quarta cifra decimale.

Vedi calcoli aggregati per maggiori dettagli ed esempi sull'uso delle funzioni foreach.

Comportamento in base alla disponibilità

La tabella seguente illustra come si comporta ciascuna funzione nei casi di disponibilità limitata di host/item e dei dati di storico.

Function	Host disabilitato	Host non disponibile con dati	Host non disponibile senza dati	Item disabilitato	Item non supportato	Errore di recupero dati (SQL)
avg_foreach	ignora	restituisce avg	ignora	ignora	ignora	ignora
bucket_rate_foreach	ignora	restituisce bucket rate	ignora	ignora	ignora	ignora
count_foreach	ignora	restituisce count	0	ignora	ignora	ignora
exists_foreach	ignora	1	1	ignora	1	n/d
last_foreach	ignora	restituisce last	ignora	ignora	ignora	ignora
max_foreach	ignora	restituisce max	ignora	ignora	ignora	ignora
min_foreach	ignora	restituisce min	ignora	ignora	ignora	ignora
sum_foreach	ignora	restituisce sum	ignora	ignora	ignora	ignora

Se l'item viene *ignorato*, non viene aggiunto nulla all'aggregazione.

Funzioni bitwise

Tutte le funzioni elencate qui sono supportate in:

- Espressioni trigger
- Formule degli item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare i dettagli completi.

Function	Description
bitand	Il valore di "AND bitwise" di un valore di item e di una maschera.
bitlshift	Lo shift bitwise a sinistra di un valore di item.
bitnot	Il valore di "NOT bitwise" di un valore di item.
bitor	Il valore di "OR bitwise" di un valore di item e di una maschera.
bitrshift	Lo shift bitwise a destra di un valore di item.
bitxor	Il valore di "OR esclusivo bitwise" di un valore di item e di una maschera.

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- Le espressioni sono accettate come parametri
- I parametri opzionali della funzione (o parti di parametro) sono indicati da < >

bitand(value,mask)

Il valore di "AND bit a bit" tra il valore di un item e una maschera.
 Tipi di valore supportati: *Integer*.

Parametri:

- **value** - il valore da controllare;

- **mask** (obbligatorio) - un intero senza segno a 64 bit (0 - 18446744073709551615).

Sebbene il confronto venga eseguito in modalità bit a bit, tutti i valori devono essere forniti e vengono restituiti in formato decimale. Ad esempio, il controllo del 3° bit viene eseguito confrontando con 4, non con 100.

Esempi:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3° o 4° bit impostato, ma non entrambi conte
bitand(last(/host/key),20)=16 #3° bit non impostato e 5° bit impostato
```

bitlshift(value,bits da spostare)

Lo spostamento bit a sinistra del valore di un item.
 Tipi di valore supportati: *Intero*.

Parametri:

- **value** - il valore da controllare;
- **bits to shift** (obbligatorio) - il numero di bit da spostare.

Sebbene il confronto venga eseguito a livello di bit, tutti i valori devono essere forniti e vengono restituiti in formato decimale. Ad esempio, il controllo del 3° bit viene eseguito confrontandolo con 4, non con 100.

bitnot(value)

Il valore di ”NOT bit a bit” di un valore di item.
 Tipi di valore supportati: *Integer*.

Parametro:

- **value** - il valore da controllare.

Sebbene il confronto venga eseguito in modalità bit a bit, tutti i valori devono essere forniti e vengono restituiti in formato decimale. Ad esempio, il controllo del 3° bit viene eseguito confrontando con 4, non con 100.

bitor(value,mask)

Il valore dell’”OR bit a bit” tra il valore di un item e la maschera.
 Tipi di valore supportati: *Intero*.

Parametri:

- **value** - il valore da controllare;
- **mask** (obbligatorio) - un intero senza segno a 64 bit (0 - 18446744073709551615).

Sebbene il confronto venga eseguito in modo bit a bit, tutti i valori devono essere forniti e vengono restituiti in formato decimale. Ad esempio, il controllo del 3° bit viene eseguito confrontando con 4, non con 100.

bitrshift(valore, bit da spostare)

Lo spostamento bit a destra del valore di un item.
 Tipi di valore supportati: *Integer*.

Parametri:

- **value** - il valore da controllare;
- **bits to shift** (obbligatorio) - il numero di bit da spostare.

Sebbene il confronto venga eseguito a livello di bit, tutti i valori devono essere forniti e vengono restituiti in formato decimale. Ad esempio, il controllo del 3° bit viene eseguito confrontando con 4, non con 100.

bitxor(value,mask)

Il valore dell’”OR esclusivo bit a bit” tra un valore di item e una maschera.
 Tipi di valore supportati: *Intero*.

Parametri:

- **value** - il valore da controllare;
- **mask** (obbligatorio) - un intero senza segno a 64 bit (0 - 18446744073709551615).

Sebbene il confronto venga eseguito a livello di bit, tutti i valori devono essere forniti e vengono restituiti in formato decimale. Ad esempio, il controllo del 3° bit viene eseguito confrontando con 4, non con 100.

Vedere tutte le funzioni supportate.

Funzioni di data e ora

Tutte le funzioni elencate qui sono supportate in:

- Espressioni di trigger
- Formule degli item calcolati

- Macro di espressione

Attention:
Le funzioni di data e ora non possono essere usate da sole nell'espressione; deve essere inclusa almeno una funzione di un altro gruppo, che faccia riferimento all'item dell'host, nell'espressione (ad eccezione della funzione nodata()). Per informazioni dettagliate sul funzionamento delle funzioni di data e ora all'interno delle espressioni, vedere Tempo di calcolo.

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare i dettagli completi.

Function	Description
date	La data corrente nel formato YYYYMMDD.
dayofmonth	Il giorno del mese nell'intervallo da 1 a 31.
dayofweek	Il giorno della settimana nell'intervallo da 1 a 7.
now	Il numero di secondi trascorsi dall'Epoch (00:00:00 UTC, 1 gennaio 1970).
time	L'ora corrente nel formato HHMMSS.

Dettagli della funzione

date

La data corrente nel formato YYYYMMDD.

Esempio:

`date()<20220101`

dayofmonth

Il giorno del mese nell'intervallo da 1 a 31.

Esempio:

`dayofmonth()=1`

dayofweek

Il giorno della settimana nell'intervallo da 1 a 7 (lun - 1, dom - 7).

Esempio (solo giorni feriali):

`dayofweek()<6`

Esempio (solo fine settimana):

`dayofweek()>5`

now

Il numero di secondi trascorsi dall'Epoch (00:00:00 UTC, 1 gennaio 1970).

Esempio:

`now()<1640998800`

time

L'ora corrente nel formato HHMMSS.

Esempio (solo ore notturne, 00:00-06:00):

`time()<060000`

Vedere tutte le funzioni supportate.

Funzioni della cronologia

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Formule degli item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Funzione	Descrizione
change	La differenza tra il valore precedente e quello più recente.
changecount	Il numero di variazioni tra valori adiacenti nel periodo di valutazione definito.
count	Il numero di valori nel periodo di valutazione definito.
countunique	Il numero di valori univoci nel periodo di valutazione definito.
find	Trova una corrispondenza di valore nel periodo di valutazione definito.
first	Il primo valore (il più vecchio) nel periodo di valutazione definito.
firstclock	Il timestamp del primo valore (il più vecchio) nel periodo di valutazione definito.
fuzzytime	Verifica di quanto l'ora dell'agent passivo differisce dall'ora del server/proxy Zabbix.
jsonpath	Restituisce un array dei risultati della query JSONPath.
last	Il valore più recente.
lastclock	Il timestamp dell'N-esimo valore più recente nel periodo di valutazione definito.
logeventid	Verifica se l'ID evento dell'ultima voce di log corrisponde a un'espressione regolare.
logseverity	La gravità del log dell'ultima voce di log.
logsource	Verifica se l'origine del log dell'ultima voce di log corrisponde a un'espressione regolare.
logtimestamp	Il timestamp del messaggio di log del valore dell'item di log N-esimo più recente.
monodec	Verifica se si è verificata una diminuzione monotona dei valori.
monoinc	Verifica se si è verificato un aumento monotono dei valori.
nodata	Verifica che non siano stati ricevuti dati.
percentile	Il percentile P di un periodo, dove P (percentuale) è specificato dal terzo parametro.
rate	La velocità media al secondo dell'aumento di un contatore in aumento monotono nel periodo di tempo definito.
xmlxpath	Restituisce un array dei risultati della query XML XPath.

Parametri comuni

- `/host/key` è un primo parametro comune obbligatorio per le funzioni che fanno riferimento alla cronologia dell'item del host.
- `(sec|#num)<:time shift>` è un secondo parametro comune per le funzioni che fanno riferimento alla cronologia dell'item del host, dove:
  - **sec** - periodo massimo di valutazione in secondi (è possibile utilizzare i suffissi temporali), oppure
  - **#num** - intervallo massimo di valutazione nei valori raccolti più recenti (se preceduto da un cancelletto).
  - **time shift** (facoltativo) consente di spostare indietro nel tempo il punto di valutazione. Per ulteriori dettagli sulla specifica del time shift.

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- I parametri opzionali della funzione (o parti di parametro) sono indicati da < >
- I parametri specifici della funzione sono descritti per ciascuna funzione
- I parametri `/host/key` e `(sec|#num)<:time shift>` non devono mai essere racchiusi tra virgolette

change(/host/key)

La quantità di differenza tra il valore precedente e l'ultimo valore.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.
 Per le stringhe restituisce: 0 - i valori sono uguali; 1 - i valori sono diversi.

Parametri: vedere parametri comuni.

Commenti:

- Verrà calcolata la differenza numerica, come mostrato in questi valori di esempio in ingresso ('valore precedente' e 'ultimo valore' = differenza):
'1' e '5' = +4
'3' e '1' = −2
'0' e '-2.5' = −2.5.
- Nei trigger, è utile per rilevare picchi improvvisi (o cali), azzeramenti dei contatori, transizioni di stato numeriche.
- Vedere anche: abs per il confronto.

Esempi:

```
change(/host/system.uptime)<0 #la variazione di system uptime è stata negativa dall'ultimo valore (indican
change(/host/system.cpu.load[all,avg1])>2 #il carico CPU (per un minuto) è aumentato di oltre 2 dall'ultim
change(/host/vfs.fs.size[/,free])<-1G #lo spazio libero su disco è diminuito di oltre 1 GB tra i controlli
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

Il numero di variazioni tra valori adiacenti entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere i parametri comuni
- **mode** (deve essere racchiuso tra doppi apici) - valori possibili: *all* - conta tutte le variazioni (predefinito); *dec* - conta le diminuzioni; *inc* - conta gli aumenti

Commenti:

- Per i tipi di valore non numerici, il parametro *mode* viene ignorato.
- Nei trigger, è utile per rilevare frequenti cambiamenti di stato (che indicano instabilità).

Esempi:

```
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>)

Il numero di valori all'interno del periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedi parametri comuni
- **operator** (deve essere tra virgolette doppie) `operator` supportati:
eq - uguale (predefinito per integer, float)
ne - non uguale
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like (predefinito per string, text, log) - corrisponde se contiene il pattern (case-sensitive)
bitand - AND bit a bit
regexp - corrispondenza case-sensitive dell'espressione regolare fornita in `pattern`
iregexp - corrispondenza case-insensitive dell'espressione regolare fornita in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra virgolette doppie)

Commenti:

- Gli item Float corrispondono con una precisione di 2.22e-16.
- *like* non è supportato come operatore per valori integer.
- *like* e *bitand* non sono supportati come operatori per valori float.
- Per valori string, text e log sono supportati solo gli operatori *eq*, *ne*, *like*, *regexp* e *iregexp*.
- Con *bitand* come operatore, il quarto parametro `pattern` può essere specificato come due numeri, separati da '/': **number_to_compare_with/mask**. count() calcola l'"AND bit a bit" dal valore e dalla *mask* e confronta il risultato con *number_to_compare_with*. Se il risultato dell'"AND bit a bit" è uguale a *number_to_compare_with*, il valore viene contato.
Se *number_to_compare_with* e *mask* sono uguali, è necessario specificare solo la *mask* (senza '/').
- Con *regexp* o *iregexp* come operatore, il quarto parametro `pattern` può essere una normale espressione regolare o una globale (che inizia con '@'). Nel caso delle espressioni regolari globali, la sensibilità alle maiuscole/minuscole viene ereditata dalle impostazioni dell'espressione regolare globale. Ai fini della corrispondenza regexp, i valori float saranno sempre rappresentati con 4 cifre decimali dopo '.'. Si noti inoltre che per numeri grandi la differenza tra la rappresentazione decimale (memorizzata nel database) e quella binaria (usata dal server Zabbix) può influire sulla quarta cifra decimale.

Esempi:

```
count(/host/icmpping,30m,,"0")>5 #ping non riuscito più di 5 volte in 30 minuti
count(/host/key,10m,"like","error") #il numero di valori degli ultimi 10 minuti fino a ora che contengono
count(/host/key,10m,,12) #il numero di valori degli ultimi 10 minuti fino a ora uguali a '12'
count(/host/key,10m,"gt",12) #il numero di valori degli ultimi 10 minuti fino a ora superiori a '12'
count(/host/key,#10,"gt",12) #il numero di valori tra gli ultimi 10 valori fino a ora superiori a '12'
count(/host/key,10m:now-1d,"gt",12) #il numero di valori compresi tra 24 ore e 10 minuti e 24 ore fa rispe
count(/host/key,10m,"bitand","6/7") #il numero di valori degli ultimi 10 minuti fino a ora che hanno '110'
count(/host/key,10m:now-1d) #il numero di valori compresi tra 24 ore e 10 minuti e 24 ore fa rispetto a or
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Il numero di valori unici all'interno del periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedi parametri comuni

- **operator** (deve essere tra virgolette doppie). `operator` supportati:
eq - uguale (predefinito per integer, float)
ne - diverso
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like (predefinito per string, text, log) - corrisponde se contiene il pattern (sensibile alle maiuscole/minuscole)
bitand - AND bit a bit
regexp - corrispondenza sensibile alle maiuscole/minuscole con l'espressione regolare indicata in `pattern`
iregexp - corrispondenza non sensibile alle maiuscole/minuscole con l'espressione regolare indicata in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra virgolette doppie)

Commenti:

- Gli item Float corrispondono con una precisione di 2.22e-16.
- *like* non è supportato come operator per valori integer.
- *like* e *bitand* non sono supportati come operator per valori float.
- Per valori string, text e log sono supportati solo gli operatori *eq*, *ne*, *like*, *regexp* e *iregexp*.
- Con *bitand* come operator, il quarto parametro `pattern` può essere specificato come due numeri, separati da '/': **number_to_compare_with/mask**. countunique() calcola l'"AND bit a bit" dal valore e dalla *mask* e confronta il risultato con *number_to_compare_with*. Se il risultato dell'"AND bit a bit" è uguale a *number_to_compare_with*, il valore viene conteggiato.
Se *number_to_compare_with* e *mask* sono uguali, è necessario specificare solo la *mask* (senza '/').
- Con *regexp* o *iregexp* come operator, il quarto parametro `pattern` può essere una normale espressione regolare o una globale (che inizia con '@'). Nel caso delle espressioni regolari globali, la sensibilità alle maiuscole/minuscole viene ereditata dalle impostazioni dell'espressione regolare globale. Ai fini della corrispondenza regexp, i valori float saranno sempre rappresentati con 4 cifre decimali dopo '.'. Si noti inoltre che per numeri grandi la differenza tra la rappresentazione decimale (memorizzata nel database) e binaria (usata dal server Zabbix) può influire sulla 4a cifra decimale.

Esempi:

```
countunique(/host/key,10m) #il numero di valori unici negli ultimi 10 minuti fino a ora
countunique(/host/key,10m,"like","error") #il numero di valori unici negli ultimi 10 minuti fino a ora che
countunique(/host/key,10m,,12) #il numero di valori unici negli ultimi 10 minuti fino a ora uguali a '12'
countunique(/host/key,10m,"gt",12) #il numero di valori unici negli ultimi 10 minuti fino a ora maggiori d
countunique(/host/key,#10,"gt",12) #il numero di valori unici negli ultimi 10 valori fino a ora maggiori d
countunique(/host/key,10m:now-1d,"gt",12) #il numero di valori unici tra 24 ore e 10 minuti e 24 ore fa ri
countunique(/host/key,10m,"bitand","6/7") #il numero di valori unici negli ultimi 10 minuti fino a ora che
countunique(/host/key,10m:now-1d) #il numero di valori unici tra 24 ore e 10 minuti e 24 ore fa rispetto a
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Trova una corrispondenza di valore entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.
 Restituisce: 1 - trovato; 0 - altrimenti.

Parametri:

- Vedi parametri comuni
- **sec** o **#num** (opzionale) - per impostazione predefinita viene usato l'ultimo valore se non specificato
- **operator** (deve essere tra virgolette doppie) Operatori supportati:
eq - uguale (predefinito per integer, float)
ne - diverso
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like (predefinito per string, text, log) - corrisponde se contiene la stringa indicata in `pattern` (sensibile alle maiuscole/minuscole)
bitand - AND bit a bit
regexp - corrispondenza sensibile alle maiuscole/minuscole con l'espressione regolare indicata in `pattern`
iregexp - corrispondenza non sensibile alle maiuscole/minuscole con l'espressione regolare indicata in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra virgolette doppie); espressione regolare Perl Compatible Regular Expression (PCRE) se `operator` è *regexp*, *iregexp*

Commenti:

- Se viene elaborato più di un valore, viene restituito '1' se esiste almeno un valore corrispondente.
- *like* non è supportato come operatore per valori integer.
- *like* e *bitand* non sono supportati come operatori per valori float.
- Per valori string, text e log sono supportati solo gli operatori *eq*, *ne*, *like*, *regexp* e *iregexp*.
- Con *regexp* o *iregexp* come operatore, il quarto parametro `pattern` può essere un'espressione regolare ordinaria o globale (che inizia con '@'). In caso di espressioni regolari globali, la sensibilità alle maiuscole/minuscole viene ereditata dalle impostazioni dell'espressione regolare globale.

Esempi:

```
find(/host/key,10m,"like","error") #trova un valore che contiene 'error' negli ultimi 10 minuti fino a ora
find(/host/agent.version,,"like","beta")=1 #Zabbix agent ha una versione beta, deve essere aggiornato
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #è stato trovato un errore interno del web
```

first(/host/key,sec<:time shift>)

Il primo valore (il più vecchio) entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere parametri comuni

Vedere anche last().

Esempio:

```
first(/host/key,1h) #recupera il valore più vecchio nell'ultima ora fino ad ora
```

firstclock(/host/key,sec<:time shift>)

Il timestamp del valore più vecchio all'interno del periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere parametri comuni.

La funzione restituisce un errore se non sono stati raccolti dati nel periodo specificato.

Vedere anche lastclock().

Esempi:

```
firstclock(/host/key,1h) #recupera il timestamp del valore più vecchio nell'ultima ora
firstclock(/host/key,1h:now-24h) #recupera il timestamp del valore più vecchio nell'ultima ora di un giorn
```

fuzzytime(/host/key,sec)

Verifica di quanto il tempo dell'agent passivo differisce dal tempo del server/proxy Zabbix.
 Tipi di valore supportati: *Float*, *Integer*.
 Restituisce: 1 - la differenza tra il valore dell'item passivo (come timestamp) e il timestamp del server/proxy Zabbix (l'orario di raccolta del valore) è minore o uguale a *sec* secondi; 0 - altrimenti.

Parametri:

- Vedi parametri comuni.

Commenti:

- In genere usato con l'item 'system.localtime' per verificare che l'ora locale sia sincronizzata con l'ora locale del server Zabbix. *Nota* che 'system.localtime' deve essere configurato come controllo passivo per Zabbix agent; su Zabbix agent 2 può essere configurato come controllo attivo.
- Può essere usato anche con la chiave `vfs.file.time[/path/file,modify]` per verificare che il file non riceva aggiornamenti da molto tempo.
- Questa funzione non è consigliata per l'uso in espressioni trigger complesse (con più item coinvolti), perché può causare risultati imprevisti (la differenza di tempo verrà misurata con la metrica più recente), ad esempio in `fuzzytime(/Host/system.localtim or last(/Host/trap)<>0`.

Esempi:

```
fuzzytime(/host/system.localtime,5m)=0 #l'ora locale del client differisce dal tempo del server/proxy Zabb
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #l'ora locale del client d
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

Restituisce un array di risultati delle query JSONPath.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- Vedere i parametri comuni
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere racchiusi tra virgolette doppie)

Commenti:

- Questa funzione viene utilizzata con le funzioni aggregate.
- Una query JSONPath viene eseguita su ogni valore storico selezionato; ogni risultato della query riuscita e non vuoto viene aggiunto all'array. Se il risultato di una query per un singolo valore storico contiene più corrispondenze, ogni corrispondenza viene aggiunta all'array separatamente.
- La funzione restituisce un errore se:
  - Il formato di `pattern` non è valido.

    - Uno qualsiasi dei valori storici contiene JSON non valido oppure la query JSONPath non riesce per quel valore.
- La funzione restituisce un array vuoto se:
    - Non è stato selezionato alcun valore storico.
    - Tutti i valori storici selezionati contengono JSON non valido.

Esempio:

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

last(/host/key,<#num<:time shift»)

Il valore più recente.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere parametri comuni
- **#num** (opzionale) - l'N-esimo valore più recente

Commenti:

- Si noti che un periodo di tempo con cancelletto (#N) qui funziona in modo diverso rispetto a molte altre funzioni. Ad esempio: `last(/host/key)` è sempre uguale a `last(/host/key,#1)`; `last(/host/key,#3)` - il terzo valore più recente (*non* gli ultimi tre valori).
- Zabbix non garantisce l'ordine esatto dei valori se nella cronologia esistono più di due valori entro un secondo.
- Vedere anche first().

Esempi:

```
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»)

Il timestamp dell'N-esimo valore più recente entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere i parametri comuni;
- **#num** (opzionale) - l'N-esimo valore più recente.

La funzione restituisce un errore se non sono stati raccolti dati nel periodo specificato o se non è stato raccolto alcun valore N-esimo.

Vedere anche firstclock().

Esempi:

```
lastclock(/host/key) #recupera il timestamp del valore più recente
lastclock(/host/key,#2) #recupera il timestamp del 2° valore più recente
now()-lastclock(/host/heartbeat.item)>300 #sono trascorsi più di 5 minuti dall'aggiornamento dell'item hea
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 #l'intervallo
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #il v
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

Verifica se l'ID evento dell'ultima voce di log corrisponde a un'espressione regolare.
 Tipi di valore supportati: *Log*.
 Restituisce: 0 - non corrisponde; 1 - corrisponde.

Parametri:

- Vedi parametri comuni
- **#num** (opzionale) - l'N-esimo valore più recente
- **pattern** (opzionale) - l'espressione regolare che descrive il pattern richiesto, in stile Perl Compatible Regular Expression (PCRE) (gli argomenti stringa devono essere racchiusi tra virgolette doppie)

Esempi:

```
logeventid(/host/eventlog[Security],,"^4625$")=1 #trovata una voce di log con ID corrispondente a "4625" (
logeventid(/host/eventlog[System],,"^(6008|41)$")=1 #trovata una voce di log con ID corrispondente a "6008
```

logseverity(/host/key,<#num<:time shift»)

Gravità del log dell'ultima voce di log.
 Tipi di valore supportati: *Log*.
 Restituisce: 0 - gravità predefinita; N - gravità (intero, utile per i log eventi di Windows: 1 - Informazione, 2 - Avviso, 4 - Errore, 7 - Controllo errori, 8 - Controllo riuscito, 9 - Critico, 10 - Dettagliato).

Parametri:

- Vedere i parametri comuni
- **#num** (opzionale) - l'N-esimo valore più recente

Zabbix ricava la gravità del log dal campo **Information** del log eventi di Windows.

Esempi:

```
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>)

Verifica se la sorgente del log dell'ultima voce di log corrisponde a un'espressione regolare.
 Tipi di valore supportati: *Log*.
 Restituisce: 0 - non corrisponde; 1 - corrisponde.

Parametri:

- Vedi parametri comuni
- **#num** (opzionale) - l'N-esimo valore più recente
- **pattern** (opzionale) - l'espressione regolare che descrive il pattern richiesto, in stile Perl Compatible Regular Expression (PCRE) (gli argomenti stringa devono essere racchiusi tra virgolette doppie)

Normalmente usato per i log eventi di Windows.

Esempi:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #trovata una voce di log con sorgente corrisponden
logsource(/host/eventlog[System],,"Service Control Manager")=1 #trovata una voce di log con sorgente corri
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^70
```

logtimestamp(/host/key,<#num<:time shift»)

Il timestamp del messaggio di log dell'N-esimo valore più recente di un item di log.
 Tipi di valore supportati: *Log*.

Parametri:

- Vedi parametri comuni;
- **#num** (opzionale) - l'N-esimo valore più recente.

Commenti:

- Il calcolo dello spostamento temporale si basa sull'orario del valore dell'item, non sul timestamp del messaggio di log;
- La funzione restituisce un errore se:
  - viene ricevuto un item di tipo non log;
  - non sono stati raccolti dati nel periodo specificato;
  - non è stato raccolto l'N-esimo valore;
  - il messaggio di log non contiene un valore di timestamp.

Esempi:

```
logtimestamp(/host/key) #recupera il timestamp dell'ultimo messaggio di log
logtimestamp(/host/key,#2) #recupera il timestamp del 2° messaggio di log più recente
logtimestamp(/host/key,#3:now-1d) #recupera il timestamp del 3° messaggio di log più recente di un giorno
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Controlla se si è verificata una diminuzione monotona dei valori.
 Tipi di valore supportati: *Integer*.
 Restituisce: 1 - se tutti gli elementi nel periodo di tempo diminuiscono continuamente; 0 - altrimenti.

Parametri:

- Vedi parametri comuni
- **mode** (deve essere tra virgolette doppie) - *weak* (ogni valore è minore o uguale al precedente; predefinito) oppure *strict* (ogni valore è diminuito)

Esempi:

```
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 #il numero di processi nginx è diminuito monotonamente nell'
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Verifica se si è verificato un aumento monotono dei valori.
 Tipi di valore supportati: *Integer*.
 Restituisce: 1 - se tutti gli elementi nel periodo di tempo aumentano continuamente; 0 - altrimenti.

Parametri:

- Vedere i parametri comuni
- **mode** (deve essere racchiuso tra doppi apici) - *weak* (ogni valore è maggiore o uguale al precedente; predefinito) oppure *strict* (ogni valore è aumentato)

Esempi:

```
monoinc(/host/system.localtime,#3,"strict")=0 #l'ora locale del sistema non è aumentata in modo costante
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #trigger se il numero di file è rimasto invariat
```

nodata(/host/key,sec,<mode>)

Verifica l'assenza di dati ricevuti.
 Tipi di valore supportati: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Restituisce: 1 - se non sono stati ricevuti dati durante il periodo di tempo definito; 0 - altrimenti.

Parametri:

- Vedere i parametri comuni
- **sec** - il periodo non deve essere inferiore a 30 secondi, poiché il processo history syncer calcola questa funzione solo ogni 30 secondi; `nodata(/host/key,0)` non è consentito
- **mode** - se impostato su *strict* (tra virgolette doppie), questa funzione non sarà sensibile alla disponibilità del proxy (vedere i commenti per i dettagli)

Commenti:

- i trigger 'nodata' monitorati dal proxy sono, per impostazione predefinita, sensibili alla disponibilità del proxy: se il proxy diventa non disponibile, i trigger 'nodata' non si attiveranno immediatamente dopo il ripristino della connessione, ma ignoreranno i dati relativi al periodo di ritardo. Nota: per i proxy passivi, la soppressione viene attivata se la connessione viene ripristinata dopo più di 15 secondi e non meno di 2 secondi. Per i proxy attivi, la soppressione viene attivata se la connessione viene ripristinata dopo più di 15 secondi. Per disattivare la sensibilità alla disponibilità del proxy, utilizzare il terzo parametro, ad esempio: `nodata(/host/key,5m,"strict")`; in questo caso la funzione si attiverà non appena sarà trascorso il periodo di valutazione (cinque minuti) senza dati.
- Questa funzione visualizzerà un errore se, durante il periodo del primo parametro:
- non ci sono dati e il server Zabbix è stato riavviato
- non ci sono dati e la manutenzione è stata completata
- non ci sono dati e l'item è stato aggiunto o riabilitato.
- Gli errori vengono visualizzati nella colonna *Info* dell'elenco dei trigger.
- Questa funzione potrebbe non funzionare correttamente se ci sono differenze di orario tra server Zabbix, proxy e agent. Vedere anche: Requisito di sincronizzazione dell'ora.

Esempio:

```
nodata(/host/agent.ping,5m)=1 #trigger se non sono stati ricevuti dati dall'agent Zabbix per 5 minuti
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

Il percentile P-esimo di un periodo, dove P (percentage) è specificato dal terzo parametro.
 Tipi di valore supportati: *Float*, *Integer*.

Parametri:

- Vedere i parametri comuni
- **percentage** - un numero in virgola mobile compreso tra 0 e 100 (inclusi) con fino a 4 cifre dopo il separatore decimale

Esempi:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #il 95° percentile dell'input di rete (byte/sec) nel
percentile(/host/system.cpu.util,5m,95)>80 #il 95° percentile della percentuale di utilizzo CPU in user-ti
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #la maggior parte delle misurazioni di latenza è in
percentile(/host/net.if.in[eth0,bytes],1h,50) #calcola il 50° percentile (il valore mediano) del traffico
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #calcola i
```

rate(/host/key,sec<:time shift>)

L'aumento medio al secondo di un contatore monotonicamente crescente entro il periodo di tempo definito.
 Tipi di valore supportati: *Float*, *Integer*.

Parametri:

- Vedere parametri comuni

Corrisponde funzionalmente a 'rate' di PromQL.

Esempi:

```
rate(/host/key,30s) #se l'aumento monotono in 30 secondi è 20, questa funzione restituisce 0.67.
rate(/host/net.if.in[eth0,bytes],5m)>500000 #la velocità del traffico in ingresso dell'interfaccia su eth0
rate(/host/app.requests.count,1m)>100 #il contatore delle richieste è aumentato di oltre 100 richieste al
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

Restituisce un array di risultati delle query XML XPath.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- Vedere i parametri comuni
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere racchiusi tra virgolette doppie)

Commenti:

- Questa funzione viene utilizzata con le funzioni aggregate.
- Una query XML XPath viene eseguita su ogni valore storico selezionato; ogni risultato non vuoto di una query eseguita correttamente viene aggiunto all'array. Se il risultato di una query per un singolo valore storico contiene più corrispondenze, ogni corrispondenza viene aggiunta all'array separatamente.
- La funzione restituisce un errore se:
  - Il formato di `pattern` non è valido.
  - Zabbix è stato compilato senza il supporto XML.
  - Uno qualsiasi dei valori storici contiene XML non valido oppure la query XPath non riesce per quel valore.
- La funzione restituisce un array vuoto se:
  - Non è stato selezionato alcun valore storico.
  - Tutti i valori storici selezionati contengono XML non valido.

Esempio:

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

Vedere tutte le funzioni supportate.

Funzioni trend

Le funzioni trend, a differenza delle funzioni history, utilizzano i dati trend per i calcoli.

I trend memorizzano valori aggregati orari. Le funzioni trend utilizzano queste medie orarie e sono quindi utili per analisi a lungo termine.

I risultati delle funzioni trend vengono memorizzati nella cache, quindi più chiamate alla stessa funzione con gli stessi parametri recuperano le informazioni dal database una sola volta. La cache delle funzioni trend è controllata dal parametro server TrendFunctionCacheSize.

Usa `zabbix[tcache,cache,paccessed]` per stimare quanto della cache è effettivamente utilizzato. Un valore basso di `paccessed`, insieme a valori bassi di `pitems` e `pmisses`, di solito indica che la dimensione della cache può essere ridotta.

I trigger che fanno riferimento **solo** a funzioni trend vengono valutati una volta per il più piccolo intervallo di tempo nell'espressione. Ad esempio, un trigger come

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

verrà valutato una volta al giorno. Se il trigger contiene sia funzioni trend sia history (o date and time e/o nodata()), viene calcolato in conformità ai principi usuali.

Tutte le funzioni elencate qui sono supportate in:

- Espressioni trigger
- Item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fai clic sulla funzione per vedere i dettagli completi.

Function	Description
baselinedev	Restituisce il numero di deviazioni (con algoritmo stddevpop) tra l'ultimo periodo di dati e gli stessi periodi di dati nelle stagioni precedenti.
baselinewma	Calcola la baseline mediando i dati dello stesso intervallo di tempo in più periodi di tempo uguali ('stagioni') usando l'algoritmo della media mobile ponderata.
trendavg	La media dei valori trend all'interno del periodo di tempo definito.
trendcount	Il numero di valori history recuperati con successo e utilizzati per calcolare il valore trend all'interno del periodo di tempo definito.
trendmax	Il valore massimo tra i valori trend all'interno del periodo di tempo definito.
trendmin	Il valore minimo tra i valori trend all'interno del periodo di tempo definito.
trendstl	Restituisce il tasso di anomalie durante il periodo di rilevamento - un valore decimale compreso tra 0 e 1 che è `((the number of anomaly values)/(total number of values))`.
trendsum	La somma dei valori trend all'interno del periodo di tempo definito.

Parametri comuni

- `/host/key` è un primo parametro obbligatorio comune
- `periodo di tempo:spostamento temporale` è un secondo parametro comune, dove:
  - **periodo di tempo** - il periodo di tempo (minimo '1h'), definito come <N><unità di tempo> dove `N` - il numero di unità di tempo, `unità di tempo` - h (ora), d (giorno), w (settimana), M (mese) o y (anno).
  - **spostamento temporale** - l'offset del periodo di tempo (vedere gli esempi di funzione)

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- I parametri opzionali della funzione (o parti di parametro) sono indicati da < >
- I parametri specifici della funzione sono descritti insieme a ciascuna funzione
- I parametri `/host/key` e `time period:time shift` non devono mai essere racchiusi tra virgolette

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Restituisce il numero di deviazioni (tramite l'algoritmo stddevpop) tra l'ultimo periodo di dati e gli stessi periodi di dati nelle stagioni precedenti.

Parametri:

- Vedere i parametri comuni
- **data period** - il periodo di raccolta dei dati all'interno di una stagione, definito come <N><time unit> dove:
`N` - il numero di unità di tempo
`time unit` - h (ora), d (giorno), w (settimana), M (mese) oppure y (anno), deve essere uguale o inferiore a season
- **season unit** - l'unità di calendario che definisce una stagione (h, d, w, M, y); non può essere più piccola di data period
- **num seasons** - il numero di stagioni da valutare

Esempi:

```
baselinedev(/host/key,1d:now/d,"M",6) #calcola il numero di deviazioni standard (popolazione) tra il giorn
baselinedev(/host/key,1h:now/h,"d",10)  #calcola il numero di deviazioni standard della popolazione tra l'
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Calcola la baseline calcolando la media dei dati dello stesso intervallo temporale in più periodi di tempo uguali (''stagioni'') utilizzando l'algoritmo della media mobile pesata.

Parametri:

- Vedi parametri comuni
- **data period** - il periodo di raccolta dei dati all'interno di una stagione, definito come <N><unità di tempo> dove:
`N` - il numero di unità di tempo
`unità di tempo` - h (ora), d (giorno), w (settimana), M (mese) o y (anno), deve essere uguale o inferiore alla stagione
Time shift - l'offset del periodo di tempo, definisce la fine dell'intervallo temporale di raccolta dei dati in stagioni (vedi esempi)
- **season unit** - l'unità di calendario che definisce una stagione (h, d, w, M, y); non può essere più piccola di data period
- **num seasons** - il numero di stagioni da valutare

Esempi:

```
baselinewma(/host/key,1h:now/h,"d",3) #calcolo della baseline sulla stessa ora del giorno negli ultimi 3 g
baselinewma(/host/key,2h:now/h,"d",3) #calcolo della baseline sullo stesso intervallo di due ore negli ult
baselinewma(/host/key,1d:now/d,"M",4) #calcolo della baseline sullo stesso giorno del mese di 'ieri' nei 4
```

trendavg(/host/key,time period:time shift)

La media dei valori di trend nel periodo di tempo definito.

Parametri:

- Vedi parametri comuni

Esempi:

```
trendavg(/host/key,1h:now/h) #la media per l'ora precedente (ad es. 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #la media per due ore fa (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #la media per tre ore fa (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #la media per il mese precedente di un anno fa
```

trendcount(/host/key,time period:time shift)

Il numero di valori della cronologia recuperati con successo utilizzati per calcolare il valore di trend nel periodo di tempo definito.

Parametri:

- Vedere i parametri comuni

Esempi:

```
trendcount(/host/key,1h:now/h) #il conteggio dei valori per l'ora precedente (ad es. 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #il conteggio dei valori per due ore fa (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #il conteggio dei valori per tre ore fa (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #il conteggio dei valori per il mese precedente di un anno fa
```

trendmax(/host/key,time period:time shift)

Il massimo nei valori di trend all'interno del periodo di tempo definito.

Parametri:

- Vedere parametri comuni

Esempi:

```
trendmax(/host/key,1h:now/h) #il massimo per l'ora precedente (ad es. 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calcola la differenza tra i valori massimo e
trendmax(/host/key,1h:now/h-1h) #il massimo per due ore fa (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #il massimo per tre ore fa (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #il massimo per il mese precedente di un anno fa
```

trendmin(/host/key,time period:time shift)

Il minimo nei valori di trend entro il periodo di tempo definito.

Parametri:

- Vedere i parametri comuni

Esempi:

```
trendmin(/host/key,1h:now/h) #il minimo per l'ora precedente (ad es. 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calcola la differenza tra i valori massimo e
trendmin(/host/key,1h:now/h-1h) #il minimo per due ore fa (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #il minimo per tre ore fa (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #il minimo per il mese precedente di un anno fa
```

trendstl(/host/key,periodo di valutazione:spostamento temporale,periodo di rilevamento,stagione,<deviazioni>,<algdev>,<finestra s>)

Restituisce il tasso di anomalie durante il periodo di rilevamento, un valore decimale compreso tra 0 e 1 che è `((il numero di valori anomali)/(numero totale di valori))`.

Parametri:

- Vedere i parametri comuni

- **periodo di valutazione** - il periodo di tempo che deve essere decomposto (minimo '1h'), definito come <N><unità di tempo> dove
`N` - il numero di unità di tempo
`unità di tempo` - h (ora), d (giorno), w (settimana), M (mese) o y (anno)
- **periodo di rilevamento** - il periodo di tempo prima della fine del periodo di valutazione per il quale vengono calcolate le anomalie (minimo '1h', non può essere più lungo del periodo di valutazione), definito come <N><unità di tempo> dove
`N` - il numero di unità di tempo
`unità di tempo` - h (ora), d (giorno), w (settimana)
- **stagione** - il periodo di tempo più breve in cui è previsto un pattern ripetitivo ("stagione") (minimo '2h', non può essere più lungo del periodo di valutazione, il numero di voci nel periodo di valutazione deve essere maggiore di due volte la frequenza risultante (stagione/h)), definito come <N><unità di tempo> dove
`N` - il numero di unità di tempo
`unità di tempo` - h (ora), d (giorno), w (settimana)
- **deviazioni** - il numero di deviazioni (calcolate da algdev) da considerare come anomalia (può essere decimale), (deve essere maggiore o uguale a 1, il valore predefinito è 3)
- **algdev** (deve essere tra virgolette doppie) - l'algoritmo di deviazione, può essere *stddevpop*, *stddevsamp* o *mad* (predefinito)
- **finestra s** - l'ampiezza (in ritardi) della finestra loess per l'estrazione stagionale (il valore predefinito è 10 * numero di voci nel periodo di valutazione + 1)

Esempi:

```
trendstl(/host/key,100h:now/h,10h,2h) #analizza le ultime 100 ore di dati di trend, trova il tasso di anom
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #analizza il periodo di 100 ore di dati di trend, fin
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #analizza 100 giorni di dati di trend fino a un giorno fa,
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #analizza il mese precedente di un anno fa, trova il t
```

trendsum(/host/key,time period:time shift)

La somma dei valori di trend all'interno del periodo di tempo definito.

Parametri:

- Vedere i parametri comuni

Esempi:

```
trendsum(/host/key,1h:now/h) #la somma per l'ora precedente (ad es. 12:00-13:00)
trendsum(/host/key,1h:now/h-1h) #la somma per due ore fa (11:00-12:00)
trendsum(/host/key,1h:now/h-2h) #la somma per tre ore fa (10:00-11:00)
trendsum(/host/key,1M:now/M-1y) #la somma per il mese precedente di un anno fa
```

Vedere tutte le funzioni supportate.

Funzioni matematiche

Tutte le funzioni elencate qui sono supportate in:

- Espressioni trigger
- Formule di item calcolati
- Macro di espressione

Le funzioni matematiche sono supportate con tipi di valore float e integer, salvo diversa indicazione.

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare i dettagli completi.

Function	Description
abs	Il valore assoluto di un valore.
acos	L'arccoseno di un valore come angolo, espresso in radianti.
asin	L'arcoseno di un valore come angolo, espresso in radianti.
atan	L'arcotangente di un valore come angolo, espresso in radianti.
atan2	L'arcotangente delle coordinate ordinate (valore) e ascissa specificate come angolo, espresso in radianti.
avg	Il valore medio dei valori dell'item referenziato.
cbrt	La radice cubica di un valore.
ceil	Arrotonda il valore per eccesso all'intero maggiore o uguale più vicino.
cos	Il coseno di un valore, dove il valore è un angolo espresso in radianti.
cosh	Il coseno iperbolico di un valore.
cot	La cotangente di un valore, dove il valore è un angolo espresso in radianti.
degrees	Converte un valore da radianti a gradi.
e	Il numero di Eulero (2.718281828459045).

Function	Description
exp	Il numero di Eulero elevato alla potenza di un valore.
expm1	Il numero di Eulero elevato alla potenza di un valore meno 1.
floor	Arrotonda il valore per difetto all'intero minore o uguale più vicino.
log	Il logaritmo naturale.
log10	Il logaritmo decimale.
max	Il valore più alto dei valori dell'item referenziato.
min	Il valore più basso dei valori dell'item referenziato.
mod	Il resto della divisione.
pi	La costante Pi (3.14159265358979).
power	La potenza di un valore.
radians	Converte un valore da gradi a radianti.
rand	Restituisce un valore intero casuale.
round	Arrotonda il valore a un numero di cifre decimali.
signum	Restituisce '-1' se un valore è negativo, '0' se un valore è zero, '1' se un valore è positivo.
sin	Il seno di un valore, dove il valore è un angolo espresso in radianti.
sinh	Il seno iperbolico di un valore, dove il valore è un angolo espresso in radianti.
sqrt	La radice quadrata di un valore.
sum	La somma dei valori dell'item referenziato.
tan	La tangente di un valore.
truncate	Tronca il valore a un numero di cifre decimali.

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- Le espressioni sono accettate come parametri
- I parametri opzionali della funzione (o parti di parametro) sono indicati da < >

abs(value)

Il valore assoluto (a partire da 0) di un valore.

Parametro:

- **value** - il valore da controllare

Ad esempio, il valore assoluto di '3' o '-3' sarà '3'.

Esempio:

```
abs(last(/host/key))>10
```

acos(value)

L'arcocoseno di un valore come angolo, espresso in radianti.

Parametro:

- **value** - il valore da controllare

Il valore deve essere compreso tra -1 e 1. Ad esempio, l'arcocoseno del valore '0.5' sarà '2.0943951'.

Esempio:

```
acos(last(/host/key))
```

asin(value)

L'arcoseno di un valore come angolo, espresso in radianti.

Parametro:

- **value** - il valore da controllare

Il valore deve essere compreso tra -1 e 1. Ad esempio, l'arcoseno del valore '0.5' sarà '-0.523598776'.

Esempio:

```
asin(last(/host/key))
```

atan(value)

L'arcotangente di un valore come angolo, espressa in radianti.

Parametro:

- **value** - il valore da verificare

Ad esempio, l'arcotangente del valore '1' sarà '0.785398163'.

Esempio:

```
atan(last(/host/key))
```

atan2(value,abscissa)

L'arcotangente delle coordinate dell'ordinata (value) e dell'ascissa specificata come angolo, espressa in radianti.

Parametri:

- **value** - il valore da controllare;
- **abscissa** - il valore dell'ascissa.

Ad esempio, l'arcotangente delle coordinate dell'ordinata e dell'ascissa di un valore '1' sarà '2.21429744'.

Esempio:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

Il valore medio dei valori degli item referenziati.

Parametro:

- **valueX** - il valore restituito da un'altra funzione che lavora con la cronologia degli item.

Esempio:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

La radice cubica di un valore.

Parametro:

- **value** - il valore da controllare

Ad esempio, la radice cubica di '64' sarà '4', quella di '63' sarà '3.97905721'.

Esempio:

```
cbrt(last(/host/key))
```

ceil(value)

Arrotonda il valore per eccesso all'intero maggiore o uguale più vicino.

Parametro:

- **value** - il valore da verificare

Ad esempio, '2.4' verrà arrotondato per eccesso a '3'. Vedi anche floor().

Esempio:

```
ceil(last(/host/key))
```

cos(value)

Il coseno di un valore, dove il valore è un angolo espresso in radianti.

Parametro:

- **value** - il valore da controllare

Per esempio, il coseno del valore '1' sarà '0.54030230586'.

Esempio:

```
cos(last(/host/key))
```

cosh(value)

Il coseno iperbolico di un valore. Restituisce il valore come numero reale, non in notazione scientifica.

Parametro:

- **value** - il valore da controllare

Ad esempio, il coseno iperbolico del valore ’1’ sarà ’1.54308063482’.

Esempio:

```
cosh(last(/host/key))
```

cot(value)

La cotangente di un valore, dove il valore è un angolo espresso in radianti.

Parametro:

- **value** - il valore da controllare

Per esempio, la cotangente del valore ’1’ sarà ’0.54030230586’.

Esempio:

```
cot(last(/host/key))
```

degrees(value)

Converte un valore da radianti a gradi.

Parametro:

- **value** - il valore da controllare

Ad esempio, un valore ’1’ convertito in gradi sarà ’57.2957795’.

Esempio:

```
degrees(last(/host/key))
```

e

Il numero di Eulero (2.718281828459045).

Esempio:

```
e()
```

exp(value)

Il numero di Eulero elevato alla potenza di un valore.

Parametro:

- **value** - il valore da controllare

Per esempio, il numero di Eulero elevato alla potenza del valore ’2’ sarà ’7.38905609893065’.

Esempio:

```
exp(last(/host/key))
```

expm1(value)

Il numero di Eulero elevato alla potenza di un valore meno 1.

Parametro:

- **value** - il valore da controllare

Ad esempio, il numero di Eulero elevato alla potenza del valore ’2’ meno 1 sarà ’6.38905609893065’.

Esempio:

```
expm1(last(/host/key))
```

floor(value)

Arrotonda il valore per difetto all’intero più piccolo o uguale più vicino.

Parametro:

- **value** - il valore da controllare

Ad esempio, ’2.6’ verrà arrotondato per difetto a ’2’. Vedi anche ceil().

Esempio:

```
floor(last(/host/key))
```

log(value)

Il logaritmo naturale.

Parametro:

- **value** - il valore da controllare

Ad esempio, il logaritmo naturale di un valore ’2’ sarà ’0.69314718055994529’.

Esempio:

```
log(last(/host/key))
```

log10(value)

Il logaritmo decimale.

Parametro:

- **value** - il valore da controllare

Per esempio, il logaritmo decimale del valore ’5’ sarà ’0.69897000433’.

Esempio:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

Il valore più alto tra i valori dell’item referenziato.

Parametro:

- **valueX** - il valore restituito da un’altra funzione che lavora con la cronologia dell’item.

Esempio:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<value1>,<value2>,...)

Il valore più basso tra i valori degli item di riferimento.

Parametro:

- **valueX** - il valore restituito da un’altra funzione che lavora con la cronologia degli item.

Esempio:

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

Il resto della divisione.

Parametri:

- **value** - il valore da controllare;
- **denominator** - il denominatore della divisione.

Ad esempio, il resto della divisione del valore ’5’ con denominatore della divisione ’2’ sarà ’1’.

Esempio:

```
mod(last(/host/key),2)
```

pi

La costante Pi (3.14159265358979).

Esempio:

```
pi()
```

power(valore,valore della potenza)

La potenza di un valore.

Parametri:

- **valore** - il valore da controllare;
- **valore della potenza** - la potenza N-esima da utilizzare.

Ad esempio, la terza potenza di un valore ’2’ sarà ’8’.

Esempio:

```
power(last(/host/key),3)
```

radians(value)

Converte un valore da gradi a radianti.

Parametro:

- **value** - il valore da controllare

Ad esempio, un valore ’1’ convertito in radianti sarà ’0.0174532925’.

Esempio:

```
radians(last(/host/key))
```

rand

Restituisce un valore intero casuale. Un numero pseudo-casuale generato usando il tempo come seed (sufficiente per scopi matematici, ma non per la crittografia).

Esempio:

```
rand()
```

round(valore, posizioni decimali)

Arrotonda il valore al numero di posizioni decimali specificato.

Parametri:

- **valore** - il valore da controllare;
- **posizioni decimali** - specifica il numero di posizioni decimali per l’arrotondamento (è possibile anche 0).

Ad esempio, un valore ’2.5482’ arrotondato a 2 posizioni decimali diventerà ’2.55’.

Esempio:

```
round(last(/host/key),2)
```

signum(value)

Restituisce ’-1’ se un valore è negativo, ’0’ se un valore è zero, ’1’ se un valore è positivo.

Parametro:

- **value** - il valore da controllare.

Esempio:

```
signum(last(/host/key))
```

sin(value)

Il seno di un valore, dove il valore è un angolo espresso in radianti.

Parametro:

- **value** - il valore da controllare

Ad esempio, il seno del valore ’1’ sarà ’0.8414709848’.

Esempio:

```
sin(last(/host/key))
```

sinh(value)

Il seno iperbolico di un valore.

Parametro:

- **value** - il valore da controllare

Ad esempio, il seno iperbolico del valore '1' sarà '1.17520119364'.

Esempio:

```
sinh(last(/host/key))
```

sqrt(value)

La radice quadrata di un valore.
 Questa funzione non riuscirà con un valore negativo.

Parametro:

- **value** - il valore da controllare

Ad esempio, la radice quadrata di un valore '3.5' sarà '1.87082869339'.

Esempio:

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

La somma dei valori degli item referenziati.

Parametro:

- **valueX** - il valore restituito da un'altra funzione che lavora con la cronologia degli item.

Esempio:

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

La tangente di un valore.

Parametro:

- **value** - il valore da controllare

Ad esempio, la tangente del valore '1' sarà '1.55740772465'.

Esempio:

```
tan(last(/host/key))
```

truncate(valore, posizioni decimali)

Tronca il valore al numero di posizioni decimali specificato.

Parametri:

- **valore** - il valore da controllare;
- **posizioni decimali** - specifica il numero di posizioni decimali per il troncamento (è possibile anche 0).

Ad esempio, un valore '2.5482' troncato a 2 posizioni decimali diventerà '2.54'.

Esempio:

```
truncate(last(/host/key),2)
```

Vedere tutte le funzioni supportate.

Funzioni degli operatori

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Formule degli item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Funzione	Descrizione
between	Verifica se il valore appartiene all'intervallo specificato.
in	Verifica se il valore è uguale ad almeno uno dei valori elencati.

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- Le espressioni sono accettate come parametri

between(value,min,max)

Verifica se il valore appartiene all'intervallo specificato.
 Tipi di valore supportati: *Integer*, *Float*.
 Restituisce: 1 - nell'intervallo; 0 - altrimenti.

Parametri:

- **value** - il valore da verificare;
- **min** - il valore minimo;
- **max** - il valore massimo.

Esempio:

```
between(last(/host/key),1,10)=1 #trigger se il valore è compreso tra 1 e 10
```

in(value,value1,value2,...valueN)

Verifica se il valore è uguale ad almeno uno dei valori elencati.
 Tipi di valore supportati: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Restituisce: 1 - se uguale; 0 - altrimenti.

Parametri:

- **value** - il valore da verificare;
- **valueX** - valori elencati (i valori stringa devono essere racchiusi tra doppi apici).

Il valore viene confrontato con i valori elencati come numero, se tutti questi valori possono essere convertiti in formato numerico; altrimenti viene confrontato come stringa.

Esempio:

```
in(last(/host/key),5,10)=1 #trigger se l'ultimo valore è uguale a 5 o 10
in("text",last(/host/key),last(/host/key,#2))=1 #trigger se "text" è uguale a uno degli ultimi 2 valori
```

Vedere tutte le funzioni supportate.

Funzioni predittive

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Formule degli item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare i dettagli completi.

Function	Description
forecast	Il valore futuro, massimo, minimo, delta o medio dell'item.
timeleft	Il tempo, in secondi, necessario affinché un item raggiunga la soglia specificata.

Parametri comuni

- `/host/key` è un primo parametro comune obbligatorio per le funzioni che fanno riferimento alla cronologia dell'item del host.
- `(sec|#num)<:time shift>` è un secondo parametro comune per le funzioni che fanno riferimento alla cronologia dell'item del host, dove:
  - **sec** - periodo massimo di valutazione in secondi (è possibile utilizzare i suffissi temporali), oppure

- **#num** - intervallo massimo di valutazione nei valori raccolti più recenti (se preceduto da un cancelletto).
- **time shift** (facoltativo) - consente di spostare indietro nel tempo il punto di valutazione. Per ulteriori dettagli sulla specifica del time shift.

Dettagli della funzione

Alcune note generali sui parametri della funzione:

- I parametri della funzione sono separati da una virgola
- I parametri opzionali della funzione (o parti di parametro) sono indicati da < >
- I parametri specifici della funzione sono descritti con ciascuna funzione
- I parametri `/host/key` e `(sec|#num)<:time shift>` non devono mai essere racchiusi tra virgolette

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

Il valore futuro, massimo, minimo, delta o medio dell'item.
 Tipi di valore supportati: *Float*, *Integer*.

Parametri:

- Vedere i parametri comuni;
- **time** - l'orizzonte di previsione in secondi (ad es. 1, 60) (è possibile utilizzare i suffissi temporali (ad es. `30s`, `1m`)); sono supportati valori negativi;
- **fit** (facoltativo; deve essere racchiuso tra virgolette doppie) - la funzione utilizzata per adattare i dati storici. Adattamenti supportati:
linear - funzione lineare (predefinita)
polynomialN - polinomio di grado N (1 <= N <= 6)
exponential - funzione esponenziale
logarithmic - funzione logaritmica
power - funzione potenza
Nota: *polynomial1* equivale a *linear*;
- **mode** (facoltativo; deve essere racchiuso tra virgolette doppie) - l'output richiesto. Modalità supportate:
value - valore (predefinita)
max - massimo
min - minimo
delta - *max-min*
avg - media
Nota: *value* stima il valore dell'item al momento `now + time`; *max*, *min*, *delta* e *avg* analizzano la stima del valore dell'item nell'intervallo tra `now` e `now + time`.

Commenti:

- Se il valore da restituire è maggiore di 1.7976931348623158E+308 o minore di -1.7976931348623158E+308, il valore restituito viene limitato rispettivamente a 1.7976931348623158E+308 o -1.7976931348623158E+308;
- Diventa non supportato solo se utilizzato in modo errato nell'espressione (tipo di item errato, parametri non validi); in caso contrario, restituisce -1 in caso di errori;
- Vedere anche le informazioni aggiuntive sulle funzioni trigger predittive.

Esempi:

```
forecast(/host/key,#10,1h) #prevede il valore dell'item tra un'ora in base agli ultimi 10 valori
forecast(/host/key,1h,30m) #prevede il valore dell'item tra 30 minuti in base ai dati dell'ultima ora
forecast(/host/key,1h:now-1d,12h) #prevede il valore dell'item tra 12 ore in base ai dati di un'ora fa, un
forecast(/host/key,1h,10m,"exponential") #prevede il valore dell'item tra 10 minuti in base ai dati dell'u
forecast(/host/key,1h,2h,"polynomial3","max") #prevede il valore massimo che l'item può raggiungere nelle
forecast(/host/key,#2,-20m) #stima il valore dell'item di 20 minuti fa in base agli ultimi due valori (può
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Il tempo, espresso in secondi, necessario affinché un item raggiunga la soglia specificata.
 Tipi di valore supportati: *Float*, *Integer*.

Parametri:

- Vedere i parametri comuni;
- **threshold** - il valore da raggiungere (è possibile utilizzare i suffissi delle unità);
- **fit** (facoltativo; deve essere racchiuso tra virgolette doppie) - vedere forecast().

Commenti:

- Se il valore da restituire è maggiore di 1.7976931348623158E+308, il valore restituito viene limitato a 1.7976931348623158E+308;
- Restituisce 1.7976931348623158E+308 se non è possibile raggiungere la soglia;
- Diventa non supportato solo se utilizzato in modo errato nell'espressione (tipo di item errato, parametri non validi); in caso contrario, restituisce -1 in caso di errori;
- Vedere anche le informazioni aggiuntive sulle funzioni trigger predittive.

Esempi:

```
timeleft(/host/key,#10,0) #il tempo necessario affinché il valore dell'item raggiunga zero in base agli ul
timeleft(/host/key,1h,100) #il tempo necessario affinché il valore dell'item raggiunga 100 in base ai dati
timeleft(/host/key,1h:now-1d,100) #il tempo necessario affinché il valore dell'item raggiunga 100 in base
```

```
timeleft(/host/key,1h,200,"polynomial2") #il tempo necessario affinché il valore dell'item raggiunga 200 i
```

Vedere tutte le funzioni supportate.

Funzioni predittive dei trigger

Panoramica

A volte ci sono segnali di un problema imminente. Questi segnali possono essere individuati in modo da poter intraprendere azioni in anticipo per prevenire o almeno ridurre al minimo l'impatto del problema.

Zabbix dispone di strumenti per prevedere il comportamento futuro del sistema monitorato sulla base dei dati storici. Questi strumenti sono realizzati tramite funzioni predittive dei trigger.

Funzioni

Prima di impostare un trigger, è necessario definire che cosa sia uno stato di problema e quanto tempo sia necessario per intervenire. Esistono quindi due modi per configurare un trigger che segnali una potenziale situazione indesiderata. Primo: il trigger deve attivarsi quando è previsto che il sistema si trovi in uno stato di problema dopo il "tempo per intervenire". Secondo: il trigger deve attivarsi quando il sistema raggiungerà lo stato di problema in meno del "tempo per intervenire". Le corrispondenti funzioni di trigger da utilizzare sono **forecast** e **timeleft**. Si noti che l'analisi statistica sottostante è sostanzialmente identica per entrambe le funzioni. È possibile configurare un trigger nel modo che si preferisce, ottenendo risultati simili.

Parametri

Entrambe le funzioni utilizzano quasi lo stesso insieme di parametri. Utilizzare l'elenco delle funzioni supportate come riferimento.

Intervallo di tempo

Prima di tutto, è necessario specificare il periodo storico che Zabbix deve analizzare per elaborare la previsione. Questo si fa nel modo consueto, tramite il parametro `time period` e un eventuale spostamento temporale, come avviene con le funzioni **avg**, **count**, **delta**, **max**, **min** e **sum**.

Orizzonte di previsione

(solo **forecast**)
 Il parametro `time` specifica quanto lontano nel futuro Zabbix deve estrapolare le dipendenze che trova nei dati storici. Indipendentemente dal fatto che si utilizzi `time_shift` o meno, `time` viene sempre conteggiato a partire dal momento attuale.

Soglia da raggiungere

(solo **timeleft**)
 Il parametro `threshold` specifica un valore che l'item analizzato deve raggiungere, indipendentemente dal fatto che vi arrivi dall'alto o dal basso. Una volta determinata f(t) (vedere sotto), si deve risolvere l'equazione f(t) = `threshold` e restituire la radice più vicina al momento attuale e situata a destra di esso oppure 1.7976931348623158E+308 se non esiste alcuna radice di questo tipo.

Note:
Quando i valori dell'item si avvicinano alla soglia e poi la superano, **timeleft** presume che l'intersezione sia già nel passato e pertanto passa all'intersezione successiva con il livello `threshold`, se presente. La procedura consigliata è utilizzare le previsioni come complemento alla normale diagnostica dei problemi, non come sostituzione.[1]

Funzioni di adattamento

La funzione `fit` predefinita è quella *lineare*. Ma se il sistema monitorato è più complesso, hai a disposizione più opzioni tra cui scegliere.

`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	$x = a*t^b$

Modalità

(solo **forecast**)
 Ogni volta che una funzione trigger viene valutata, ottiene i dati dal periodo di storico specificato e adatta ai dati una funzione specificata. Quindi, se i dati sono leggermente diversi, la funzione adattata sarà leggermente diversa. Se

calcoliamo semplicemente il valore della funzione adattata in un momento specificato nel futuro, non saprai nulla su come ci si aspetta che l'item analizzato si comporti tra ora e quel momento nel futuro. Per alcune opzioni di `fit` (come *polynomial*) un semplice valore futuro può essere fuorviante.

mode	Risultato di **forecast**
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	media di f(t) (now <= t <= now + `time`) secondo la definizione

Dettagli

Per evitare calcoli con numeri enormi, consideriamo il timestamp del primo valore nel periodo specificato più 1 ns come un nuovo tempo zero corrente (il tempo epoch corrente è dell'ordine di 10^9, l'epoch al quadrato è 10^{18}, la precisione double è circa 10^{-16}). Viene aggiunto 1 ns per fornire tutti i valori di tempo positivi per gli adattamenti *logaritmici* e *di potenza*, che comportano il calcolo di log(t). Lo spostamento temporale non influisce sugli adattamenti *lineari*, *polinomiali* ed *esponenziali* (a parte rendere i calcoli più semplici e precisi), ma modifica la forma delle funzioni *logaritmiche* e *di potenza*.

Errori potenziali

Le funzioni restituiscono -1 nelle seguenti situazioni:

- il periodo di valutazione specificato non contiene dati;
- il risultato dell'operazione matematica non è definito[3];
- complicazioni numeriche (purtroppo, per alcuni insiemi di dati di input, l'intervallo e la precisione del formato in virgola mobile a doppia precisione diventano insufficienti)[4].

Note:
Non vengono segnalati avvisi o errori se l'adattamento scelto descrive male i dati forniti oppure se i dati sono semplicemente troppo pochi per una previsione accurata.

Esempi e gestione degli errori

Per ricevere un avviso quando stai per esaurire lo spazio libero su disco del tuo host, puoi usare un'espressione trigger come questa:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

Tuttavia, il codice di errore -1 può entrare in gioco e portare il trigger in uno stato di problema. In generale è una buona cosa, perché ricevi un avviso che le tue previsioni non funzionano correttamente e dovresti esaminarle più a fondo per capire il motivo. Ma a volte è un male, perché -1 può semplicemente significare che nell'ultima ora non sono stati ottenuti dati sullo spazio libero su disco dell'host. Se ricevi troppi avvisi di falsi positivi, considera l'uso di un'espressione trigger più complessa [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

La situazione è un po' più difficile con **forecast**. Prima di tutto, -1 può o meno portare il trigger in uno stato di problema a seconda del fatto che tu abbia un'espressione come `forecast(/host/item,(...))<...` oppure come `forecast(/host/item,(...))>...`

Inoltre, -1 può essere una previsione valida se è normale che il valore dell'item sia negativo. Ma la probabilità di questa situazione nel mondo reale è trascurabile (vedi come funziona l'operatore **=**). Quindi aggiungi `... or forecast(/host/item,(...))=-1` oppure `... and forecast(/host/item,(...))<>-1` se vuoi o non vuoi trattare rispettivamente -1 come un problema.

Note

[1] Ad esempio, un semplice trigger come `timeleft(/host/item,1h,X) < 1h` può entrare in stato di problema quando il valore dell'item si avvicina a X e poi improvvisamente riprendersi una volta raggiunto il valore X. Se il problema è che il valore dell'item è inferiore a X, usa: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Se il problema è che il valore dell'item è superiore a X usa: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] Il grado del polinomio può andare da 1 a 6, *polynomial1* è equivalente a *linear*. Tuttavia, usa polinomi di grado superiore con cautela. Se il periodo di valutazione contiene meno punti di quelli necessari per determinare i coefficienti del polinomio, il grado del polinomio verrà ridotto (ad esempio, viene richiesto *polynomial5*, ma ci sono solo 4 punti, quindi verrà adattato *polynomial3*).

[3] Ad esempio, l'adattamento di funzioni *exponential* o *power* comporta il calcolo di log() dei valori dell'item. Se i dati contengono zeri o numeri negativi, otterrai un errore poiché log() è definito solo per valori positivi.

[4] Per gli adattamenti *linear*, *exponential*, *logarithmic* e *power* tutti i calcoli necessari possono essere scritti esplicitamente. Per *polynomial* solo *value* può essere calcolato senza passaggi aggiuntivi. Il calcolo di *avg* comporta il calcolo della primitiva del

polinomio (analiticamente). Il calcolo di *max*, *min* e *delta* comporta il calcolo della derivata del polinomio (analiticamente) e la ricerca delle sue radici (numericamente). La risoluzione di f(t) = 0 comporta la ricerca delle radici del polinomio (numericamente).

[5] Ma in questo caso -1 può far sì che il trigger esca dallo stato di problema. Per essere completamente protetti usa: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free], or {TRIGGER.VALUE}=1)`

Funzioni sulle stringhe

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Formule degli item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Funzione	Descrizione
ascii	Il codice ASCII del carattere più a sinistra del valore.
bitlength	La lunghezza del valore in bit.
bytelength	La lunghezza del valore in byte.
char	Restituisce il carattere interpretando il valore come codice ASCII.
concat	La stringa risultante dalla concatenazione dei valori degli item referenziati o dei valori costanti.
contains	Restituisce 1 se il valore contiene la sottostringa specificata da `search value`, altrimenti 0.
insert	Inserisce i caratteri o gli spazi specificati nella stringa di caratteri, a partire dalla posizione specificata nella stringa.
jsonpath	Restituisce il risultato JSONPath.
left	Restituisce i caratteri più a sinistra del valore.
length	La lunghezza del valore in caratteri.
ltrim	Rimuove i caratteri specificati dall'inizio della stringa.
mid	Restituisce una sottostringa di N caratteri a partire dalla posizione del carattere specificata da 'start'. Questa funzione è un alias della funzione `substring()`.
repeat	Ripete una stringa.
replace	Trova il modello nel valore e lo sostituisce con la sostituzione.
right	Restituisce i caratteri più a destra del valore.
rtrim	Rimuove i caratteri specificati dalla fine della stringa.
substring	Restituisce una sottostringa di N caratteri a partire dalla posizione del carattere specificata da 'start'.
trim	Rimuove i caratteri specificati dall'inizio e dalla fine della stringa.
xmlxpath	Restituisce il risultato XML XPath.

Dettagli della funzione

Alcune note generali sui parametri delle funzioni:

- I parametri delle funzioni sono separati da una virgola
- Le espressioni sono accettate come parametri
- I parametri di tipo stringa devono essere racchiusi tra doppi apici; in caso contrario potrebbero essere interpretati in modo errato
- I parametri opzionali delle funzioni (o parti di parametro) sono indicati da < >

ascii(value)

Il codice ASCII del carattere più a sinistra del valore.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametro:

- **value** - il valore da controllare

Per esempio, un valore come 'Abc' restituirà '65' (codice ASCII di 'A').

Esempio:

```
ascii(last(/host/key))
```

bitlength(value)

La lunghezza di value in bit.
 Tipi di valore supportati: *String*, *Text*, *Log*, *Integer*.

Parametro:

- **value** - il valore da controllare

Esempio:

`bitlength(last(/host/key))`

bytelength(value)

La lunghezza di value in byte.
 Tipi di valore supportati: *String*, *Text*, *Log*, *Integer*.

Parametro:

- **value** - il valore da controllare

Esempio:

`bytelength(last(/host/key))`

char(value)

Restituisce il carattere interpretando il valore come codice ASCII.
 Tipi di valore supportati: *Integer*.

Parametro:

- **value** - il valore da controllare

Il valore deve essere compreso nell'intervallo 0-255. Per esempio, un valore come '65' (interpretato come codice ASCII) restituirà 'A'.

Esempio:

`char(last(/host/key))`

concat(<value1>,<value2>,...)

La stringa risultante dalla concatenazione dei valori degli item referenziati o di valori costanti.
 Tipi di valore supportati: *String*, *Text*, *Log*, *Float*, *Integer*.

Parametro:

- **valueX** - il valore restituito da una delle funzioni della cronologia o un valore costante (stringa, numero intero o in virgola mobile). Deve contenere almeno due parametri.

Ad esempio, un valore come 'Zab' concatenato a 'bix' (la stringa costante) restituirà 'Zabbix'.

Esempi:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15])
```

contains(value, search value)

Restituisce 1 se il valore contiene la sottostringa specificata da `search value`, altrimenti 0.| Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da verificare;
- **search value** - il valore da cercare.

Commenti:

- La ricerca distingue tra maiuscole e minuscole.
- Quando il valore è valido e `search value` è vuoto, la funzione restituisce 1.

Ad esempio, la funzione restituisce 1 se il valore contiene 'Zabbix'.

Esempio:

`contains(last(/host/key),"Zabbix")`

insert(value,start,length,replacement)

Inserisce i caratteri o gli spazi specificati nella stringa di caratteri a partire dalla posizione specificata nella stringa.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;

- **start** - posizione iniziale;
- **length** - posizioni da sostituire;
- **replacement** - stringa di sostituzione.

Ad esempio, un valore come 'Zabbbix' verrà sostituito da 'Zabbix' se 'bb' (posizione iniziale 3, posizioni da sostituire 2) viene sostituito con 'b'.

Esempio:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

Restituisce il risultato JSONPath.
 Tipi di valore supportati: *Stringa*, *Testo*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **path** - il percorso (deve essere tra virgolette);
- **default** - il valore di fallback opzionale se la query JSONPath non restituisce dati. Si noti che in caso di altri errori viene restituito un errore (ad es. "unsupported construct").

Esempio:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Restituisce i caratteri più a sinistra del valore.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **count** - il numero di caratteri da restituire.

Ad esempio, è possibile restituire 'Zab' da 'Zabbix' specificando 3 caratteri più a sinistra da restituire. Vedere anche right().

Esempio:

```
left(last(/host/key),3) #restituisce i tre caratteri più a sinistra
```

length(value)

La lunghezza di value in caratteri.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametro:

- **value** - il valore da controllare.

Esempi:

```
length(last(/host/key)) #la lunghezza dell'ultimo valore
length(last(/host/key,#3)) #la lunghezza del terzo valore più recente
length(last(/host/key,#1:now-1d)) #la lunghezza del valore più recente di un giorno fa
```

ltrim(value,<chars>)

Rimuove i caratteri specificati dall'inizio della stringa.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **chars** (opzionale) - specifica i caratteri da rimuovere.

Per impostazione predefinita, vengono rimossi gli spazi vuoti iniziali (se non vengono specificati caratteri opzionali). Vedi anche: rtrim(), trim().

Esempi:

```
ltrim(last(/host/key)) #rimuove gli spazi vuoti dall'inizio della stringa
ltrim(last(/host/key),"Z") #rimuove qualsiasi 'Z' dall'inizio della stringa
ltrim(last(/host/key)," Z") #rimuove qualsiasi spazio e 'Z' dall'inizio della stringa
```

mid(value,start,length)

Restituisce una sottostringa di N caratteri a partire dalla posizione del carattere specificata da 'start'.
Questa funzione è un alias della funzione `substring()`.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da verificare;
- **start** - posizione iniziale della sottostringa;
- **length** - posizioni da restituire nella sottostringa.

Ad esempio, è possibile restituire 'abbi' da un valore come 'Zabbix' se la posizione iniziale è 2 e le posizioni da restituire sono 4.

Esempio:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Ripete una stringa.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **count** - il numero di volte da ripetere.

Esempio:

```
repeat(last(/host/key),2) #ripete il valore due volte
```

replace(value,pattern,replacement)

Trova il pattern nel valore e sostituiscilo con replacement. Tutte le occorrenze del pattern verranno sostituite.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **pattern** - il pattern da trovare;
- **replacement** - la stringa con cui sostituire il pattern.

Esempio:

```
replace(last(/host/key),"ibb","abb") #sostituisce tutte le occorrenze di 'ibb' con 'abb'
```

right(value,count)

Restituisce i caratteri più a destra del valore.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **count** - il numero di caratteri da restituire.

Ad esempio, è possibile restituire 'bix' da 'Zabbix' specificando i 3 caratteri più a destra da restituire. Vedere anche left().

Esempio:

```
right(last(/host/key),3) #restituisce i tre caratteri più a destra
```

rtrim(value,<chars>)

Rimuove i caratteri specificati dalla fine della stringa.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **chars** (opzionale) - specifica i caratteri da rimuovere.

Per impostazione predefinita, gli spazi vuoti vengono rimossi da destra (se non vengono specificati caratteri opzionali). Vedi anche: ltrim(), trim().

Esempi:

```
rtrim(last(/host/key)) #rimuove gli spazi vuoti dalla fine della stringa
rtrim(last(/host/key),"x") #rimuove qualsiasi 'x' dalla fine della stringa
rtrim(last(/host/key),"x ") #rimuove qualsiasi 'x' e spazio dalla fine della stringa
```

substring(value,start,length)

Restituisce una sottostringa di N caratteri a partire dalla posizione del carattere specificata da "start".
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;

- **start** - posizione iniziale della sottostringa;
- **length** - numero di posizioni da restituire nella sottostringa.

Ad esempio, è possibile restituire ”abbi” da un valore come ”Zabbix” se la posizione iniziale è 2 e il numero di posizioni da restituire è 4.

Esempio:

```
substring(last(/host/key),2,4)="abbi"
```

trim(value,<chars>)

Rimuove i caratteri specificati dall’inizio e dalla fine della stringa.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **chars** (opzionale) - specifica i caratteri da rimuovere.

Per impostazione predefinita, gli spazi vuoti vengono rimossi da entrambi i lati (se non vengono specificati caratteri opzionali). Vedi anche: ltrim(), rtrim().

Esempi:

```
trim(last(/host/key)) #rimuove gli spazi vuoti dall'inizio e dalla fine della stringa
trim(last(/host/key),"_") #rimuove '_' dall'inizio e dalla fine della stringa
```

xmlxpath(value,path,<default>)

Restituisce il risultato XML XPath.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **path** - il percorso (deve essere tra virgolette);
- **default** - il valore di fallback opzionale se la query XML XPath restituisce un nodeset vuoto. Non verrà restituito se il risultato vuoto non è un nodeset (ad es. una stringa vuota). In caso di altri errori viene restituito un errore (ad es. ”invalid expression”).

Esempio:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

Vedere tutte le funzioni supportate.

Best practice per i trigger

I trigger sono uno strumento potente, ma possono anche generare rumore di alert indesiderato. Per vedere più segnali reali e meno rumore, segui questi suggerimenti:

1. Desensibilizza i trigger. Invece di generare un alert sull’ultimo valore (troppo alto/troppo basso), analizza la media su un periodo esteso usando funzioni come **avg**, **min** e **max**.
2. Valuta l’uso della funzione **percentile** (impostata al 95% o al 5%) nei trigger, se vuoi evitare alert su picchi e cali casuali.
3. Usa l’isteresi per evitare il flapping dei trigger, cioè i cambiamenti frequenti dello stato del trigger (OK ↔ Problem). È possibile definire una continuità per lo stato di problema aggiungendo un’espressione di recupero (una condizione separata per la risoluzione del problema).
4. Usa le dipendenze dei trigger per evitare alert non correlati alla causa principale.
5. Usa la gravità dei trigger per ricevere alert solo sui problemi più gravi.
6. Definisci finestre di manutenzione.

Isteresi

A volte è necessario un intervallo tra gli stati di problema e di ripristino, anziché una semplice soglia. Ad esempio, se vogliamo definire un trigger che segnali un problema quando la temperatura della sala server supera i 20°C e vogliamo che rimanga nello stato di problema finché la temperatura non scende sotto i 15°C, una semplice soglia del trigger a 20°C non sarà sufficiente.

Invece, dobbiamo prima definire un’espressione del trigger per l’evento di problema (temperatura sopra i 20°C). Poi dobbiamo definire una condizione di ripristino aggiuntiva (temperatura sotto i 15°C). Questo viene fatto definendo una *Recovery expression* durante la configurazione di un trigger.

In questo caso, il ripristino del problema avviene in due passaggi:

- Per prima cosa, l'espressione di problema (temperatura sopra i 20°C) deve essere valutata come FALSE
- In secondo luogo, l'espressione di ripristino (temperatura sotto i 15°C) deve essere valutata come TRUE

L'espressione di ripristino viene valutata solo dopo che l'evento di problema è stato risolto. Il solo fatto che l'espressione di ripristino sia TRUE non risolve un problema se l'espressione di problema è ancora TRUE!

Esempio

La temperatura nella sala server è troppo alta.

Espressione del problema:

```
last(/server/temp)>20
```

Espressione di ripristino:

```
last(/server/temp)<=15
```

> **Note:**
> Non è produttivo utilizzare la macro {TRIGGER.VALUE} in un'espressione di ripristino perché questa espressione viene valutata solo quando il trigger si trova nello stato "Problema". Di conseguenza, {TRIGGER.VALUE} verrà sempre risolta come "1" (che indica uno stato di "Problema") durante la valutazione dell'espressione.

Dipendenze dei trigger

Le dipendenze dei trigger possono essere utilizzate per evitare avvisi che non sono correlati alla causa principale.

Vedi tutte le best practice.

Panoramica

A volte la disponibilità di un host dipende da un altro. Un server che si trova dietro un router diventerà irraggiungibile se il router si arresta. Con i trigger configurati per entrambi, potreste ricevere notifiche relative a due host non disponibili, mentre l'unico responsabile era il router.

In questo caso può essere utile impostare una dipendenza tra host. Una volta configurata la dipendenza, le notifiche dei dipendenti possono essere trattenute e può essere inviata solo la notifica relativa al problema principale.

Sebbene Zabbix non supporti direttamente le dipendenze tra host, queste possono essere definite con un metodo diverso e più flessibile: le dipendenze tra trigger. Un trigger può dipendere da uno o più trigger.

Nel nostro semplice esempio, apriamo il modulo di configurazione del trigger del server e impostiamo la dipendenza dal trigger corrispondente del router. Con questa dipendenza, il trigger del server non cambierà stato finché il trigger da cui dipende si trova nello stato 'PROBLEM'; di conseguenza, non verranno eseguite azioni dipendenti e non verranno inviate notifiche.

Se sia il server sia il router non sono disponibili e viene impostata una dipendenza tra il trigger "server non disponibile" e il trigger "router non disponibile", Zabbix non eseguirà azioni per il trigger dipendente.

Finché il trigger principale si trova nello stato PROBLEM, i relativi trigger dipendenti potrebbero segnalare valori non attendibili. Pertanto, i trigger dipendenti non verranno rivalutati finché il trigger principale (il router nell'esempio precedente):

- non torna dallo stato 'PROBLEM' allo stato 'OK';
- non cambia il proprio stato da 'PROBLEM' a 'UNKNOWN';
- non viene chiuso manualmente, tramite correlazione o con l'aiuto delle funzioni date and time e/o nodata();
- non viene risolto dal valore di un item non coinvolto nel trigger dipendente;
- non viene disabilitato, oppure non viene disabilitato un item o un host con item disabilitati.

In tutti i casi indicati sopra, il trigger dipendente (il server) verrà rivalutato solo quando verrà ricevuta una nuova metrica per tale trigger. Ciò significa che il trigger dipendente potrebbe non essere aggiornato immediatamente.

Inoltre:

- Una dipendenza tra trigger può essere aggiunta dal trigger di qualsiasi host al trigger di qualsiasi altro host, purché ciò non comporti una dipendenza circolare.
- Una dipendenza tra trigger può essere aggiunta da un template a un altro. Se un trigger del template A dipende da un trigger del template B, il template A può essere collegato a un host (o a un altro template) solo insieme al template B, mentre il template B può essere collegato da solo a un host (o a un altro template).
- Una dipendenza tra trigger può essere aggiunta da un trigger di template a un trigger di host. In questo caso, collegando tale template a un host verrà creato un trigger dell'host che dipende dallo stesso trigger di template da cui dipendeva il trigger originale. Ciò consente, ad esempio, di avere un template in cui alcuni trigger dipendono dai trigger del router (host). Tutti gli host collegati a questo template dipenderanno da quel router specifico.

- Non è possibile aggiungere una dipendenza tra trigger da un trigger di host a un trigger di template.
- Una dipendenza tra trigger può essere aggiunta da un prototipo di trigger a un altro prototipo di trigger (all'interno della stessa regola di low-level discovery) o a un trigger reale. Un prototipo di trigger non può dipendere da un prototipo di trigger appartenente a una regola LLD diversa né da un trigger creato a partire da un prototipo di trigger. Un prototipo di trigger dell'host non può dipendere da un trigger di un template.

Configurazione

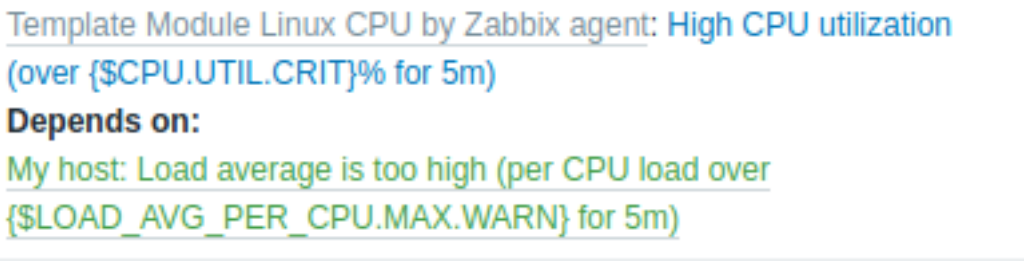
Per definire una dipendenza:

1. Aprire la scheda *Dipendenze* nel modulo di configurazione del trigger.
2. Fare clic su *Aggiungi* nella sezione *Dipendenze* e selezionare uno o più trigger da cui dipenderà il trigger.



3. Fare clic su *Aggiorna*.

A questo punto, nell'elenco il trigger indica la relativa dipendenza.



Esempio di più dipendenze

Ad esempio, l'host si trova dietro Router2 e Router2 si trova dietro Router1.

```
Zabbix - Router1 - Router2 - Host
```

Se Router1 è inattivo, allora ovviamente anche l'host e Router2 non sono raggiungibili, ma ricevere tre notifiche che indicano che l'host, Router1 e Router2 sono tutti inattivi è eccessivo.

Quindi, in questo caso definiamo due dipendenze:

```
il trigger 'Host is down' dipende dal trigger 'Router2 is down'
il trigger 'Router2 is down' dipende dal trigger 'Router1 is down'
```

Prima di modificare lo stato del trigger 'Host is down', Zabbix verificherà le dipendenze dei trigger corrispondenti. Se ne vengono trovate e uno di quei trigger è nello stato **Problem**, lo stato del trigger non verrà modificato, le azioni non verranno eseguite e non verrà inviata alcuna notifica.

Zabbix esegue questa verifica in modo ricorsivo. Se Router1 o Router2 non sono raggiungibili, il trigger dell'host non verrà aggiornato.

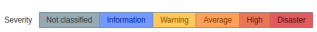
Gravità del trigger

La gravità del trigger può essere utilizzata per evitare rumore negli avvisi per problemi meno seri.

Vedi tutte le best practice.

Panoramica

La gravità rappresenta l'importanza relativa di un trigger.



L'assegnazione della gravità non è utile solo per la rappresentazione visiva, ma anche per evitare notifiche su problemi trascurabili.

Le gravità dei trigger vengono utilizzate per:

- Rappresentazione visiva (colore distinto per ogni problema creato, in base alla gravità originale del trigger)
- Limitazione delle notifiche in base alla gravità (vedi condizioni delle azioni)
- Supporti utente - supporti diversi (canali di notifica) per gravità diverse (ad esempio, SMS per trigger con gravità *Alta* e *Disastro*, ed Email per trigger con altre gravità)
- Audio negli allarmi globali - audio diversi per gravità diverse

Le seguenti gravità dei trigger sono supportate per impostazione predefinita:

Gravità	Colore	Descrizione
Non classificato	Grigio	Può essere utilizzata quando il livello di gravità di un evento è sconosciuto, non è stato determinato, non rientra nel normale ambito di monitoraggio, ecc.; ad esempio, durante la configurazione iniziale, come segnaposto per una valutazione futura o come parte di un processo di integrazione.
Informazione	Azzurro chiaro	Può essere utilizzata per eventi informativi che non richiedono attenzione immediata, ma che possono comunque fornire indicazioni utili.
Avviso	Giallo	Può essere utilizzata per indicare un potenziale problema che potrebbe richiedere indagine o intervento, ma che non è critico.
Media	Arancione	Può essere utilizzata per indicare un problema significativo che dovrebbe essere affrontato in tempi relativamente brevi per prevenire ulteriori problemi.
Alta	Rosso chiaro	Può essere utilizzata per indicare problemi critici che richiedono attenzione immediata per evitare interruzioni significative.
Disastro	Rosso	Può essere utilizzata per indicare un incidente grave che richiede un intervento immediato per prevenire, ad esempio, interruzioni del sistema o perdita di dati.

I nomi e i colori delle gravità dei trigger possono essere **personalizzati** in *Administration > General > Trigger displaying options*. I colori sono condivisi tra tutti i temi del frontend.

Traduzione dei nomi di gravità personalizzati

> **Attention:**
> Se vengono utilizzate le traduzioni del frontend di Zabbix, i nomi di gravità personalizzati sovrascriveranno per impostazione predefinita i nomi tradotti.

I nomi di gravità predefiniti dei trigger sono disponibili per la traduzione in tutte le lingue.
Se un nome di gravità viene modificato, il nome personalizzato viene utilizzato in tutte le lingue ed è necessaria un'ulteriore traduzione manuale.

Procedura di traduzione del nome di gravità personalizzato:

- impostare il nome di gravità personalizzato richiesto, ad esempio 'Important'
- modificare <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po
- aggiungere 2 righe:

```
msgid "Important"
msgstr "<stringa tradotta>"
```

e salvare il file.

- creare i file .mo come descritto in <frontend_dir>/locale/README

Qui **msgid** deve corrispondere al nuovo nome di gravità personalizzato e **msgstr** deve essere la relativa traduzione nella lingua specifica.

Questa procedura deve essere eseguita dopo ogni modifica del nome di gravità.

Visualizzazione

Panoramica

La visualizzazione offre diversi modi per mostrare i dati di monitoraggio raccolti da Zabbix. I grafici mostrano tendenze e metriche, le mappe di rete visualizzano la topologia dell'infrastruttura e le dashboard combinano widget personalizzabili per fornire viste di monitoraggio complete.

Strumenti di visualizzazione

- Grafici - visualizza i dati degli item come semplici grafici generati automaticamente, crea grafici personalizzati oppure confronta più item in grafici ad hoc.
- Mappe di rete - crea rappresentazioni visive dell'infrastruttura di rete con elementi e indicatori personalizzabili.
- Dashboard - personalizza i layout combinando una varietà di widget della dashboard che visualizzano i dati in diversi formati.

Grafici

Panoramica

Con una grande quantità di dati che confluisce in Zabbix, per gli utenti diventa molto più semplice osservare una rappresentazione visiva di ciò che sta accadendo, anziché limitarsi ai soli numeri.

È qui che entrano in gioco i grafici. I grafici consentono di comprendere il flusso dei dati a colpo d'occhio, correlare i problemi, scoprire quando è iniziato qualcosa o illustrare quando qualcosa potrebbe trasformarsi in un problema.

Zabbix offre agli utenti:

- Grafici semplici integrati per i dati di un singolo item.
- La possibilità di creare grafici personalizzati più complessi.
- Accesso a un confronto rapido di diversi item nei grafici ad hoc.
- Grafici vettoriali moderni e personalizzabili e grafici a torta.

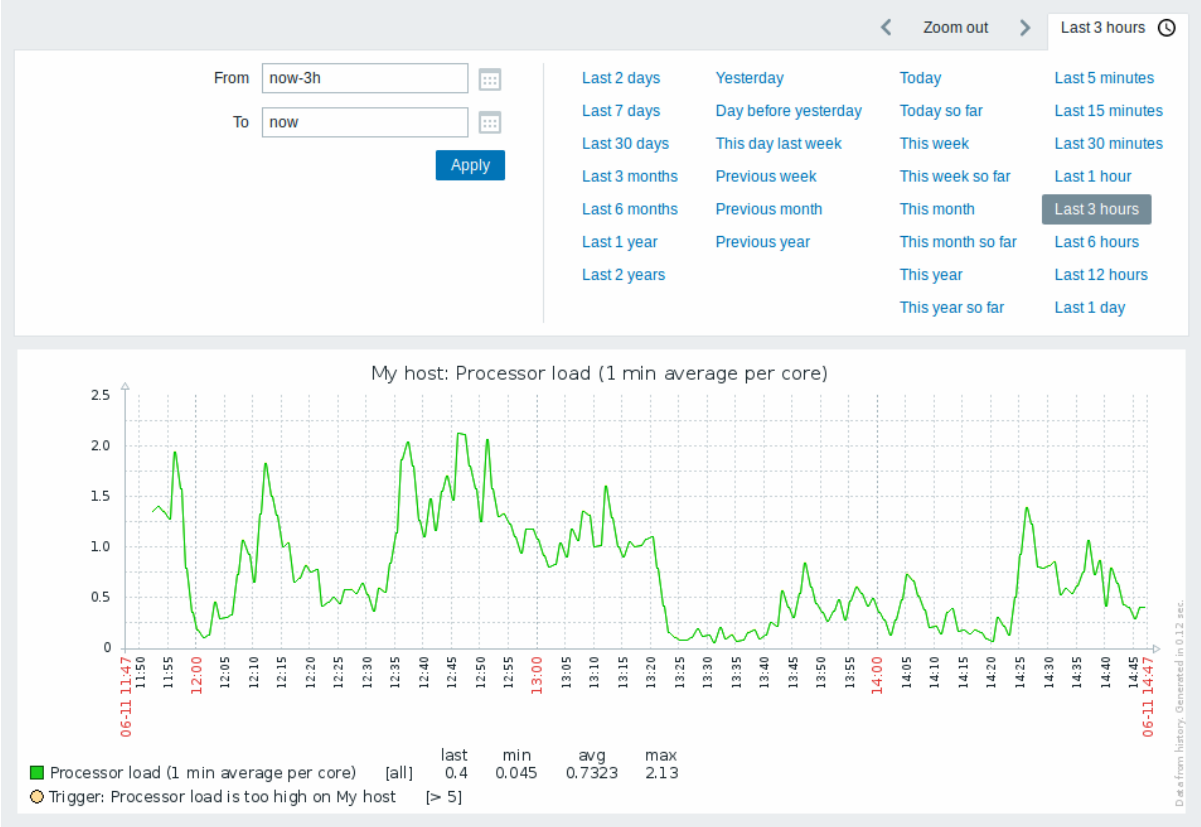
Grafici semplici

Panoramica

I grafici semplici sono disponibili per la visualizzazione dei dati raccolti dagli item.

Per visualizzare i grafici semplici non è richiesta alcuna configurazione da parte dell'utente. Sono messi liberamente a disposizione da Zabbix.

È sufficiente accedere a *Monitoring* > *Latest data* e fare clic sul collegamento *Graph* dell'item desiderato per visualizzare un grafico.



Note:
I grafici semplici sono disponibili per tutti gli item numerici. Per gli item testuali, in *Monitoring* > *Latest data* è disponibile un collegamento a *History*.

Selettore del periodo di tempo

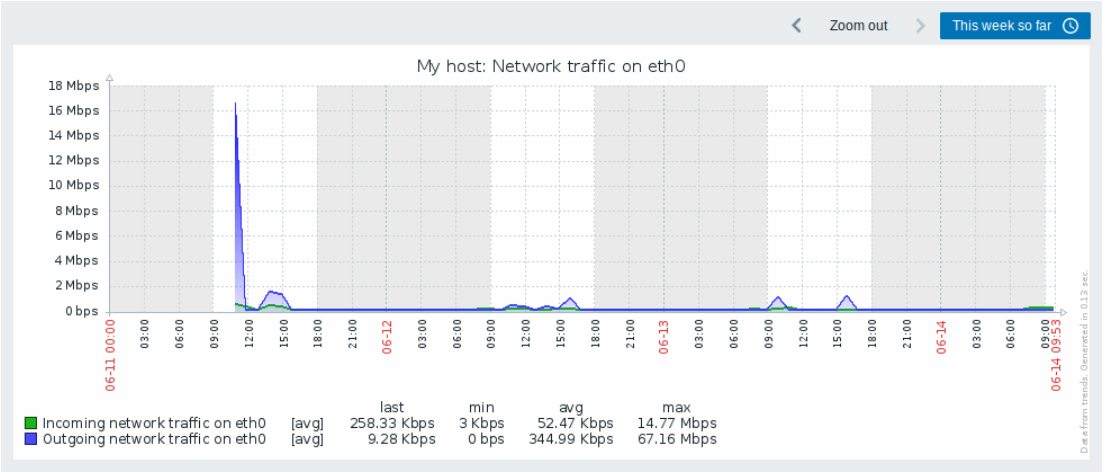
Il selettore del periodo di tempo sopra il grafico consente di selezionare con un solo clic del mouse i periodi utilizzati più frequentemente. Per ulteriori informazioni, vedere Selettori del periodo di tempo e dell'host.

Dati recenti vs periodi più lunghi

Per dati molto recenti viene tracciata una **singola** linea che collega ogni valore ricevuto. La singola linea viene tracciata finché è disponibile almeno un pixel orizzontale per un valore.

Per i dati che coprono un periodo più lungo vengono tracciate **tre linee**: una verde scuro mostra la media, mentre una linea rosa chiaro e una verde chiaro mostrano i valori massimo e minimo in quel momento. Lo spazio tra i massimi e i minimi è riempito con uno sfondo giallo.

L'orario di lavoro (giorni lavorativi) viene visualizzato nei grafici con uno sfondo bianco, mentre il tempo non lavorativo viene visualizzato in grigio (con il tema frontend predefinito *Original blue*).



L'orario di lavoro viene sempre visualizzato nei grafici semplici, mentre la sua visualizzazione nei grafici personalizzati dipende dalle preferenze dell'utente.

L'orario di lavoro non viene visualizzato se il grafico mostra più di 3 mesi.

Linee dei trigger

I trigger semplici vengono visualizzati come linee tratteggiate nere sovrapposte al colore della severità del trigger; si noti la linea blu sul grafico e le informazioni del trigger visualizzate nella legenda. Nel grafico possono essere visualizzate fino a 3 linee di trigger; se i trigger sono di più, vengono prioritizzati quelli con severità inferiore. I trigger vengono sempre visualizzati nei grafici semplici, mentre la loro visualizzazione nei grafici personalizzati dipende dalle preferenze dell'utente.



Generazione dalla cronologia/dai trend

I grafici possono essere tracciati in base alla cronologia o ai trend di un item.

Per gli utenti che hanno attivato la modalità di debug nel frontend, nell'angolo inferiore destro del grafico viene visualizzata una didascalia verticale grigia che indica la provenienza dei dati.

Diversi fattori influenzano l'utilizzo della cronologia o dei trend:

- Durata di conservazione della cronologia dell'item. Ad esempio, la cronologia di un item può essere conservata per 14 giorni. In tal caso, tutti i dati più vecchi di quattordici giorni proverranno dai trend.
- Densità dei dati nel grafico. Se il numero di secondi da visualizzare in un pixel orizzontale del grafico supera 3600/16, vengono visualizzati i dati dei trend (anche se la cronologia dell'item è ancora disponibile per lo stesso periodo).
- Se i trend sono disabilitati, per la creazione del grafico viene utilizzata la cronologia dell'item, se disponibile per quel periodo.

Assenza di dati

Per gli item con un intervallo di aggiornamento regolare, non viene visualizzato nulla nel grafico se i dati dell'item non vengono raccolti.

Tuttavia, per gli item trapper e gli item con un intervallo di aggiornamento pianificato (e intervallo di aggiornamento regolare impostato a 0), viene tracciata una linea retta fino al primo valore raccolto e dall'ultimo valore raccolto fino alla fine del grafico; la linea si trova rispettivamente al livello del primo/ultimo valore.

Passaggio ai valori grezzi

Un menu a discesa in alto a destra consente di passare dal grafico semplice agli elenchi *Valori/500 valori più recenti*. Questa opzione può essere utile per visualizzare i valori numerici che compongono il grafico.

I valori qui rappresentati sono grezzi, ovvero non vengono utilizzate unità né operazioni di post-elaborazione sui valori. Tuttavia, viene applicata la mappatura dei valori.

Problemi conosciuti

Consulta problemi noti per i grafici.

Grafici personalizzati

Panoramica

I grafici personalizzati, come suggerisce il nome, offrono possibilità di personalizzazione.

Mentre i grafici semplici sono adatti per visualizzare i dati di un singolo item, non offrono possibilità di configurazione.

Pertanto, se si desidera modificare lo stile del grafico o il modo in cui vengono visualizzate le linee, oppure confrontare diversi item, ad esempio il traffico in entrata e in uscita in un unico grafico, è necessario un grafico personalizzato.

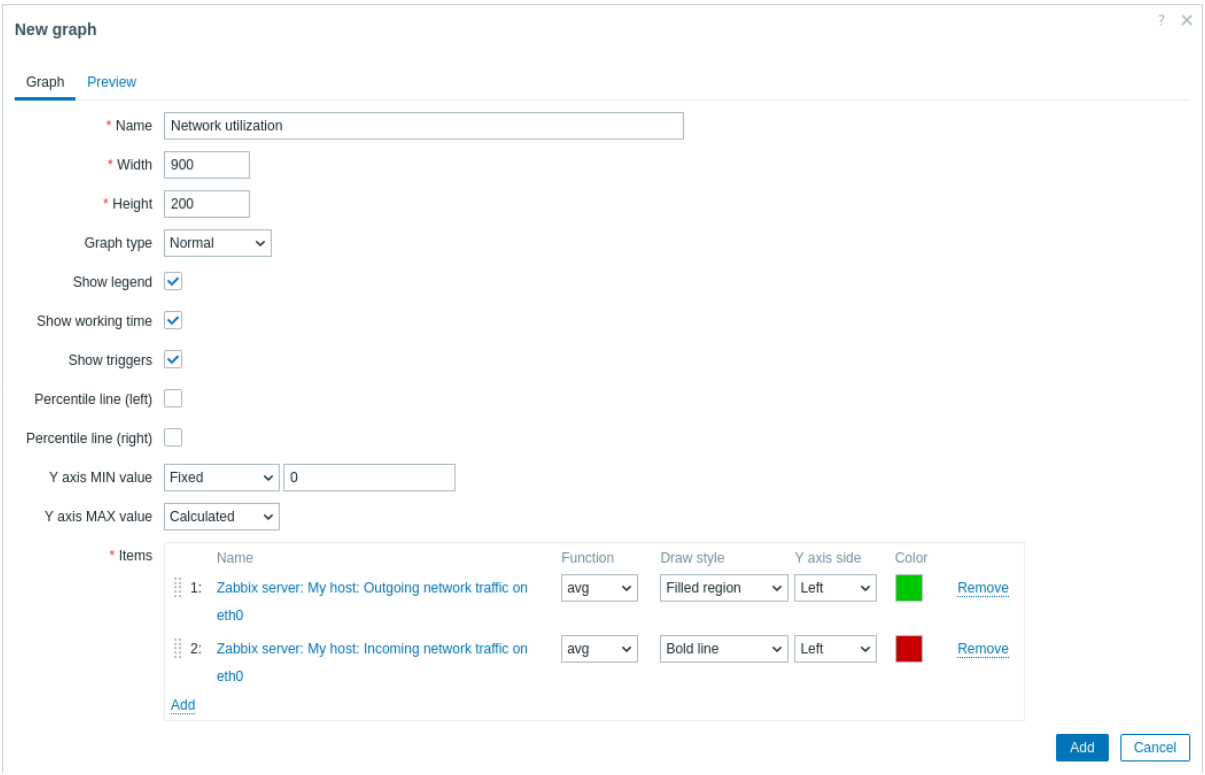
I grafici personalizzati vengono configurati manualmente.

Possono essere creati per un host o più host oppure per un singolo template.

Configurazione di grafici personalizzati

Per creare un grafico personalizzato, procedere come segue:

1. Andare in *Raccolta dati* > *Host* (o *template*).
2. Fare clic su *Grafici* nella riga accanto all'host o al template desiderato.
3. Nella schermata Grafici, fare clic su *Crea grafico*.
4. Modificare gli attributi del grafico.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

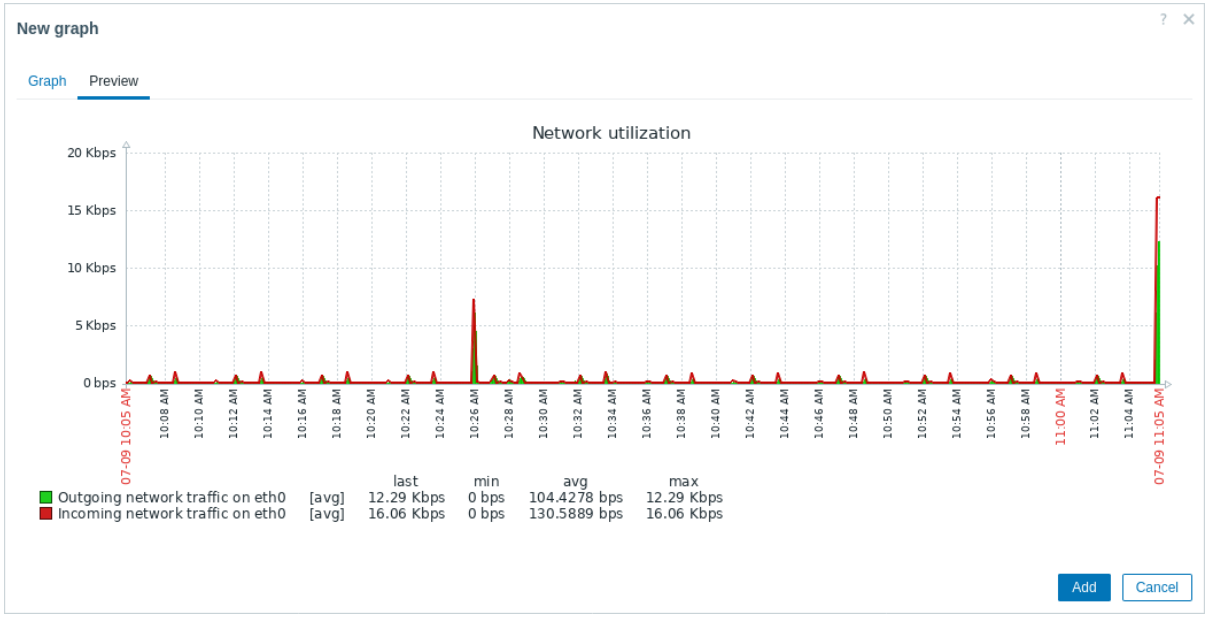
Attributi del grafico:

Parametro	Descrizione
Name	Nome univoco del grafico. In questo campo sono supportate le espressioni macros, ma solo con le funzioni `avg`, `last`, `min` e `max`, con il tempo come parametro (ad esempio, `{?avg(/host/key,1h)}`). Le macro {HOST.HOST<1-9>} sono supportate per l'uso all'interno di questa macro, facendo riferimento al primo, secondo, terzo, ecc. host nel grafico, ad esempio, `{?avg(/{HOST.HOST2}/key,1h)}`. Si noti che il riferimento al primo host con questa macro è ridondante, poiché il primo host può essere richiamato implicitamente, ad esempio, `{?avg(//key,1h)}`.
Width	Larghezza del grafico in pixel (solo per l'anteprima e per i grafici a torta/esplosi).
Height	Altezza del grafico in pixel.
Graph type	Tipo di grafico: **Normal** - grafico normale, i valori sono visualizzati come linee **Stacked** - grafico in pila, le aree riempite sono visualizzate **Pie** - grafico a torta **Exploded** - grafico a torta "esploso", le porzioni sono visualizzate come "ritagliate" dalla torta
Show legend	Selezionando questa casella, verrà visualizzata la legenda del grafico.

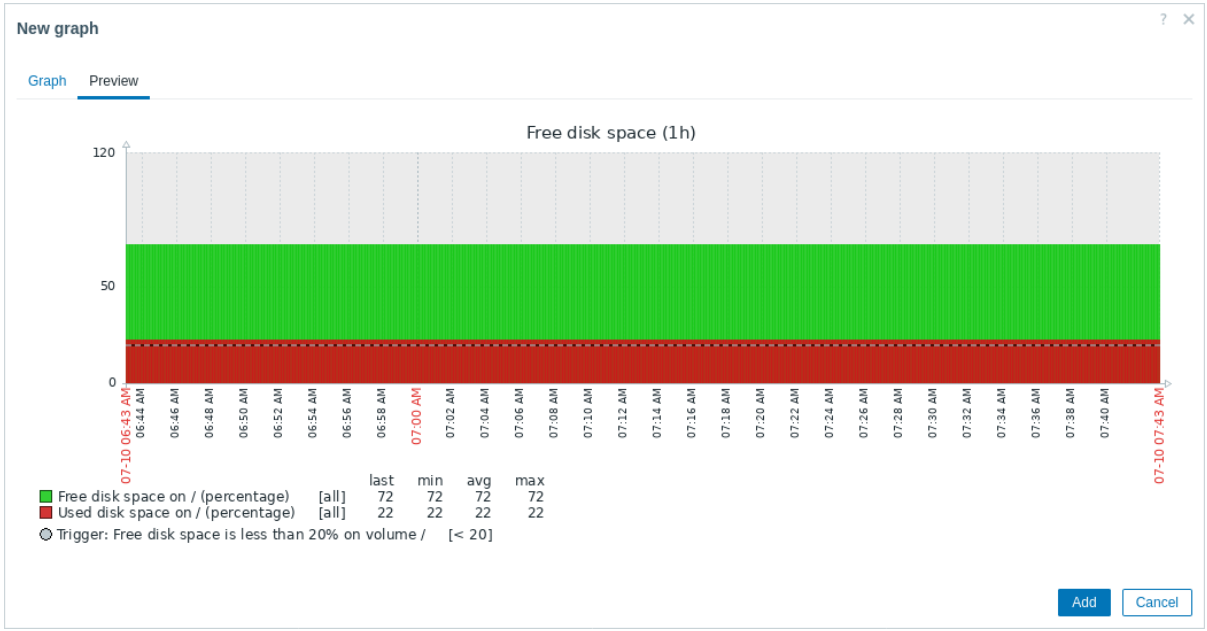
Parametro	Descrizione
Show working time	Se selezionato, le ore non lavorative saranno mostrate con uno sfondo grigio. Questo parametro non è disponibile per i grafici a torta e a torta esplosa.
Show triggers	Se selezionato, i simple triggers saranno visualizzati come linee con trattini neri sopra il colore della severità del trigger. Questo parametro non è disponibile per i grafici a torta e a torta esplosa.
Percentile line (left)	Visualizza il percentile per l'asse Y sinistro. Se, ad esempio, è impostato il percentile del 95%, la linea del percentile sarà al livello sotto il quale rientra il 95% dei valori. Visualizzata come una linea verde brillante. Disponibile solo per i grafici normali.
Percentile line (right)	Visualizza il percentile per l'asse Y destro. Se, ad esempio, è impostato il percentile del 95%, la linea del percentile sarà al livello sotto il quale rientra il 95% dei valori. Visualizzata come una linea rosso brillante. Disponibile solo per i grafici normali.
Y axis MIN value	Valore minimo dell'asse Y: **Calculated** - il valore minimo dell'asse Y verrà calcolato automaticamente. **Fixed** - valore minimo fisso per l'asse Y. **Item** - l'ultimo valore dell'item selezionato sarà il valore minimo. Questo parametro non è disponibile per i grafici a torta e a torta esplosa.
Y axis MAX value	Valore massimo dell'asse Y: **Calculated** - il valore massimo dell'asse Y verrà calcolato automaticamente. **Fixed** - valore massimo fisso per l'asse Y. **Item** - l'ultimo valore dell'item selezionato sarà il valore massimo Questo parametro non è disponibile per i grafici a torta e a torta esplosa.
3D view	Abilita lo stile 3D. Solo per grafici a torta e a torta esplosa.
Items	Item i cui dati devono essere visualizzati in questo grafico. Fare clic su *Add* per selezionare gli item. È inoltre possibile selezionare varie opzioni di visualizzazione (funzione, stile di disegno, visualizzazione asse sinistro/destro, colore).
Sort order (0→100)	Ordine di disegno. 0 verrà elaborato per primo. Può essere usato per disegnare linee o aree dietro (o davanti a) un'altra. È possibile trascinare e rilasciare gli item usando l'icona all'inizio di una riga per impostare l'ordine di ordinamento o per stabilire quale item viene visualizzato davanti all'altro.
Name	Il nome dell'item selezionato viene visualizzato come collegamento. Facendo clic sul collegamento si apre l'elenco degli altri item disponibili.
Type	Tipo (disponibile solo per grafici a torta e a torta esplosa): **Simple** - il valore dell'item è rappresentato proporzionalmente nella torta **Graph sum** - il valore dell'item rappresenta l'intera torta Si noti che la colorazione dell'item "graph sum" sarà visibile solo nella misura in cui non è occupata dagli item "proportional".
Function	Selezionare quali valori verranno visualizzati quando per un item esiste più di un valore per pixel verticale del grafico: **all** - visualizza tutti i valori possibili (minimo, massimo, medio) nel grafico. Si noti che per periodi più brevi questa impostazione non ha effetto; solo per periodi più lunghi, quando la congestione dei dati in un pixel verticale del grafico aumenta, 'all' inizia a visualizzare i valori minimo, massimo e medio. Questa funzione è disponibile solo per il tipo di grafico *Normal*. Vedere anche: Generazione di grafici da history/trends. **avg** - visualizza i valori medi **last** - visualizza gli ultimi valori. Questa funzione è disponibile solo se come tipo di grafico è selezionato *Pie/Exploded pie*. **max** - visualizza i valori massimi **min** - visualizza i valori minimi
Draw style	Selezionare lo stile di disegno (disponibile solo per i grafici normali; per i grafici in pila viene sempre usata un'area riempita) da applicare ai dati dell'item - *Line*, *Bold line*, *Filled region*, *Dot*, *Dashed line*, *Gradient line*.
Y axis side	Selezionare il lato dell'asse Y su cui visualizzare i dati dell'item - *Left*, *Right*.
Color	Selezionare il colore da applicare ai dati dell'item.

Anteprima del grafico

Nella scheda *Anteprima* viene visualizzata un'anteprima del grafico, così puoi vedere immediatamente ciò che stai creando.



Tieni presente che l'anteprima non mostrerà alcun dato per gli item del template.



In questo esempio, presta attenzione alla linea tratteggiata in grassetto che mostra il livello del trigger e alle informazioni del trigger visualizzate nella legenda.

Note:
Non possono essere visualizzate più di 3 linee di trigger. Se sono presenti più trigger, per la visualizzazione vengono prioritizzati i trigger con gravità inferiore.

 Se l'altezza del grafico è impostata a meno di 120 pixel, nella legenda non verrà visualizzato alcun trigger.

Grafici ad hoc

Panoramica

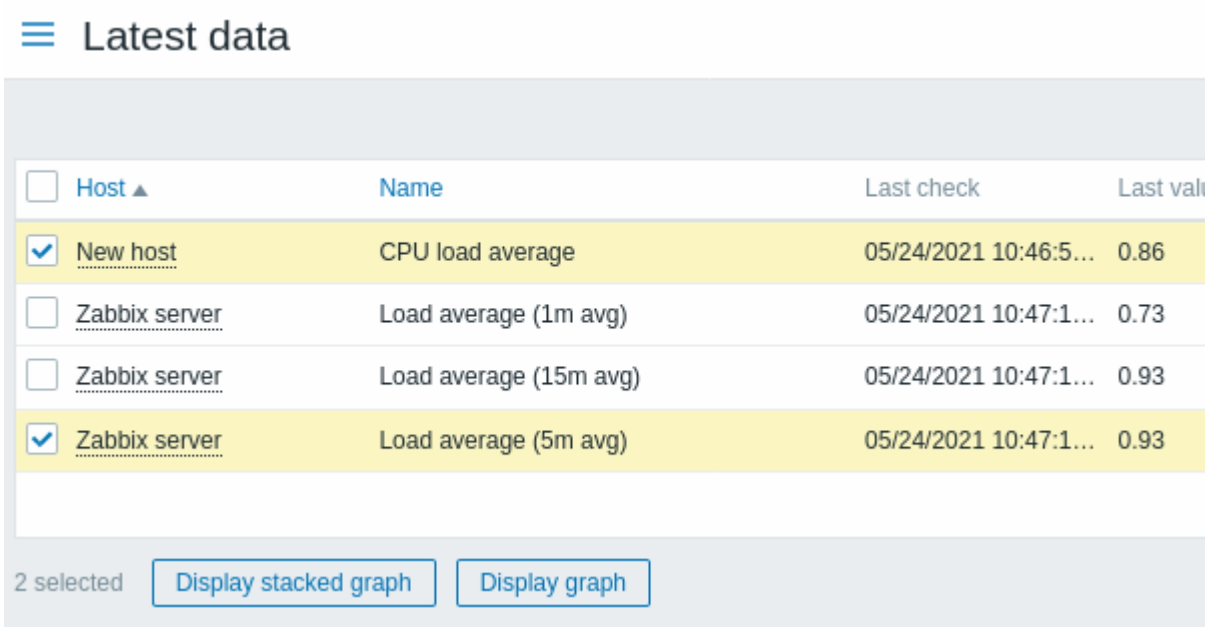
Sebbene un grafico semplice sia utile per visualizzare i dati di un singolo item e i grafici personalizzati offrano opzioni di personalizzazione, nessuno dei due consente di creare rapidamente un grafico di confronto per più item con il minimo sforzo e manutenzione.

Per risolvere questo problema, è possibile creare grafici ad hoc per diversi item in modo molto rapido.

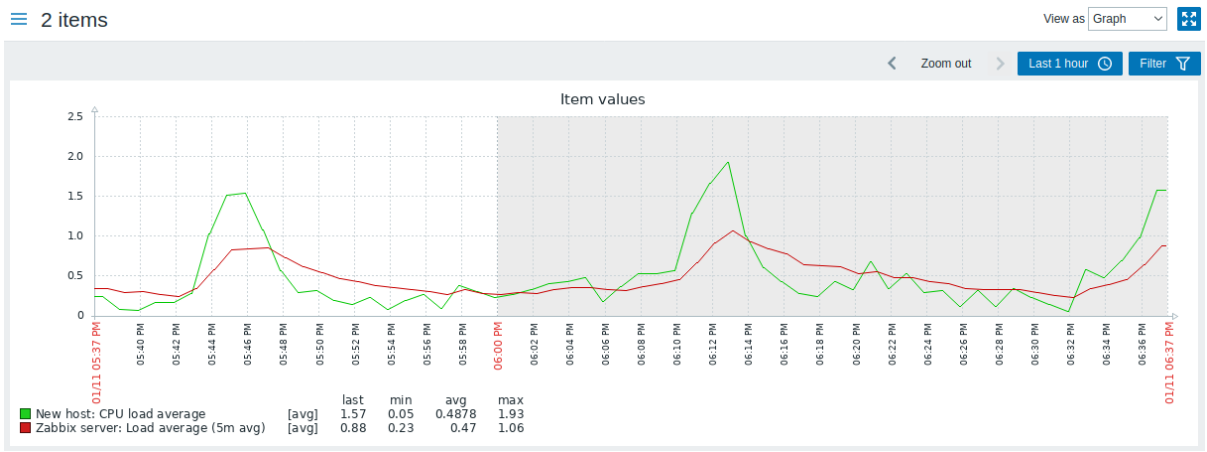
Configurazione

Per creare un grafico ad hoc, procedere come segue:

1. Andare in *Monitoring* > *Latest data*.
2. Usare il filtro per visualizzare gli item desiderati.
3. Selezionare le caselle di controllo degli item che si desidera rappresentare nel grafico.
4. Fare clic sui pulsanti *Display stacked graph* o *Display graph*.

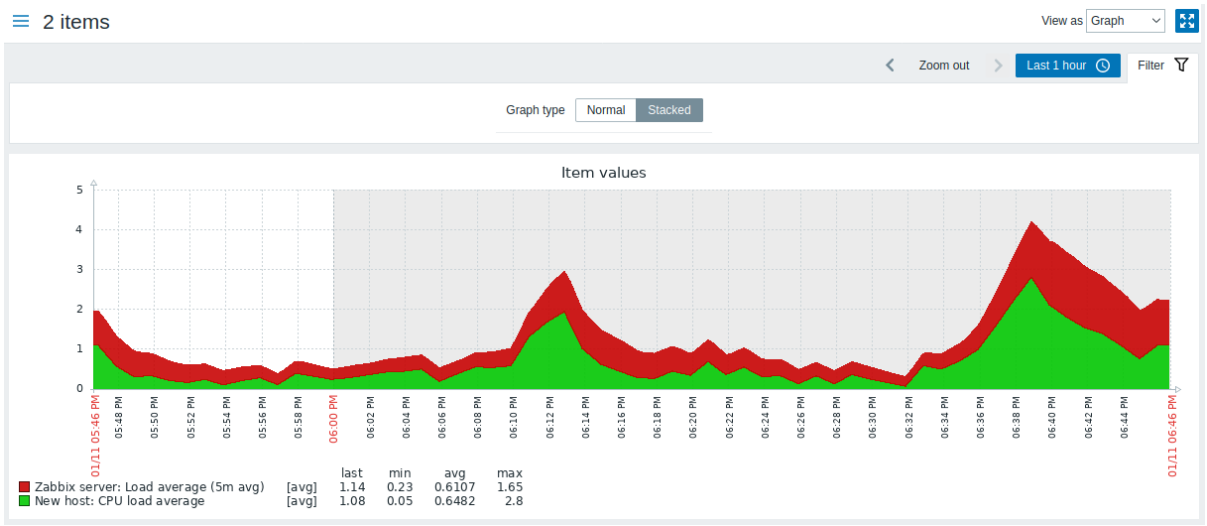


Il grafico viene creato immediatamente:



Si noti che, per evitare di visualizzare troppe linee nel grafico, viene mostrato solo il valore medio di ciascun item (le linee dei valori min/max non vengono visualizzate). I trigger e le informazioni sui trigger non vengono visualizzati nel grafico.

Nella finestra del grafico creato è disponibile il selettore dell'intervallo di tempo e la possibilità di passare dal grafico a linee "normale" a uno impilato e viceversa.



Mappe di rete

Panoramica

Se devi gestire una rete, potresti voler avere una panoramica della tua infrastruttura da qualche parte. A tale scopo, puoi creare mappe in Zabbix, sia di reti sia di qualsiasi altra cosa desideri.

Tutti gli utenti possono creare mappe di rete. Le mappe possono essere pubbliche (disponibili per tutti gli utenti) oppure private (disponibili per utenti selezionati).

Procedi a configurare una mappa di rete.

Configurazione di una mappa di rete

Panoramica

La configurazione di una mappa in Zabbix richiede innanzitutto la creazione di una mappa, definendone i parametri generali, e poi il popolamento effettivo della mappa con elementi e relativi collegamenti.

È possibile popolare la mappa con elementi che siano un host, un gruppo di host, un trigger, un'immagine o un'altra mappa.

Le icone vengono utilizzate per rappresentare gli elementi della mappa. È possibile definire le informazioni che verranno visualizzate con le icone e impostare che i problemi recenti vengano mostrati in modo speciale. È possibile collegare le icone e definire le informazioni da visualizzare sui collegamenti.

È possibile aggiungere URL personalizzati a cui accedere facendo clic sulle icone. In questo modo, si può collegare l'icona di un host alle proprietà dell'host oppure l'icona di una mappa a un'altra mappa.

Il conteggio dei problemi nelle mappe viene visualizzato solo per i problemi causa.

Le mappe sono gestite in *Monitoring* > *Maps*, dove possono essere configurate, gestite e visualizzate. Nella vista di monitoraggio, è possibile fare clic sulle icone e sfruttare i collegamenti ad alcuni script e URL.

Le mappe di rete si basano su grafica vettoriale (SVG).

Mappe pubbliche e private

Tutti gli utenti in Zabbix (inclusi gli utenti non amministratori) possono creare mappe di rete. Le mappe hanno un proprietario: l'utente che le ha create. Le mappe possono essere pubbliche o private.

- Le mappe *pubbliche* sono visibili a tutti gli utenti, anche se, per visualizzarle, l'utente deve avere accesso in lettura ad almeno un elemento della mappa. Le mappe pubbliche possono essere modificate nel caso in cui un utente/gruppo di utenti disponga di permessi di lettura-scrittura per questa mappa e di almeno permessi di lettura per tutti gli elementi della mappa corrispondente, inclusi i trigger nei collegamenti.
- Le mappe *private* sono visibili solo al loro proprietario e agli utenti/gruppi di utenti con cui la mappa è condivisa dal proprietario. Gli utenti normali (non Super admin) possono condividere solo con i gruppi a cui appartengono e con gli utenti che appartengono a tali gruppi. Gli utenti di livello Admin possono vedere le mappe private indipendentemente dal fatto di esserne i proprietari o di appartenere all'elenco degli utenti con cui sono condivise. Le mappe private possono essere modificate dal proprietario della mappa e nel caso in cui un utente/gruppo di utenti disponga di permessi di lettura-scrittura per questa mappa e di almeno permessi di lettura per tutti gli elementi della mappa corrispondente, inclusi i trigger nei collegamenti.

Gli elementi della mappa per i quali l'utente non dispone di permessi di lettura vengono visualizzati con un'icona in grigio e tutte le informazioni testuali sull'elemento sono nascoste. Tuttavia, l'etichetta del trigger è visibile anche se l'utente non dispone di permessi per il trigger.

Per aggiungere un elemento alla mappa, l'utente deve avere almeno i permessi di lettura per l'elemento.

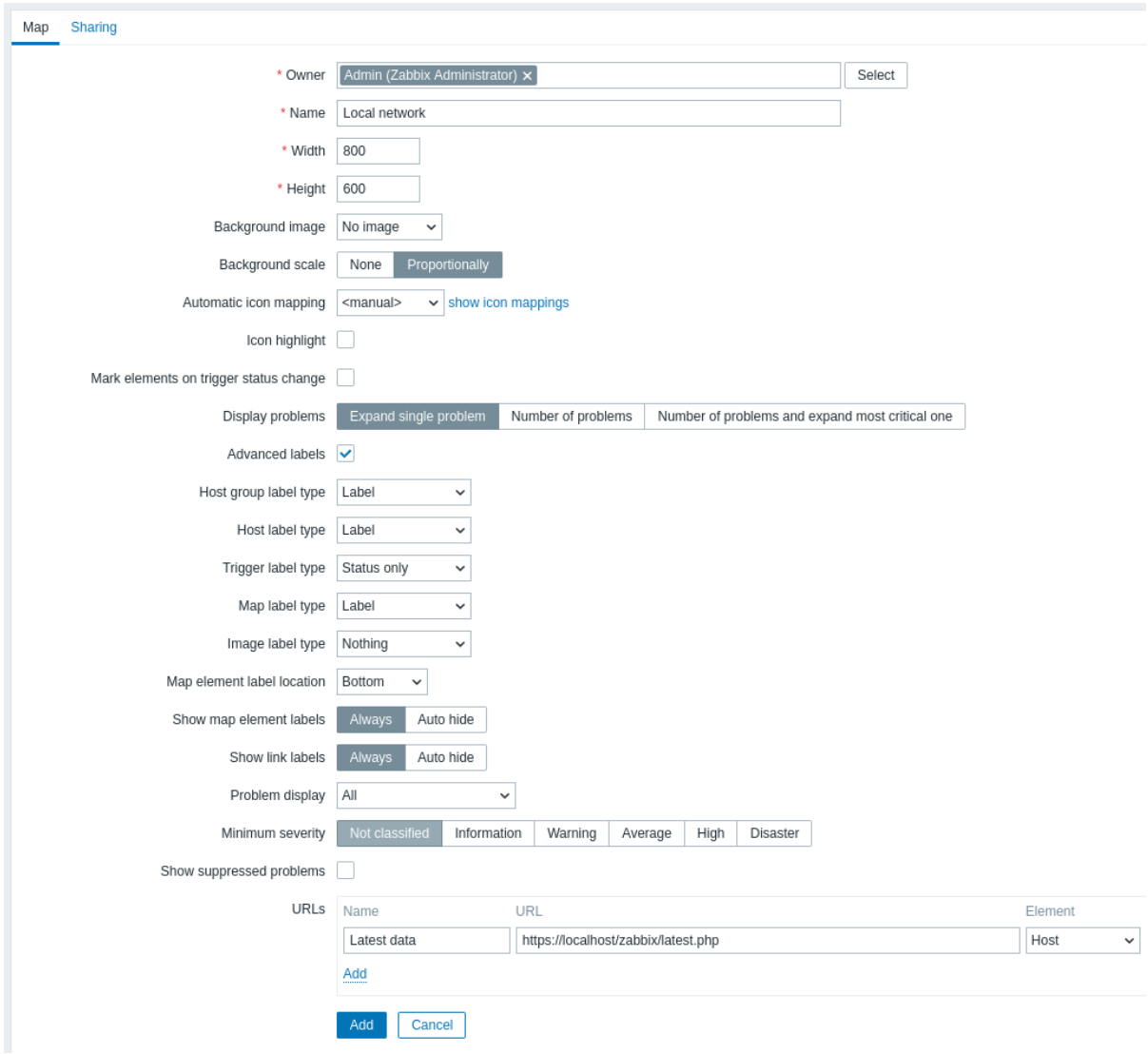
Creazione di una mappa

Per creare una mappa, procedere come segue:

1. Andare a *Monitoring* > *Maps*.
2. Andare alla vista con tutte le mappe.
3. Fare clic su *Create map*.
4. Configurare i dettagli della mappa.

È inoltre possibile utilizzare il pulsante *Clone* nel modulo di configurazione di una mappa esistente per creare una nuova mappa. Questa mappa avrà tutte le proprietà della mappa esistente, inclusi gli attributi generali del layout e gli elementi della mappa esistente.

La scheda **Map** contiene gli attributi generali della mappa:



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Attributi generali della mappa:

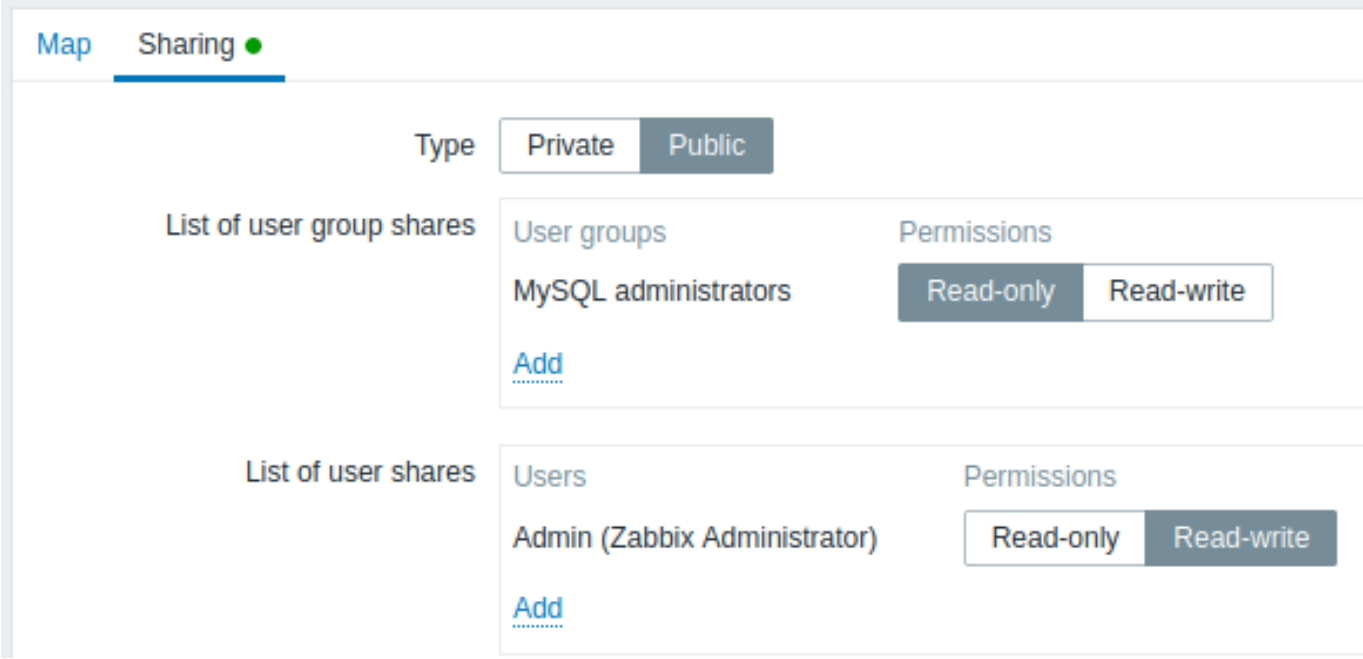
Parametro	Descrizione
Owner	Nome del proprietario della mappa.
Name	Nome univoco della mappa.
Width	Larghezza della mappa in pixel.
Height	Altezza della mappa in pixel.
Background image	Utilizzare un'immagine di sfondo: **No image** - nessuna immagine di sfondo (sfondo bianco) **Image** - immagine selezionata da utilizzare come immagine di sfondo. È possibile utilizzare una mappa geografica o qualsiasi altra immagine per migliorare la mappa.
Background scale	Utilizzare il ridimensionamento dello sfondo: **None** - nessun ridimensionamento; **Proportionally** - copre l'intero sfondo della mappa mantenendo le proporzioni dell'immagine (impostazione predefinita).
Automatic icon mapping	È possibile impostare l'utilizzo di una mappatura automatica delle icone, configurata in *Administration > General > Icon mapping*. La mappatura delle icone consente di associare determinate icone a specifici campi dell'inventario del host.

Parametro	Descrizione
Icon highlight	Selezionando questa casella di controllo, gli elementi della mappa verranno evidenziati. Gli elementi con un trigger attivo riceveranno uno sfondo circolare dello stesso colore del trigger con la severità più alta. Inoltre, attorno al cerchio verrà visualizzata una spessa linea verde se tutti i problemi sono stati riconosciuti. Gli elementi con stato ”disabled” o ”in maintenance” riceveranno rispettivamente uno sfondo quadrato grigio o arancione. Vedere anche: Visualizzazione delle mappe
Mark elements on trigger status change	Una modifica recente dello stato del trigger (problema recente o risoluzione) verrà evidenziata con indicatori (triangoli rossi rivolti verso l’interno) sui tre lati dell’icona dell’elemento privi dell’etichetta. Gli indicatori vengono visualizzati per 30 minuti.
Display problems	Selezionare come visualizzare i problemi con un elemento della mappa: **Expand single problem** - se è presente un solo problema, viene visualizzato il nome del problema. In caso contrario, viene visualizzato il numero totale di problemi. **Number of problems** - viene visualizzato il numero totale di problemi **Number of problems and expand most critical one** - vengono visualizzati il nome del problema più critico e il numero totale di problemi. Il problema ”più critico” viene determinato in base alla severità del problema e, in caso di parità, all’ID dell’evento del problema (l’ID più alto o il problema più recente viene visualizzato per primo). Per un *trigger map element*, viene determinato in base alla severità del problema e, in caso di parità, alla posizione del trigger nell’elenco dei trigger. In caso di più problemi dello stesso trigger, verrà visualizzato quello più recente.
Advanced labels	Selezionando questa casella di controllo, sarà possibile definire tipi di etichetta separati per tipi di elemento separati.
Host group label type	Tipo di etichetta utilizzato per l’elemento della mappa: **Label** - etichetta dell’elemento della mappa **IP address** - indirizzo IP **Element name** - nome dell’elemento (ad esempio, nome del host) **Status only** - solo stato (OK o PROBLEM) **Nothing** - non vengono visualizzate etichette
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	Posizione dell’etichetta rispetto all’elemento della mappa: **Bottom** - sotto l’elemento della mappa **Left** - a sinistra **Right** - a destra **Top** - sopra l’elemento della mappa
Show map element labels	Selezionare come visualizzare le etichette degli elementi della mappa: **Always** - sempre visualizzate (impostazione predefinita); **Auto hide** - nasconde l’etichetta se non vi si passa sopra con il puntatore o non viene selezionata.
Show link labels	Selezionare come visualizzare le etichette dei collegamenti: **Always** - sempre visualizzate (impostazione predefinita); **Auto hide** - nasconde l’etichetta se non vi si passa sopra con il puntatore o non viene selezionata.
Problem display	Visualizzare il numero di problemi come: **All** - viene visualizzato il numero completo di problemi **Separated** - il numero di problemi non riconosciuti viene visualizzato separatamente come parte del numero totale di problemi **Unacknowledged only** - viene visualizzato solo il numero di problemi non riconosciuti
Minimum severity	I problemi al di sotto del livello di severità minimo selezionato non verranno visualizzati sulla mappa. Ad esempio, selezionando *Warning*, le modifiche con trigger di livello *Information* e *Not classified* non verranno riflesse nella mappa.
Show suppressed problems	Selezionare la casella di controllo per visualizzare i problemi che altrimenti verrebbero soppressi (non visualizzati) a causa della manutenzione del host.

Parametro	Descrizione
URLs	Qui è possibile definire gli URL per ogni tipo di elemento, dove: **Name** - un'etichetta per l'URL; **URL** - un URL (fino a 2048 caratteri); **Element** - il tipo di elemento (ad esempio, *Host*, *Host group*, ecc.). Questi verranno visualizzati come collegamenti quando un utente fa clic sull'elemento nella modalità di visualizzazione della mappa. Le macro possono essere utilizzate nei nomi e nei valori degli URL della mappa. Per un elenco completo, vedere le macro supportate e cercare 'map URL names and values'.

Condivisione

La scheda **Sharing** contiene il tipo di mappa e le opzioni di condivisione (gruppi di utenti, utenti) per le mappe private:



Parameter	Description
Type	Seleziona il tipo di mappa: **Private** - la mappa è visibile solo ai gruppi di utenti e agli utenti selezionati **Public** - la mappa è visibile a tutti
List of user group shares	Seleziona i gruppi di utenti a cui la mappa è accessibile. Puoi consentire l'accesso in sola lettura o in lettura e scrittura.
List of user shares	Seleziona gli utenti a cui la mappa è accessibile. Puoi consentire l'accesso in sola lettura o in lettura e scrittura.

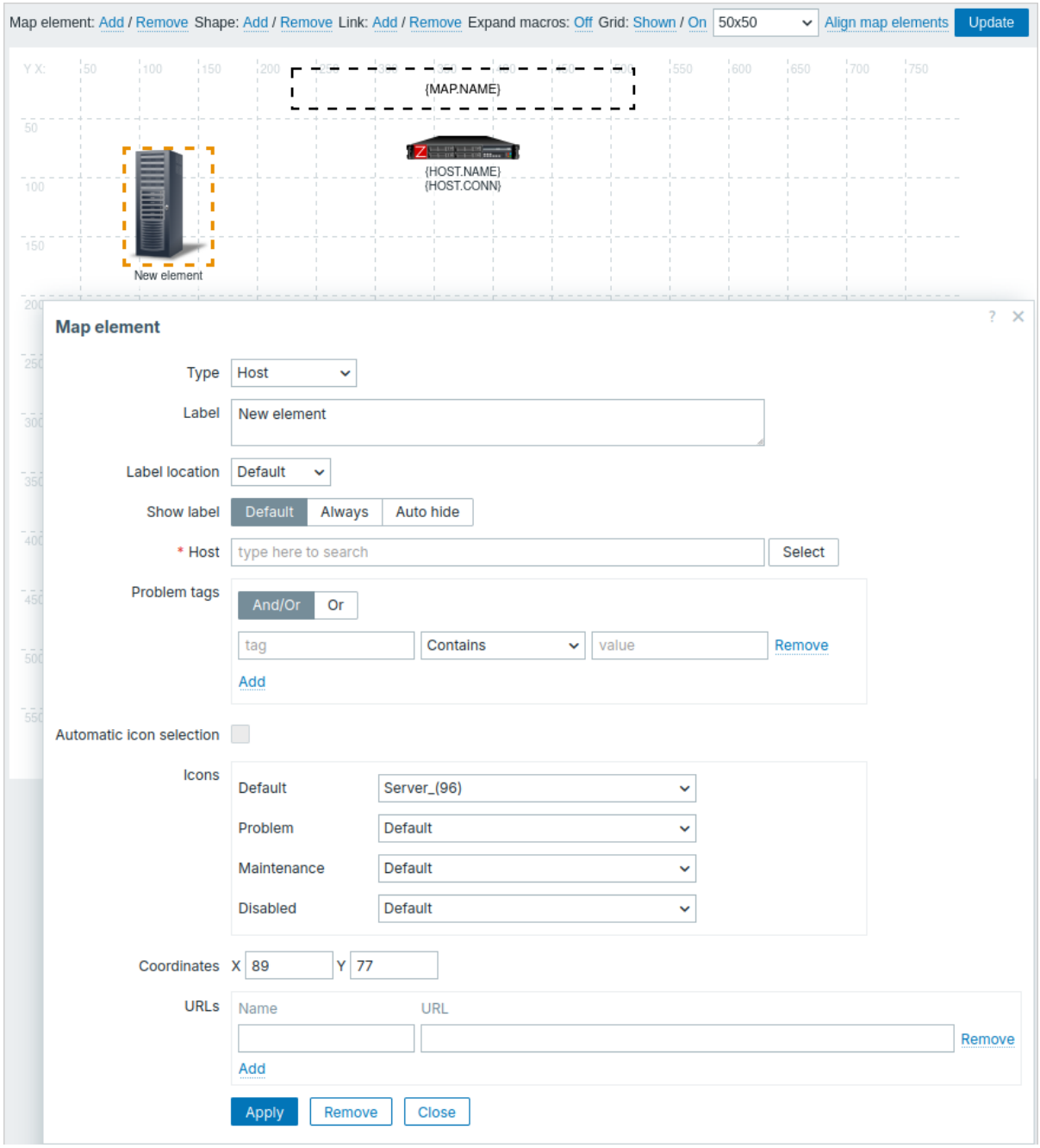
Quando fai clic su *Add* per salvare questa mappa, hai creato una mappa vuota con un nome, dimensioni e alcune preferenze. Ora devi aggiungere alcuni elementi. Per farlo, fai clic su *Edit* nell'elenco delle mappe per aprire l'area modificabile.

Aggiunta di elementi

Per aggiungere un elemento, fai clic su *Add* accanto a `Map element`. Il nuovo elemento apparirà nell'angolo in alto a sinistra della mappa. Trascinalo e rilascialo dove preferisci.

Nota che con l'opzione Grid impostata su "On", gli elementi si allineeranno sempre alla griglia (puoi scegliere varie dimensioni della griglia dal menu a discesa, e anche nascondere/mostrare la griglia). Se vuoi posizionare gli elementi ovunque senza allineamento, imposta l'opzione su "Off". (Puoi allineare in seguito gli elementi casuali alla griglia facendo clic su *Align map elements*.)

Ora che hai inserito alcuni elementi, potresti voler iniziare a distinguerli assegnando nomi, ecc. Facendo clic sull'elemento, viene visualizzato un modulo in cui puoi impostare il tipo di elemento, assegnare un nome, scegliere un'icona diversa, ecc.



Attributi dell'elemento della mappa:

Parameter	Description
Type	Tipo dell'elemento: **Host** - icona che rappresenta lo stato di tutti i trigger dell'host selezionato **Map** - icona che rappresenta lo stato di tutti gli elementi di una mappa **Trigger** - icona che rappresenta lo stato di uno o più trigger **Host group** - icona che rappresenta lo stato di tutti i trigger di tutti gli host appartenenti al gruppo selezionato **Image** - un'icona, non collegata ad alcuna risorsa
Label	Etichetta dell'icona, qualsiasi stringa. È possibile usare macro e stringhe multilinea. In questo campo sono supportate le macro di espressione macros, ma solo con le funzioni `avg`, `last`, `min` e `max`, con il tempo come parametro (ad esempio, `{?avg(/host/key,1h)}`). Per un elenco completo delle macro supportate, vedi supported macros e cerca 'map element labels'.

Parameter	Description
Label location	Posizione dell'etichetta rispetto all'icona: **Default** - posizione predefinita dell'etichetta della mappa; **Bottom** - sotto l'icona; **Left** - a sinistra; **Right** - a destra; **Top** - sopra l'icona.
Show label	Seleziona come visualizzare l'etichetta dell'elemento: **Default** - usa le impostazioni *Show map element labels* dalla configurazione della mappa (predefinito); **Always** - sempre visualizzata (predefinito); **Auto hide** - nasconde l'etichetta se non è sotto il puntatore o selezionata.
Host	Seleziona un host; in alternativa, inserisci il nome dell'host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà visualizzato un elenco a discesa degli host corrispondenti. Scorri verso il basso per selezionarne uno. Fai clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Host*.
Map	Seleziona una mappa; in alternativa, inserisci il nome della mappa. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di una mappa verrà visualizzato un elenco a discesa delle mappe corrispondenti. Scorri verso il basso per selezionarne una. Fai clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Map*.
Triggers	Seleziona uno o più trigger nel campo *New triggers* qui sotto e fai clic su *Add*.In alternativa, inizia a digitare il nome di un trigger e selezionane uno dall'elenco a discesa dei trigger corrispondenti. Fai clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Trigger*. L'ordine dei trigger selezionati può essere modificato, ma solo all'interno della stessa gravità dei trigger. La selezione multipla dei trigger influisce anche sulla risoluzione delle macro {HOST.*} sia in modalità di modifica sia in modalità di visualizzazione. In modalità di modifica, le prime macro {HOST.*} visualizzate verranno risolte in base al primo trigger nell'elenco (in base alla gravità del trigger). La modalità di visualizzazione dipende dal parametro Display problems negli attributi generali della mappa: - Se viene scelta la modalità *Expand single problem*, le prime macro {HOST.*} visualizzate verranno risolte in base all'ultimo trigger di problema rilevato (indipendentemente dalla gravità) oppure al primo trigger nell'elenco (nel caso in cui non sia stato rilevato alcun problema). - Se viene scelta la modalità *Number of problems and expand most critical one*, le prime macro {HOST.*} visualizzate verranno risolte in base alla gravità del trigger.
Host group	Seleziona un host group; in alternativa, inserisci l'host group. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host group verrà visualizzato un elenco a discesa degli host group corrispondenti. Scorri verso il basso per selezionarne uno. Fai clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Host group*.
Problem tags	Specifica i tag per limitare il numero di problemi visualizzati nel widget. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole/minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate tramite la condizione Or **Or** - è sufficiente che una condizione sia soddisfatta Questo campo è disponibile solo per elementi di tipo *Host* e *Host group*.
Automatic icon selection	Quando la casella di controllo è selezionata, il mapping delle icone verrà usato per determinare quale icona visualizzare.

Parameter	Description
Icons	Puoi scegliere di visualizzare icone diverse per l'elemento nei seguenti casi: *Default*, *Problem*, *Maintenance* e *Disabled*. Questo campo è disponibile solo per elementi di tipo *Host*.
Coordinates	Coordinate **X** e **Y** dell'elemento della mappa.
URLs	Qui è possibile impostare URL specifici per l'elemento, dove: **Name** - un'etichetta per l'URL; **URL** - un URL (fino a 2048 caratteri). Questi verranno visualizzati come link quando un utente fa clic sull'elemento in modalità di visualizzazione della mappa. Se l'elemento ha i propri URL e sono definiti URL a livello di mappa per il suo tipo, verranno combinati nello stesso menu. Le macro possono essere usate nei nomi e nei valori degli elementi della mappa. Per un elenco completo, vedi supported macros e cerca 'map URL names and values'.

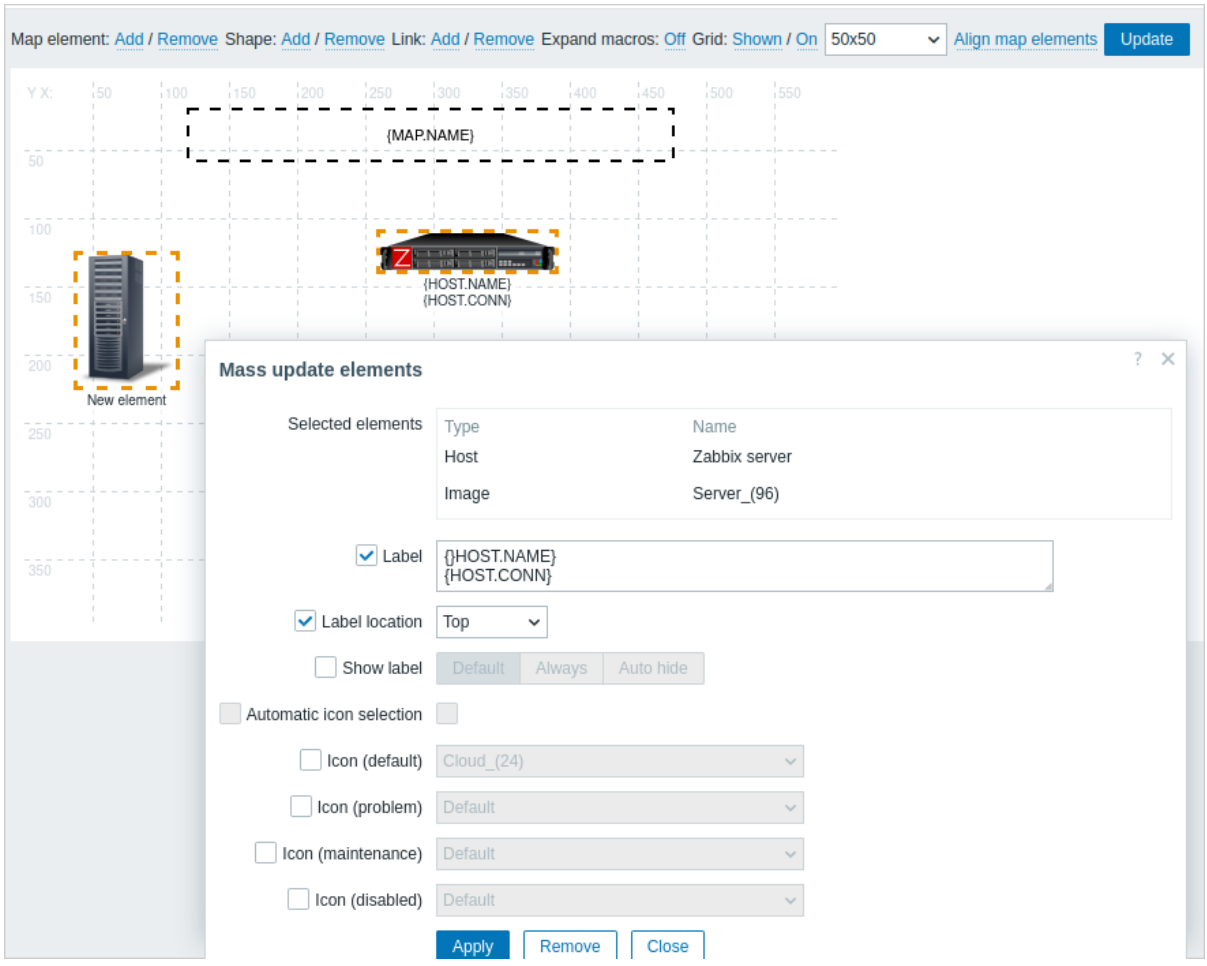
Attention:
Gli elementi aggiunti non vengono salvati automaticamente. Se lasci la pagina, tutte le modifiche potrebbero andare perse.
Per questo motivo, è consigliabile fare clic sul pulsante **Update** nell'angolo in alto a destra. Una volta fatto clic, le modifiche vengono salvate indipendentemente da ciò che scegli nel popup successivo.
Le opzioni della griglia selezionate vengono salvate anche con ogni mappa.

Selezione degli elementi

Per selezionare gli elementi, selezionane uno e poi tieni premuto *Ctrl* per selezionare gli altri.

Puoi anche selezionare più elementi trascinando un rettangolo nell'area modificabile e selezionando tutti gli elementi al suo interno.

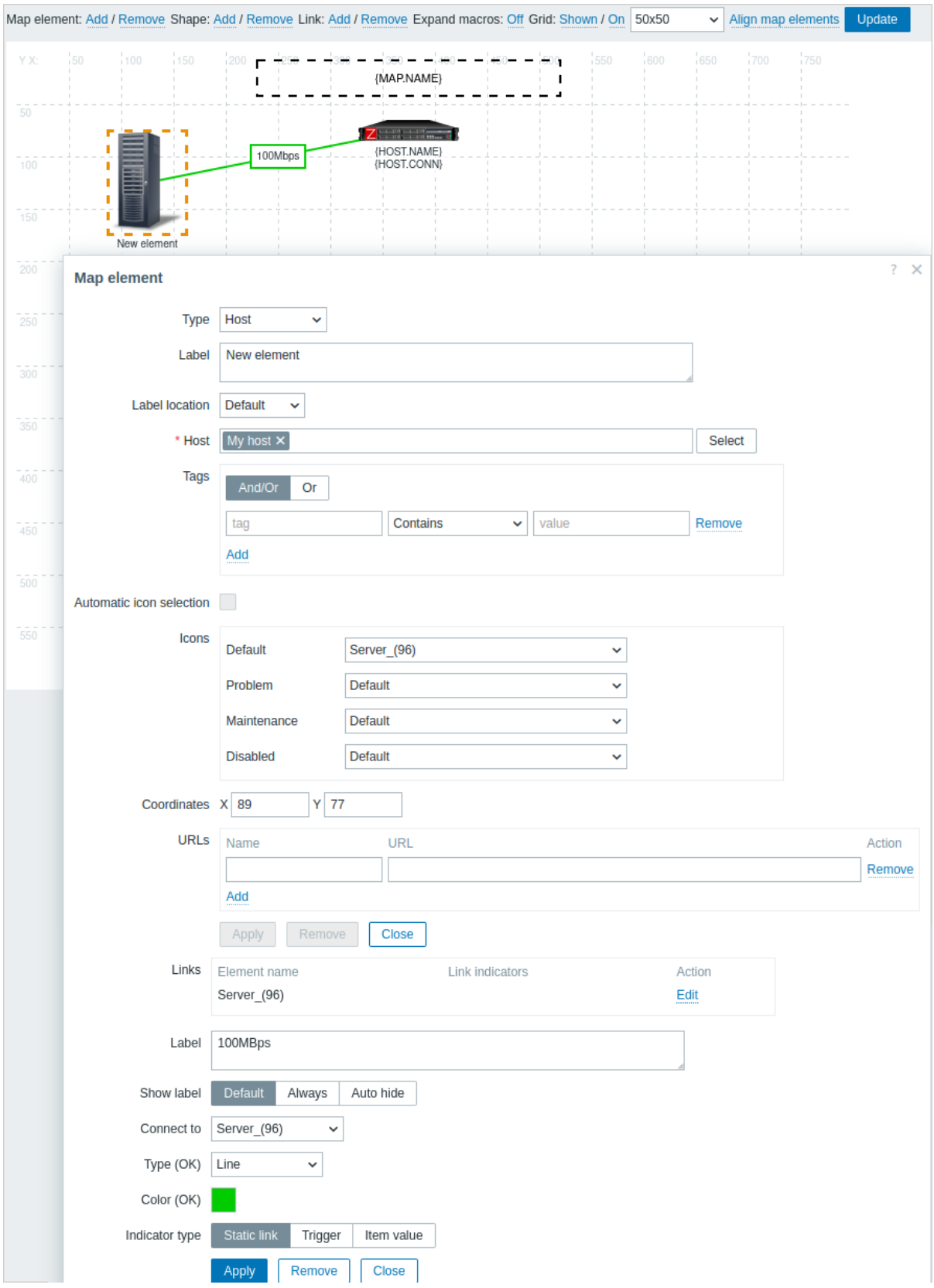
Una volta selezionato più di un elemento, il modulo delle proprietà dell'elemento passa alla modalità di aggiornamento di massa, così puoi modificare in un'unica volta gli attributi degli elementi selezionati. Per farlo, contrassegna l'attributo usando la casella di controllo e inserisci un nuovo valore. Qui puoi usare le macro (ad esempio, {HOST.NAME} per l'etichetta dell'elemento).



Collegamento degli elementi

Una volta posizionati alcuni elementi sulla mappa, è il momento di iniziare a collegarli. Per collegare due elementi, devi prima selezionarli. Con gli elementi selezionati, fai clic su *Aggiungi* accanto a Link.

Una volta creato un link, il modulo del singolo elemento contiene ora una sezione aggiuntiva *Link*. Fai clic su *Modifica* per modificare gli attributi del link.



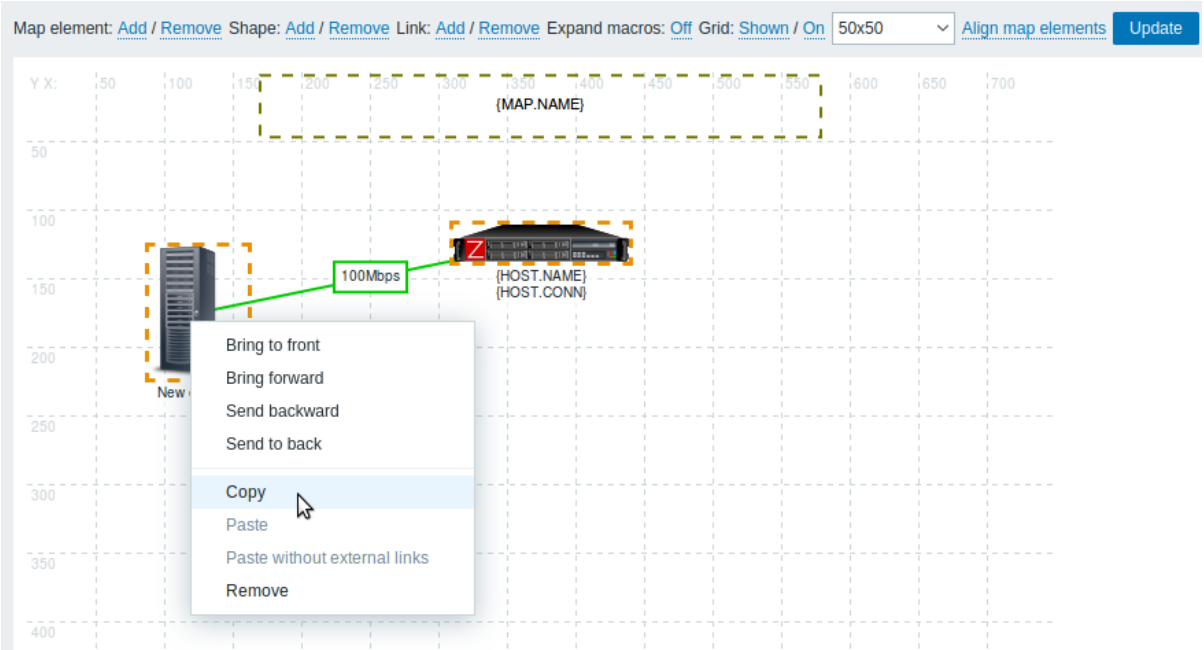
Attributi del link:

Parameter	Description
Label	Etichetta che verrà visualizzata sopra il link. In questo campo sono supportate le espressioni macros, ma solo con le funzioni `avg`, `last`, `min` e `max`, con il tempo come parametro (ad esempio, `{?avg(/host/key,1h)}`).
Show label	Seleziona come visualizzare l'etichetta del link: **Default** - usa le impostazioni *Show map element labels* dalla configurazione della mappa (predefinito); **Always** - visualizzata sempre (predefinito); **Auto hide** - nasconde l'etichetta se non è sotto il puntatore o selezionata.
Connect to	L'elemento a cui il link si connette.
Type (OK)	Stile predefinito del link. Seleziona lo stile: *Line*, *Bold line*, *Dot* o *Dashed line*.
Color (OK)	Colore predefinito del link. Fai clic sul campione colore per selezionare un altro colore.
Indicator type	Seleziona il tipo di indicatore del link: **Static link** - nessun indicatore; **Trigger** - consente indicatori del link basati su trigger; **Item value** - consente indicatori del link basati sul valore di item.
Item	Seleziona l'item. Il valore di questo item influenzerà lo stile del link. Questo campo è disponibile quando *Item value* è selezionato per *Indicator type*.
Indicators	L'elenco degli indicatori del link. Questo campo è disponibile quando *Indicator type* è impostato su *Trigger* o *Item value*. Se impostato su *Trigger*, i trigger possono essere assegnati come indicatori del link. Quando un trigger è nello stato *Problem*, il suo stile (colore e tipo di linea selezionati) verrà applicato al link. Se impostato su *Item value* e un item è selezionato nel campo *Item*, i valori dell'item possono essere assegnati come indicatori del link. Quando il valore dell'item raggiunge la soglia specificata (per un item numerico) o corrisponde al pattern (per un item testuale), il suo stile (colore e tipo di linea selezionati) viene applicato al link.

Spostamento e copia-incolla degli elementi

Diversi elementi selezionati possono essere **spostati** in un'altra posizione nella mappa facendo clic su uno degli elementi selezionati, tenendo premuto il pulsante del mouse e spostando il cursore nella posizione desiderata.

Uno o più elementi possono essere **copiati** selezionando gli elementi, quindi facendo clic su un elemento selezionato con il pulsante destro del mouse e scegliendo *Copy* dal menu.

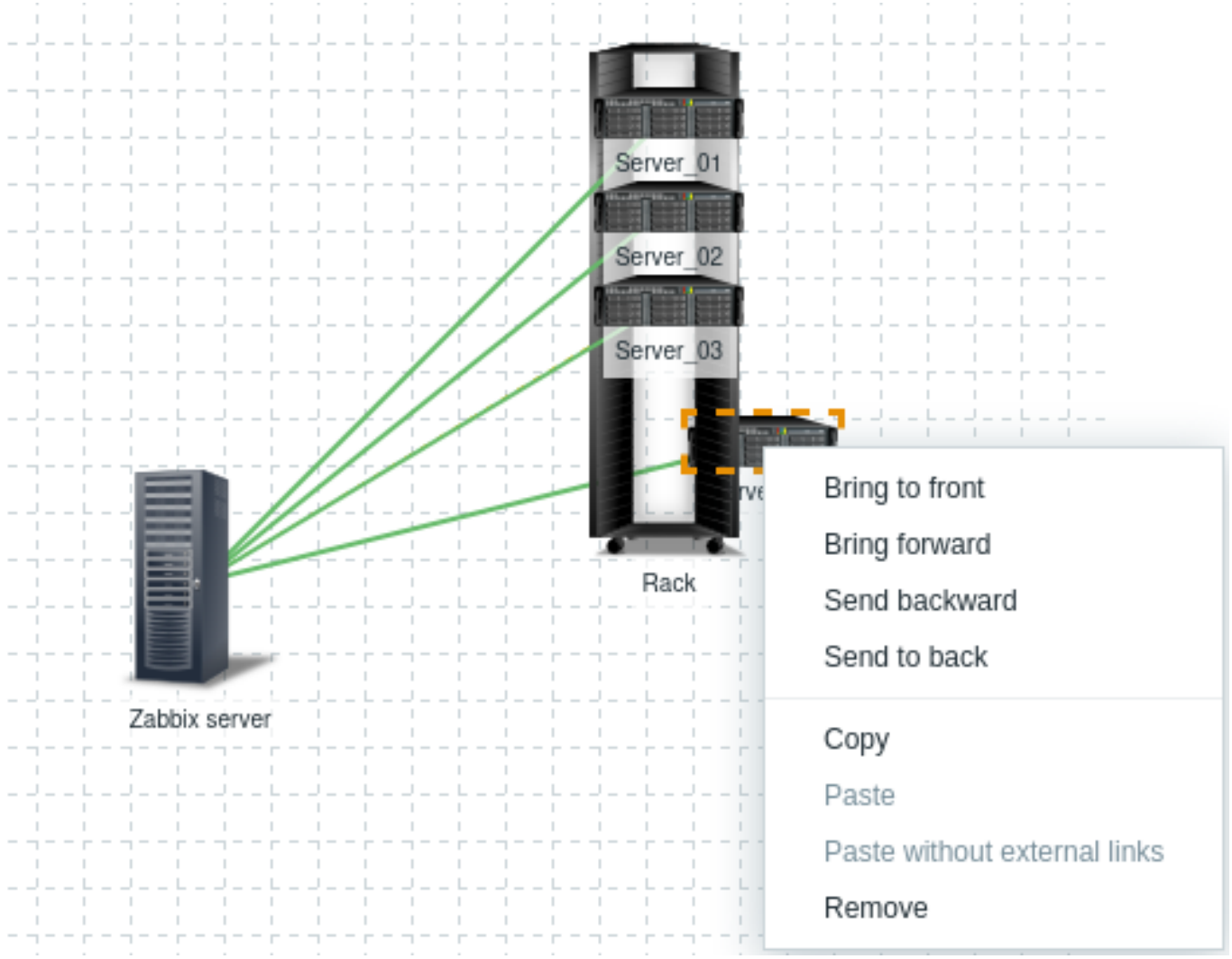


Per incollare gli elementi, fare clic su un'area della mappa con il pulsante destro del mouse e selezionare *Paste* dal menu. L'opzione *Paste without external links* incollerà gli elementi mantenendo solo i collegamenti presenti tra gli elementi selezionati.

La funzione di copia-incolla funziona all'interno della stessa finestra del browser. Le scorciatoie da tastiera non sono supportate.

Ordinamento degli elementi

Per portare un elemento davanti all'altro (o viceversa), fai clic con il pulsante destro del mouse sull'elemento e seleziona *Porta avanti*/*Porta in primo piano* oppure *Porta indietro*/*Porta sullo sfondo*.

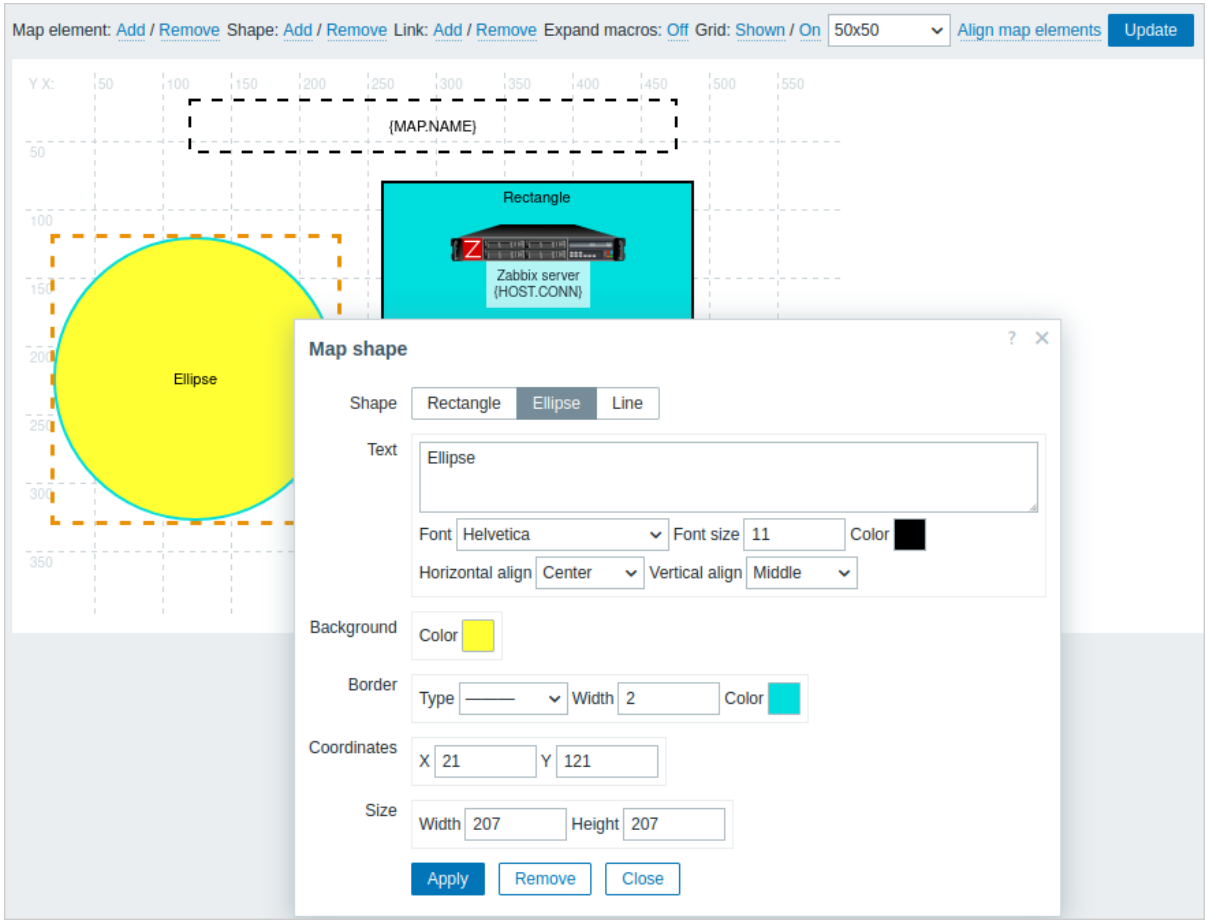


Aggiunta di forme

Oltre agli elementi della mappa, è anche possibile aggiungere alcune forme. Le forme non sono elementi della mappa; sono solo una rappresentazione visiva. Per esempio, una forma rettangolare può essere usata come sfondo per raggruppare alcuni host. È possibile aggiungere forme rettangolari ed ellittiche.

Per aggiungere una forma, fai clic su *Add* accanto a Shape. La nuova forma apparirà nell'angolo in alto a sinistra della mappa. Trascinala e rilasciala dove preferisci.

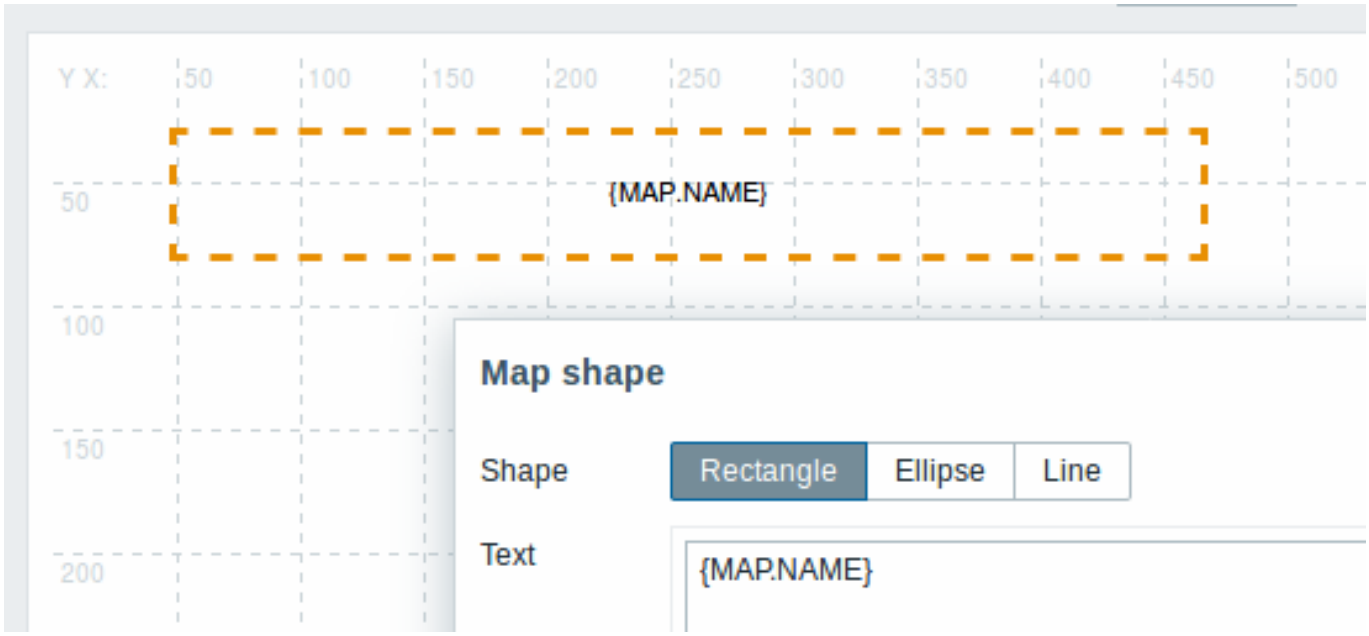
Una nuova forma viene aggiunta con i colori predefiniti. Facendo clic sulla forma, viene visualizzato un modulo e puoi personalizzare l'aspetto della forma, aggiungere testo, ecc.



Per selezionare le forme, selezionane una e poi tieni premuto *Ctrl* per selezionare le altre. Con più forme selezionate, le proprietà comuni possono essere aggiornate in massa, analogamente agli elementi.

È possibile aggiungere testo nelle forme. Nel testo sono supportate le macro di espressione, ma solo con le funzioni `avg`, `last`, `min` e `max`, con il tempo come parametro (ad esempio, `{?avg(/host/key,1h)}`).

Per visualizzare solo il testo, la forma può essere resa invisibile rimuovendo il bordo della forma (seleziona 'None' nel campo *Border*). Per esempio, nota come la macro {MAP.NAME}, visibile nello screenshot sopra, sia in realtà una forma rettangolare con testo, che si può vedere facendo clic sulla macro:



{MAP.NAME} viene risolta nel nome della mappa configurato quando si visualizza la mappa.

Se nel testo vengono usati collegamenti ipertestuali, diventano cliccabili durante la visualizzazione della mappa.

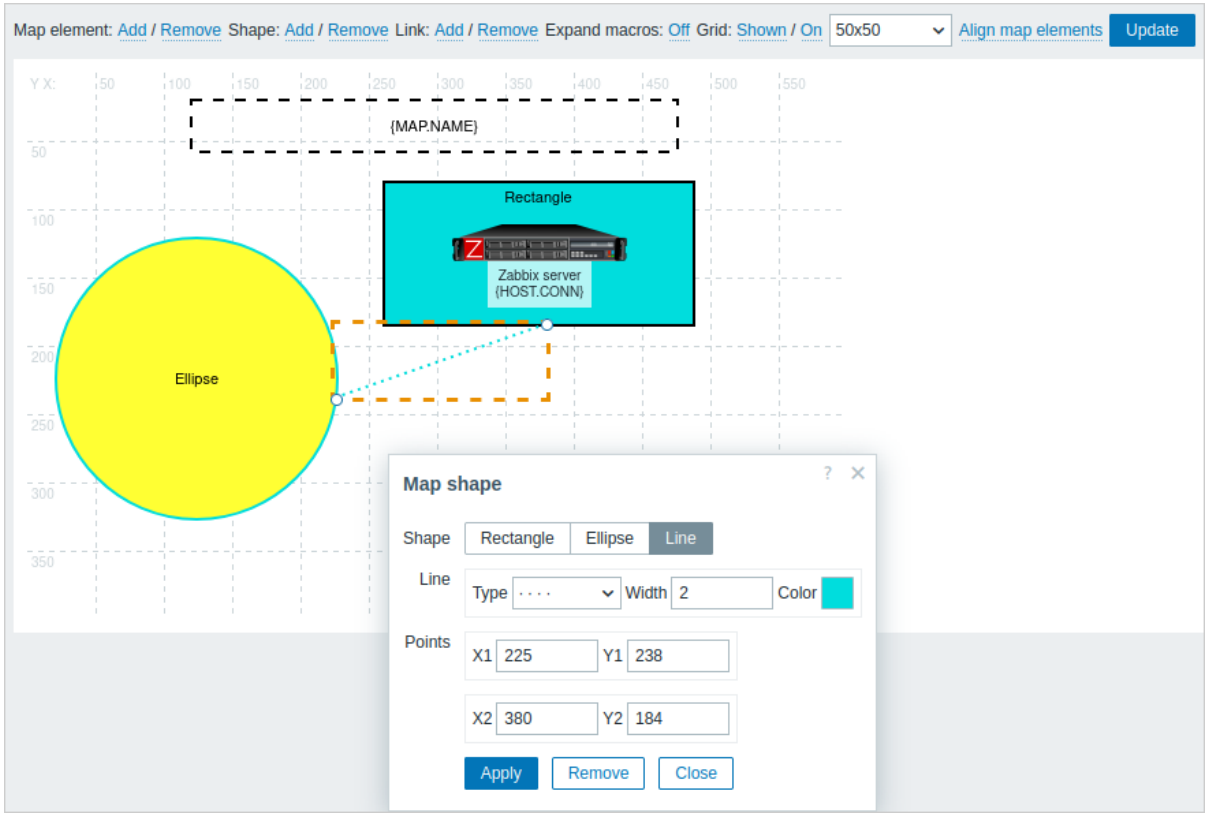
L'andata a capo del testo è sempre attiva all'interno delle forme. Tuttavia, all'interno di un'ellisse, le righe vengono mandate a capo come se l'ellisse fosse un rettangolo. La suddivisione delle parole non è implementata, quindi le parole lunghe (parole che non

rientrano nella forma) non vengono mandate a capo, ma vengono mascherate (nella pagina di modifica della mappa) o ritagliate (altre pagine con mappe).

Aggiunta di linee

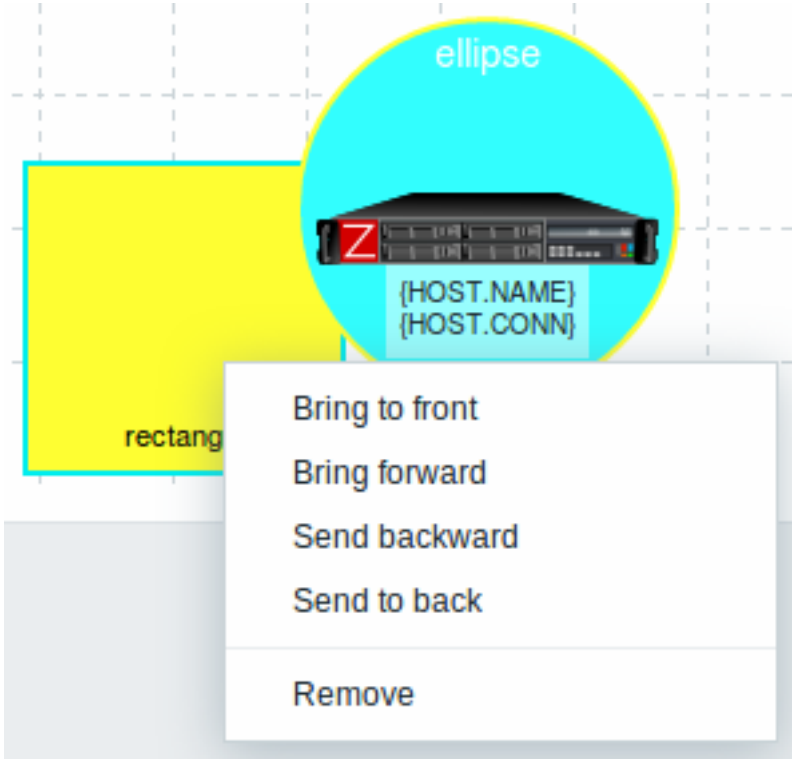
Oltre alle forme, è anche possibile aggiungere alcune linee. Le linee possono essere utilizzate per collegare elementi o forme in una mappa.

Per aggiungere una linea, fai clic su *Add* accanto a Shape. Una nuova forma apparirà nell'angolo in alto a sinistra della mappa. Selezionala e fai clic su *Line* nel modulo di modifica per trasformare la forma in una linea. Quindi regola le proprietà della linea, come tipo di linea, larghezza, colore, ecc.



Ordinamento di forme e linee

Per portare una forma davanti all'altra (o viceversa), fai clic sulla forma con il tasto destro del mouse per aprire il menu della forma della mappa.

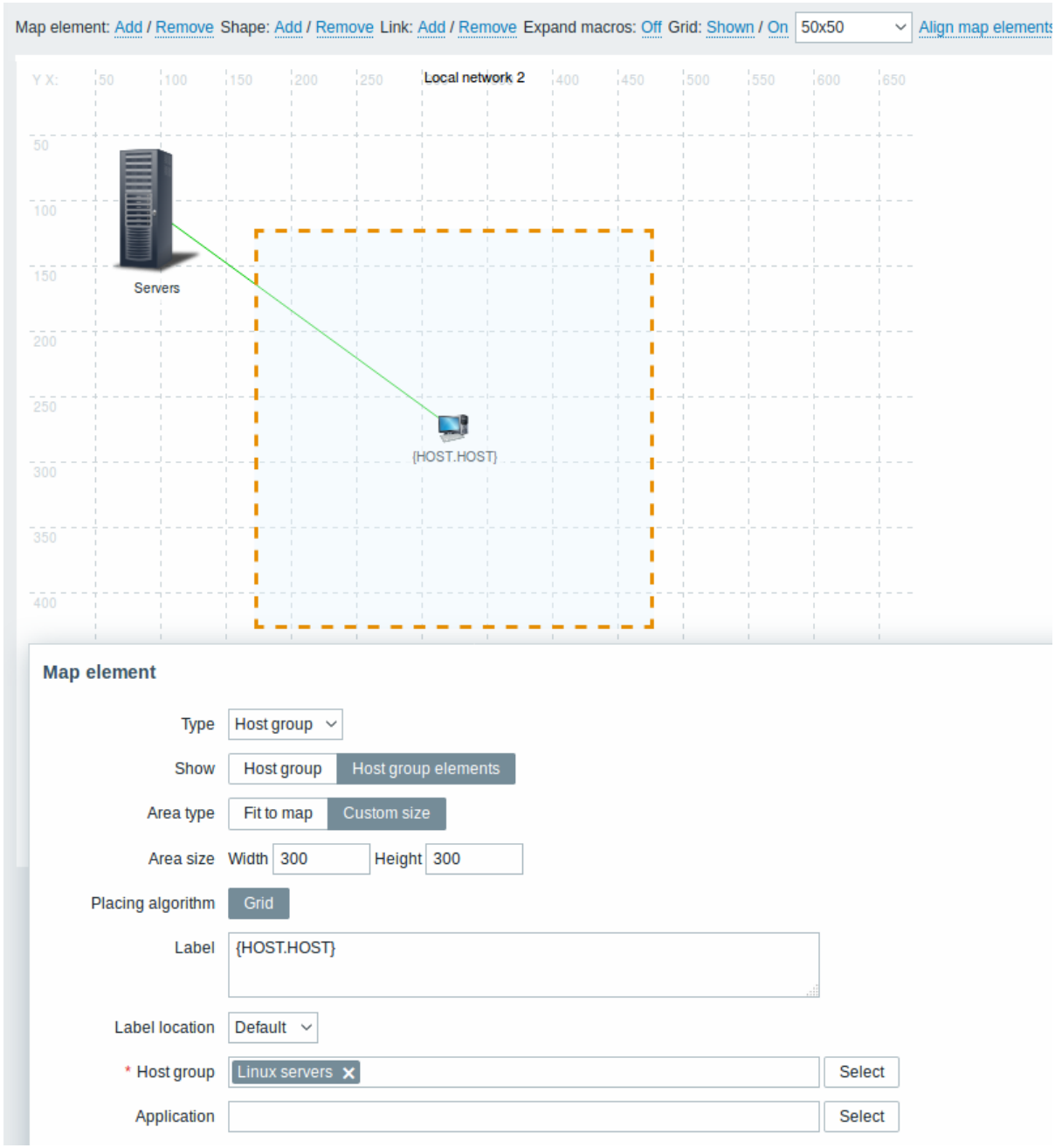


Elementi del gruppo host

Panoramica

Questa sezione spiega come aggiungere un elemento di tipo *Host group* durante la configurazione di una mappa di rete.

Configurazione



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

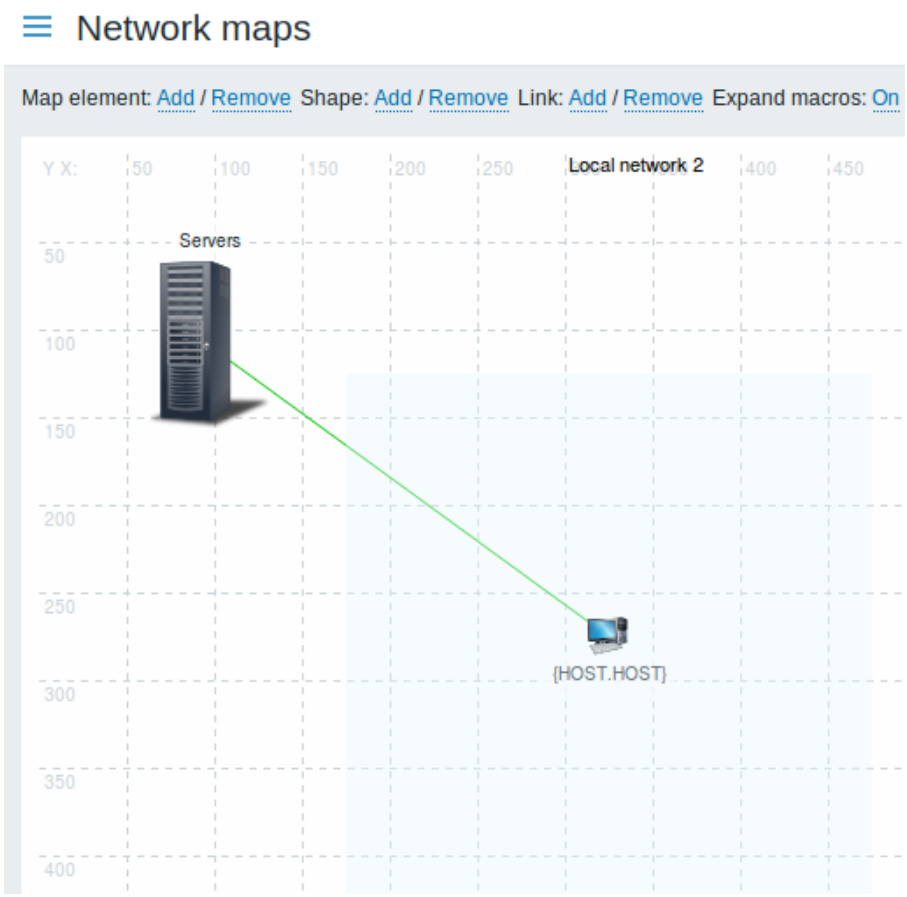
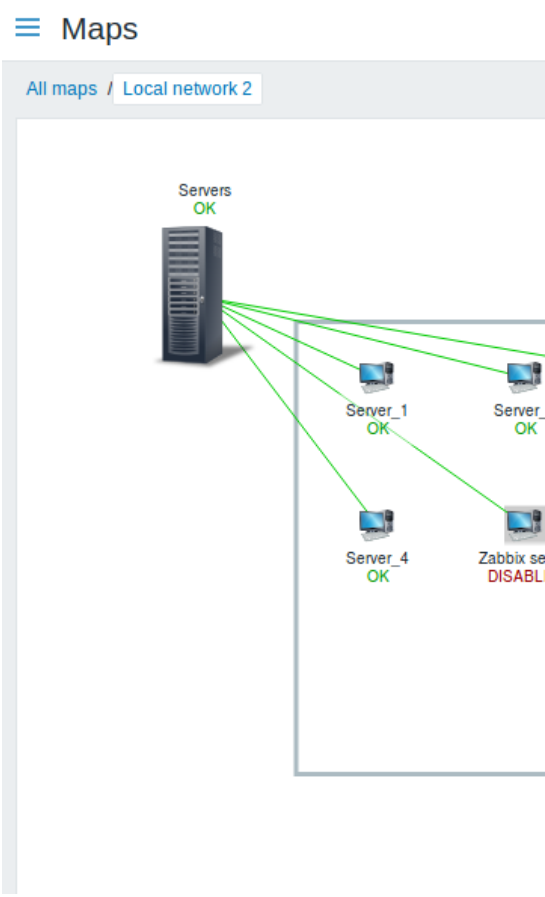
Questa tabella è composta da parametri tipici per il tipo di elemento *Host group*:

Parametro	Descrizione
Type	Selezionare il tipo di elemento: **Host group** - Icona che rappresenta lo stato di tutti i trigger di tutti gli host appartenenti al gruppo selezionato.
Show	Opzioni di visualizzazione: **Host group** - Selezionando questa opzione verrà visualizzata un'unica icona contenente le informazioni corrispondenti a un determinato gruppo di host. **Host group elements** - Selezionando questa opzione verranno visualizzate più icone contenenti le informazioni corrispondenti a ogni singolo elemento (host) del determinato gruppo di host.
Area type	Questa impostazione è disponibile se *Show* è impostato su *Host group elements*: **Fit to map** - Tutti gli elementi del gruppo di host vengono posizionati in modo uniforme all'interno della mappa. **Custom size** - Impostazione manuale dell'area della mappa per tutti gli elementi del gruppo di host da visualizzare.
Area size	Questa impostazione è disponibile se *Area type* è impostato su *Custom size*: **Width** - Valore in pixel che specifica la larghezza dell'area della mappa. **Height** - Valore in pixel che specifica l'altezza dell'area della mappa.

Parametro	Descrizione
Placing algorithm	**Grid** - Unica opzione disponibile per visualizzare tutti gli elementi del gruppo di host.
Label	Etichetta dell'icona, qualsiasi stringa. Nelle etichette è possibile utilizzare macro e stringhe multilinea. Se il tipo di elemento della mappa è *Host group*, la specifica di determinate macro influisce sulla visualizzazione della mappa, mostrando le informazioni corrispondenti a ogni singolo host. Ad esempio, se viene utilizzata la macro `{HOST.IP}`, la visualizzazione di modifica della mappa mostrerà solo la macro `{HOST.IP}` stessa, mentre la visualizzazione della mappa includerà e mostrerà l'indirizzo IP univoco di ciascun host.

Visualizzazione degli elementi del gruppo di host

Questa opzione è disponibile se è selezionata l'opzione di visualizzazione *Elementi del gruppo di host*.
Quando si seleziona *Elementi del gruppo di host* come opzione *mostra*, inizialmente viene visualizzata una sola icona per il gruppo di host.
Tuttavia, dopo aver salvato la mappa e aver aperto la visualizzazione della mappa, si noterà che la mappa include tutti gli elementi (host) del gruppo di host specifico:

Visualizzazione della modifica della mappa	Visualizzazione della mappa
	

Si noti come viene utilizzata la macro `{HOST.HOST}`.
Nella modifica della mappa, il nome della macro non viene risolto, mentre nella visualizzazione della mappa vengono visualizzati i nomi univoci degli host.

Indicatori di collegamento

Panoramica

Puoi assegnare indicatori a un collegamento tra elementi in una mappa di rete.

Gli indicatori possono essere basati su trigger o valori di item. È possibile visualizzare uno stile e un colore del collegamento diversi:

- quando i trigger passano a uno stato di problema;

- quando il valore dell'item:
  - raggiunge una soglia (per gli item numerici);
  - corrisponde a un'espressione regolare (per gli item di testo).

Quando configuri un collegamento, imposti il tipo e il colore predefiniti del collegamento. Assegnando indicatori a un collegamento, diventa possibile fare in modo che lo stile e il colore del collegamento dipendano dallo stato del trigger o dal valore dell'item.

Ad esempio, se uno qualsiasi dei trigger assegnati passa a uno stato di problema, lo stile e il colore del collegamento cambieranno per rifletterlo. Quindi, il tuo collegamento predefinito potrebbe essere una linea verde. Ora, con il trigger nello stato di problema, il collegamento può diventare rosso e in grassetto (se lo hai definito così).

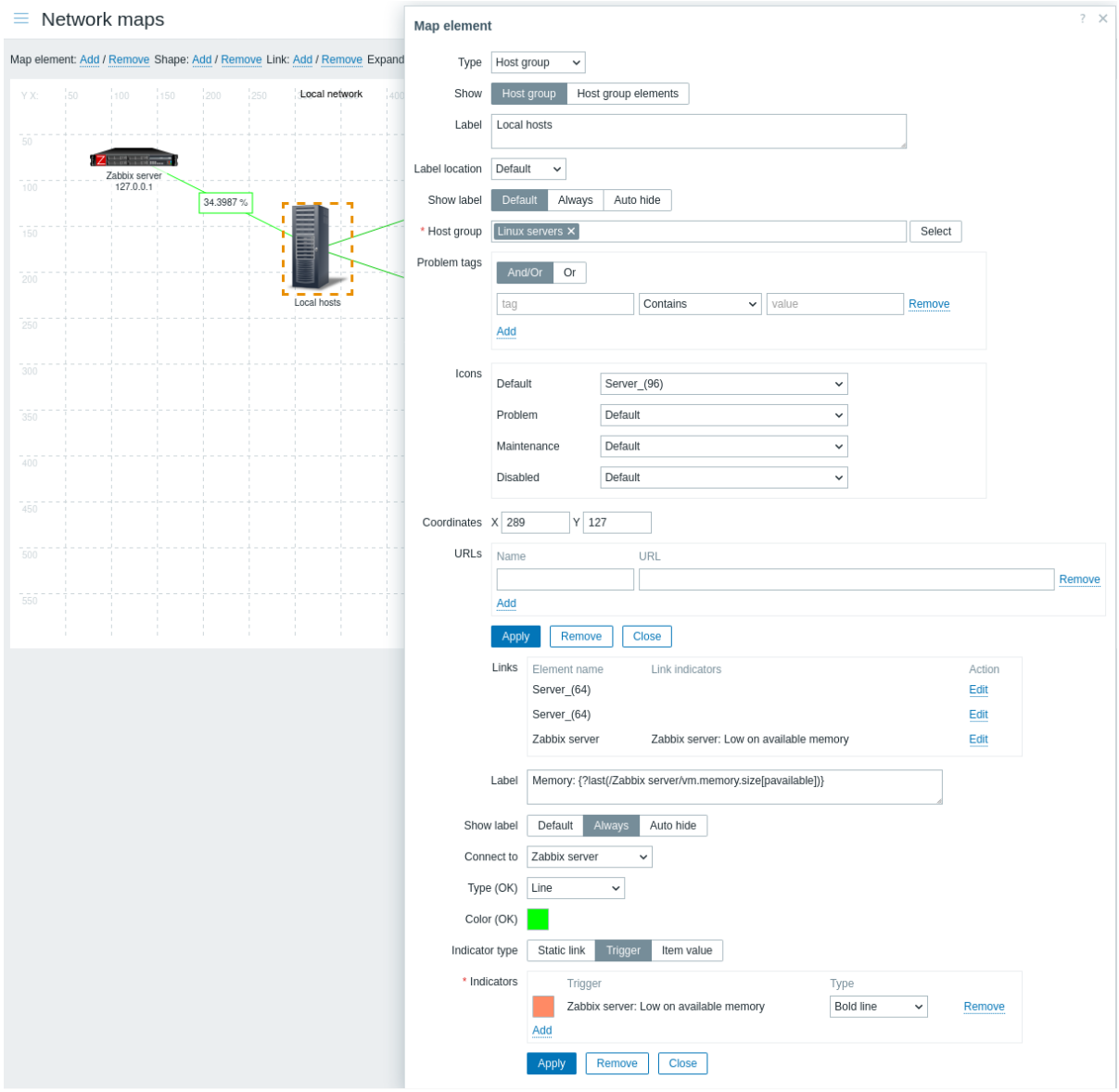
Allo stesso modo, se il valore di un item raggiunge una soglia specificata o corrisponde a un'espressione regolare specificata, lo stile del collegamento può rifletterlo.

Configurazione

Trigger

Per assegnare i trigger come indicatori del collegamento, procedere come segue:

- selezionare un elemento della mappa
- fare clic su *Edit* nella sezione *Links* per il collegamento appropriato
- selezionare *Trigger* come tipo di indicatore
- fare clic su *Add* nel blocco *Indicators* e selezionare uno o più trigger

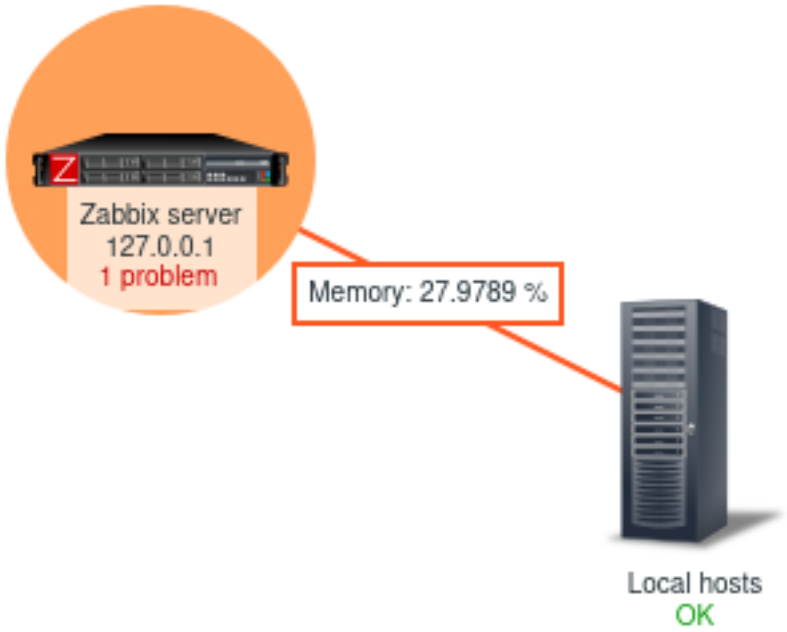


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I trigger aggiunti sono visibili nell'elenco *Indicators*.

È possibile impostare direttamente dall'elenco il tipo di collegamento e il colore per ciascun indicatore. Al termine, fare clic su *Apply*, chiudere il modulo e fare clic su *Update* per salvare le modifiche alla mappa.

In *Monitoring* > *Maps* il relativo tipo di collegamento e colore verranno visualizzati se il **trigger** entra in uno stato di problema.



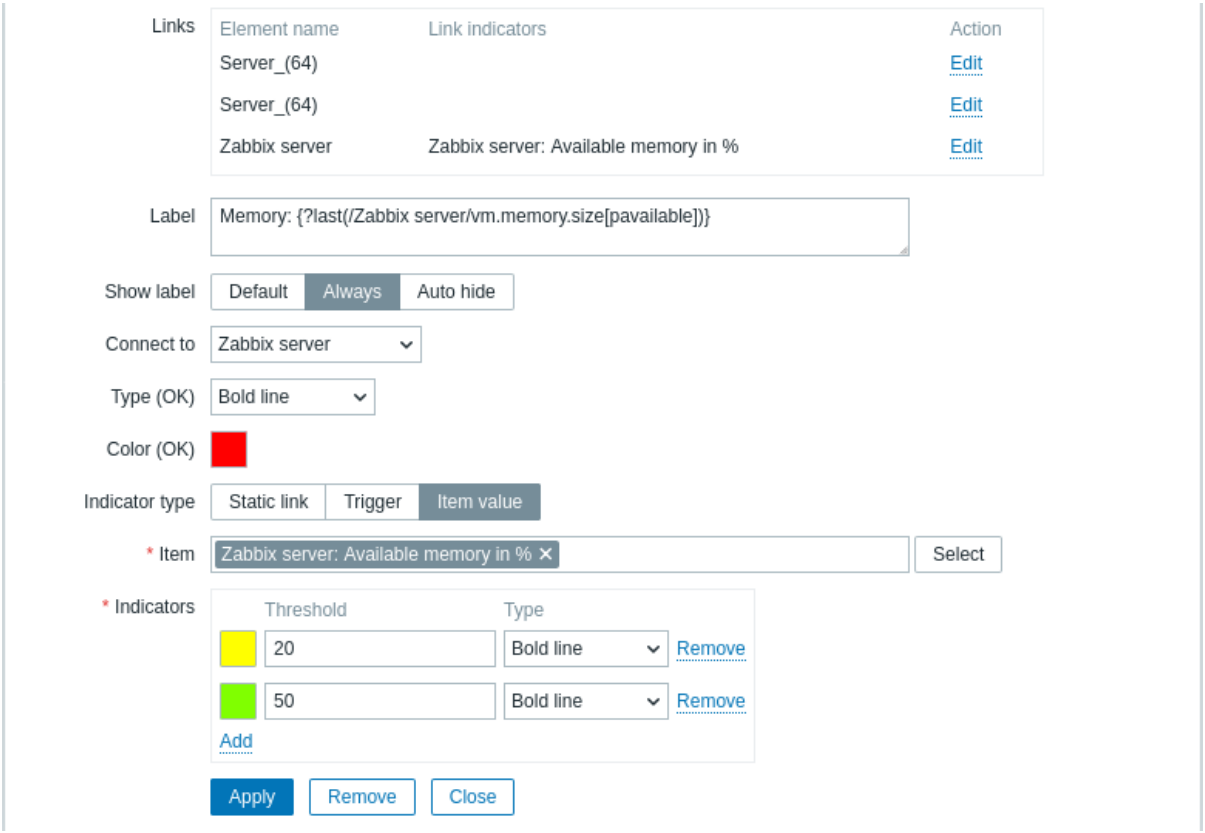
Note:
Se più trigger entrano in uno stato di problema, il problema con la gravità più alta determinerà lo stile e il colore del collegamento. Se più trigger con la stessa gravità sono assegnati allo stesso collegamento della mappa, ha la precedenza quello con l'ID più basso. Si noti inoltre che:

1. Le impostazioni *Minimum trigger severity* e *Show suppressed problem* della configurazione della mappa influiscono su quali problemi vengono presi in considerazione.
2. Nel caso di trigger con più problemi (generazione di più problemi), ciascun problema può avere una gravità diversa dalla gravità del trigger (modificata manualmente), può avere tag diversi (a causa delle macro) ed essere soppresso.

Valori degli item

Per assegnare i valori degli item come indicatori del collegamento, procedere come segue:

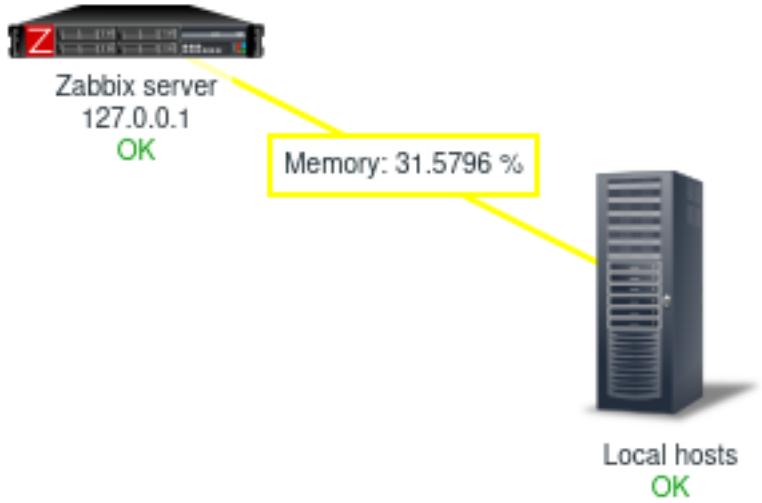
- selezionare un elemento della mappa
- fare clic su *Edit* nella sezione *Links* per il collegamento appropriato
- selezionare *Item value* come tipo di indicatore
- selezionare l'item
- aggiungere una o più soglie o pattern del valore dell'item nel blocco *Indicators*



Le soglie/pattern degli item aggiunti sono visibili nell'elenco *Indicators*.

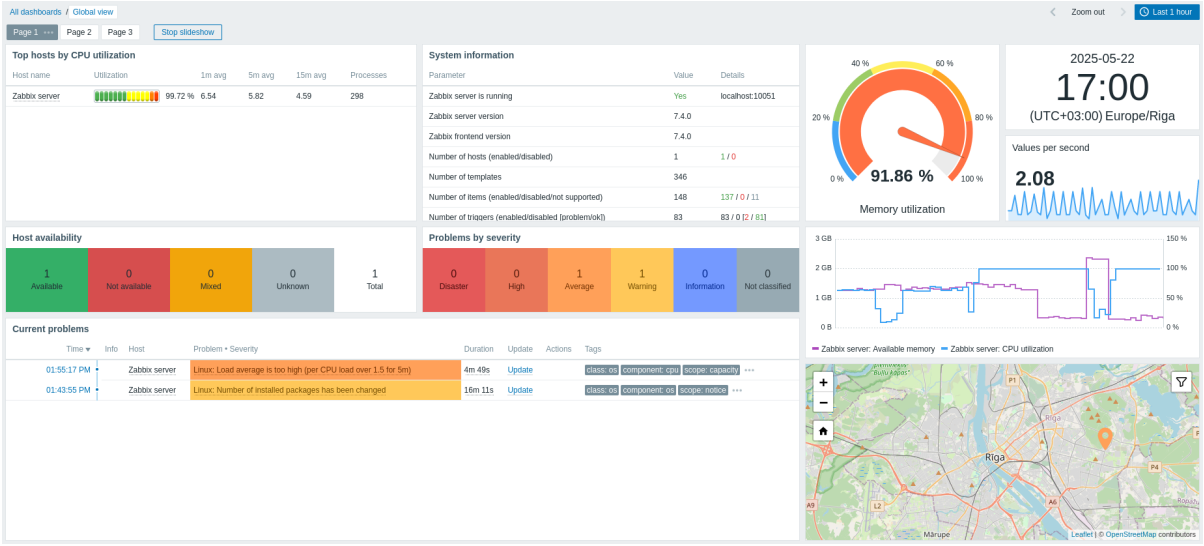
È possibile impostare direttamente dall'elenco il tipo di collegamento e il colore per ciascun indicatore. Al termine, fare clic su *Apply*, chiudere il modulo e fare clic su *Update* per salvare le modifiche alla mappa.

In *Monitoring* > *Maps* il relativo tipo di collegamento e colore verranno visualizzati se il **valore dell'item** raggiunge la soglia impostata (per i tipi di dati numerici) oppure corrisponde al pattern dell'espressione regolare (per i tipi di dati testuali).



Dashboard

I dashboard - sia i global dashboards sia i host dashboards - offrono una solida piattaforma di visualizzazione con widget e strumenti come grafici moderni, mappe, presentazioni e altro ancora.



Template e gruppi di template

Panoramica

L'uso dei template è un modo eccellente per ridurre il carico di lavoro e semplificare la configurazione di Zabbix. Un template è un insieme di entità che può essere applicato comodamente a più host.

Le entità possono essere:

- item
- trigger
- grafici
- dashboard
- regole di low-level discovery
- scenari web

Poiché nella pratica molti host sono identici o piuttosto simili, ne consegue naturalmente che l'insieme di entità (item, trigger, grafici,...) creato per un host può essere utile per molti altri. Naturalmente, potresti copiarle su ogni nuovo host, ma ciò richiederebbe molto lavoro manuale. Invece, con i template puoi copiarle in un template e poi applicare il template a tutti gli host necessari.

Quando un template è collegato a un host, tutte le entità (item, trigger, grafici,...) del template vengono aggiunte all'host. I template vengono assegnati direttamente a ciascun singolo host (e non a un gruppo di host).

I template vengono spesso utilizzati per raggruppare entità relative a servizi o applicazioni specifici (come Apache, MySQL, PostgreSQL, Postfix...) e poi applicati agli host che eseguono tali servizi.

Un altro vantaggio dell'uso dei template si ha quando è necessario modificare qualcosa per tutti gli host. Modificando una sola volta qualcosa a livello di template, la modifica verrà propagata a tutti gli host collegati.

I template sono organizzati in gruppi di template.

Procedi con la creazione e configurazione di un template.

Template predefiniti pronti all'uso

Panoramica

Zabbix fornisce un insieme in continua crescita di template preconfigurati per semplificare e velocizzare la configurazione degli obiettivi di monitoraggio.

Tutti i template pronti all'uso sono disponibili in *Raccolta dati* > *Template*.

Quando si aggiorna Zabbix, i template non vengono aggiornati automaticamente. In questo modo si evita di sovrascrivere eventuali modifiche personalizzate. Per aggiornare o aggiungere manualmente i template, vedere la sezione *Aggiornamento dei template* riportata di seguito.

Utilizzare la barra laterale per accedere alle informazioni sui tipi specifici di template e sui requisiti operativi.

I dashboard globali progettati per funzionare con i template Zabbix esistenti possono essere scaricati dal repository dei dashboard e importati nel proprio ambiente Zabbix. Una descrizione dettagliata di ogni dashboard, inclusi i prerequisiti, è disponibile nel file `README` del dashboard.

Vedere anche:

- Collegamento di un template
- Problemi noti relativi ai template

Aggiornamento dei template

Per aggiornare o aggiungere un **singolo template**:

1. Aprire la directory `templates` nel repository Git di Zabbix.

2. Scegliere il branch (sotto l'etichetta *Source*) corrispondente alla propria versione di Zabbix (ad esempio, *master* per la versione in sviluppo).

3. Individuare il template da aggiungere o aggiornare (ad esempio, MySQL by Zabbix agent).

4. Copiare il contenuto del file del template (ad esempio, `template_db_mysql_agent.yaml`) e salvarlo in un file locale.

5. Nell'interfaccia utente di Zabbix, andare a *Raccolta dati* > *Template* e importare il file locale.

> **Note:**
> Anche i tipi di supporto webhook possono essere aggiornati in modo analogo: recuperarli dalla directory `media` e importarli in *Avvisi* > *Tipi di supporto*.

Per aggiornare o aggiungere **più template**:

1. Scaricare l'archivio dei bundle di template in base al proprio web server:

- Nginx - `templates_bundles-8.0-1M-nginx.zip`
- Apache - `templates_bundles-8.0-2M-apache.zip`

2. Estrarre l'archivio. Contiene file YAML, ciascuno con più template, e un file indice che elenca i template presenti in ogni file YAML.

3. Nell'interfaccia utente di Zabbix, andare a *Raccolta dati* > *Template* e importare i file YAML uno alla volta.

> **Note:**
> Anche i tipi di supporto webhook possono essere aggiornati in modo analogo: prelevare il file `media_bundle-8.0-000.yaml` dall'archivio dei bundle di template e importarlo in *Avvisi* > *Tipi di supporto*.

Per aggiornare o aggiungere **tutti i template** contemporaneamente, è possibile utilizzare lo script `import_templates.sh` incluso nell'archivio dei bundle di template. Lo script richiede le utilità `curl` e `jq` e utilizza il metodo dell'API Zabbix `configuration.import`. Quando si esegue lo script, specificare la directory contenente i file YAML, l'URL dell'API Zabbix e il token di autenticazione.

> **Warning:**
> Lo script `import_templates.sh` sovrascrive tutti i template esistenti, senza offrire la possibilità di esaminare o annullare le modifiche ai template.

Esempio:

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

Compatibilità del template con Host Wizard

Host Wizard è compatibile con tutti i template disponibili immediatamente.

È inoltre possibile rendere i propri template personalizzati compatibili con Host Wizard:

1. Nel frontend di Zabbix, accedere a *Raccolta dati* > *Template* ed esportare il template personalizzato.

2. Modificare il file esportato:

- Aggiungere l'elemento template `wizard_ready` per indicare la compatibilità con Host Wizard.

- Facoltativamente, aggiungere l'elemento template `readme` o `config` secondo necessità.

3. Nel frontend di Zabbix, accedere a *Raccolta dati* > *Template* e importare il file del template aggiornato.

Operazioni del template Zabbix agent

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche con Zabbix agent:

1. Assicurarsi che Zabbix agent sia installato sull'host. Per i controlli attivi, assicurarsi inoltre che l'indirizzo del server/proxy Zabbix sia aggiunto al parametro 'ServerActive' del file di configurazione dell'agent.
2. Collegare il template a un host di destinazione (se il template non è disponibile nella propria installazione di Zabbix, potrebbe essere necessario importare prima il template).
3. Se necessario, modificare i valori delle macro del template.
4. Configurare l'istanza monitorata per consentire la condivisione dei dati con Zabbix.

Una descrizione dettagliata di un template, incluso l'elenco completo di macro, item e trigger, è disponibile nel file README del template (accessibile facendo clic sul nome del template).

Sono disponibili i seguenti template:

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- MariaDB by Zabbix agent
- MariaDB by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- Percona by Zabbix agent
- Percona by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- Podman by Zabbix agent
- Podman by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

Operazione del template Zabbix agent 2

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche con Zabbix agent 2:

1. Assicurarsi che agent 2 sia installato sull'host e che la versione installata contenga il plugin richiesto. In alcuni casi, potrebbe essere necessario aggiornare prima agent 2.
2. Collegare il template a un host di destinazione (se il template non è disponibile nella tua installazione di Zabbix, potrebbe essere necessario importare prima il template).
3. Se necessario, modificare i valori delle macro del template. Nota che le macro utente possono essere usate per sovrascrivere i parametri di configurazione.
4. Configurare l'istanza monitorata per consentire la condivisione dei dati con Zabbix.

> **Attention:**
> I template di Zabbix agent 2 funzionano in combinazione con i plugin. Mentre la configurazione di base può essere eseguita semplicemente modificando le macro utente, una personalizzazione più approfondita può essere ottenuta configurando direttamente il plugin. Ad esempio, se un plugin supporta sessioni denominate, è possibile monitorare più entità dello stesso tipo (ad esempio, MySQL1 e MySQL2) specificando una sessione denominata con URI, nome utente e password propri per ciascuna entità nel file di configurazione.

Una descrizione dettagliata di un template, inclusa l'elenco completo di macro, item e trigger, è disponibile nel file README del template (accessibile facendo clic sul nome del template).

Sono disponibili i seguenti template:

- 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

Operazione del template HTTP

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche con l'HTTP agent:

1. Creare un host in Zabbix corrispondente all'oggetto monitorato. Specificare l'indirizzo IP o il nome DNS dell'oggetto monitorato come interfaccia principale dell'host. Questo è necessario affinché la macro `{HOST.CONN}` venga risolta correttamente negli item del template.
2. Collegare il template all'host creato nel passaggio 1 (se il template non è disponibile nell'installazione di Zabbix, potrebbe essere necessario importarlo prima).
3. Se necessario, modificare i valori delle macro a livello di template.
4. Configurare l'istanza monitorata per consentire la condivisione dei dati con Zabbix.

Una descrizione dettagliata di un template, inclusa l'elenco completo delle macro, degli item e dei trigger, è disponibile nel file `README` del template (accessibile facendo clic sul nome del template).

Sono disponibili i seguenti template:

- Acronis Cyber Protect Cloud by HTTP
- Apache by HTTP
- Asterisk by HTTP
- AWS by HTTP
- AWS Cost Explorer by HTTP
- AWS EC2 by HTTP
- AWS ECS Cluster by HTTP
- AWS ECS Serverless Cluster by HTTP
- AWS ELB Application Load Balancer by HTTP
- AWS ELB Network Load Balancer by HTTP
- AWS Lambda by HTTP
- AWS RDS instance by HTTP
- AWS S3 bucket by HTTP
- Azure by HTTP
- Cisco Meraki organization by HTTP
- Cisco SD-WAN by HTTP
- Cisco Secure Firewall Threat Defense by HTTP
- Claude API by HTTP
- ClickHouse by HTTP
- Cloudflare by HTTP
- CockroachDB by HTTP
- Control-M enterprise manager by HTTP
- Control-M server by HTTP
- Cradlepoint NCM v2 by HTTP
- DELL PowerEdge R720 by HTTP
- DELL PowerEdge R740 by HTTP
- DELL PowerEdge R820 by HTTP
- DELL PowerEdge R840 by HTTP
- Domain RDAP by HTTP
- Elasticsearch Cluster by HTTP

- Envoy Proxy by HTTP
- Etcd by HTTP
- FortiGate by HTTP
- GitHub repository by HTTP
- GitLab by HTTP
- GLPI by HTTP
- Google Cloud Platform (GCP) by HTTP
- Hadoop by HTTP
- HAProxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP
- HashiCorp Nomad by HTTP
- HashiCorp Vault by HTTP
- Hikvision camera by HTTP
- HPE iLO by HTTP
- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP
- HPE Synergy by HTTP
- InfluxDB by HTTP
- Jenkins by HTTP
- Kubernetes API server by HTTP, Kubernetes cluster state by HTTP, Kubernetes Controller manager by HTTP, Kubernetes kubelet by HTTP, Kubernetes nodes by HTTP, Kubernetes Scheduler by HTTP sono deprecati e sono stati sostituiti dai template *Kubernetes Cluster by HTTP* e *Kubernetes Node by HTTP*.
- Kubernetes Cluster by HTTP
- Kubernetes Node by HTTP
- MantisBT by HTTP
- Microsoft 365 reports by HTTP
- Microsoft SharePoint by HTTP
- NetApp AFF A700 by HTTP
- Nextcloud by HTTP
- NGINX by HTTP
- NGINX Plus by HTTP
- Nutanix Prism Element by HTTP
- OpenAI Platform by HTTP
- OpenStack by HTTP
- OpenWeatherMap by HTTP
- Oracle Cloud by HTTP
- Palo Alto PA-440 by HTTP
- PHP-FPM by HTTP
- Podman by HTTP
- Proxmox VE by HTTP
- Pure Storage FlashArray v1 and v2 by HTTP
- RabbitMQ cluster by HTTP
- Ribbon SBC Edge by HTTP
- Ribbon SBC SWe CE by HTTP
- Ribbon SBC SWe Core by HTTP
- TiDB by HTTP
- TiDB PD by HTTP
- TiDB TiKV by HTTP
- Travis CI by HTTP
- Veeam Backup Enterprise Manager by HTTP
- Veeam Backup and Replication by HTTP
- VeloCloud SD-WAN by HTTP
- VeloCloud SD-WAN Edge by HTTP
- YugabyteDB by HTTP
- ZooKeeper by HTTP

Operazione del template IPMI

I template IPMI non richiedono alcuna configurazione specifica. Per iniziare il monitoraggio, collega il template a un host di destinazione (se il template non è disponibile nella tua installazione di Zabbix, potrebbe essere necessario importare prima il template).

Una descrizione dettagliata di un template, inclusa l'elenco completo di macro, item e trigger, è disponibile nel file README del template (accessibile facendo clic sul nome di un template).

Template disponibile:

- Chassis by IPMI

Operazione del template JMX

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche tramite JMX:

1. Assicurarsi che Zabbix Java gateway sia installato e configurato correttamente.
2. Collegare il template all'host di destinazione. L'host deve avere l'interfaccia JMX configurata. Se il template non è disponibile nella vostra installazione di Zabbix, potrebbe essere necessario importare prima il template.
3. Se necessario, modificare i valori delle macro del template.
4. Configurare l'istanza monitorata per consentire la condivisione dei dati con Zabbix.

Una descrizione dettagliata di un template, inclusa l'elenco completo di macro, item e trigger, è disponibile nel file README del template (accessibile facendo clic sul nome di un template).

Sono disponibili i seguenti template:

- 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

Operazione del template ODBC

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche tramite monitoraggio ODBC:

1. Assicurarsi che il driver ODBC richiesto sia installato su server o proxy Zabbix.
2. Collegare il template a un host di destinazione (se il template non è disponibile nella vostra installazione di Zabbix, potrebbe essere necessario importare prima il template).
3. Se necessario, modificare i valori delle macro del template.
4. Configurare l'istanza monitorata per consentire la condivisione dei dati con Zabbix.

Una descrizione dettagliata di un template, inclusa l'elenco completo di macro, item e trigger, è disponibile nel file README del template (accessibile facendo clic sul nome del template).

Sono disponibili i seguenti template:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

Template standardizzati per dispositivi di rete

Panoramica

Per fornire il monitoraggio dei dispositivi di rete come switch e router, abbiamo creato due cosiddetti modelli: uno per il dispositivo di rete stesso (in pratica il suo chassis) e uno per l'interfaccia di rete.

Sono disponibili template per molte famiglie di dispositivi di rete. Tutti i template coprono, ove possibile ottenere questi elementi dal dispositivo:

- Monitoraggio dei guasti dello chassis (alimentatori, ventole e temperatura, stato generale)
- Monitoraggio delle prestazioni dello chassis (item di CPU e memoria)
- Raccolta dell'inventario dello chassis (numeri di serie, nome del modello, versione del firmware)

- Monitoraggio delle interfacce di rete con IF-MIB ed EtherLike-MIB (stato dell'interfaccia, carico del traffico dell'interfaccia, stato duplex per Ethernet)

Se il template non è disponibile nella tua installazione di Zabbix, potrebbe essere necessario importare prima il template.

Se stai importando i nuovi template predefiniti, potresti voler aggiornare anche l'espressione regolare globale `@Network interfaces for discovery` in:

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

per filtrare le interfacce loopback e null sulla maggior parte dei sistemi.

Dispositivi

Elenco delle famiglie di dispositivi per cui sono disponibili template:

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certificato
Aruba CX 8300s by SNMP	HPE Aruba Networking	Serie switch Aruba CX	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certificato
Brocade FC SNMP	Brocade	Switch FC Brocade	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Prestazioni Guasto
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certificato
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Prestazioni Guasto
Check Point Next Generation Firewall by SNMP	Check Point	Firewall di nuova generazione	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certificato
Ciena 3906 by SNMP	Ciena	CPE Ciena	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certificato

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
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	Certificato
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certificato
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T e 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certificato
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	Certificato
D-Link DES_DGS Switch SNMP	D-Link	Switch DES/DGX	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certificato
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Prestazioni Guasto Interfacce
Dell Force S-Series SNMP	Dell	Dell Force serie S	S4810		F10-S-SERIES-CHASSIS-MIB	Certificato
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certificato
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Prestazioni Inventario
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certificato
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Certificato

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
Huawei AR600 by SNMP	Huawei	Router Huawei NetEngine serie AR600	Router Huawei serie AR600/AR6100/AR6200/AR6300 con software VRP (V300R019/V300R021/V300R023 o successivo)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificato
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado con software V5		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificato
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500/5600/6800/18500/18800 con software V6/V700		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificato
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certificato
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Dispositivi Infiniband Intel/QLogic	Infiniband 12300		ICS-CHASSIS-MIB	Guasto Inventario
Juniper SNMP	Juniper	Modelli MX,SRX,EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certificato
Juniper MX NETCONF	Juniper	Modelli MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Non applicabile (usa NETCONF su SSH)*	Certificato
Juniper MX SNMP	Juniper	Modelli MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certificato
Mellanox SNMP	Mellanox	Dispositivi Infiniband Mellanox	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certificato

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Routers (serie CCR)	Sono disponibili template dedicati separati per 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	Certificato

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (serie CRS)	Sono disponibili template dedicati separati per 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	Certificato
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (serie CSS)	Sono disponibili template dedicati separati per MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificato
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificato
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Sono disponibili template dedicati separati per MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificato

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	Tag
MikroTik netPower <device model> SNMP	MikroTik	MikroTik netPower	Sono disponibili template dedicati separati per MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificato
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik PowerBox	Sono disponibili template dedicati separati per MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificato
MikroTik RB<device model> SNMP	MikroTik	Router MikroTik serie RB	Sono disponibili template dedicati separati per 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	Certificato
MikroTik SNMP	MikroTik	Dispositivi MikroTik RouterOS	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificato
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Guasto Inventario
QTech QSW SNMP	QTech	Dispositivi Qtech	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Prestazioni Inventario

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
Stormshield SNS by SNMP	Stormshield	Firewall Stormshield Network Security (SNS)	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Certificato
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Prestazioni Inventario
Ubiquiti AirOS SNMP	Ubiquiti	Dispositivi wireless Ubiquiti AirOS	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Prestazioni
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	Prestazioni Inventario

Progettazione dei template

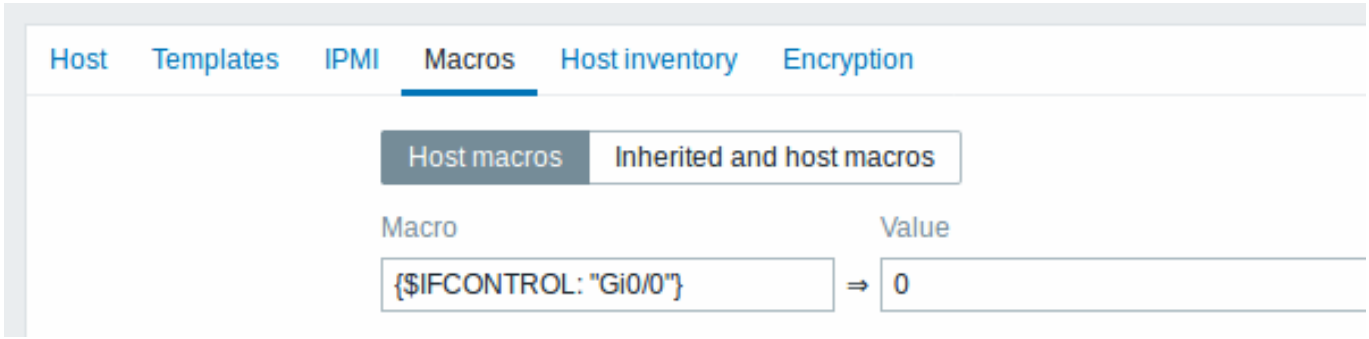
I template sono stati progettati tenendo presenti i seguenti aspetti:

- Le macro utente vengono utilizzate il più possibile, in modo che i trigger possano essere regolati dall'utente;
- Il low-level discovery viene utilizzato il più possibile per ridurre al minimo il numero di item non supportati;
- Tutti i template dipendono da Template ICMP Ping, quindi tutti i dispositivi vengono controllati anche tramite ICMP;
- Gli item non utilizzano alcun MIB: negli item e nei low-level discovery vengono usati gli OID SNMP. Pertanto, non è necessario caricare alcun MIB in Zabbix affinché i template funzionino;
- Le interfacce di rete loopback vengono filtrate durante il rilevamento, così come le interfacce con ifAdminStatus = down(2)
- Ove possibile, vengono utilizzati contatori a 64 bit da IF-MIB::ifXTable. Se non sono supportati, vengono invece utilizzati i contatori predefiniti a 32 bit.

Tutte le interfacce di rete rilevate hanno un trigger che ne monitora lo stato operativo (link), ad esempio:

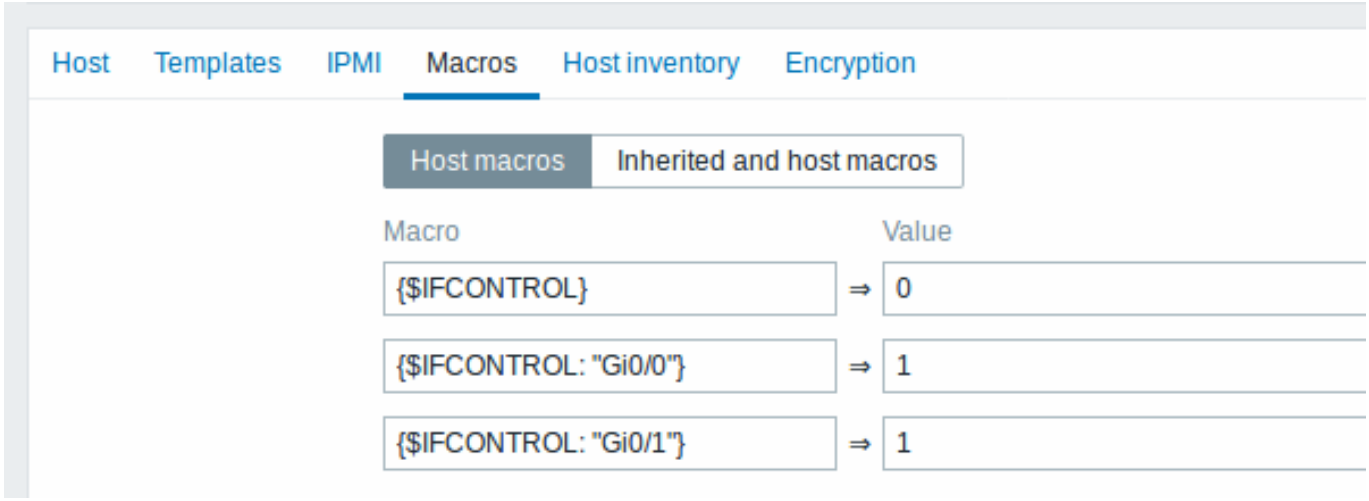
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- Se non si desidera monitorare questa condizione per una specifica interfaccia, creare una macro utente con contesto con valore 0. Ad esempio:



dove Gi0/0 è {#IFNAME}. In questo modo il trigger non verrà più utilizzato per questa specifica interfaccia.

- È anche possibile modificare il comportamento predefinito in modo che tutti i trigger non si attivino e attivare questo trigger solo per un numero limitato di interfacce, come gli uplink:



Tag

- Prestazioni - le MIB della famiglia di dispositivi forniscono un modo per monitorare gli item di CPU e memoria;
- Guasti - le MIB della famiglia di dispositivi forniscono un modo per monitorare almeno un sensore di temperatura;
- Inventario - le MIB della famiglia di dispositivi forniscono un modo per raccogliere almeno il numero di serie del dispositivo e il nome del modello;
- Certificato - tutte e tre le categorie principali sopra indicate sono coperte.

Operazione del template VMware

Panoramica

Zabbix include una serie di template pronti all'uso per il monitoraggio delle macchine virtuali, progettati per gli hypervisor VMware vCenter ed ESXi. I template disponibili sono contenuti in due set di template separati:

- VMware - utilizza dati UUID per le macro corrispondenti
- VMware FQDN - utilizza dati FQDN per le macro corrispondenti

I template progettati per il monitoraggio degli hypervisor VMware vCenter o ESXi contengono regole di low-level discovery preconfigurate insieme a vari controlli integrati per il monitoraggio delle installazioni virtuali.

Note:
Per il corretto funzionamento del template *VMware FQDN*, ogni macchina virtuale monitorata deve avere un nome del sistema operativo univoco conforme alle regole FQDN. Inoltre, VMware Tools/Open Virtual Machine Tools deve essere installato su ogni macchina. Se questi prerequisiti sono soddisfatti, si consiglia di utilizzare il template *VMware FQDN*. Il template *VMware FQDN* è disponibile da Zabbix 5.2 con l'introduzione della possibilità di creare host con interfacce personalizzate.

 È disponibile anche un classico template *VMware*, che può essere utilizzato se i requisiti FQDN non sono soddisfatti. Tuttavia, il template *VMware* presenta un problema noto. Gli host per le macchine virtuali rilevate vengono creati con i nomi salvati in vCenter (ad esempio, "VM1", "VM2", ecc.). Se Zabbix agent è installato su questi host ed è abilitata l'autoregistrazione attiva di Zabbix agent, il processo di autoregistrazione leggerà i nomi host così come sono stati registrati all'avvio (ad esempio, "vm1.example.com", "vm2.example.com", ecc.). Ciò può portare alla creazione di nuovi host per macchine virtuali esistenti (poiché non è stata trovata alcuna corrispondenza di nome), con conseguente duplicazione degli host con nomi diversi.

Tenere presente quanto segue:

- Il template *VMware Hypervisor* può essere collegato manualmente a un host e utilizzato anche nel discovery;
- Il template *VMware Guest* non deve essere collegato manualmente a un host e può essere utilizzato solo nel discovery.

Configurazione delle macro host

Per utilizzare i controlli semplici VMware, l'host deve avere definite le seguenti macro utente:

- {$VMWARE.URL} - URL dell'SDK del servizio VMware (vCenter o hypervisor ESXi) (https://servername/sdk)
- {$VMWARE.USERNAME} - nome utente del servizio VMware
- {$VMWARE.PASSWORD} - password dell'utente {$VMWARE.USERNAME} del servizio VMware

Operatività del template OTLP

Zabbix fornisce una serie di template pronti all'uso per raccogliere e monitorare tracce, metriche e log OpenTelemetry utilizzando gli item *Telemetry query*.

Passaggi per garantire il corretto funzionamento dei template:

1. Controllare il file `README` del template per verificare eventuali requisiti di configurazione specifici del template.
2. Creare un host in Zabbix corrispondente alla destinazione del monitoraggio.
3. Assicurarsi che il parametro `TelemetryProvider` sia configurato nel file di configurazione del server Zabbix o del proxy Zabbix.
4. Collegare il template all'host creato (se il template non è disponibile nell'installazione Zabbix, potrebbe essere necessario importare prima il template).
5. Se necessario, modificare i valori delle macro a livello di template.

Una descrizione dettagliata dei template, inclusa la lista completa di macro, item e trigger, è disponibile nel file `README` del template, accessibile facendo clic sul nome del template.

Sono disponibili i seguenti template:

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

Notifiche sugli eventi

Panoramica

Supponendo di aver configurato alcuni item e trigger e di ricevere ora degli eventi come risultato del cambiamento di stato dei trigger, è tempo di considerare alcune azioni.

Per cominciare, non vorremmo dover osservare continuamente l'elenco dei trigger o degli eventi. Sarebbe molto meglio ricevere una notifica se è accaduto qualcosa di significativo (come un problema). Inoltre, quando si verificano problemi, vorremmo vedere che tutte le persone coinvolte siano informate.

Per questo motivo, l'invio di notifiche è una delle azioni principali offerte da Zabbix. È possibile definire chi deve essere notificato e quando in seguito a un determinato evento.

Per poter inviare e ricevere notifiche da Zabbix è necessario:

- definire alcuni supporti
- configurare un'azione che invii un messaggio a uno dei supporti definiti

Le azioni sono costituite da *condizioni* e *operazioni*. In pratica, quando le condizioni sono soddisfatte, vengono eseguite le operazioni. Le due operazioni principali sono l'invio di un messaggio (notifica) e l'esecuzione di un comando remoto.

Per gli eventi creati da discovery e autoregistrazione, sono disponibili alcune operazioni aggiuntive. Queste includono l'aggiunta o la rimozione di un host, il collegamento di un template, ecc.

Tipi di media

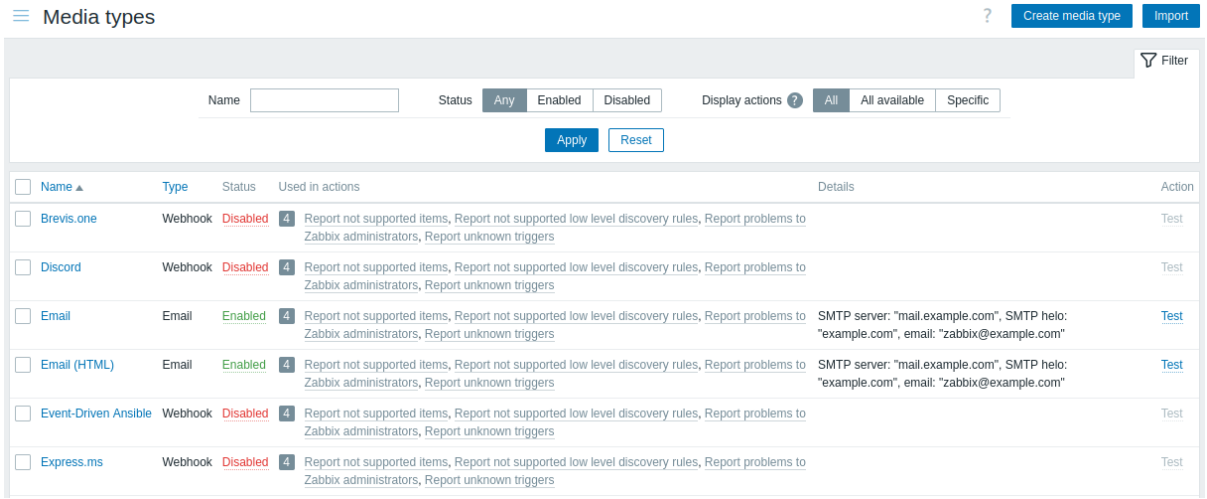
Panoramica

I media type sono i canali di distribuzione utilizzati per inviare notifiche e avvisi da Zabbix.

I media type supportano i seguenti metodi di distribuzione:

- E-mail
- Notifica push
- SMS
- Script personalizzato
- Webhook

I media type vengono gestiti in *Avvisi* > *Media type*. Alcuni media type sono predefiniti nel dataset predefinito. È sufficiente perfezionarne i parametri per renderli operativi.



Per vedere come i media type si inseriscono nel processo di generazione degli avvisi, esaminiamo i tre requisiti necessari per distribuire le notifiche da Zabbix agli utenti finali:

1. Deve essere definita un'operazione dell'azione che invii le notifiche.
2. Deve essere definito un media type funzionante, ad esempio *E-mail*, che invii gli avvisi utilizzando SMTP.
3. I dettagli di distribuzione a livello utente, come indirizzi e-mail, numeri di telefono e così via, devono essere definiti nei media dell'utente.

Test del tipo di media

Per verificare se un tipo di media configurato funziona, fare clic su *Test* nell'elenco dei tipi di media.

La richiesta di test verrà inviata al server Zabbix. Il server Zabbix tenterà di inviare un avviso utilizzando il tipo di media specificato e restituirà il risultato al frontend. Il frontend attenderà che il server restituisca i risultati. Per impostazione predefinita, il test del tipo di media ha un timeout di 65 secondi (configurabile in *Amministrazione* > *Generale* > *Timeout*).

Vedere anche i dettagli dei test per:

- Email
- Notifica push
- Webhook
- Script

Configurazione

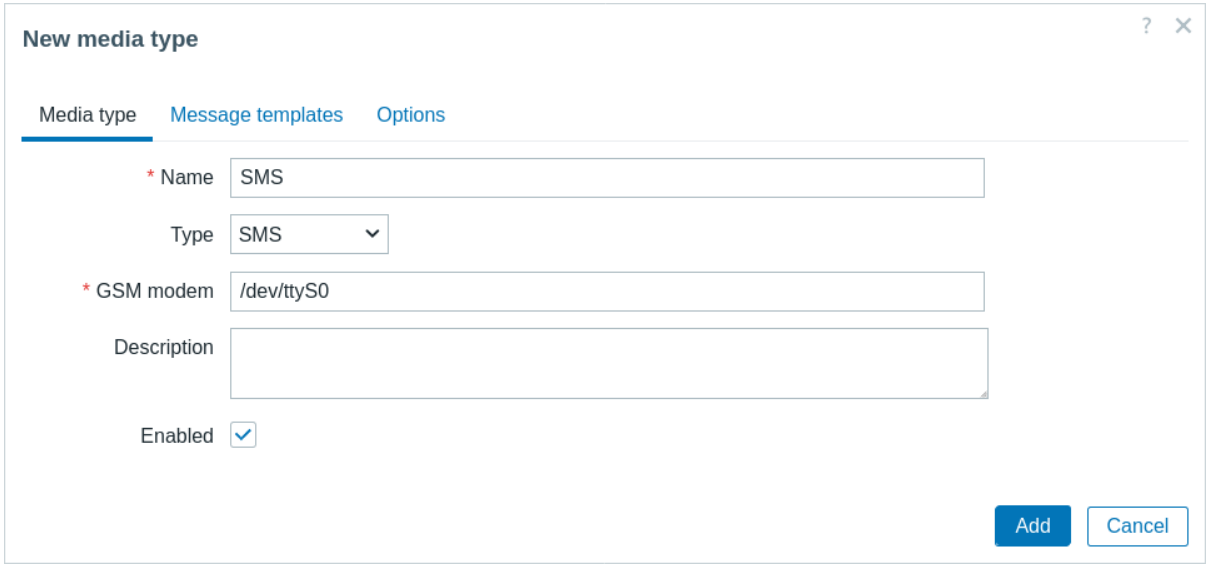
Per creare un tipo di supporto nel frontend di Zabbix:

1. Vai a *Avvisi* > *Tipi di supporto*.

2. Fai clic su *Crea tipo di supporto*.
3. Inserisci i parametri del tipo di supporto nel modulo.

Alcuni parametri sono **comuni** a tutti i metodi di consegna.

Note:
Se uno o più metodi di consegna sono specificati nel flag delle funzionalità `$ZBX_FEATURE_FLAGS['media_type_denylist']` nel file di configurazione del frontend (`zabbix.conf.php`), tali metodi di consegna non sono disponibili nell'elenco *Tipo*.

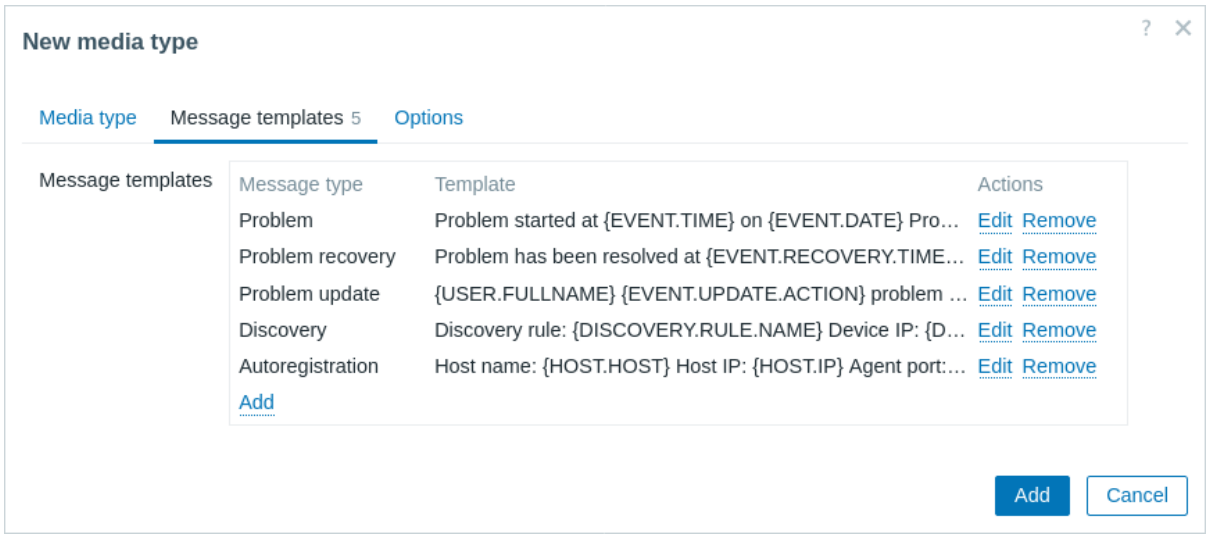


Parametro	Descrizione
Nome	Nome del tipo di supporto.
Tipo	Seleziona il metodo di consegna per il tipo di supporto.
Descrizione	Inserisci una descrizione per il tipo di supporto.
Abilitato	Seleziona la casella di controllo per abilitare il tipo di supporto.

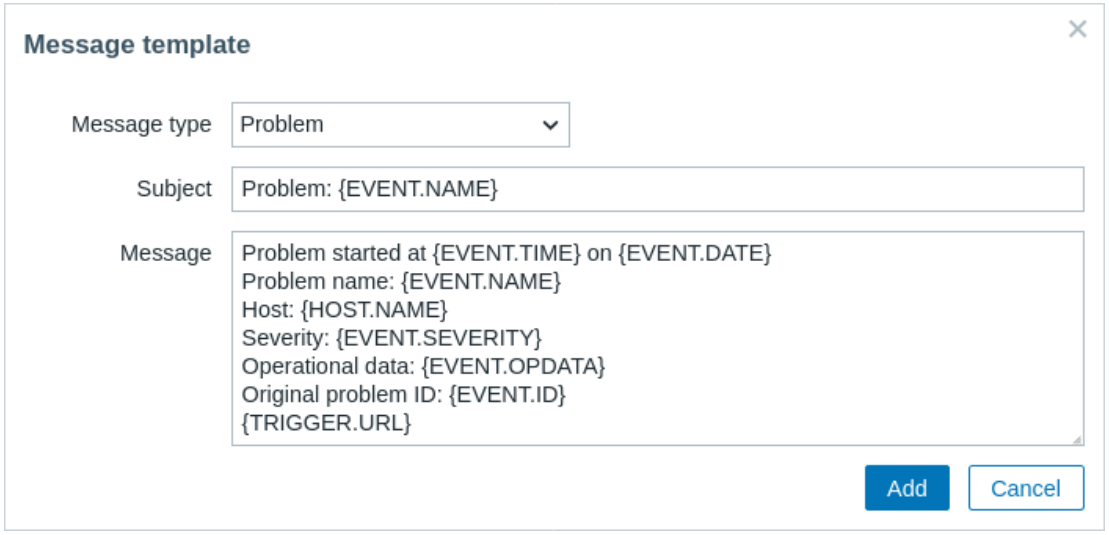
Per i parametri specifici del metodo, consulta le pagine relative a email, notifica push, SMS, alertscript personalizzato o webhook.

Template dei messaggi

La scheda **Template dei messaggi** contiene i messaggi predefiniti per tipo di evento (problema, ripristino del problema, discovery, ecc.).



Fai clic su **Aggiungi** per definire un messaggio predefinito (oppure su **Modifica** per aggiornare un messaggio esistente):



Parameter	Description
Message type	Tipo di evento per il quale deve essere utilizzato il messaggio predefinito. Per ogni tipo di evento può essere definito un solo messaggio predefinito.
Subject	Oggetto del messaggio predefinito. L'oggetto può contenere macro. È limitato a 255 caratteri. L'oggetto non è disponibile per il tipo di media SMS.
Message	Il messaggio predefinito. È limitato a un certo numero di caratteri a seconda del tipo di database (vedi Invio dei messaggi per ulteriori informazioni). Il messaggio può contenere macro supportate. Nei messaggi di problema e di aggiornamento del problema, sono supportate le macro di espressione (ad esempio, `{?avg(/host/key,1h)}`).

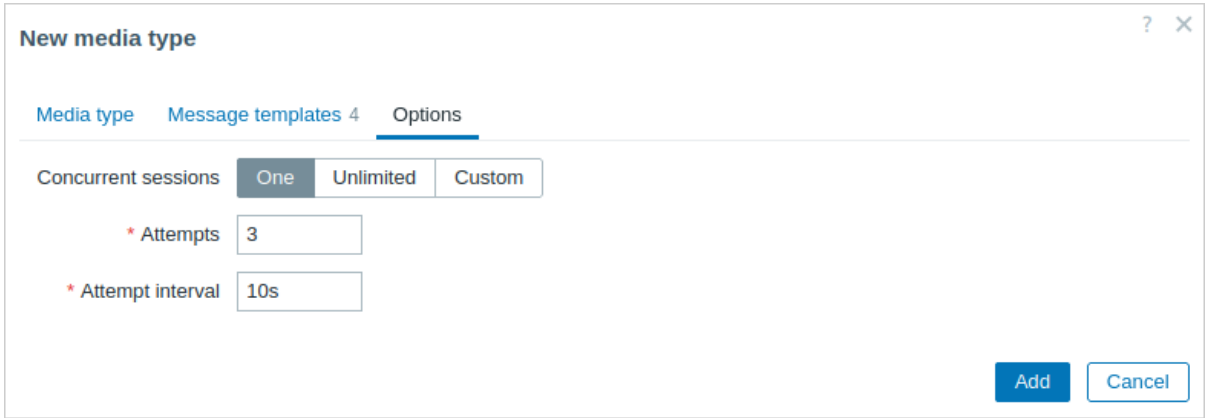
Nota che i messaggi predefiniti vengono sovrascritti dai messaggi personalizzati, se definiti nelle operazioni dell'azione.

Warning:
La definizione dei template dei messaggi è obbligatoria per tutti i metodi di consegna, inclusi webhook o script di avviso personalizzati che non usano messaggi predefiniti per le notifiche. Ad esempio, l'azione "Send message to Pushover webhook" non riuscirà a inviare notifiche di problema, se il messaggio di problema per il webhook Pushover non è definito.

Opzioni

La scheda **Opzioni** contiene le impostazioni per l'elaborazione degli avvisi.
Lo stesso insieme di opzioni è configurabile per ogni tipo di media.

Tutti i tipi di media vengono elaborati in parallelo.
Sebbene il numero massimo di sessioni simultanee sia configurabile per ogni tipo di media,
il numero totale di processi alerter sul server può essere limitato solo dal parametro StartAlerters.
Gli avvisi generati da un singolo trigger vengono elaborati in sequenza.
Pertanto, più notifiche possono essere elaborate simultaneamente solo se sono generate da più trigger.



Parametro	Descrizione
Sessioni simultanee	Selezionare il numero di sessioni alerter parallele per il tipo di media: **Una** - una sessione **Illimitate** - numero illimitato di sessioni **Personalizzato** - selezionare un numero personalizzato di sessioni Valori illimitati/elevati indicano un numero maggiore di sessioni parallele e una capacità aumentata per l'invio delle notifiche. I valori illimitati/elevati devono essere utilizzati in ambienti di grandi dimensioni, dove potrebbe essere necessario inviare simultaneamente molte notifiche. Se è necessario inviare più notifiche rispetto al numero di sessioni simultanee, le notifiche rimanenti verranno accodate; non andranno perse.
Tentativi	Numero di tentativi per l'invio di una notifica. È possibile specificare fino a 100 tentativi; il valore predefinito è '3'. Se viene specificato '1', Zabbix invierà la notifica una sola volta e non effettuerà nuovi tentativi se l'invio non riesce.
Intervallo tra i tentativi	Frequenza con cui si tenta di inviare nuovamente una notifica nel caso in cui l'invio non sia riuscito, espressa in secondi (0-3600). Se viene specificato '0', Zabbix effettuerà nuovamente il tentativo immediatamente. Sono supportati i suffissi temporali (ad esempio, `30s`, `1m`, `1h`).

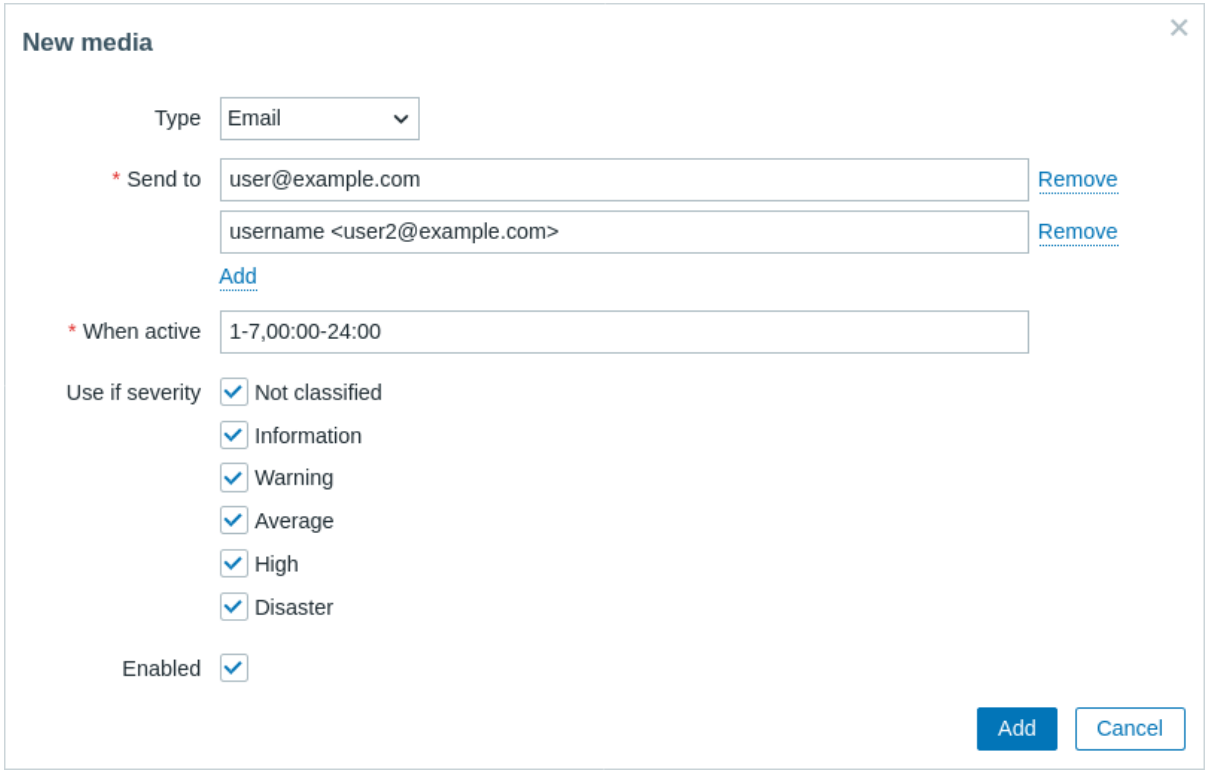
Media dell'utente

Mentre i tipi di media definiscono **come** verrà inviata una notifica, i media dell'utente definiscono **dove** deve essere inviata.

I media dell'utente (ad esempio, l'indirizzo email, l'ID utente del webhook, ecc.) devono essere definiti nel profilo dell'utente, indipendentemente dal metodo di consegna. Un'azione che invia messaggi all'utente *Admin* utilizzando il webhook *X* non riuscirà a consegnarli se i dettagli di consegna del webhook *X* non sono definiti nel profilo dell'utente Admin.

Per definire i media dell'utente:

1. Vai a *Users* > *Users* e apri il modulo delle proprietà dell'utente.
   In alternativa, vai a *User settings* > *Notifications* del tuo profilo utente.
2. Fai clic su *Add* nella scheda *Media*.
3. Inserisci i parametri del media dell'utente nel modulo.



Parameter	Description
Type	L'elenco a discesa contiene i nomi dei tipi di media abilitati. Nota che, quando si modifica un media di un tipo di media disabilitato, il tipo viene visualizzato in rosso.

Parameter	Description
Send to	Inserisci le informazioni di contatto a cui devono essere inviati i messaggi. Per il tipo di media email, è possibile aggiungere più indirizzi facendo clic sul pulsante Add sotto il campo dell'indirizzo. In questo caso, le notifiche verranno inviate a tutti gli indirizzi elencati. Per esempi di indirizzi, consulta la descrizione del parametro *Email* per il tipo di media email. Per il tipo di media delle notifiche push, puoi selezionare i dispositivi mobili attivi dell'utente oppure aggiungere uno o più dispositivi specifici.
When active	Puoi limitare l'orario di invio dei messaggi, ad esempio impostando solo i giorni lavorativi (1-5,09:00-18:00). Nota che questo limite si basa sul fuso orario dell'utente. Se il fuso orario dell'utente viene modificato ed è diverso da quello del sistema, potrebbe essere necessario modificare di conseguenza questo limite per non perdere messaggi importanti. Per una descrizione del formato, consulta la pagina Specifica del periodo di tempo. Sono supportate le macro utente.
Use if severity	Seleziona le caselle di controllo delle gravità dei trigger per le quali desideri ricevere notifiche. *Nota* che la gravità predefinita ('Not classified') **deve essere** selezionata se desideri ricevere notifiche per eventi non correlati ai trigger. Dopo il salvataggio, le gravità dei trigger selezionate verranno visualizzate con i colori di gravità corrispondenti, mentre quelle non selezionate saranno disattivate.
Status	Stato del media dell'utente. **Enabled** - in uso. **Disabled** - non in uso.

Email

Panoramica

Per configurare l'email come canale di consegna dei messaggi, è necessario configurare l'email come tipo di supporto e assegnare indirizzi specifici agli utenti.

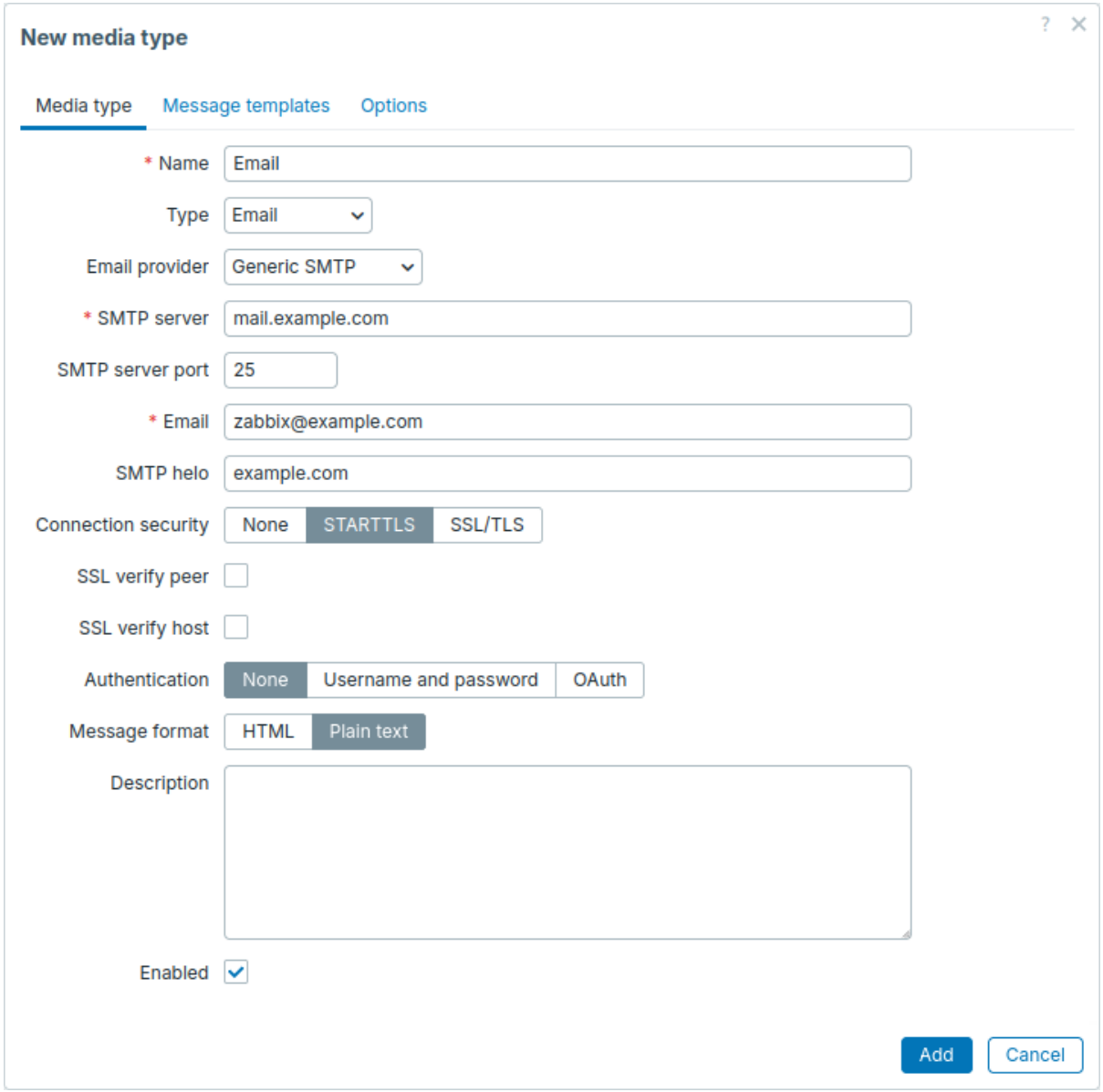
> **Note:**
> Le notifiche multiple per un singolo evento verranno raggruppate nello stesso thread email.

Configurazione

Per configurare l'email come tipo di supporto:

1. Vai a *Avvisi* > *Tipi di supporto*.
2. Fai clic su *Crea tipo di supporto* (oppure fai clic su *Email* nell'elenco dei tipi di supporto predefiniti).
3. Inserisci i parametri del tipo di supporto email nel modulo.

La scheda **Tipo di supporto** contiene gli attributi generali del tipo di supporto:



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Warning:
La password viene esportata in testo non crittografato durante l'esportazione della configurazione del tipo di supporto email.

I parametri seguenti sono specifici del tipo di supporto email:

Parametro	Descrizione
Provider email	Seleziona il provider email: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* o *Office365 relay*. Se selezioni le opzioni relative a Gmail/Office365, dovrai fornire solo l'indirizzo email del mittente e la password; opzioni come *Server SMTP*, *Porta del server SMTP*, *SMTP helo* e *Sicurezza della connessione* verranno compilate automaticamente da Zabbix. Vedi anche: Tipi di supporto Gmail/Office365 automatizzati.
Server SMTP	Imposta un server SMTP per gestire i messaggi in uscita. Questo campo è disponibile se *Generic SMTP* è selezionato come provider email.
Porta del server SMTP	Imposta la porta del server SMTP per gestire i messaggi in uscita. Questo campo è disponibile se *Generic SMTP* è selezionato come provider email.

Parametro	Descrizione
Email	L'indirizzo inserito qui verrà utilizzato come indirizzo **Da** per i messaggi inviati. È supportata l'aggiunta di un nome visualizzato del mittente con l'indirizzo email effettivo (ad esempio, `Zabbix_info <zabbix@company.com>`). Esistono alcune restrizioni sui nomi visualizzati nelle email Zabbix rispetto a quelli consentiti da RFC 5322, come illustrato negli esempi seguenti. Esempi validi: `zabbix@company.com` (solo un indirizzo email; le parentesi angolari non sono necessarie) `Zabbix_info <zabbix@company.com>` (un nome visualizzato seguito dall'indirizzo email tra parentesi angolari) `Ω-monitoring <zabbix@company.com>` (caratteri UTF-8 nel nome visualizzato) Esempi non validi: `Zabbix HQ zabbix@company.com` (è presente un nome visualizzato, ma mancano le parentesi angolari intorno all'indirizzo email) `"Zabbix@<H(comment)Q>" <zabbix@company.com>` (sebbene valido secondo RFC 5322, le coppie tra virgolette e i commenti non sono supportati nelle email Zabbix).
SMTP helo	Imposta un valore SMTP helo corretto, normalmente un nome di dominio. Se vuoto, verrà inviato il nome di dominio dell'email (ovvero ciò che segue `@` nel campo *Email*). Se non è possibile recuperare il nome di dominio, verrà registrato un avviso a livello di debug e il nome host del server verrà inviato come dominio per il comando HELO. Questo campo è disponibile se *Generic SMTP* è selezionato come provider email.
Sicurezza della connessione	Seleziona il livello di sicurezza della connessione: **Nessuna** - non utilizzare l'opzione CURLOPT_USE_SSL **STARTTLS** - utilizzare l'opzione CURLOPT_USE_SSL con il valore CURLUSESSL_ALL **SSL/TLS** - l'utilizzo di CURLOPT_USE_SSL è facoltativo.
Verifica peer SSL	Seleziona la casella per verificare il certificato SSL del server SMTP. Il valore della direttiva di configurazione del server "SSLCALocation" deve essere inserito in CURLOPT_CAPATH per la convalida del certificato. Questa opzione imposta l'opzione cURL CURLOPT_SSL_VERIFYPEER.
Verifica host SSL	Seleziona la casella per verificare che il campo *Common Name* o il campo *Subject Alternate Name* del certificato del server SMTP corrisponda. Questa opzione imposta l'opzione cURL CURLOPT_SSL_VERIFYHOST.
Autenticazione	Seleziona il livello di autenticazione: **Nessuna** - non vengono impostate opzioni cURL **Nome utente e password** - implica "AUTH=*" lasciando a cURL la scelta del meccanismo di autenticazione **OAuth** - autenticazione OAuth L'autenticazione OAuth non è supportata per il provider email *Office365 relay*.
Nome utente	Nome utente da utilizzare per l'autenticazione. Imposta il valore di CURLOPT_USERNAME. Sono supportate le macro utente.
Password	Password da utilizzare per l'autenticazione. Imposta il valore di CURLOPT_PASSWORD. Sono supportate le macro utente. Il valore inserito qui viene esportato così com'è quando viene esportata la configurazione del tipo di supporto. Per evitare di esporre credenziali sensibili nei file esportati, utilizza una macro utente (ad esempio, `{$EMAIL_SMTP_PASSWORD}`) invece di una password letterale: verrà esportato il riferimento alla macro, mentre il valore segreto dovrà essere configurato nuovamente nel sistema di destinazione.
Token OAuth	Fai clic su *Configura* per impostare i parametri per il recupero dei token OAuth in una nuova finestra. Questo campo è disponibile solo se nel campo *Autenticazione* è selezionato "OAuth".
Formato del messaggio	Seleziona il formato del messaggio: **HTML** - invia come HTML; **Testo normale** - invia come testo normale.

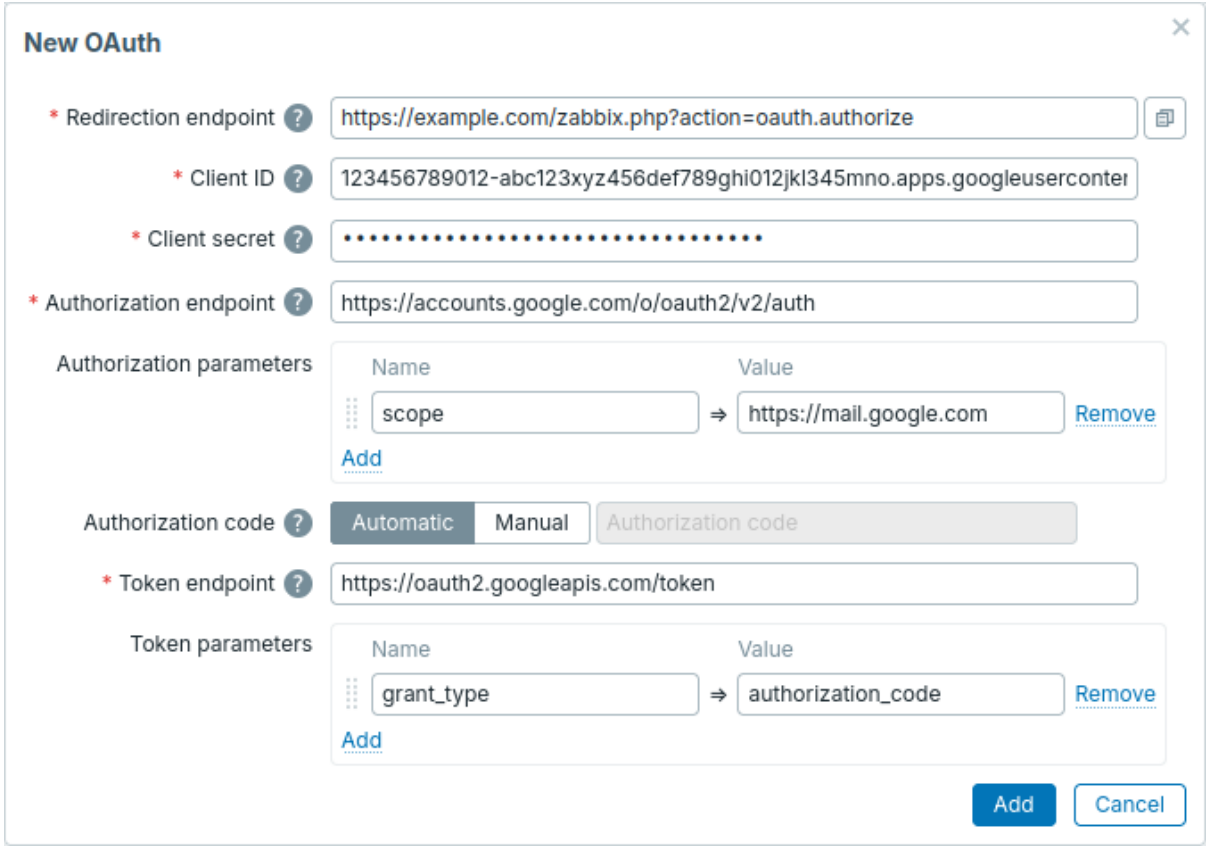
Attention:
Per abilitare l'autenticazione SMTP, il server Zabbix deve essere compilato con l'opzione `--with-libcurl` di compilazione, che richiede libcurl (vedi i requisiti del server per i dettagli sulle versioni).

Vedi anche i parametri comuni dei tipi di supporto per informazioni dettagliate su come configurare i messaggi predefiniti e le

opzioni di elaborazione degli avvisi.

Token OAuth

I seguenti parametri sono necessari per recuperare i token OAuth:



Warning:
Il client secret viene esportato in testo in chiaro quando si esporta la configurazione del tipo di media email.

Parameter	Description
Redirection endpoint	Inserisci l'URL del frontend di Zabbix in cui il servizio OAuth reindirizzerà la richiesta di autorizzazione OAuth (usando il formato `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Viene impostato automaticamente per un nuovo tipo di media email quando è definito il frontend URL.
Client ID	Inserisci l'identificatore univoco dell'app client registrata nel server di autorizzazione OAuth.
Client secret	Inserisci il segreto privato dell'app client registrata nel server di autorizzazione OAuth.
Authorization endpoint	Inserisci l'URL del server di autorizzazione OAuth per richiedere l'autorizzazione dell'utente.
Authorization parameters	Inserisci i parametri per l'endpoint di autorizzazione.
Authorization code	Inserisci il codice di autorizzazione: **Automatic** - il codice verrà recuperato automaticamente tramite una pagina di reindirizzamento; **Manual** - inserisci il codice manualmente se il recupero automatico non riesce.
Token endpoint	Inserisci l'URL del server di autorizzazione OAuth per scambiare il codice di autorizzazione con i token di accesso e di refresh.
Token parameters	Inserisci i parametri per il token di accesso.

Note:
Il recupero dei token OAuth dispone di funzionalità automatizzate per i provider email **Gmail**, **Gmail relay** e **Office365**. È necessario fornire solo i valori dei parametri *Redirection endpoint*, *Client ID* e *Client secret*. Zabbix compilerà automaticamente gli altri valori richiesti (vedi OAuth URL defaults by provider).

Recupero dei token di accesso e di aggiornamento OAuth

Dopo aver inviato i parametri OAuth:

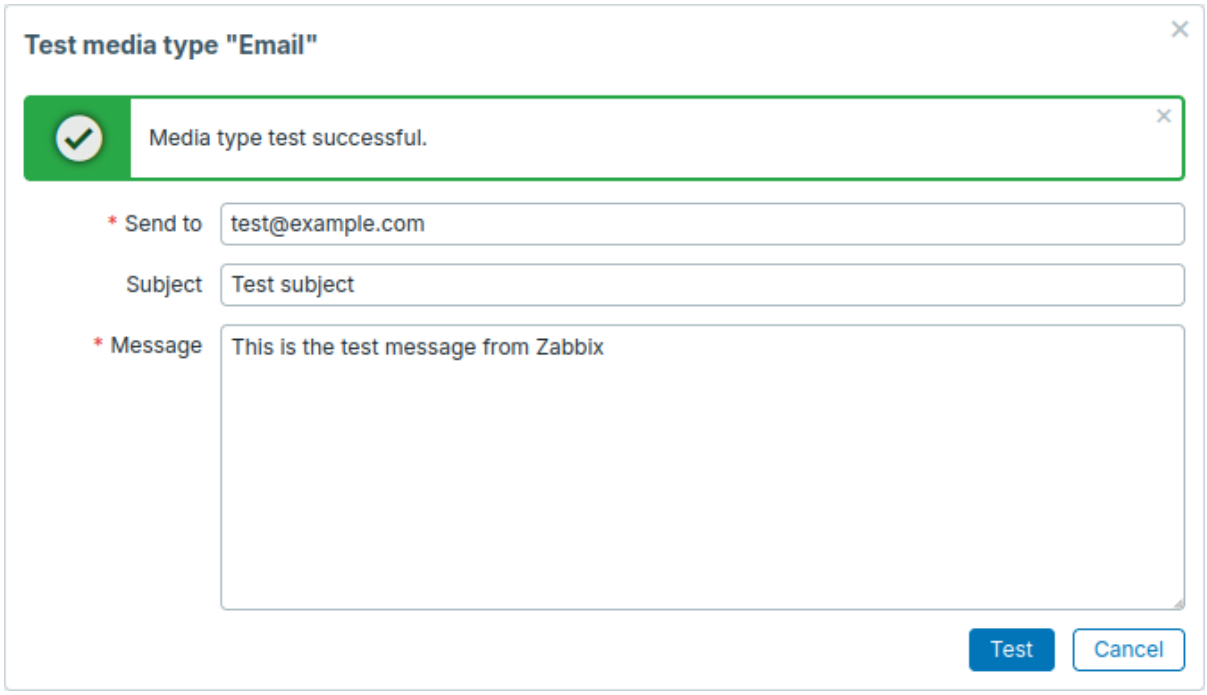
1. Si apre una finestra popup del browser che reindirizza l'utente all'*endpoint di autorizzazione*.
2. L'utente autorizza Zabbix nel servizio OAuth.
3. Il servizio OAuth reindirizza l'utente all'azione `oauth.authorize` del frontend di Zabbix con il codice di autorizzazione e il valore dello scope.
4. In risposta, Zabbix invierà una richiesta all'*endpoint del token* per scambiare il codice di autorizzazione con i token di accesso e di aggiornamento.

Test

Per verificare se un tipo di media email configurato funziona correttamente:

1. Individua l'email pertinente nell'elenco dei tipi di media.
2. Fai clic su *Test* nell'ultima colonna dell'elenco (si aprirà una finestra di test).
3. Inserisci un indirizzo destinatario in *Send to*, il corpo del messaggio e, facoltativamente, l'oggetto.
4. Fai clic su *Test* per inviare un messaggio di prova.

Nella stessa finestra verrà visualizzato un messaggio di esito positivo o negativo del test:



Media utente

Una volta configurato il tipo di media email, vai alla sezione *Users* > *Users* e modifica il profilo utente per assegnare il media email all'utente. I passaggi per configurare i media utente, comuni a tutti i tipi di media, sono descritti nella pagina Media types.

Tipi di media automatizzati per Gmail/Office365

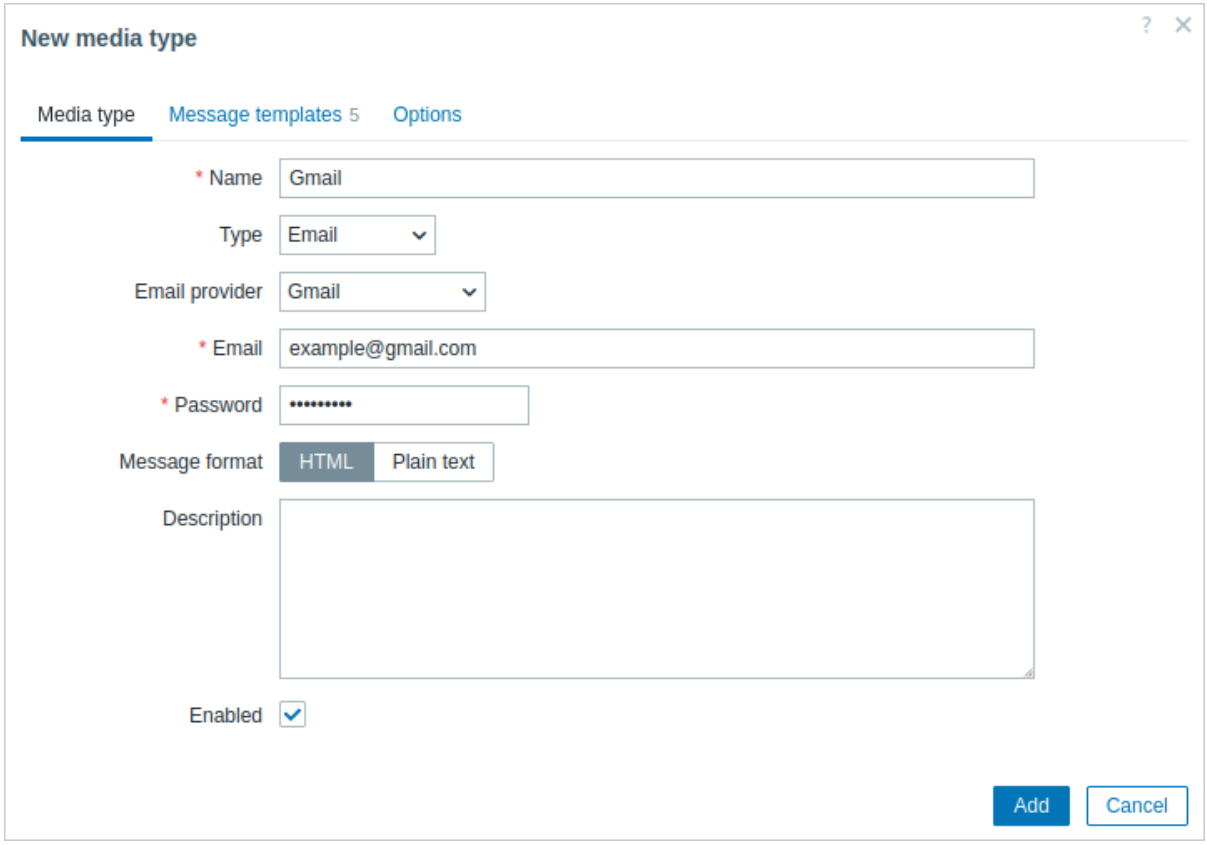
Panoramica

Gli utenti di Gmail o Office365 possono trarre vantaggio dalle funzionalità automatizzate nella configurazione del tipo di supporto.

Configurazione

Il campo *Provider email* nella configurazione del tipo di supporto email consente di selezionare opzioni preconfigurate per Gmail e Office 365.

Quando si selezionano le opzioni relative a Gmail/Office365, è sufficiente fornire l'indirizzo email e la password del mittente per creare un tipo di supporto funzionante.



Non appena vengono forniti l'indirizzo email e la password, Zabbix può compilare automaticamente tutte le impostazioni necessarie per i tipi di supporto Gmail/Office365 con i valori effettivi/consigliati, ovvero *server SMTP*, *porta del server SMTP*, *SMTP helo* e *Sicurezza della connessione*. Grazie a questa automazione, questi campi non vengono nemmeno visualizzati; tuttavia, è possibile visualizzare i dettagli del server SMTP e dell'email nell'elenco dei tipi di supporto (vedere la colonna *Dettagli*).

Si noti inoltre che:

- La password non è necessaria per le opzioni relay.
- Per il relay di Office365, il nome di dominio dell'indirizzo email fornito verrà utilizzato per compilare dinamicamente il server SMTP (ovvero, per sostituire "example.com" in `example-com.mail.protection.outlook.com` con il valore effettivo).

Token OAuth

Il recupero dei token OAuth dispone di funzionalità automatizzate per i provider email *Gmail*, *Gmail relay* e *Office365*.



È necessario fornire solo un insieme limitato di parametri per l'accesso OAuth e il recupero del token di aggiornamento: *Endpoint di reindirizzamento*, *Client ID* e *Client secret*. Zabbix compilerà automaticamente gli altri valori richiesti (vedere Valori predefiniti degli URL OAuth per provider).

Si noti che è anche possibile utilizzare il modulo Generic SMTP per configurare l'autorizzazione OAuth per questi provider.

Note:
SmtpClientAuthentication deve essere abilitato per singolo utente/casella di posta condivisa oppure per l'intero tenant in Office365 (non è abilitato nelle impostazioni predefinite).

URL OAuth predefiniti per provider

La seguente tabella elenca i valori predefiniti degli URL OAuth e i parametri per ciascun provider utilizzati da Zabbix.

Parametro	Gmail	Office365	SMTP generico
Endpoint di autorizzazione	`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authorize`	Nessun valore predefinito
scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send offline_access`	Nessun valore predefinito
access_type	`offline`	Non utilizzato	Nessun valore predefinito
prompt	`consent`	Non utilizzato	Nessun valore predefinito
redirect_uri	Il valore del campo *Endpoint di reindirizzamento* viene aggiunto automaticamente dal codice. Non è elencato nei parametri dell'*Endpoint di autorizzazione*.		
state	Un hash univoco per identificare il supporto aggiornato con il token OAuth viene aggiunto automaticamente dal codice. Non è elencato nei parametri dell'*Endpoint di autorizzazione*.		
Endpoint del token	`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	Nessun valore predefinito
grant_type	`authorization_code`	`authorization_code`	Nessun valore predefinito
redirect_uri	Il valore del campo *Endpoint di reindirizzamento* viene aggiunto automaticamente dal codice. Non è elencato nei parametri dell'*Endpoint del token*.		
state	Un hash univoco per identificare il supporto aggiornato con il token OAuth viene aggiunto automaticamente dal codice. Non è elencato nei parametri dell'*Endpoint del token*.		

Soluzione alternativa di Office365 per gli account personali

Gli account personali di Office365 non supportano le stringhe di query nell'URL di reindirizzamento.

Per aggirare questo problema, l'amministratore del server web Apache può aggiungere la seguente regola di riscrittura al file `.htaccess`:

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

Ciò consente di utilizzare un URL di reindirizzamento semplificato come:

```
http://server-name/zabbix/oauth.authorize
```

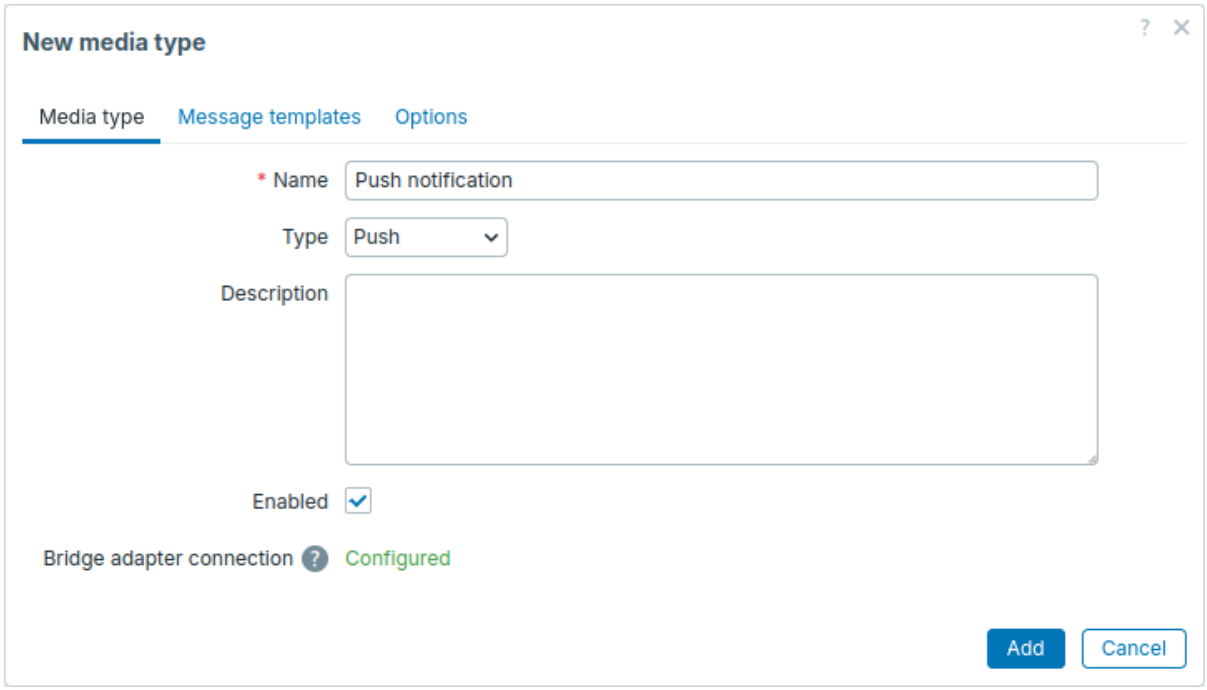
invece di un URL con parametri di query, garantendo la compatibilità con gli account personali di Office365.

Notifica push

Questo tipo di supporto consente al server Zabbix di inviare notifiche push all'app mobile Zabbix sul dispositivo mobile.

Configurazione

Quando si crea un tipo di media, selezionare *Push* come *Type*:



Il modulo di configurazione contiene i parametri comuni del tipo di media, i modelli di messaggio e le opzioni.

Il modulo mostra anche se l'adattatore Zabbix bridge è configurato.
 Se l'adattatore non è configurato, è comunque possibile creare il tipo di media, ma non funzionerà.

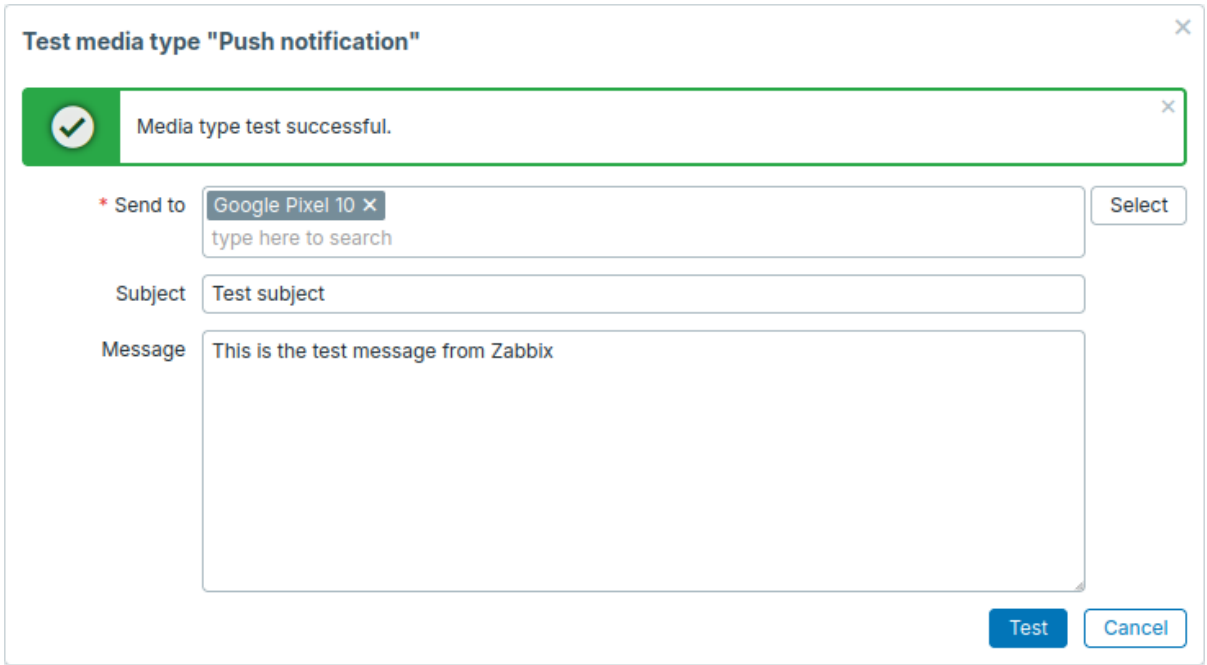
Dopo aver creato e abilitato il tipo di media, aggiungere un dispositivo per un utente per aggiungere automaticamente il tipo di media a tale utente (oppure aggiungere manualmente i media dell'utente). In questo modo l'utente potrà ricevere notifiche push sul proprio dispositivo mobile.

Test

Per verificare se questo tipo di media funziona:

1. Individuare il tipo di media nell'elenco dei tipi di media.
2. Fare clic su *Test* nell'ultima colonna dell'elenco per aprire il modulo di test.
3. Nel campo *Invia a*, selezionare un dispositivo.
4. Fare clic su *Test* per inviare una notifica push di test al dispositivo selezionato.

Nella stessa finestra verrà visualizzato un messaggio di esito positivo o negativo:



SMS

Panoramica

Zabbix supporta l'invio di messaggi SMS utilizzando un modem GSM seriale collegato alla porta seriale del server Zabbix.

Assicurarsi che:

- La velocità del dispositivo seriale (normalmente /dev/ttyS0 in Linux) corrisponda a quella del modem GSM. Zabbix non imposta la velocità del collegamento seriale. Utilizza le impostazioni predefinite.
- L'utente 'zabbix' disponga dell'accesso in lettura/scrittura al dispositivo seriale. Eseguire il comando ls -l /dev/ttyS0 per visualizzare i permessi correnti del dispositivo seriale.
- Il modem GSM abbia il PIN inserito e lo mantenga dopo il riavvio dell'alimentazione. In alternativa, è possibile disabilitare il PIN sulla scheda SIM. Il PIN può essere inserito impartendo il comando AT+CPIN="NNNN" (NNNN è il proprio numero PIN, le virgolette devono essere presenti) in un software terminale, come Unix minicom o Windows HyperTerminal.

Zabbix è stato testato con questi modem GSM:

- Siemens MC35
- Teltonika ModemCOM/G10

Per configurare gli SMS come canale di consegna dei messaggi, è inoltre necessario configurare gli SMS come tipo di supporto e inserire i rispettivi numeri di telefono per gli utenti.

Configurazione

Per configurare gli SMS come tipo di supporto:

1. Vai a *Avvisi* > *Tipi di supporto*.
2. Fai clic su *Crea tipo di supporto* (oppure fai clic su *SMS* nell'elenco dei tipi di supporto predefiniti).
3. Inserisci i parametri del tipo di supporto SMS nel modulo.

I seguenti parametri sono specifici per il tipo di supporto SMS:

Parametro	Descrizione
Modem GSM	Imposta il nome del dispositivo seriale del modem GSM. Il percorso inserito verrà convalidato rispetto al parametro del server SMSDevices (se specificato).

Per informazioni dettagliate su come configurare i messaggi predefiniti e le opzioni di elaborazione degli avvisi, consulta i parametri comuni dei tipi di supporto. Tieni presente che non è possibile elaborare in parallelo l'invio delle notifiche SMS.

Media dell'utente

Dopo aver configurato il tipo di media SMS, accedere alla sezione *Utenti* > *Utenti* e modificare il profilo dell'utente per assegnargli il media SMS. I passaggi per configurare i media dell'utente, comuni a tutti i tipi di media, sono descritti nella pagina Tipi di media.

Script di avviso personalizzati

Panoramica

Se non sei soddisfatto dei media type esistenti per l'invio degli alert, esiste un modo alternativo per farlo. Puoi creare uno script che gestisca la notifica nel modo desiderato.

Gli script di alert personalizzati vengono eseguiti su Zabbix server. Questi script devono trovarsi nella directory specificata nel parametro AlertScriptsPath del file di configurazione di server.

Ecco un esempio di script di alert personalizzato:

```
#####!/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 controlla il codice di uscita dei comandi e degli script eseguiti. Qualsiasi codice di uscita diverso da **0** viene considerato un errore di esecuzione del comando. In questi casi, Zabbix tenterà di ripetere l'esecuzione non riuscita.

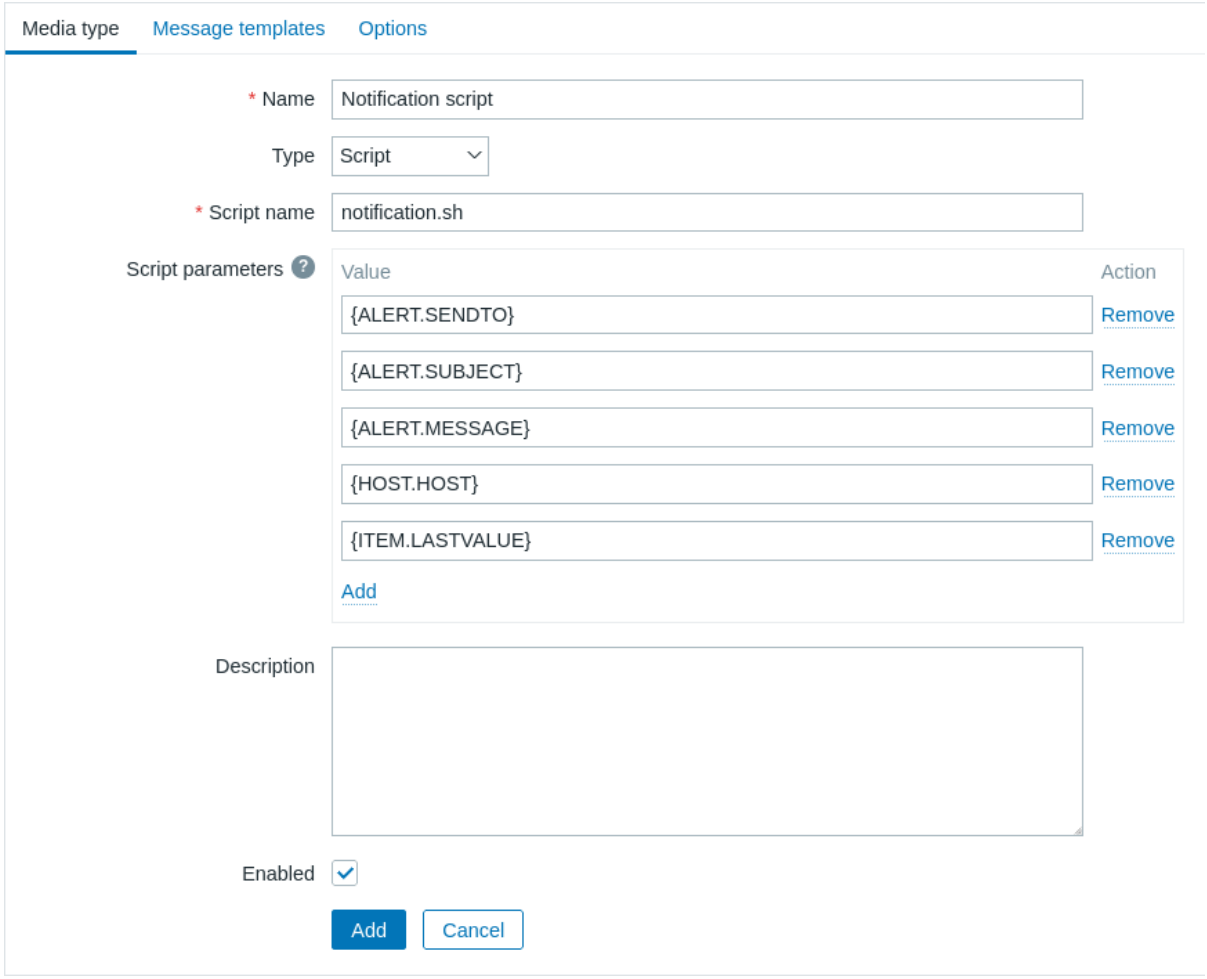
Le variabili d'ambiente non vengono preservate né create per lo script, quindi devono essere gestite esplicitamente.

Configurazione

Per configurare script di avviso personalizzati come tipo di media:

1. Vai a *Avvisi* > *Tipi di media*.
2. Fai clic su *Crea tipo di media*.
3. Inserisci nel modulo i parametri del tipo di media script.

La scheda **Tipo di media** contiene gli attributi generali del tipo di media:



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I seguenti parametri sono specifici per il tipo di media script:

Parametro	Descrizione
Nome dello script	Inserisci il nome del file script (ad esempio, notification.sh) che si trova nella directory specificata nel parametro di configurazione del server AlertScriptsPath.

Parametro	Descrizione
Parametri dello script	Aggiungi parametri script facoltativi che verranno passati allo script come argomenti della riga di comando nell'ordine in cui sono definiti. I parametri dello script supportano le macro {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e tutte le macro supportate nelle notifiche, nonché le macro utente.

Per informazioni dettagliate su come configurare i messaggi predefiniti e le opzioni di elaborazione degli avvisi, consulta i parametri comuni dei tipi di media.

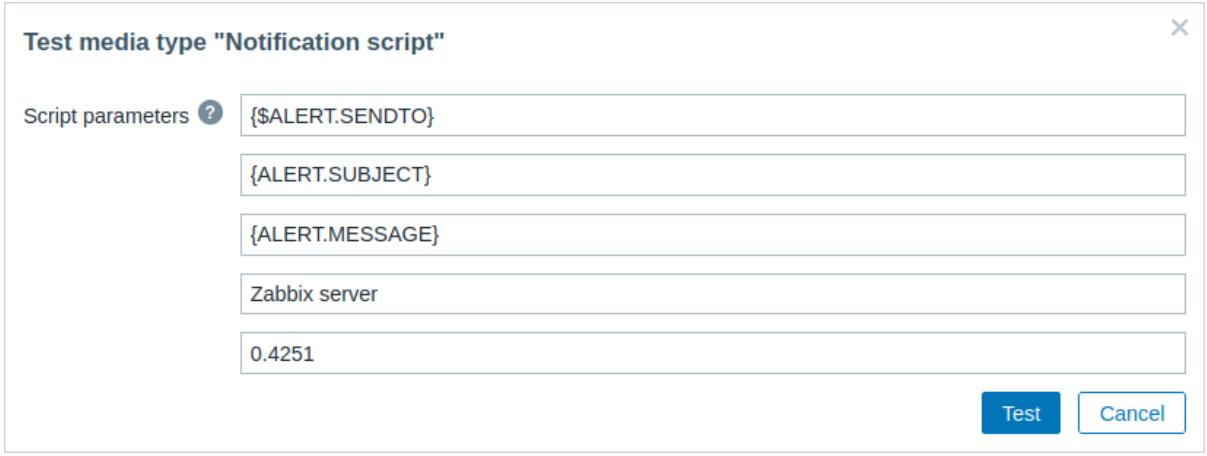
Warning:
Anche se uno script di avviso non utilizza messaggi predefiniti, i modelli di messaggio per i tipi di operazione utilizzati da questo tipo di media devono comunque essere definiti. In caso contrario, non verrà inviata alcuna notifica.

Attention:
Se è configurato più di un tipo di media script, questi script possono essere elaborati in parallelo dai processi alerter. Il numero totale di processi alerter è limitato dal parametro `StartAlerters` nel file di configurazione del server.

Testing

Per testare un media type script configurato:

1. Individua lo script pertinente nell'elenco dei media type.
2. Fai clic su *Test* nell'ultima colonna dell'elenco; si aprirà un modulo di test in una finestra pop-up. Il modulo di test conterrà lo stesso numero di parametri configurati per il media type script.
3. Modifica i valori dei parametri dello script, se necessario. La modifica influisce solo sulla procedura di test; i valori effettivi non verranno modificati.
4. Fai clic su *Test*.



Note:
Quando si testa un media type script configurato, {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e le macro utente verranno risolte nei rispettivi valori, ma le macro correlate agli eventi (ad es. {HOST.HOST}, {ITEM.LASTVALUE}, ecc.) non verranno risolte, poiché durante il test non è presente alcun evento correlato da cui ottenere i dettagli. Nota che anche le macro all'interno delle macro {ALERT.SUBJECT} e {ALERT.MESSAGE} non verranno risolte. Ad esempio, se il valore di {ALERT.SUBJECT} è composto da "Problem: {EVENT.NAME}", la macro {EVENT.NAME} non verrà risolta.

Media utente

Una volta configurato il tipo di media, vai alla sezione *Users* > *Users* ed esegui la modifica di un profilo utente assegnando questo tipo di media all'utente. I passaggi per configurare il media utente, comuni a tutti i tipi di media, sono descritti nella pagina Media types.

Nota che, quando si definisce il media utente, il campo *Send to* non può essere vuoto. Se questo campo non viene utilizzato nello script di allarme, inserisci qualsiasi combinazione di caratteri supportati per aggirare i requisiti di convalida.

webhook

Panoramica

Il tipo di supporto webhook è utile per effettuare chiamate HTTP utilizzando codice JavaScript personalizzato per un'integrazione semplice con software esterni come sistemi di helpdesk, chat o messenger. È possibile scegliere di importare un'integrazione fornita da Zabbix oppure creare un'integrazione personalizzata da zero.

Integrazioni

Sono disponibili le seguenti integrazioni, che consentono di utilizzare tipi di supporto webhook predefiniti per inviare notifiche di Zabbix a:

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
Oltre ai servizi elencati qui, Zabbix può essere integrato con **Spiceworks** (non è richiesto alcun webhook). Per convertire le notifiche di Zabbix in ticket di Spiceworks, creare un tipo di supporto email e inserire l'indirizzo email dell'helpdesk di Spiceworks (ad esempio, help@zabbix.on.spiceworks.com) nelle impostazioni del profilo di un utente Zabbix designato.

Configurazione

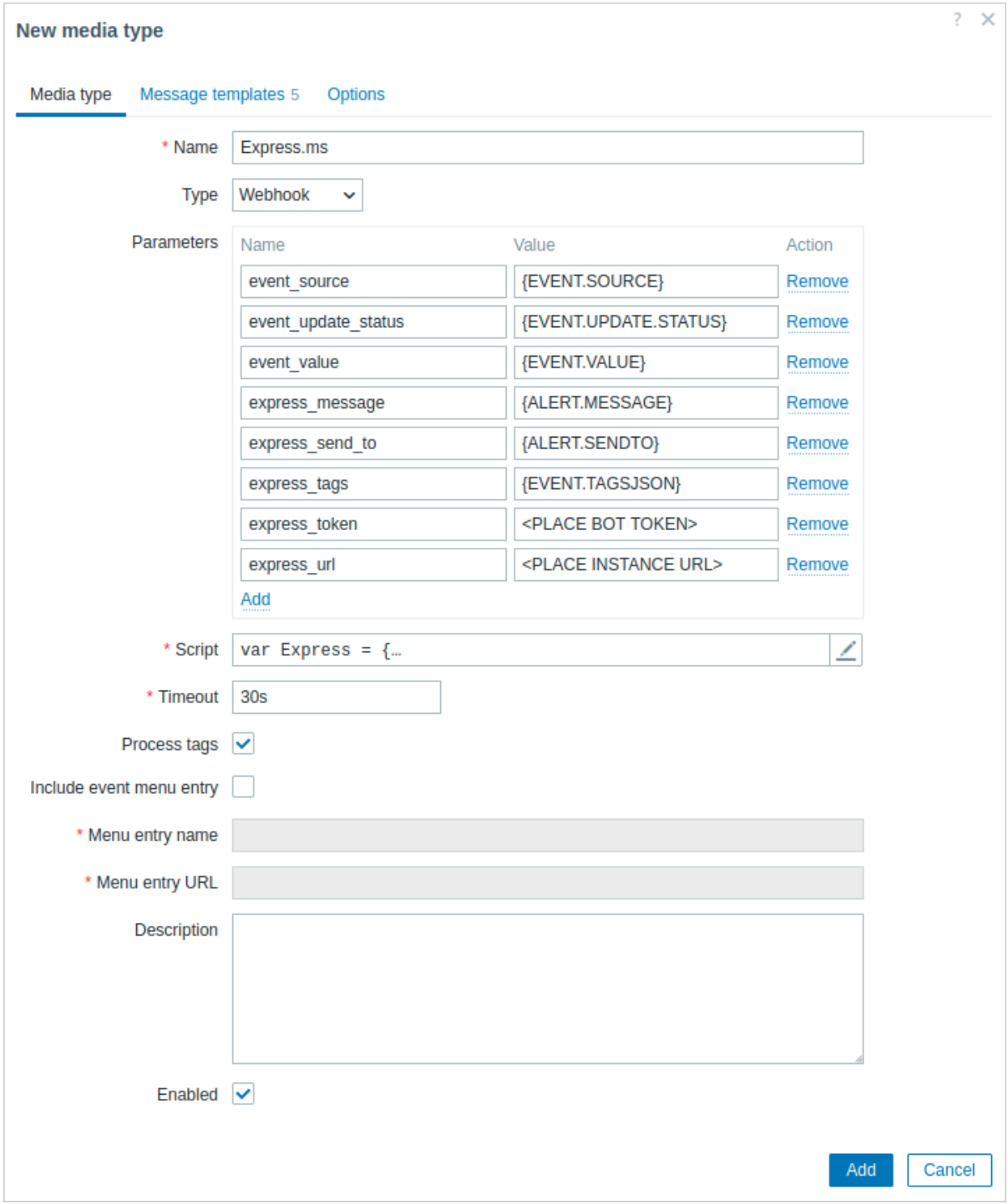
Per iniziare a utilizzare un'integrazione webhook:

1. Individuare il file `.yaml` richiesto nella directory `templates/media` della versione di Zabbix scaricata oppure scaricarlo dal repository git di Zabbix.
2. Importare il file nell'installazione di Zabbix. Il webhook verrà visualizzato nell'elenco dei tipi di supporto.
3. Configurare il webhook seguendo le istruzioni nel file `Readme.md` (è possibile fare clic sul nome di un webhook qui sopra per accedere rapidamente al file `Readme.md`).

Per creare un webhook personalizzato da zero:

1. Accedere a *Avvisi* > *Tipi di supporto*.
2. Fare clic su *Crea tipo di supporto*.
3. Inserire i parametri del tipo di supporto webhook nel modulo.

La scheda **Tipo di supporto** contiene vari attributi specifici per questo tipo di supporto:



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I seguenti parametri sono specifici per il tipo di supporto webhook:

Parametro	Descrizione
Parametri	Specificare le variabili del webhook come coppie attributo e valore. Per i webhook preconfigurati, l'elenco dei parametri varia in base al servizio. Consultare il file *Readme.md* del webhook per la descrizione dei parametri. Per i nuovi webhook, diverse variabili comuni sono incluse per impostazione predefinita (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}); è possibile mantenerle o rimuoverle. I parametri del webhook supportano le macro utente, tutte le macro supportate nelle notifiche dei problemi e, inoltre, le macro {ALERT.SENDTO}, {ALERT.SUBJECT} e {ALERT.MESSAGE}. Se si specifica un proxy HTTP, il campo supporta le stesse funzionalità del campo Proxy HTTP nella configurazione dell'item. La stringa del proxy può essere preceduta da `[scheme]://` per specificare il tipo di proxy utilizzato (ad esempio https, socks4, socks5; vedere la documentazione).
Script	Inserire il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona a forma di matita accanto a esso. Questo codice eseguirà l'operazione del webhook. Lo script è una funzione che accetta coppie parametro-valore. I valori devono essere convertiti in oggetti JSON utilizzando il metodo JSON.parse(), ad esempio: `var params = JSON.parse(value);`. Il codice ha accesso a tutti i parametri, può eseguire richieste HTTP GET, POST, PUT e DELETE, supportare metodi aggiuntivi come CONNECT, PATCH, HEAD, OPTIONS e TRACE e gestire le intestazioni HTTP e il corpo della richiesta. Lo script deve contenere un operatore return, altrimenti non sarà valido. Può restituire lo stato OK insieme a un elenco facoltativo di tag e valori dei tag (vedere l'opzione *Elabora tag*) oppure una stringa di errore. Gli eventi di ripristino (generati automaticamente o in seguito a una chiusura manuale) vengono creati dal server e includono i tag dell'evento risolto, compresi quelli ereditati da template, host e trigger. Gli script webhook vengono eseguiti dopo la creazione dell'avviso; pertanto, i tag restituiti da uno script webhook vengono aggiunti solo dopo la creazione iniziale dell'avviso e non saranno presenti nelle macro `{EVENT.TAGS}` e `{EVENT.RECOVERY.TAGS}` del messaggio iniziale del problema o del messaggio di ripristino immediato. *Nota*: si consiglia di utilizzare variabili locali (ad esempio `var local = 1`) invece di variabili globali (ad esempio `global = 1`) per garantire che ogni script operi sui propri dati ed evitare conflitti tra chiamate simultanee (vedere i problemi noti). Vedere anche: Linee guida per lo sviluppo dei webhook, Esempi di script webhook, Oggetti JavaScript aggiuntivi.
Timeout	Timeout di esecuzione JavaScript (1-60 s, valore predefinito 30 s). Sono supportati i suffissi temporali (ad esempio `30s`, `1m`).
Elabora tag	Selezionare la casella per elaborare i valori delle proprietà JSON restituiti come tag. Questi tag vengono aggiunti ai tag esistenti del problema. Quando si utilizzano i tag webhook, il webhook deve restituire un oggetto JSON contenente almeno un oggetto tags vuoto: `var result = {tags: {}};` Esempi di tag che possono essere restituiti: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Includi voce nel menu degli eventi	Selezionare la casella per includere nel menu degli eventi una voce che rimanda a un ticket esterno creato. Verrà inclusa una voce per ogni webhook abilitato con questa casella selezionata. Se i parametri *Nome voce di menu* e *URL voce di menu* contengono macro {EVENT.TAGS.<tag name>}, la voce verrà inclusa solo se tali macro possono essere risolte, ovvero se l'evento ha questi tag definiti. Se selezionata, il webhook non deve essere utilizzato per inviare notifiche a utenti diversi (valutare la creazione di un utente dedicato) e non deve essere utilizzato in più azioni di avviso per un singolo evento problema.
Nome voce di menu	Specificare il nome della voce di menu. La macro {EVENT.TAGS.<tag name>} è supportata. Questo campo è obbligatorio solo se è selezionata l'opzione *Includi voce nel menu degli eventi*.

Parametro	Descrizione
URL voce di menu	Specificare l'URL sottostante della voce di menu. La macro {EVENT.TAGS.<tag name>} è supportata. Questo campo è obbligatorio solo se è selezionata l'opzione *Includi voce nel menu degli eventi*.

Per informazioni dettagliate sulla configurazione dei messaggi predefiniti e delle opzioni di elaborazione degli avvisi, vedere i parametri comuni dei tipi di supporto.

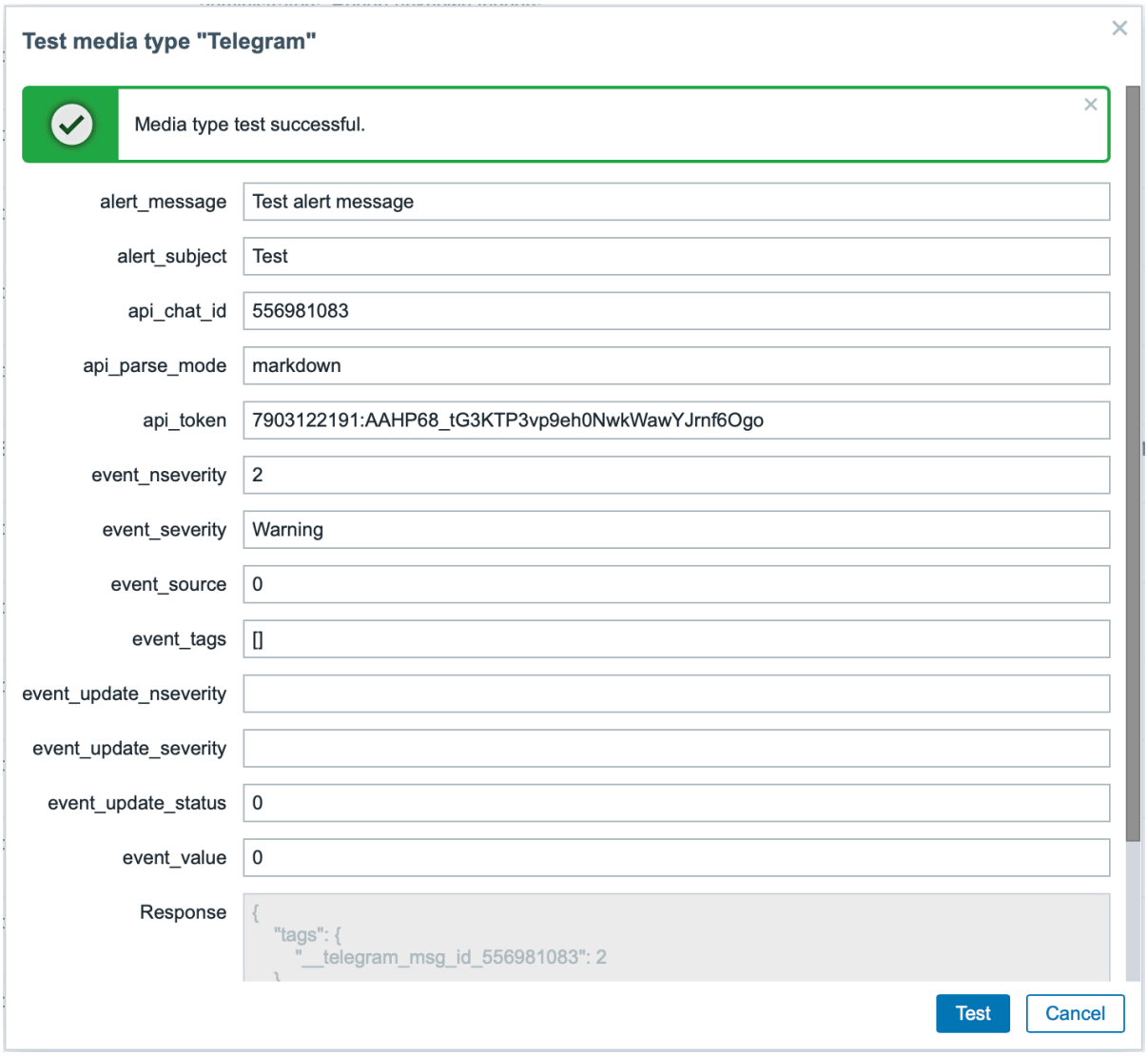
Warning:
Anche se un webhook non utilizza messaggi predefiniti, è comunque necessario definire i template dei messaggi per i tipi di operazione utilizzati da questo webhook.

Test

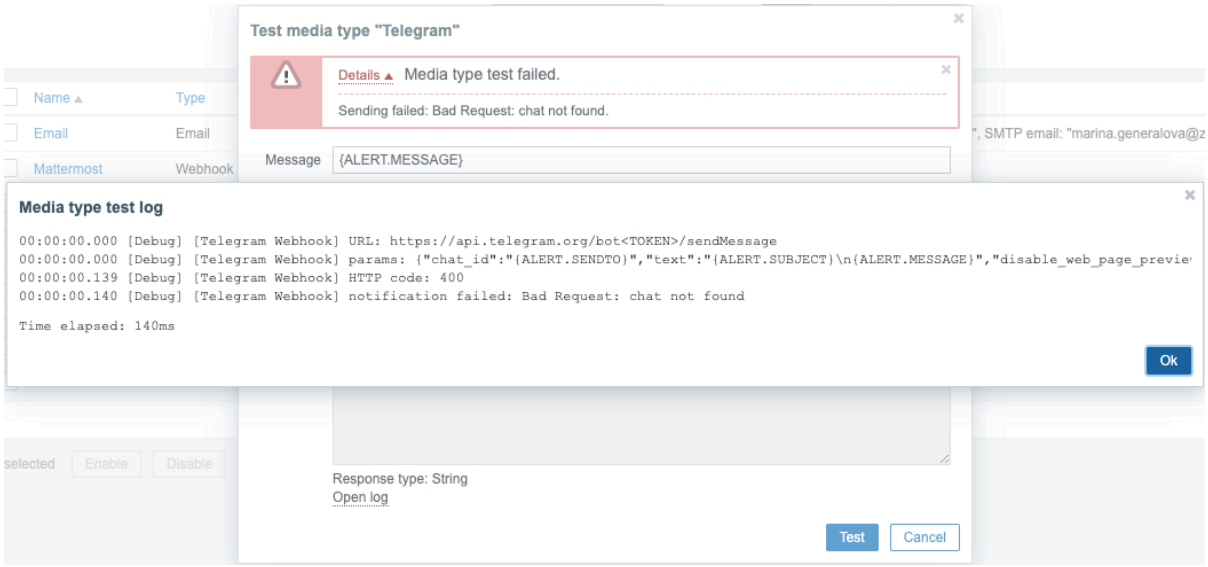
Per testare un tipo di supporto webhook configurato:

1. Individuare il webhook pertinente nell'elenco dei tipi di supporto.
2. Fare clic su *Test* nell'ultima colonna dell'elenco (si aprirà una finestra di test).
3. Modificare i valori dei parametri del webhook secondo necessità.
   Sostituire le macro con valori di esempio; in caso contrario, le macro non verranno risolte e il test avrà esito negativo.
4. Fare clic su *Test*.

La sostituzione o l'eliminazione dei valori nella finestra di test influisce solo sulla procedura di test; i valori effettivi degli attributi del webhook rimarranno invariati.



Per visualizzare le voci del registro dei test del tipo di supporto senza uscire dalla finestra di test, fare clic su *Apri registro* (si aprirà una nuova finestra pop-up).



Se il test del webhook ha esito positivo:

- viene visualizzato il messaggio *"Media type test successful."*.
- la risposta del server appare nel campo grigio *Response*.
- il tipo di risposta (JSON o String) è specificato sotto il campo *Response*.

Se il test del webhook fallisce:

- viene visualizzato il messaggio *"Media type test failed."*, seguito da ulteriori dettagli sull'errore.

Media dell'utente

Una volta configurato il tipo di media, accedere alla sezione *Users* > *Users* e assegnare il media webhook a un utente esistente oppure creare un nuovo utente che rappresenti il webhook. I passaggi per configurare il media dell'utente esistente, comuni a tutti i tipi di media, sono descritti nella pagina Media types.

Se un webhook utilizza i tag per memorizzare l'ID del ticket\messaggio, evitare di assegnare lo stesso webhook come media a utenti diversi, poiché ciò potrebbe causare errori del webhook (vale per la maggior parte dei webhook che utilizzano l'opzione *Include event menu entry*). In questo caso, è consigliabile creare un utente dedicato che rappresenti il webhook:

1. Dopo aver configurato il tipo di media webhook, accedere alla sezione *Users* > *Users* e creare un utente Zabbix dedicato che rappresenti il webhook, ad esempio con il nome utente *Slack* per il webhook Slack. Tutte le impostazioni, ad eccezione del media, possono essere lasciate ai valori predefiniti, poiché questo utente non accederà a Zabbix.
2. Nel profilo dell'utente, accedere alla scheda *Media* e aggiungere un webhook con le informazioni di contatto richieste. Se il webhook non utilizza un campo *Send to*, inserire una combinazione qualsiasi di caratteri supportati per soddisfare i requisiti di convalida.
3. Concedere a questo utente almeno autorizzazioni di lettura per tutti gli host per i quali deve inviare gli avvisi.

Quando si configura un'azione di avviso, aggiungere questo utente nel campo *Send to users* nei dettagli dell'operazione: in questo modo si indicherà a Zabbix di utilizzare il webhook per le notifiche generate da questa azione.

Configurazione delle azioni di allerta

Le azioni determinano quali notifiche devono essere inviate tramite il webhook. I passaggi per configurare le azioni che coinvolgono i webhook sono gli stessi di tutti gli altri tipi di media, con queste eccezioni:

- Se un webhook usa i tag del webhook per memorizzare l'ID del ticket\messaggio e gestire le operazioni di aggiornamento\risoluzione, evita di usare lo stesso webhook in più azioni di allerta per un singolo evento di problema. Se {EVENT.TAGS.<tag name>} esiste e viene aggiornato nel webhook, il suo valore risultante sarà indefinito. Per evitare questo, usa un nuovo nome di tag nel webhook per memorizzare i valori aggiornati. Questo si applica ai webhook Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad e Zendesk forniti da Zabbix e alla maggior parte dei webhook che utilizzano l'opzione *Include event menu entry*. Tuttavia, un singolo webhook può essere usato in più operazioni o fasi di escalation della stessa azione, così come in azioni diverse che non verranno attivate dallo stesso evento di problema a causa di diverse condizioni.
- Quando usi un webhook nelle azioni per eventi interni, assicurati di selezionare la casella *Custom message* e di definire un messaggio personalizzato nella configurazione dell'operazione dell'azione. In caso contrario, non verrà inviata alcuna notifica.

Esempi di script webhook

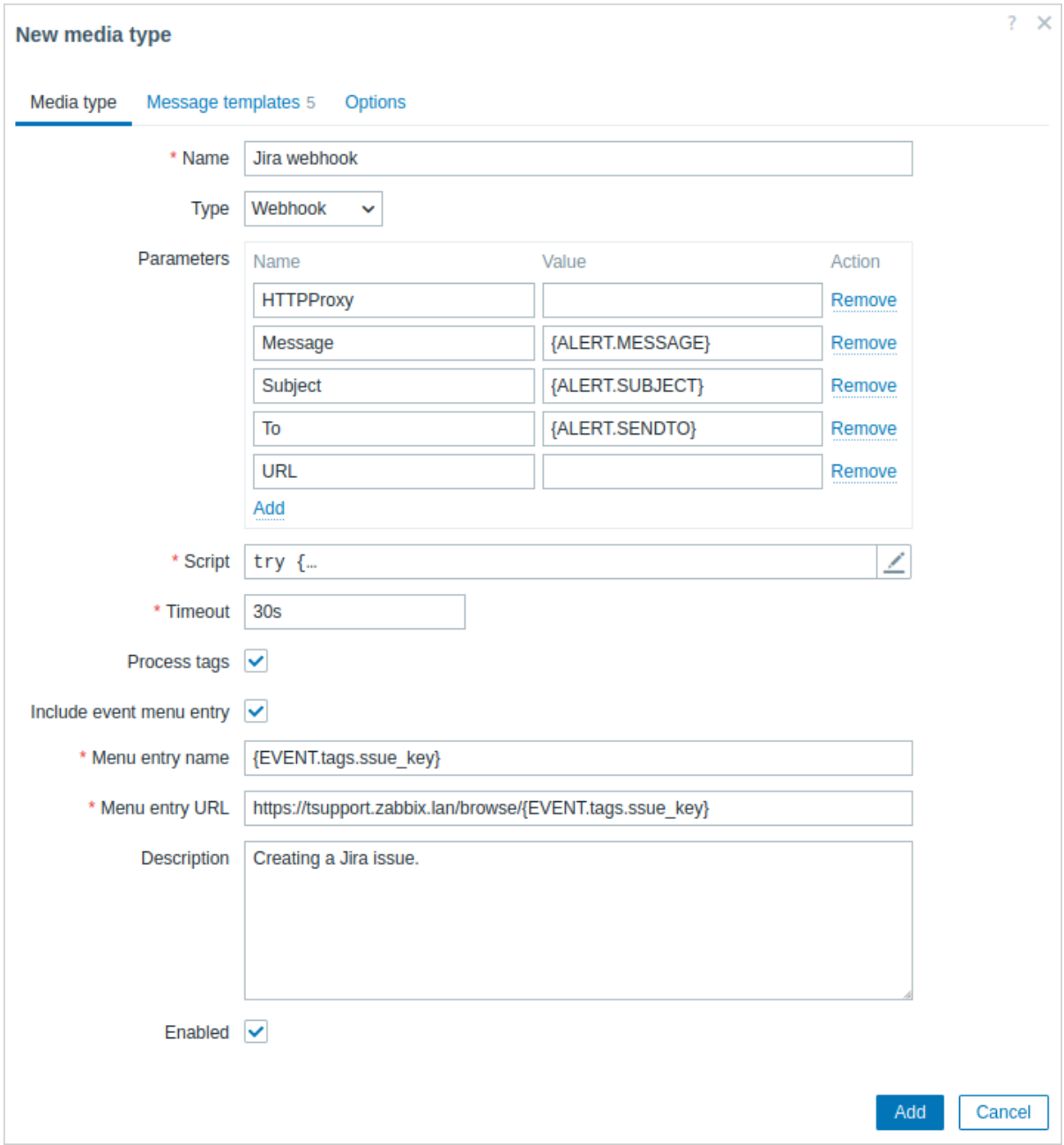
Panoramica

Sebbene Zabbix offra un gran numero di integrazioni webhook disponibili già pronte all'uso, potresti voler creare invece i tuoi webhook personalizzati. Questa sezione fornisce esempi di script webhook personalizzati (utilizzati nel parametro *Script*). Vedere la sezione webhook per la descrizione degli altri parametri del webhook.

Attention:
Non utilizzare assegnazioni non dichiarate in JavaScript di preprocessing. Usa `var` per dichiarare le variabili locali.

Gli eventi di ripristino (generati automaticamente o a seguito di una chiusura manuale) includono i tag dell'evento risolto (compresi i tag ereditati da template, host e trigger). Gli script webhook vengono eseguiti dopo la creazione dell'avviso; i tag restituiti da uno script webhook vengono quindi applicati dopo la creazione iniziale dell'avviso e potrebbero non essere presenti nell'avviso che ha attivato il webhook. Se un'integrazione richiede che i tag prodotti da un webhook siano presenti nella notifica iniziale, recupera i tag dell'evento dal server (ad esempio, usando l'API Event) oppure memorizza i tag prodotti dal webhook in un archivio persistente esterno e correlali lì.

Webhook Jira (personalizzato)



Questo script creerà una issue di JIRA e restituirà alcune informazioni sulla issue creata.

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
                'endpoint': 'jira'
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

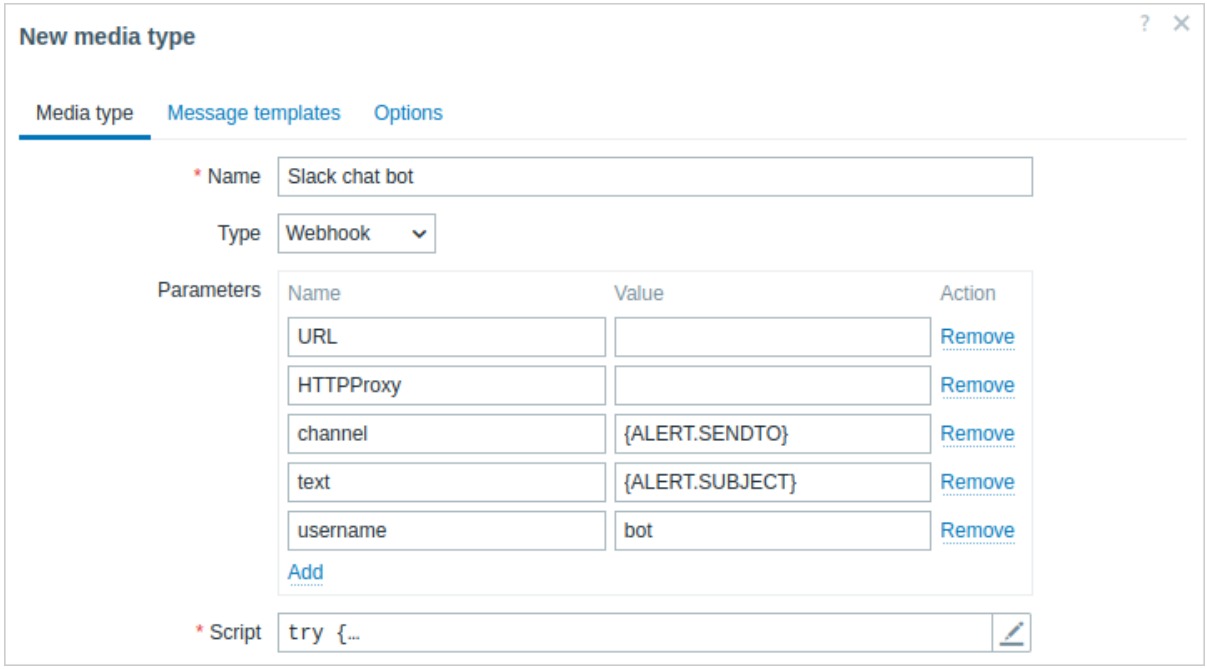
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Slack webhook (custom)

Questo webhook inoltrerà le notifiche da Zabbix a un canale Slack.




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);

    response = req.post(params.URL, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

Azioni

Panoramica

Se si desidera che alcune operazioni vengano eseguite come risultato di eventi (ad esempio, l'invio di notifiche), è necessario configurare le azioni.

Le azioni possono essere definite in risposta a eventi di tutti i tipi supportati:

- Azioni trigger - per eventi in cui lo stato del trigger cambia tra `OK` e `Problem`
- Azioni di servizio - per eventi in cui lo stato del servizio cambia tra `OK` e `Problem`
- Azioni di discovery - per eventi in cui viene eseguito il discovery di rete
- Azioni di autoregistrazione - per eventi in cui nuovi agent attivi si autoregistrano (oppure cambiano i metadati del host per quelli già registrati)
- Azioni interne - per eventi in cui gli item diventano non supportati o i trigger passano a uno stato sconosciuto

Le principali differenze delle azioni di servizio sono:

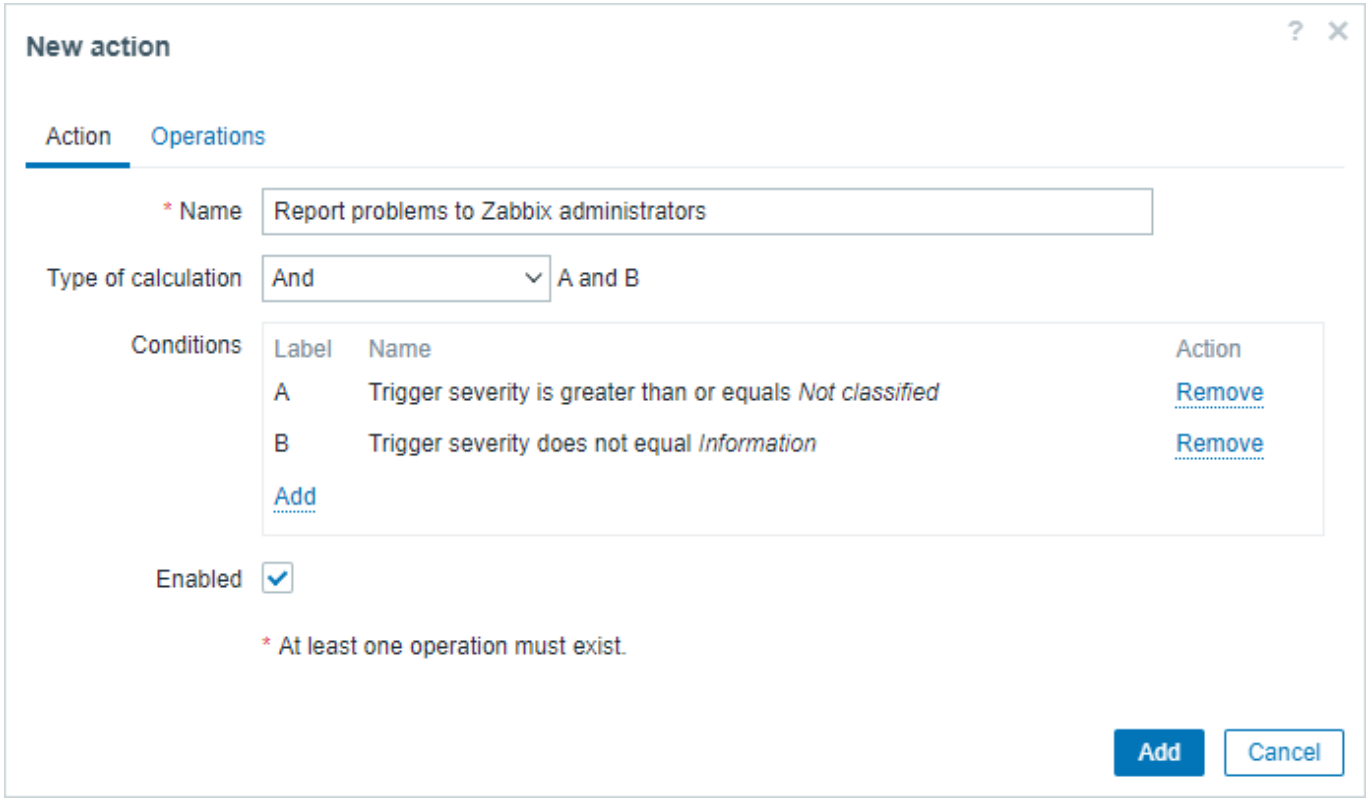
- L'accesso dell'utente alle azioni di servizio dipende dai diritti di accesso ai servizi concessi dal ruolo dell'utente
- Le azioni di servizio supportano un insieme diverso di condizioni

Configurazione di un'azione

Per configurare un'azione, procedere come segue:

1. Andare su *Alerts* > *Actions* e selezionare il tipo di azione richiesto dal sottomenu (in seguito è possibile passare a un altro tipo usando il menu a discesa del titolo).
2. Fare clic su *Create action*.
3. Assegnare un nome all'azione.
4. Scegliere le conditions in base alle quali vengono eseguite le operazioni.
5. Scegliere le operations da eseguire.

Attributi generali dell'azione:



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parameter	Description
Name	Nome univoco dell'azione.

Parameter	Description
Type of calculation	Selezionare l'opzione di valutazione per le condizioni dell'azione (con più di una condizione): **And** - tutte le condizioni devono essere soddisfatte. **Or** - è sufficiente che una condizione sia soddisfatta. **And/Or** - combinazione delle due: AND con tipi di condizione diversi e OR con lo stesso tipo di condizione. **Custom expression** - formula di calcolo definita dall'utente per valutare le condizioni dell'azione.
Conditions	Elenco delle condizioni dell'azione. Fare clic su *Add* per aggiungere una nuova condition. Se non sono configurate condizioni, l'azione verrà eseguita per ogni evento che corrisponde al tipo di azione configurato.
Enabled	Selezionare la casella di controllo per abilitare l'azione. In caso contrario, sarà disabilitata.

Condizioni

Panoramica

È possibile definire che un'azione venga eseguita solo se l'evento corrisponde a un insieme definito di condizioni. Le condizioni vengono impostate durante la configurazione dell'action.

La corrispondenza delle condizioni distingue tra maiuscole e minuscole.

Azioni dei trigger

Le seguenti condizioni possono essere utilizzate nelle azioni basate sui trigger:

Tipo di condizione	Operatori supportati	Descrizione
Gruppo host	uguale diverso da	Specificare gruppi host o gruppi host da escludere. **uguale** - l'evento appartiene a questo gruppo host. **diverso da** - l'evento non appartiene a questo gruppo host. Specificando un gruppo host padre vengono implicitamente selezionati tutti i gruppi host annidati. Per specificare solo il gruppo padre, tutti i gruppi annidati devono essere impostati aggiuntivamente con l'operatore **diverso da**.
Template	uguale diverso da	Specificare template o template da escludere. **uguale** - l'evento appartiene a un trigger ereditato da questo template. **diverso da** - l'evento non appartiene a un trigger ereditato da questo template.
Host	uguale diverso da	Specificare host o host da escludere. **uguale** - l'evento appartiene a questo host. **diverso da** - l'evento non appartiene a questo host.
Nome tag	uguale diverso da contiene non contiene	Specificare un tag evento o un tag evento da escludere. **uguale** - l'evento ha questo tag. **diverso da** - l'evento non ha questo tag. **contiene** - l'evento ha un tag che contiene questa stringa. **non contiene** - l'evento non ha un tag che contiene questa stringa.
Valore tag	uguale diverso da contiene non contiene	Specificare una combinazione di tag evento e valore oppure una combinazione di tag e valore da escludere. **uguale** - l'evento ha questo tag e valore. **diverso da** - l'evento non ha questo tag e valore. **contiene** - l'evento ha un tag e un valore che contengono queste stringhe. **non contiene** - l'evento non ha un tag e un valore che contengono queste stringhe.
Trigger	uguale diverso da	Specificare trigger o trigger da escludere. **uguale** - l'evento è generato da questo trigger. **diverso da** - l'evento è generato da qualsiasi altro trigger, eccetto questo.

Tipo di condizione	Operatori supportati	Descrizione
Nome evento	contiene non contiene	Specificare una stringa nel nome dell'evento generato dal trigger oppure una stringa da escludere. Per impostazione predefinita, il nome dell'evento corrisponde al nome del trigger, a meno che non venga specificato un nome evento personalizzato nella configurazione del trigger. **contiene** - il nome dell'evento contiene questa stringa. **non contiene** - questa stringa è esclusa dal nome dell'evento. Nota: il valore inserito verrà confrontato con il nome dell'evento con tutte le macro espanse.
Gravità del trigger	uguale diverso da maggiore o uguale a minore o uguale a	Specificare la gravità del trigger. **uguale** - uguale alla gravità del trigger. **diverso da** - diverso dalla gravità del trigger. **maggiore o uguale a** - maggiore o uguale alla gravità del trigger. **minore o uguale a** - minore o uguale alla gravità del trigger.
Periodo di tempo	in non in	Specificare un periodo di tempo o un periodo di tempo da escludere. **in** - l'ora dell'evento rientra nel periodo di tempo. **non in** - l'ora dell'evento non rientra nel periodo di tempo. Vedere la pagina specifica del periodo di tempo per la descrizione del formato. Le macro utente sono supportate.
Il problema è soppresso	no sì	Specificare se il problema è soppresso (non mostrato) a causa della manutenzione dell'host. **no** - il problema non è soppresso. **sì** - il problema è soppresso.

Azioni di servizio

Le seguenti condizioni possono essere usate nelle azioni di servizio:

Condition type	Supported operators	Description
Service	equals does not equal	Specificare un servizio o un servizio da escludere. **equals** - l'evento appartiene a questo servizio. **does not equal** - l'evento non appartiene a questo servizio. La specifica di un servizio padre seleziona implicitamente tutti i servizi figlio. Per specificare solo il servizio padre, tutti i servizi nidificati devono essere impostati anche con l'operatore **does not equal**.
Service name	contains does not contain	Specificare una stringa nel nome del servizio o una stringa da escludere. **contains** - l'evento è generato da un servizio che contiene questa stringa nel nome. **does not contain** - questa stringa non può essere trovata nel nome del servizio.
Service tag name	equals does not equal contains does not contain	Specificare un tag evento o un tag evento da escludere. I tag evento del servizio possono essere definiti nella sezione di configurazione del servizio *Tags*. **equals** - l'evento ha questo tag. **does not equal** - l'evento non ha questo tag. **contains** - l'evento ha un tag che contiene questa stringa. **does not contain** - l'evento non ha un tag che contiene questa stringa.

Condition type	Supported operators	Description
Service tag value	equals does not equal contains does not contain	Specificare una combinazione di tag evento e valore o una combinazione di tag e valore da escludere. I tag evento del servizio possono essere definiti nella sezione di configurazione del servizio *Tags*. **equals** - l'evento ha questo tag e valore. **does not equal** - l'evento non ha questo tag e valore. **contains** - l'evento ha un tag e un valore che contengono queste stringhe. **does not contain** - l'evento non ha un tag e un valore che contengono queste stringhe.

Attention:
Assicurarsi di definire i message templates per le azioni di servizio nel menu *Alerts* > *Media types*. In caso contrario, le notifiche non verranno inviate.

Azioni di discovery

Le seguenti condizioni possono essere utilizzate negli eventi basati sulla discovery:

Tipo di condizione	Operatori supportati	Descrizione
Host IP	uguale diverso da	Specificare un intervallo di indirizzi IP o un intervallo da escludere per un host rilevato. **uguale** - l'IP dell'host è nell'intervallo. **diverso da** - l'IP dell'host non è nell'intervallo. Può avere i seguenti formati: IP singolo: 192.168.1.33 Intervallo di indirizzi IP: 192.168.1-10.1-254 Maschera IP: 192.168.4.0/24 Elenco: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Gli spazi nel formato elenco sono supportati.
Tipo di servizio	uguale diverso da	Specificare il tipo di servizio di un servizio rilevato o un tipo di servizio da escludere. **uguale** - corrisponde al servizio rilevato. **diverso da** - non corrisponde al servizio rilevato. Tipi di servizio disponibili: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ping ICMP, telnet.
Porta del servizio	uguale diverso da	Specificare un intervallo di porte TCP di un servizio rilevato o un intervallo da escludere. **uguale** - la porta del servizio è nell'intervallo. **diverso da** - la porta del servizio non è nell'intervallo.
Regola di discovery	uguale diverso da	Specificare una regola di discovery o una regola di discovery da escludere. **uguale** - utilizza questa regola di discovery. **diverso da** - utilizza qualsiasi altra regola di discovery, tranne questa.
Controllo di discovery	uguale diverso da	Specificare un controllo di discovery o un controllo di discovery da escludere. **uguale** - utilizza questo controllo di discovery. **diverso da** - utilizza qualsiasi altro controllo di discovery, tranne questo.
Oggetto di discovery	uguale	Specificare l'oggetto rilevato. **uguale** - uguale all'oggetto rilevato (un dispositivo o un servizio).
Stato della discovery	uguale	**Up** - corrisponde agli eventi 'Host Up' e 'Service Up'. **Down** - corrisponde agli eventi 'Host Down' e 'Service Down'. **Discovered** - corrisponde agli eventi 'Host Discovered' e 'Service Discovered'. **Lost** - corrisponde agli eventi 'Host Lost' e 'Service Lost'.

Tipo di condizione	Operatori supportati	Descrizione
Uptime/Downtime	maggiore o uguale a minore o uguale a	Uptime per gli eventi 'Host Up' e 'Service Up'. Downtime per gli eventi 'Host Down' e 'Service Down'. **maggiore o uguale a** - maggiore o uguale a. Il parametro è espresso in secondi. **minore o uguale a** - minore o uguale a. Il parametro è espresso in secondi.
Valore ricevuto	uguale diverso da maggiore o uguale a minore o uguale a contiene non contiene	Specificare il valore ricevuto da un controllo dell'agent (Zabbix, SNMP) in una regola di discovery. Confronto tra stringhe. Se per una regola sono configurati più controlli dell'agent Zabbix o SNMP, vengono verificati i valori ricevuti per ciascuno di essi (ogni controllo genera un nuovo evento che viene confrontato con tutte le condizioni). **uguale** - uguale al valore. **diverso da** - diverso dal valore. **maggiore o uguale a** - maggiore o uguale al valore. **minore o uguale a** - minore o uguale al valore. **contiene** - contiene la sottostringa. Il parametro è espresso come stringa. **non contiene** - non contiene la sottostringa. Il parametro è espresso come stringa.
Proxy	uguale diverso da	Specificare un proxy o un proxy da escludere. **uguale** - utilizza questo proxy. **diverso da** - utilizza qualsiasi altro proxy, tranne questo.
Gruppo di proxy	uguale diverso da	Specificare un gruppo di proxy o un gruppo di proxy da escludere. **uguale** - utilizza questo gruppo di proxy. **diverso da** - utilizza qualsiasi altro gruppo di proxy, tranne questo.

Note:
I controlli dei servizi in una regola di discovery, che generano eventi di discovery, non vengono eseguiti simultaneamente. Pertanto, se nell'azione sono configurati **più** valori per le condizioni `Tipo di servizio`, `Porta del servizio` o `Valore ricevuto`, questi verranno confrontati con un evento di discovery alla volta, ma **non** con più eventi simultaneamente. Di conseguenza, le azioni con più valori per gli stessi tipi di controllo potrebbero non essere eseguite correttamente.

Azioni di autoregistrazione

Le seguenti condizioni possono essere utilizzate nelle azioni basate sull'autoregistrazione tramite agent attivo:

Tipo di condizione	Operatori supportati	Descrizione
Metadati dell'host	contiene non contiene corrisponde non corrisponde	Specificare i metadati dell'host o i metadati dell'host da escludere. **contiene** - i metadati dell'host contengono la stringa. **non contiene** - i metadati dell'host non contengono la stringa. I metadati dell'host possono essere specificati in un file di configurazione dell'agent. **corrisponde** - i metadati dell'host corrispondono all'espressione regolare. **non corrisponde** - i metadati dell'host non corrispondono all'espressione regolare.
Nome dell'host	contiene non contiene corrisponde non corrisponde	Specificare un nome dell'host o un nome dell'host da escludere. **contiene** - il nome dell'host contiene la stringa. **non contiene** - il nome dell'host non contiene la stringa. **corrisponde** - il nome dell'host corrisponde all'espressione regolare. **non corrisponde** - il nome dell'host non corrisponde all'espressione regolare.
Proxy	è uguale a non è uguale a	Specificare un proxy o un proxy da escludere. **è uguale a** - utilizza questo proxy. **non è uguale a** - utilizza qualsiasi altro proxy (o server) ad eccezione di questo proxy.

Tipo di condizione	Operatori supportati	Descrizione
Gruppo di proxy	è uguale a non è uguale a	Specificare un gruppo di proxy o un gruppo di proxy da escludere. **è uguale a** - utilizza questo gruppo di proxy. **non è uguale a** - utilizza qualsiasi altro gruppo di proxy ad eccezione di questo.

Azioni per eventi interni

Le seguenti condizioni possono essere impostate per le azioni basate su eventi interni:

Tipo di condizione	Operatori supportati	Descrizione
Tipo di evento	uguale	**Item nello stato ”non supportato”** - corrisponde agli eventi in cui un item passa da uno stato ”normale” a uno stato ”non supportato”. **Regola di low-level discovery nello stato ”non supportato”** - corrisponde agli eventi in cui una regola di low-level discovery passa da uno stato ”normale” a uno stato ”non supportato”. **Trigger nello stato ”sconosciuto”** - corrisponde agli eventi in cui un trigger passa da uno stato ”normale” a uno stato ”sconosciuto”.
Gruppo host	uguale diverso da	Specificare gruppi host o gruppi host da escludere. **uguale** - l’evento appartiene a questo gruppo host. **diverso da** - l’evento non appartiene a questo gruppo host.
Nome tag	uguale diverso da contiene non contiene	Specificare un tag evento o un tag evento da escludere. **uguale** - l’evento ha questo tag. **diverso da** - l’evento non ha questo tag. **contiene** - l’evento ha un tag che contiene questa stringa. **non contiene** - l’evento non ha un tag che contiene questa stringa.
Valore tag	uguale diverso da contiene non contiene	Specificare una combinazione di tag evento e valore oppure una combinazione di tag e valore da escludere. **uguale** - l’evento ha questo tag e valore. **diverso da** - l’evento non ha questo tag e valore. **contiene** - l’evento ha un tag e un valore che contengono queste stringhe. **non contiene** - l’evento non ha un tag e un valore che contengono queste stringhe.
Template	uguale diverso da	Specificare template o template da escludere. **uguale** - l’evento appartiene a un item/trigger/regola di low-level discovery ereditato da questo template. **diverso da** - l’evento non appartiene a un item/trigger/regola di low-level discovery ereditato da questo template.
Host	uguale diverso da	Specificare host o host da escludere. **uguale** - l’evento appartiene a questo host. **diverso da** - l’evento non appartiene a questo host.

Tipo di calcolo

Sono disponibili le seguenti opzioni per il calcolo delle condizioni:

- **And** - tutte le condizioni devono essere soddisfatte

Si noti che l’uso del calcolo ”And” non è consentito tra più trigger quando sono selezionati come condizione `Trigger=`. Le azioni possono essere eseguite solo in base all’evento di un singolo trigger.

- **Or** - è sufficiente che una sola condizione sia soddisfatta
- **And/Or** - combinazione delle due: AND con tipi di condizione diversi e OR con lo stesso tipo di condizione, ad esempio:

Gruppo host uguale a Oracle servers
 Gruppo host uguale a MySQL servers
 Nome evento contiene ’Database is down’
 Nome evento contiene ’Database is unavailable’

viene valutato come

(Gruppo host uguale a Oracle servers **or** Gruppo host uguale a MySQL servers**) and (**Nome evento contiene ’Database is down’ **or** Nome evento contiene ’Database is unavailable’**)**

- **Espressione personalizzata** - una formula di calcolo definita dall'utente per valutare le condizioni dell'azione. Deve includere tutte le condizioni (rappresentate come lettere maiuscole A, B, C, ...) e può includere spazi, tabulazioni, parentesi ( ), **and** (sensibile alle maiuscole/minuscole), **or** (sensibile alle maiuscole/minuscole), **not** (sensibile alle maiuscole/minuscole).

Mentre l'esempio precedente con `And/Or` sarebbe rappresentato come (A or B) and (C or D), in un'espressione personalizzata è anche possibile avere molti altri modi di calcolo:

(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
 ecc.

Eliminazione degli oggetti utilizzati nelle azioni

Quando si tenta di eliminare oggetti utilizzati in una condizione/operazione di un'azione, l'utente riceverà un messaggio di errore corrispondente.

Le azioni non vengono disabilitate se le condizioni o le operazioni dell'azione:

- contengono oggetti eliminati;
- contengono oggetti che non vengono più rilevati dal low-level discovery e che sono stati rimossi (dopo il periodo *Delete lost resources*).

Le condizioni/operazioni dell'azione non vengono rimosse se contengono oggetti mancanti. Invece, gli oggetti eliminati vengono visualizzati come *Deleted host group*, *Deleted host*, *Deleted trigger*, ecc. Nella colonna "Info" dell'elenco delle azioni viene visualizzata un'icona di avviso con il messaggio "This action has conditions or operations referencing deleted object(s).".

Operazioni

Panoramica

Puoi definire le seguenti operazioni per tutti gli eventi:

- Inviare un messaggio
- Eseguire un comando remoto

> **Attention:**
> Zabbix server non crea avvisi se l'accesso al host è esplicitamente "negato" per l'utente definito come destinatario dell'operazione dell'azione oppure se all'utente non sono stati definiti alcun diritto sul host.

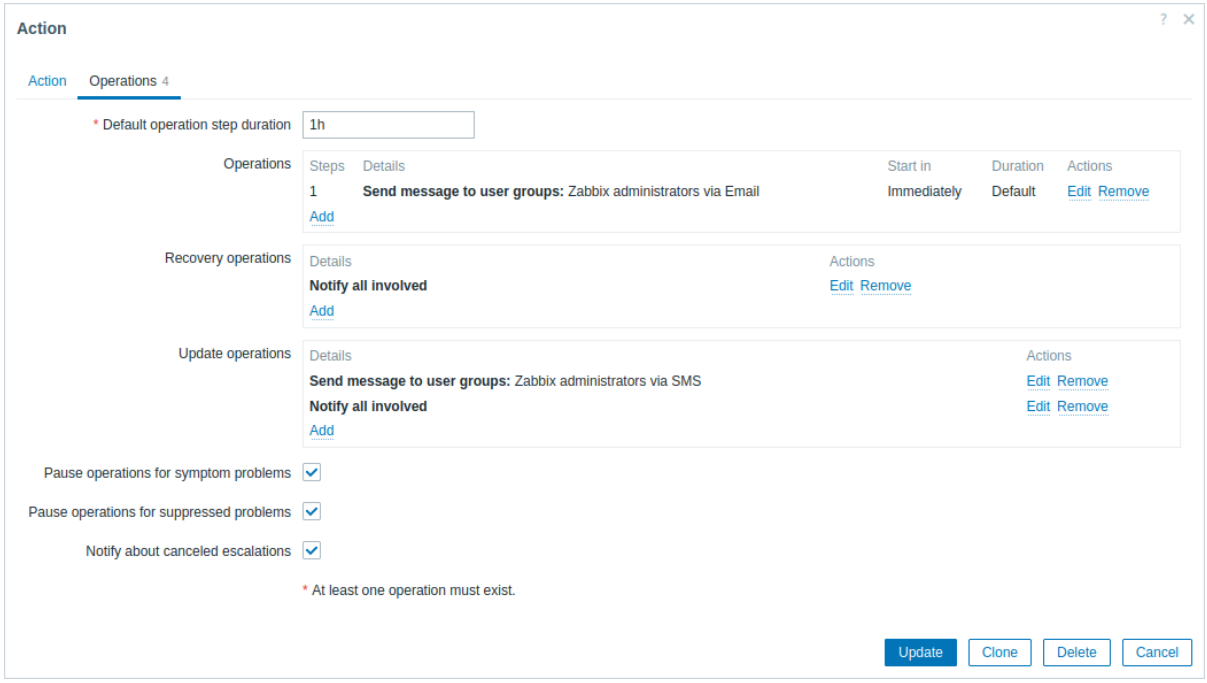
Per gli eventi di discovery e autoregistrazione, sono disponibili operazioni aggiuntive:

- Aggiungere host
- Rimuovere host
- Abilitare host
- Disabilitare host
- Aggiungere a un gruppo di host
- Rimuovere da un gruppo di host
- Aggiungere tag host
- Rimuovere tag host
- Collegare template
- Scollegare template
- Impostare la modalità di inventario host

Configurazione di un'operazione

Per configurare un'operazione:

1. Accedere alla scheda *Operations* nella configurazione dell'action.
2. Configurare i dettagli dell'operazione.



Attributi generali dell'operazione:

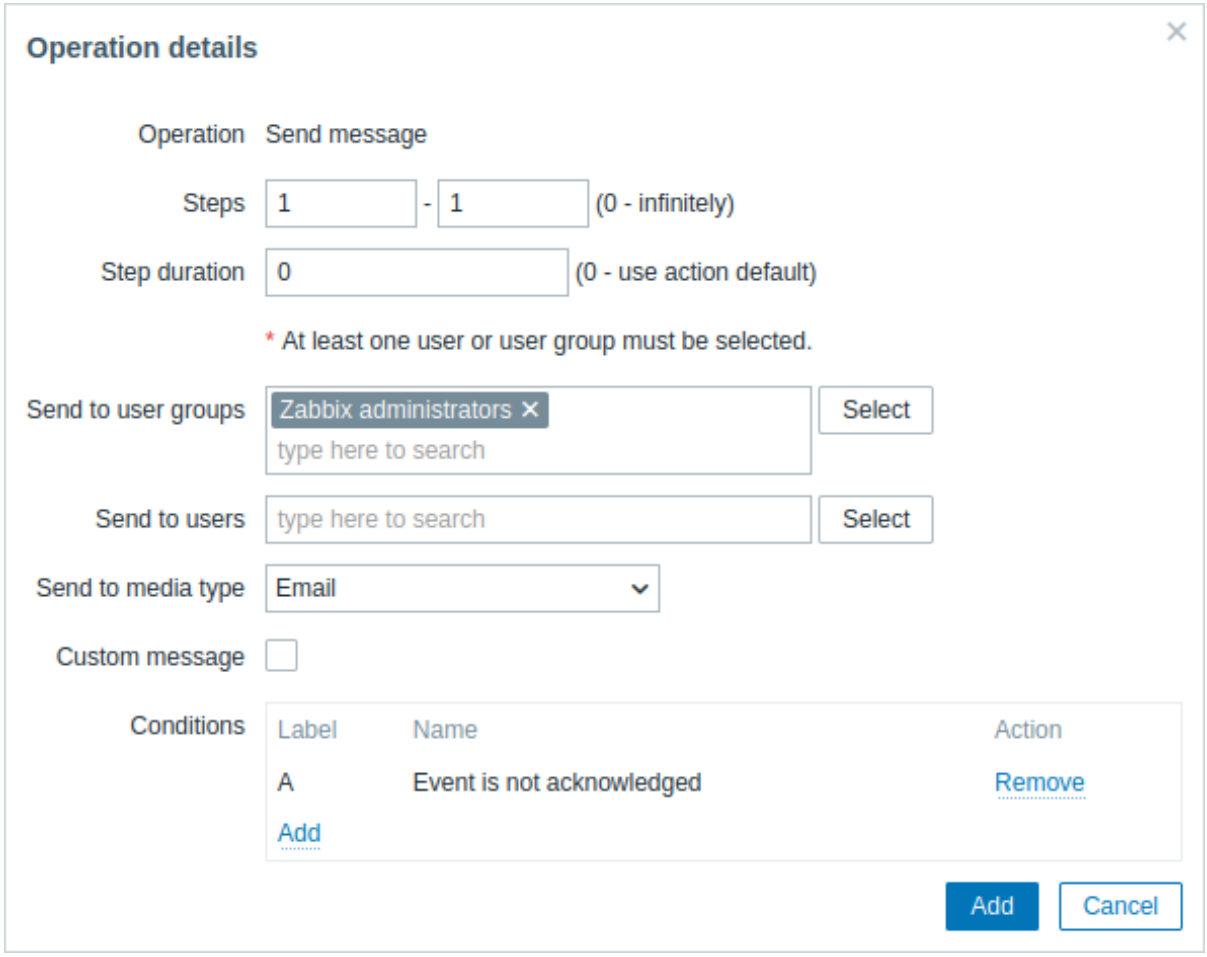
Parametro	Descrizione
Durata predefinita del passaggio dell'operazione	Durata predefinita di un passaggio dell'operazione (da 60 secondi a 1 settimana). Ad esempio, una durata del passaggio di un'ora significa che, se un'operazione viene eseguita, passerà un'ora prima del passaggio successivo. Sono supportati i suffissi temporali, ad esempio 60s, 1m, 2h, 1d. Sono supportate le macro utente.
Operazioni	Vengono visualizzate le operazioni dell'azione, se presenti, con i seguenti dettagli: **Passaggi** - passaggio o passaggi di escalation a cui è assegnata l'operazione. **Dettagli** - tipo di operazione e relativo destinatario/target. L'elenco delle operazioni mostra anche il tipo di media (email, SMS o script) utilizzato, nonché nome e cognome (tra parentesi dopo il nome utente) del destinatario della notifica. **Inizia tra** - tempo che deve trascorrere da un evento prima che l'operazione venga eseguita. **Durata (sec)** - viene visualizzata la durata del passaggio. Viene visualizzato *Predefinita* se il passaggio utilizza la durata predefinita, mentre viene visualizzato un intervallo di tempo se viene utilizzata una durata personalizzata. **Azioni** - vengono visualizzati i collegamenti per modificare e rimuovere un'operazione.
Operazioni di ripristino	Vengono visualizzate le operazioni dell'azione, se presenti, con i seguenti dettagli: **Dettagli** - tipo di operazione e relativo destinatario/target. L'elenco delle operazioni mostra anche il tipo di media (email, SMS o script) utilizzato, nonché nome e cognome (tra parentesi dopo il nome utente) del destinatario della notifica. **Azioni** - vengono visualizzati i collegamenti per modificare e rimuovere un'operazione.
Operazioni di aggiornamento	Vengono visualizzate le operazioni dell'azione, se presenti, con i seguenti dettagli: **Dettagli** - tipo di operazione e relativo destinatario/target. L'elenco delle operazioni mostra anche il tipo di media (email, SMS o script) utilizzato, nonché nome e cognome (tra parentesi dopo il nome utente) del destinatario della notifica. **Azioni** - vengono visualizzati i collegamenti per modificare e rimuovere un'operazione.
Metti in pausa le operazioni per i problemi sintomatici	Selezionare questa casella per mettere in pausa le operazioni (dopo la prima operazione) per i problemi sintomatici. Questa impostazione influisce solo sulle escalation dei problemi; le operazioni di ripristino e aggiornamento non saranno interessate. Questa opzione è disponibile solo per le *azioni trigger*.
Metti in pausa le operazioni per i problemi soppressi	Selezionare questa casella per ritardare l'avvio delle operazioni per la durata di un periodo di manutenzione. Quando le operazioni vengono avviate, al termine della manutenzione vengono eseguite tutte le operazioni, comprese quelle relative agli eventi verificatisi durante la manutenzione. Questa impostazione influisce solo sulle escalation dei problemi; le operazioni di ripristino e aggiornamento non saranno interessate. Se si deseleziona questa casella, le operazioni verranno eseguite senza ritardi anche durante un periodo di manutenzione. Questa opzione non è disponibile per le *azioni di servizio*.

Parametro	Descrizione
Notifica delle escalation annullate	Deselezionare questa casella per disabilitare le notifiche relative alle escalation annullate (quando host, item, trigger o action è disabilitato).

Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Per configurare i dettagli di una nuova operazione, fare clic su Add nel blocco *Operations*. Per modificare un'operazione esistente, fare clic su Edit accanto all'operazione. Si aprirà una finestra pop-up in cui è possibile modificare i dettagli del passaggio dell'operazione.

Dettagli dell'operazione



Parametro	Descrizione
Operazione	Selezionare l'operazione: **Invia messaggio** - invia un messaggio all'utente. **<nome del comando remoto>** - esegue un comando remoto. I comandi sono disponibili per l'esecuzione se precedentemente definiti negli script globali con *Operazione dell'azione* selezionata come ambito. Sono disponibili altre operazioni per gli eventi di discovery e autoregistrazione (vedere sopra).
Passaggi	Selezionare i passaggi da assegnare all'operazione in una pianificazione di escalation: **Da** - esegue a partire da questo passaggio. **A** - esegue fino a questo passaggio (0=infinito, l'esecuzione non sarà limitata).
Durata del passaggio	Durata personalizzata per questi passaggi (0=usa la durata predefinita del passaggio). Sono supportati i suffissi temporali, ad esempio 60s, 1m, 2h, 1d. Sono supportate le macro utente. È possibile assegnare più operazioni allo stesso passaggio. Se per queste operazioni sono definite durate del passaggio diverse, viene presa in considerazione e applicata al passaggio quella più breve.

Parametro	Descrizione
Tipo di operazione: invia messaggio	
Invia ai gruppi di utenti	Selezionare i gruppi di utenti a cui inviare il messaggio. Il gruppo di utenti deve disporre almeno delle autorizzazioni di ”lettura” per il host per poter ricevere la notifica.
Invia agli utenti	Selezionare gli utenti a cui inviare il messaggio. L’utente deve disporre almeno delle autorizzazioni di ”lettura” per il host per poter ricevere la notifica.
Invia al tipo di supporto	Invia il messaggio a tutti i tipi di supporto disponibili (configurati e abilitati) oppure solo a uno specifico.
Messaggio personalizzato	Se selezionato, è possibile configurare il messaggio personalizzato. Per le notifiche relative agli eventi interni tramite webhook, il messaggio personalizzato è obbligatorio.
Oggetto	Oggetto del messaggio personalizzato. L’oggetto può contenere macro. È limitato a 255 caratteri.
Messaggio	Messaggio personalizzato. Il messaggio può contenere macro. È limitato a una determinata quantità di caratteri a seconda del tipo di database (per ulteriori informazioni, vedere Invio di un messaggio).
Tipo di operazione: comando remoto	
Elenco destinazioni	Selezionare le destinazioni su cui eseguire il comando: **Host corrente** - il comando viene eseguito sul host del trigger che ha causato l’evento di problema. Questa opzione non funziona se nel trigger sono presenti più host. **Host** - selezionare gli host su cui eseguire il comando. **Gruppo di host** - selezionare i gruppi di host su cui eseguire il comando. Specificando un gruppo di host padre, vengono selezionati implicitamente tutti i gruppi di host nidificati. Pertanto, il comando remoto verrà eseguito anche sugli host appartenenti ai gruppi nidificati. Un comando su un host viene eseguito una sola volta, anche se l’host corrisponde più di una volta (ad esempio, da diversi gruppi di host; singolarmente e da un gruppo di host). L’elenco delle destinazioni non ha effetto se viene eseguito uno script personalizzato sul server Zabbix. In questo caso, selezionare più destinazioni comporta solo l’esecuzione dello script più volte sul server. Per gli script globali, la selezione della destinazione dipende anche dall’impostazione *Gruppo di host* nella configurazione dello script globale. L’opzione *Elenco destinazioni* non è disponibile per le *azioni dei servizi*, poiché in questo caso i comandi remoti vengono sempre eseguiti sul server Zabbix.

Parametro	Descrizione
Condizioni	Condizione per eseguire l'operazione: **L'evento non è riconosciuto** - solo quando l'evento non è riconosciuto. **L'evento è riconosciuto** - solo quando l'evento è riconosciuto. L'opzione *Condizioni* è disponibile solo per le *azioni dei trigger*.

Al termine, fare clic su *Aggiungi* per aggiungere l'operazione all'elenco delle *Operazioni*.

Invio del messaggio

Panoramica

L'invio di un messaggio è uno dei modi migliori per notificare alle persone un problema. Per questo è una delle azioni principali offerte da Zabbix.

Configurazione

Per poter inviare e ricevere notifiche da Zabbix è necessario:

- Definire il media a cui inviare un messaggio.

Se l'operazione si verifica al di fuori del periodo Quando attivo definito per il media selezionato nella configurazione dell'utente, il messaggio non verrà inviato.

La gravità predefinita del trigger ('Non classificato') **deve essere** selezionata nella configurazione del media dell'utente se si desidera ricevere notifiche per eventi non correlati ai trigger, come gli eventi di discovery, l'autoregistrazione attiva dell'agent o gli eventi interni.

- Configurare un'operazione di un'azione che invii un messaggio a uno dei media definiti.

Attention:
Zabbix invia notifiche solo agli utenti che dispongono almeno dei permessi di 'lettura' sull'host che ha generato l'evento. Almeno un host dell'espressione del trigger deve essere accessibile.

È possibile configurare scenari personalizzati per l'invio dei messaggi utilizzando le escalation.

Per ricevere e leggere correttamente le email da Zabbix, i server/client email devono supportare il formato standard 'SMTP/MIME email', poiché Zabbix invia dati UTF-8 (se l'oggetto contiene solo caratteri ASCII, non viene codificato in UTF-8). L'oggetto e il corpo del messaggio vengono codificati in base64 per rispettare lo standard del formato 'SMTP/MIME email'.

Il limite dei messaggi dopo l'espansione di tutte le macro è uguale al limite dei messaggi per i comandi remoti.

Tracciamento dei messaggi

Puoi visualizzare lo stato dei messaggi inviati in *Monitoring* > *Problems*.

Nella colonna *Actions* puoi vedere informazioni riepilogative sulle azioni eseguite. In essa, i numeri verdi rappresentano i messaggi inviati, quelli rossi i messaggi non riusciti. *In progress* indica che un'azione è stata avviata. *Failed* informa che nessuna azione è stata eseguita con successo.

Se fai clic sull'ora dell'evento per visualizzare i dettagli dell'evento, potrai vedere i dettagli dei messaggi inviati (o non inviati) a causa dell'evento nel blocco *Actions*.

In *Reports* > *Action log* vedrai i dettagli di tutte le azioni eseguite per quegli eventi per i quali è configurata un'azione.

Comandi remoti

Panoramica

Con i comandi remoti è possibile definire che un determinato comando predefinito venga eseguito automaticamente sull'host monitorato al verificarsi di una certa condizione.

I comandi remoti sono quindi un potente meccanismo per il monitoraggio proattivo intelligente.

Tra gli utilizzi più ovvi di questa funzionalità è possibile:

- Riavviare automaticamente un'applicazione (web server, middleware, CRM) se non risponde

- Utilizzare il comando IPMI 'reboot' per riavviare un server remoto se non risponde alle richieste
- Liberare automaticamente spazio su disco (rimuovendo i file più vecchi, pulendo /tmp) quando lo spazio su disco sta per esaurirsi
- Migrare una VM da un sistema fisico a un altro in base al carico della CPU
- Aggiungere nuovi nodi a un ambiente cloud quando le risorse CPU (disco, memoria o altro) sono insufficienti

La configurazione di un'azione per i comandi remoti è simile alla configurazione di un'azione per l'invio di un messaggio; l'unica differenza è che Zabbix esegue un comando invece di inviare un messaggio.

I comandi remoti possono essere eseguiti da Zabbix server, Zabbix proxy o Zabbix agent. Tenere presente quanto segue:

- I comandi remoti su Zabbix agent possono essere eseguiti direttamente da Zabbix server o tramite Zabbix proxy.
- I comandi remoti eseguiti da Zabbix server vengono eseguiti come descritto in Esecuzione dei comandi, incluso il controllo del codice di uscita.
- I comandi remoti vengono eseguiti anche se l'host di destinazione è in modalità manutenzione.

I comandi remoti sono disabilitati per impostazione predefinita sia su Zabbix agent sia su Zabbix proxy. Possono essere abilitati come segue:

- Per i comandi remoti eseguiti da Zabbix agent, aggiungere un parametro `AllowKey=system.run[<command>,*]` o `AllowKeyRegexp=^system\.run\[<command>,.*\]$` per ogni comando consentito nella configurazione dell'agent, dove `*` corrisponde a una delle modalità di esecuzione (`wait` o `nowait`).
- Per i comandi remoti eseguiti da Zabbix proxy, impostare il parametro `EnableRemoteCommands` su 1 nella configurazione del proxy. Tenere presente che non è necessario abilitare i comandi remoti su Zabbix proxy quando vengono eseguiti da Zabbix agent su un host monitorato dal proxy.

Attention:
Non utilizzare `AllowKey=system.run[*]` o `AllowKeyRegexp=^system\.run\[.*\]$` se non è necessario, poiché consentono l'esecuzione di qualsiasi comando sull'host. Per motivi di sicurezza, si consiglia di consentire solo comandi esatti, mentre `system.run[*]` è negato per impostazione predefinita (`DenyKey=system.run[*]`).

Vedere anche Limitazione dei controlli dell'agent.

Limite dei comandi remoti

Il limite dei comandi remoti, dopo la risoluzione di tutte le macro, dipende dal tipo di database e dal set di caratteri (i caratteri non ASCII richiedono più di un byte per essere memorizzati):

Database	Limite in caratteri	Limite in byte
MySQL	65535	65535
PostgreSQL	65535	non limitato
SQLite (solo Zabbix proxy)	65535	non limitato

L'output dell'esecuzione dei comandi remoti (valore restituito) è limitato a 16 MB (inclusi gli spazi finali che vengono troncati). Il limite del comando remoto IPMI si basa sulla libreria IPMI installata. Si noti che i limiti del database si applicano a tutti i comandi remoti.

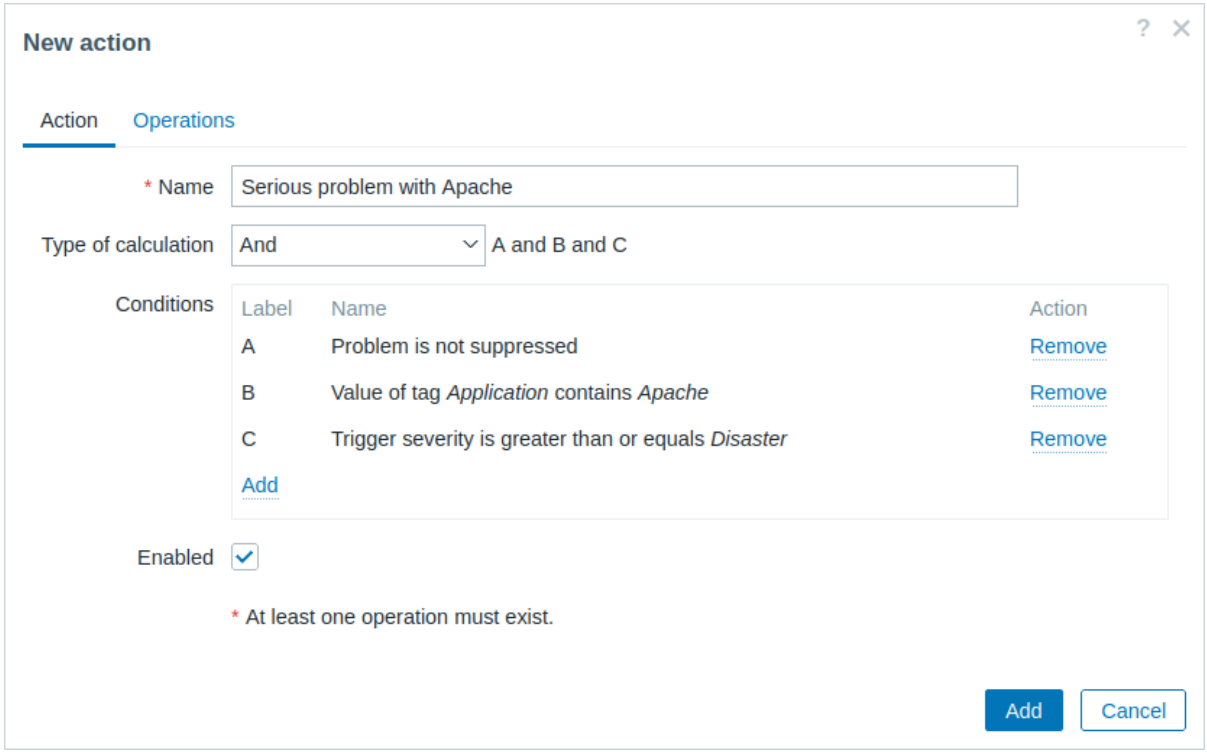
Configurazione

I comandi remoti eseguiti da Zabbix agent devono essere prima autorizzati nella configurazione dell'agent utilizzando il parametro `AllowKey` o `AllowKeyRegexp`.

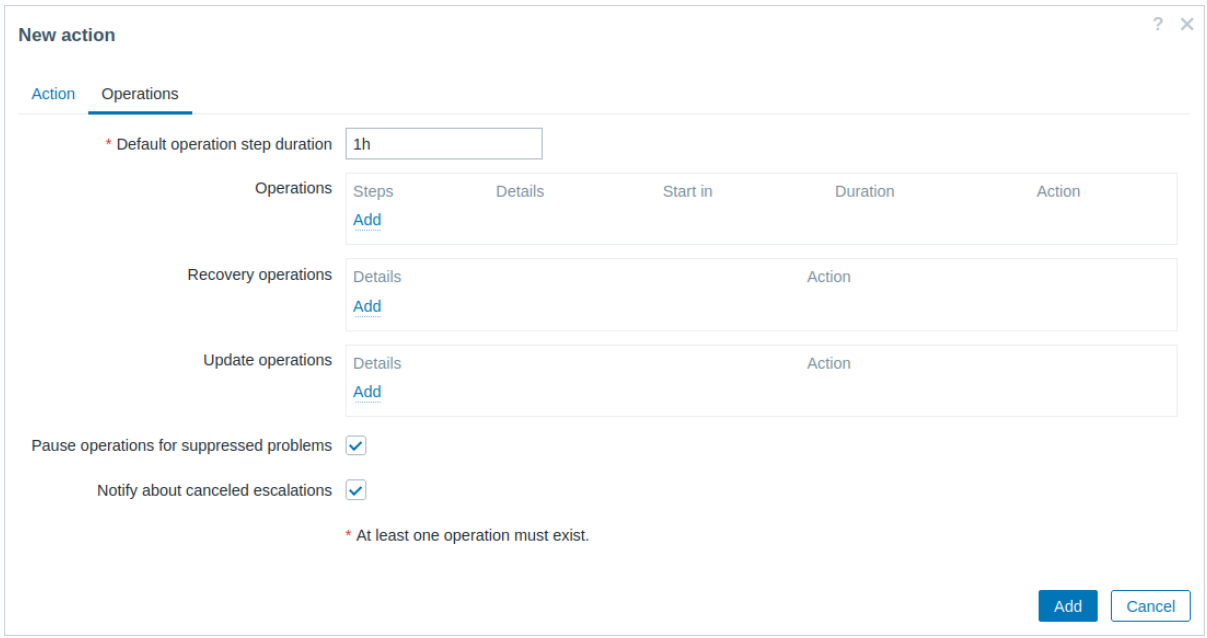
Assicurarsi che il parametro `AllowKey=system.run[<command>,*]` o `AllowKeyRegexp=^system\.run\[<command>,.*\]$` sia aggiunto alla configurazione dell'agent per ogni comando autorizzato, per consentirne l'esecuzione in modalità `nowait`. Riavviare il demone dell'agent dopo aver modificato il parametro.

Quindi, quando si configura una nuova action in *Alerts* > *Actions* > *Trigger actions*:

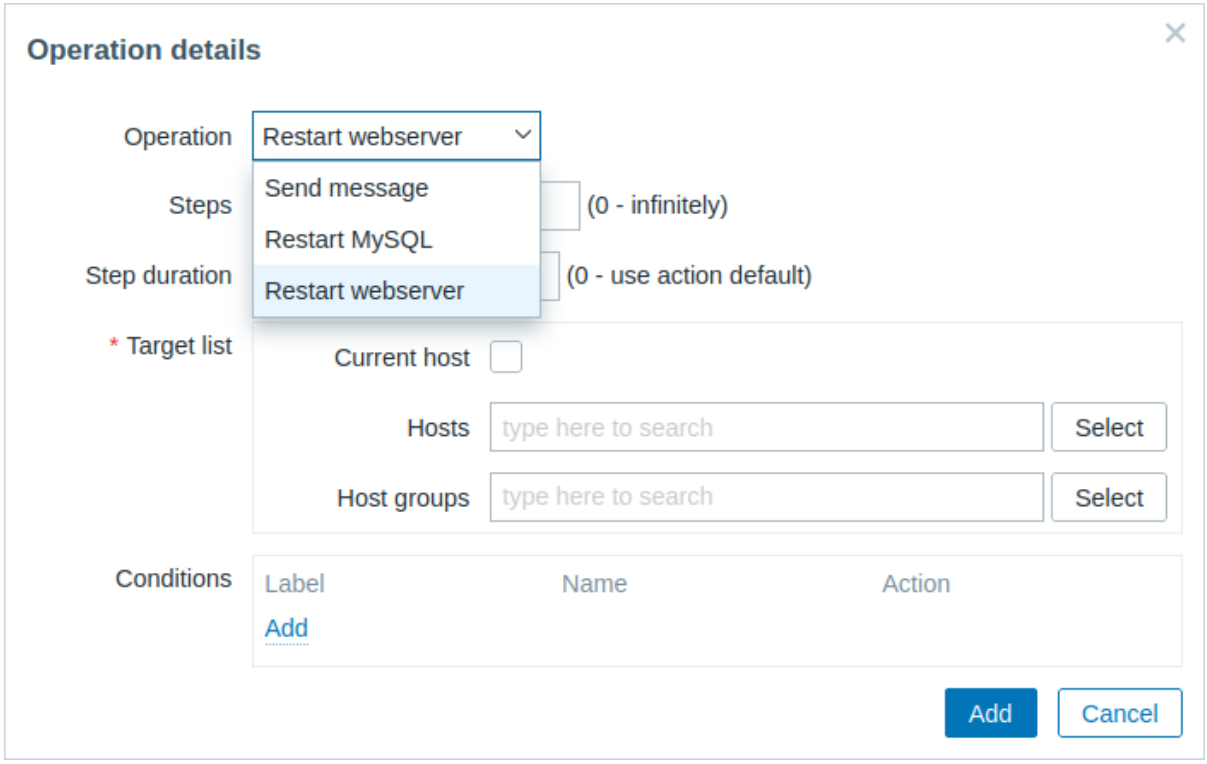
1. Definire le condizioni appropriate, ad esempio impostare l'attivazione dell'action in caso di problemi di tipo disaster relativi a una qualsiasi applicazione Apache.



2. Nella scheda *Operations*, fare clic su *Add* nella sezione *Operations*, *Recovery operations* o *Update operations*.



3. Selezionare uno degli script predefiniti dall'elenco a discesa *Operation* e impostare il *Target list* per lo script.



Script predefiniti

Gli script disponibili per le operazioni degli action (webhook, script, SSH, Telnet, IPMI) sono definiti negli script globali con *Action operation* selezionato come ambito.

Ad esempio:

```
sudo /etc/init.d/apache restart
```

In questo caso, Zabbix tenterà di riavviare un processo Apache. Con questo comando, assicurati che il comando venga eseguito su Zabbix agent (fai clic sul pulsante *Zabbix agent* accanto a *Execute on*).

Attention:
Nota l'uso di sudo - l'utente Zabbix non dispone per impostazione predefinita dei permessi necessari per riavviare i servizi di sistema. Vedi sotto per suggerimenti su come configurare sudo.

Note:
A partire da Zabbix agent 7.0, i comandi remoti possono essere eseguiti anche su un agent che opera in modalità attiva. Zabbix agent - sia attivo sia passivo - deve essere in esecuzione sull'host remoto ed esegue i comandi in background.

I comandi remoti su Zabbix agent vengono eseguiti senza timeout dalla chiave `system.run[,nowait]` e non vengono controllati per i risultati dell'esecuzione. Su Zabbix server e Zabbix proxy, i comandi remoti vengono eseguiti con il timeout impostato nel parametro `TrapperTimeout` del file `zabbix_server.conf` o `zabbix_proxy.conf` e vengono verificati per i risultati dell'esecuzione. Per ulteriori informazioni, consulta *Script timeout*.

Permessi di accesso

Assicurarsi che l'utente `zabbix` disponga dei permessi di esecuzione per i comandi configurati. Potrebbe essere utile usare `sudo` per concedere accesso a comandi con privilegi elevati. Per configurare l'accesso, eseguire come root:

```
visudo
```

Esempi di righe che potrebbero essere usate nel file `sudoers`:

```
# consente all'utente 'zabbix' di eseguire tutti i comandi senza password.
zabbix ALL=NOPASSWD: ALL

# consente all'utente 'zabbix' di riavviare apache senza password.
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
Su alcuni sistemi, il file `sudoers` impedirà agli utenti non locali di eseguire comandi. Per modificare questo comportamento, commentare l'opzione `requiretty` in `/etc/sudoers`.

Comandi remoti con più interfacce

Se il sistema di destinazione ha più interfacce del tipo selezionato (Zabbix agent o IPMI), i comandi remoti verranno eseguiti sull'interfaccia predefinita.

È possibile eseguire comandi remoti tramite SSH e Telnet utilizzando un'interfaccia diversa da quella di Zabbix agent. L'interfaccia disponibile da utilizzare viene selezionata nel seguente ordine:

- Interfaccia predefinita di Zabbix agent
- Interfaccia predefinita SNMP
- Interfaccia predefinita JMX
- Interfaccia predefinita IPMI

Comandi remoti IPMI

Per i comandi remoti IPMI deve essere utilizzata la seguente sintassi:

```
<command> [<value>]
```

dove

- <command> - uno dei comandi IPMI senza spazi
- <value> - 'on', 'off' o qualsiasi intero senza segno. <value> è un parametro facoltativo.

Esempi

Esempi di script globali che possono essere utilizzati come comandi remoti nelle operazioni delle azioni.

Esempio 1

Riavvio di Windows in determinate condizioni.

Per riavviare automaticamente Windows in caso di un problema rilevato da Zabbix, definire il seguente script:

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`c:\windows\system32\shutdown.exe -r -f`

Esempio 2

Riavviare l'host utilizzando il controllo IPMI.

Parametro dello script	Valore
Ambito	'Operazione dell'azione'
Tipo	'IPMI'
Comando	reset

Esempio 3

Spegnere il host utilizzando il controllo IPMI.

Parametro dello script	Valore
Ambito	'Operazione dell'azione'
Tipo	'IPMI'
Comando	power off

Operazioni aggiuntive

Panoramica

In questa sezione puoi trovare alcuni dettagli sulle operazioni aggiuntive per gli eventi di discovery/autoregistrazione.

Aggiunta di host

Gli host vengono aggiunti durante il processo di discovery; non appena un host viene individuato, viene aggiunto, anziché alla fine del processo di discovery.

> **Note:**
> Poiché la discovery di rete può richiedere del tempo, a causa di molti host/servizi non disponibili, si consiglia di avere pazienza e di utilizzare intervalli IP ragionevoli.

Quando si aggiunge un host, il suo nome viene determinato dalla funzione standard **gethostbyname**. Se l'host può essere risolto, viene utilizzato il nome risolto. In caso contrario, viene utilizzato l'indirizzo IP. Inoltre, se per il nome di un host deve essere utilizzato un indirizzo IPv6, tutti i ":" (due punti) vengono sostituiti con "_" (trattini bassi), poiché i due punti non sono consentiti nei nomi host.

> **Attention:**
> Se la discovery viene eseguita da un proxy, attualmente la risoluzione del nome host avviene comunque sullo Zabbix server.

> **Attention:**
> Se nella configurazione di Zabbix esiste già un host con lo stesso nome di uno appena individuato, Zabbix aggiungerà **_N** al nome host, dove **N** è un numero crescente, a partire da 2.

Uso delle macro nei messaggi

Panoramica

Negli oggetti dei messaggi e nel testo dei messaggi è possibile utilizzare le macro per una segnalazione dei problemi più efficiente.

Oltre a un certo numero di macro integrate, sono supportate anche le macro utente e le macro di espressione. È disponibile un elenco completo delle macro supportate da Zabbix.

Esempi

Gli esempi qui illustrano come è possibile utilizzare le macro nei messaggi.

Esempio 1

Oggetto del messaggio:

```
Problema: {TRIGGER.NAME}
```

Quando si riceve il messaggio, l'oggetto del messaggio verrà sostituito con qualcosa di simile a:

```
Problema: Il carico del processore è troppo elevato sul server Zabbix
```

Esempio 2

Messaggio:

```
Il carico del processore è: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

Quando si riceve il messaggio, il messaggio verrà sostituito con qualcosa di simile a:

```
Il carico del processore è: 1.45
```

Esempio 3

Messaggio:

```
Ultimo valore: {?last(/{HOST.HOST}/{ITEM.KEY})}
MAX per 15 minuti: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MIN per 15 minuti: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

Quando si riceve il messaggio, questo verrà sostituito con qualcosa di simile:

```
Ultimo valore: 1.45
MAX per 15 minuti: 2.33
MIN per 15 minuti: 1.01
```

Esempio 4

Messaggio:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Quando si riceve il messaggio, questo conterrà un collegamento alla pagina *Dettagli evento*, che fornisce informazioni sull'evento, sul relativo trigger e un elenco degli ultimi eventi generati dallo stesso trigger.

Esempio 5

Informare sui valori di diversi host in un'espressione di trigger.

Messaggio:

```
Nome del problema: {TRIGGER.NAME}
Espressione del trigger: {TRIGGER.EXPRESSION}

1. Valore dell'item su {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Valore dell'item su {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

Quando si riceve il messaggio, questo verrà sostituito con qualcosa di simile:

```
Nome del problema: Il carico del processore è troppo elevato su un host locale
Espressione del trigger: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load

1. Valore dell'item su Myhost: 0.83 (Carico del processore (media di 1 minuto per core))
2. Valore dell'item su Myotherhost: 5.125 (Carico del processore (media di 1 minuto per core))
```

Esempio 6

Ricezione dei dettagli sia dell'evento di problema che dell'evento di ripristino in un messaggio di ripristino:

Messaggio:

```
Problema:

ID evento: {EVENT.ID}
Valore evento: {EVENT.VALUE}
Stato evento: {EVENT.STATUS}
Ora evento: {EVENT.TIME}
Data evento: {EVENT.DATE}
Età evento: {EVENT.AGE}
Riconoscimento evento: {EVENT.ACK.STATUS}
Cronologia aggiornamenti evento: {EVENT.UPDATE.HISTORY}

Ripristino:

ID evento: {EVENT.RECOVERY.ID}
Valore evento: {EVENT.RECOVERY.VALUE}
Stato evento: {EVENT.RECOVERY.STATUS}
Ora evento: {EVENT.RECOVERY.TIME}
Data evento: {EVENT.RECOVERY.DATE}
Dati operativi: {EVENT.OPDATA}
```

Quando si riceve il messaggio, le macro verranno sostituite con qualcosa di simile:

```
Problema:

ID evento: 21874
Valore evento: 1
Stato evento: PROBLEM
Ora evento: 13:04:30
Data evento: 2018.01.02
Età evento: 5m 0s
Riconoscimento evento: Sì
Cronologia aggiornamenti evento: 2018.01.02 13:05:51 "John Smith (Admin)"
Azioni: riconosciuto.

Ripristino:
```

```
ID evento: 21896
Valore evento: 0
Stato evento: OK
Ora evento: 13:10:07
Data evento: 2018.01.02
Dati operativi: Il valore corrente è 0.83
```

Operazioni di ripristino

Panoramica

Le operazioni di ripristino consentono di ricevere una notifica quando i problemi vengono risolti.

Nelle operazioni di ripristino sono supportati sia i messaggi sia i comandi remoti. Sebbene sia possibile aggiungere più operazioni, l'escalation non è supportata: tutte le operazioni sono assegnate a un singolo passaggio e pertanto verranno eseguite simultaneamente.

Casi d'uso

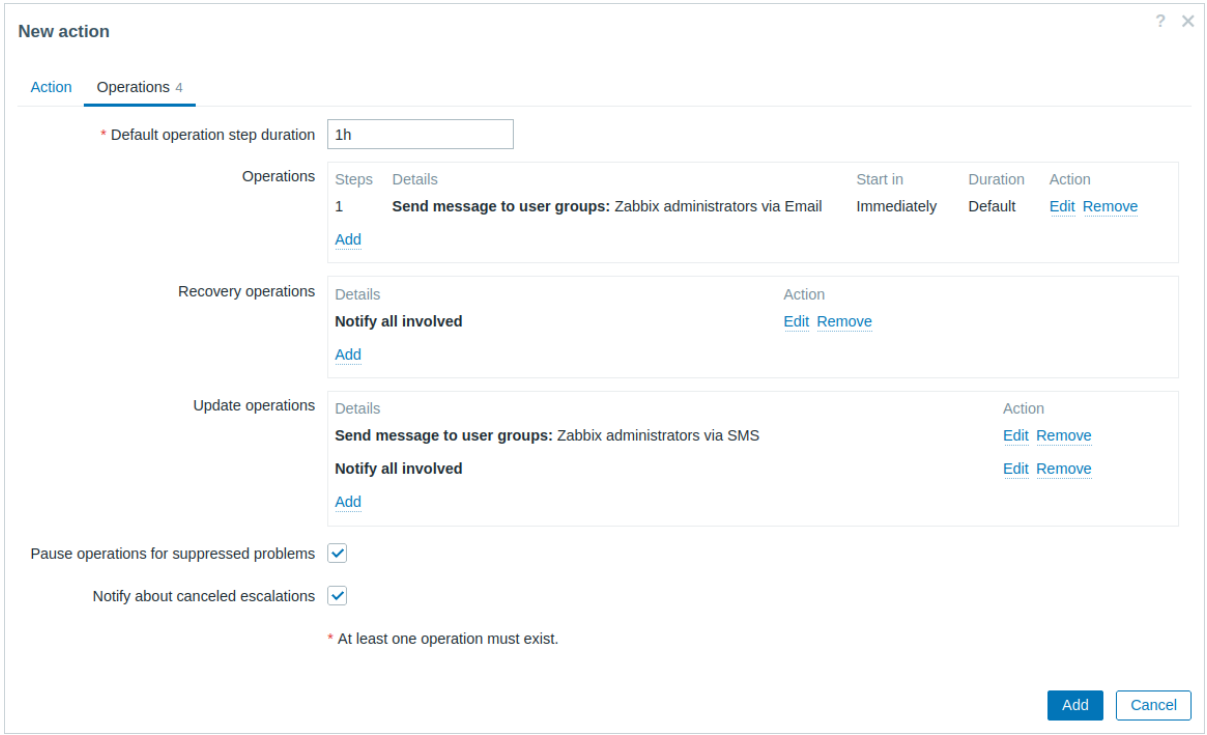
Alcuni casi d'uso per le operazioni di ripristino sono i seguenti:

1. Notificare al ripristino tutti gli utenti che sono stati notificati del problema:
   - Selezionare *Notifica tutti i coinvolti* come tipo di operazione.
2. Avere più operazioni al ripristino: inviare una notifica ed eseguire un comando remoto:
   - Aggiungere i tipi di operazione per l'invio di un messaggio e l'esecuzione di un comando.
3. Aprire un ticket in un sistema esterno di helpdesk/ticketing e chiuderlo quando il problema viene risolto:
   - Creare uno script esterno che comunichi con il sistema di helpdesk.
   - Creare un'azione con un'operazione che esegua questo script e apra quindi un ticket.
   - Avere un'operazione di ripristino che esegua questo script con altri parametri e chiuda il ticket.
   - Utilizzare la macro {EVENT.ID} per fare riferimento al problema originale.

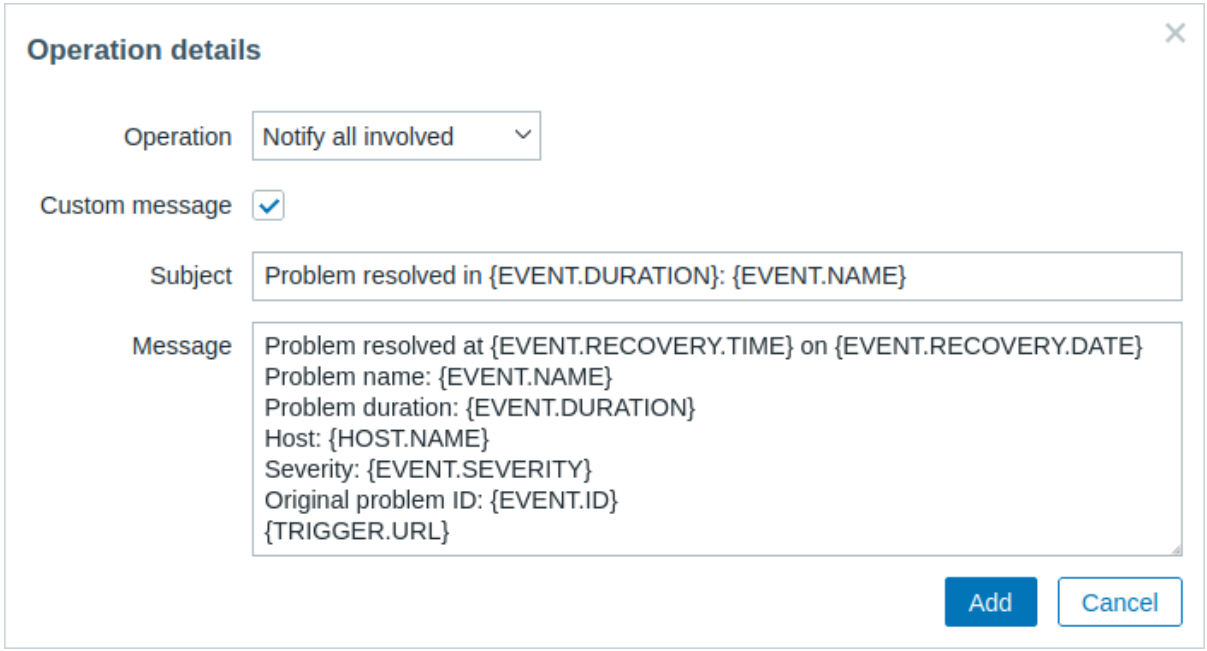
Configurazione di un'operazione di ripristino

Per configurare un'operazione di ripristino, seguire questi passaggi:

1. Aprire la scheda *Operations* nel modulo di configurazione dell'azione.
2. Per configurare una nuova operazione di ripristino, fare clic su *Add* nella sezione *Recovery operations*. Per modificare un'operazione esistente, fare clic su *Edit* accanto all'operazione.
3. Configurare i dettagli dell'operazione.



Dettagli dell'operazione di ripristino



Per gli eventi di ripristino sono disponibili tre tipi di operazione:

- **Invia messaggio** - invia un messaggio di ripristino all'utente specificato;
- **Notifica tutti i coinvolti** - invia un messaggio di ripristino a tutti gli utenti che sono stati notificati per l'evento del problema;
- **<nome del comando remoto>** - esegue un comando remoto. I comandi sono disponibili per l'esecuzione se precedentemente definiti negli script globali con *Action operation* selezionato come ambito.

I parametri per ciascun tipo di operazione sono descritti di seguito. Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso. Al termine, fare clic su *Aggiungi* per aggiungere l'operazione all'elenco delle *Operazioni di ripristino*.

Note:
Si noti che, se lo stesso destinatario è definito in diversi tipi di operazione senza un *Messaggio personalizzato* specificato, non vengono inviate notifiche duplicate.

Tipo di operazione: invia messaggio

Parametro	Descrizione
Invia ai gruppi di utenti	Selezionare i gruppi di utenti a cui inviare il messaggio di ripristino. Per ricevere la notifica, il gruppo di utenti deve disporre almeno dei permessi di "lettura" per il host.
Invia agli utenti	Selezionare gli utenti a cui inviare il messaggio di ripristino. Per ricevere la notifica, l'utente deve disporre almeno dei permessi di "lettura" per il host.
Invia al tipo di supporto	Inviare il messaggio di ripristino predefinito a tutti i tipi di supporto disponibili (configurati e abilitati) oppure solo a uno specifico.
Messaggio personalizzato	Se selezionato, è possibile definire un messaggio personalizzato.
Oggetto	Oggetto del messaggio personalizzato. L'oggetto può contenere macro.
Messaggio	Il messaggio personalizzato. Il messaggio può contenere macro.

Tipo di operazione: comando remoto

Parametro	Descrizione
Elenco delle destinazioni	Selezionare le destinazioni su cui eseguire il comando: **Host corrente** - il comando viene eseguito sull'host del trigger che ha causato l'evento di problema. Questa opzione non funziona se nel trigger sono presenti più host. **Host** - selezionare gli host su cui eseguire il comando. **Gruppo di host** - selezionare i gruppi di host su cui eseguire il comando. Specificando un gruppo di host principale, vengono selezionati implicitamente tutti i gruppi di host nidificati. Pertanto, il comando remoto verrà eseguito anche sugli host appartenenti ai gruppi nidificati. Un comando su un host viene eseguito una sola volta, anche se l'host corrisponde più di una volta (ad esempio, da diversi gruppi di host, singolarmente e da un gruppo di host). L'elenco delle destinazioni non ha significato se il comando viene eseguito sul server Zabbix. In questo caso, selezionare più destinazioni comporta solo l'esecuzione del comando sul server un numero maggiore di volte. Per gli script globali, la selezione della destinazione dipende anche dall'impostazione *Gruppo di host* nella configurazione dello script globale.

Tipo di operazione: notifica tutti i coinvolti

Parametro		Descrizione
Messaggio personalizzato		Se selezionato, è possibile definire un messaggio personalizzato.
	Oggetto	Oggetto del messaggio personalizzato. L'oggetto può contenere macro.
	Messaggio	Il messaggio personalizzato. Il messaggio può contenere macro.

Operazioni di aggiornamento

Panoramica

Le operazioni di aggiornamento sono disponibili nelle azioni con le seguenti sorgenti di eventi:

- *Triggers* - quando i problemi vengono aggiornati da altri utenti, ad esempio con un commento, riconosciuti, con la severità modificata, chiusi (manualmente);
- *Services* - quando la severità di un servizio è cambiata ma il servizio non è ancora ripristinato.

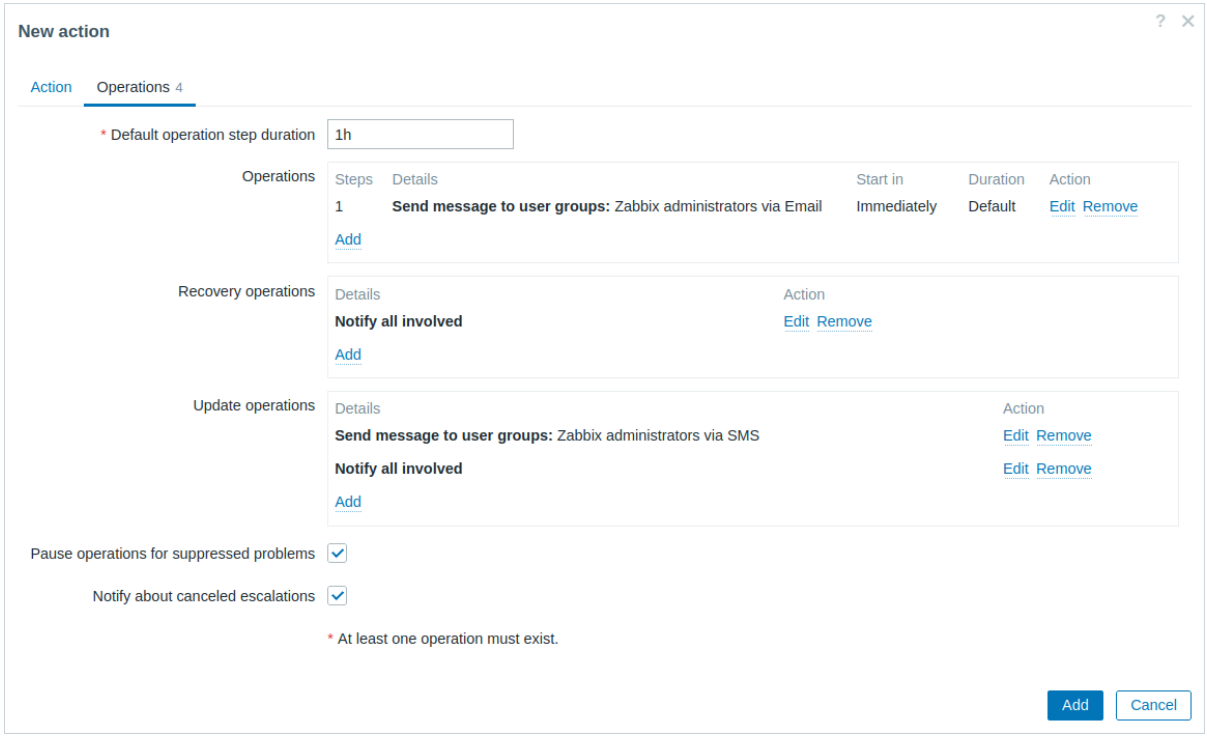
Si noti che gli utenti non ricevono notifiche sui propri aggiornamenti.

Sia i messaggi sia i comandi remoti sono supportati nelle operazioni di aggiornamento. Sebbene possano essere aggiunte più operazioni, l'escalation non è supportata: tutte le operazioni sono assegnate a un singolo passaggio e quindi verranno eseguite simultaneamente.

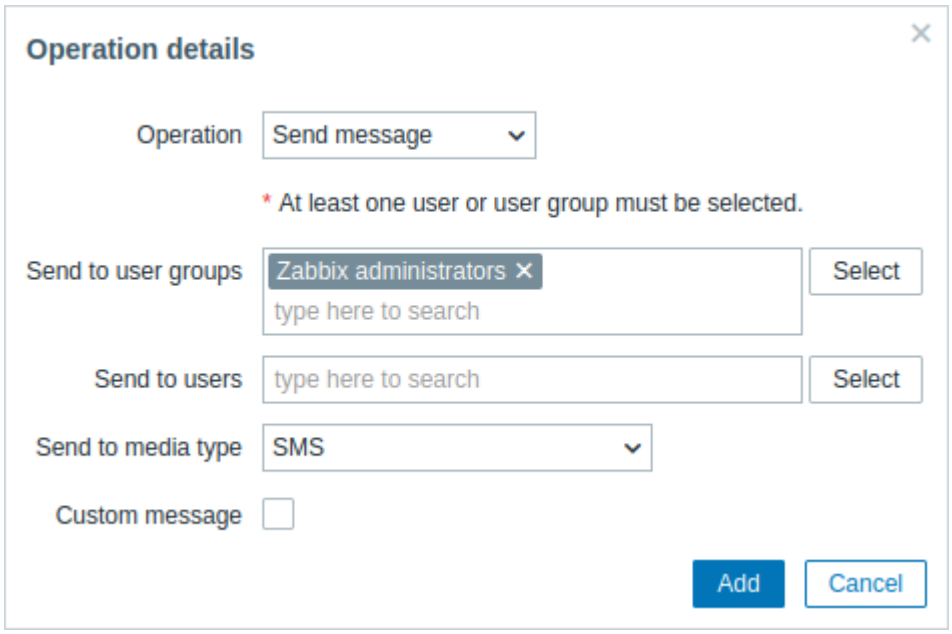
Configurazione di un'operazione di aggiornamento

Per configurare un'operazione di aggiornamento, segui questi passaggi:

1. Vai alla scheda *Operations* nel modulo di configurazione dell'azione.
2. Per configurare una nuova operazione di aggiornamento, fai clic su *Add* nella sezione *Update operations*. Per modificare un'operazione esistente, fai clic su *Edit* accanto all'operazione.
3. Configura i dettagli dell'operazione.



Dettagli dell'operazione di aggiornamento



Le operazioni di aggiornamento offrono lo stesso insieme di parametri delle operazioni di ripristino.

Escalationi

Panoramica

Con le escalation è possibile creare scenari personalizzati per l'invio di notifiche o l'esecuzione di comandi remoti.

In termini pratici, ciò significa che:

- Gli utenti possono essere informati immediatamente dei nuovi problemi.
- Le notifiche possono essere ripetute finché il problema non viene risolto.
- L'invio di una notifica può essere ritardato.
- Le notifiche possono essere escalate a un altro gruppo di utenti di livello "superiore".
- I comandi remoti possono essere eseguiti immediatamente oppure quando un problema non viene risolto per un lungo periodo.

Le azioni vengono escalate in base al **passo di escalation**. Ogni passo ha una durata temporale.

È possibile definire sia la durata predefinita sia una durata personalizzata di un singolo passo. La durata minima di un passo di escalation è di 60 secondi.

È possibile avviare azioni, come l'invio di notifiche o l'esecuzione di comandi, da qualsiasi passo. Il primo passo è destinato alle azioni immediate. Se si desidera ritardare un'azione, è possibile assegnarla a un passo successivo. Per ogni passo, possono essere definite diverse azioni.

Il numero di passi di escalation non è limitato.

Le escalation vengono definite durante la configurazione di un'operazione. Le escalation sono supportate solo per le operazioni sui problemi, non per il ripristino.

Aspetti vari del comportamento delle escalation

Consideriamo cosa accade in circostanze diverse se un'azione contiene diversi passaggi di escalation.

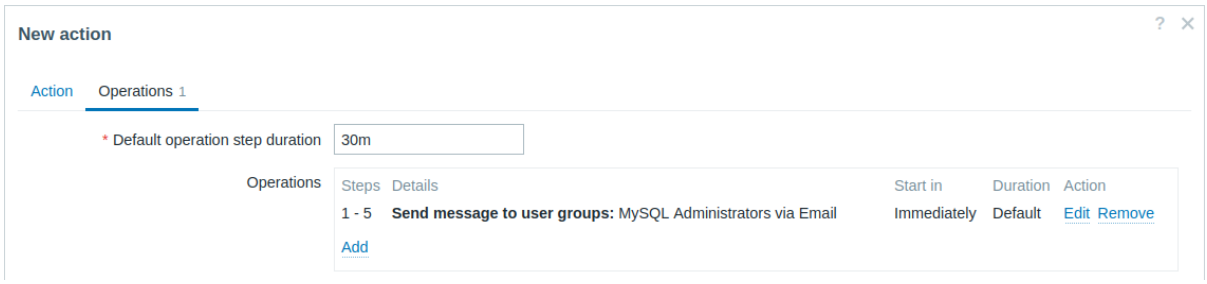
Situazione	Comportamento
Il host in questione entra in manutenzione dopo l'invio della notifica iniziale	A seconda dell'impostazione *Metti in pausa le operazioni per i problemi soppressi* nella configurazione dell'azione, tutti i passaggi di escalation rimanenti vengono eseguiti con un ritardo causato dal periodo di manutenzione oppure senza ritardo. Un periodo di manutenzione non annulla le operazioni.
Il periodo definito nella condizione ***Periodo di tempo*** *dell'azione termina dopo l'invio della notifica iniziale*	Tutti i passaggi di escalation rimanenti vengono eseguiti. La condizione *Periodo di tempo* non può interrompere le operazioni; influisce sul momento in cui le azioni vengono avviate o non avviate, non sulle operazioni.
Un problema inizia durante la manutenzione e continua (non viene risolto) dopo la fine della manutenzione	A seconda dell'impostazione *Metti in pausa le operazioni per i problemi soppressi* nella configurazione dell'azione, tutti i passaggi di escalation vengono eseguiti dal momento in cui termina la manutenzione oppure immediatamente.
Un problema inizia durante una manutenzione senza dati e continua (non viene risolto) dopo la fine della manutenzione	Deve attendere che il trigger si attivi prima che vengano eseguiti tutti i passaggi di escalation.
Escalation diverse si susseguono a breve distanza e si sovrappongono	L'esecuzione di ogni nuova escalation sostituisce quella precedente, ma almeno un passaggio di escalation viene sempre eseguito nell'escalation precedente. Questo comportamento è rilevante nelle azioni relative a eventi creati a ogni valutazione del problema del trigger.
Durante un'escalation in corso (ad esempio, mentre viene inviato un messaggio), in base a qualsiasi tipo di evento: - l'azione viene disabilitata In base a un evento del trigger: - il trigger viene disabilitato - il host o l'item viene disabilitato In base a un evento interno relativo ai trigger: - il trigger viene disabilitato In base a un evento interno relativo agli item/regole di discovery di basso livello: - l'item viene disabilitato - il host viene disabilitato	Il messaggio in corso viene inviato, quindi viene inviato un ulteriore messaggio relativo all'escalation. Il messaggio successivo conterrà all'inizio del corpo del messaggio il testo di annullamento (*NOTA: Escalation annullata*) con l'indicazione del motivo (ad esempio, *NOTA: Escalation annullata: azione '<Action name>' disabilitata*). In questo modo il destinatario viene informato che l'escalation è stata annullata e che non verranno eseguiti altri passaggi. Questo messaggio viene inviato a tutti coloro che hanno ricevuto le notifiche precedenti. Il motivo dell'annullamento viene inoltre registrato nel file di log del server (a partire dal livello di debug 3=Warning). Si noti che il messaggio *Escalation annullata* viene inviato anche se le operazioni sono terminate, ma sono configurate operazioni di ripristino che non sono ancora state eseguite.
Durante un'escalation in corso (ad esempio, mentre viene inviato un messaggio) l'azione viene eliminata	Non vengono inviati altri messaggi. L'informazione viene registrata nel file di log del server (a partire dal livello di debug 3=Warning), ad esempio: `escalation canceled: action id:334 deleted`

Esempi di escalation

Esempio 1

Invio di una notifica ripetuta una volta ogni 30 minuti (5 volte in totale) a un gruppo "MySQL Administrators". Per configurarlo:

- Nella scheda *Operations*, impostare *Default operation step duration* su "30m" (30 minuti).
- Impostare i *Steps* dell'escalation da "1" a "5".
- Selezionare il gruppo "MySQL Administrators" come destinatario del messaggio.



Le notifiche verranno inviate a 0:00, 0:30, 1:00, 1:30, 2:00 ore dopo l'inizio del problema (a meno che, naturalmente, il problema non venga risolto prima).

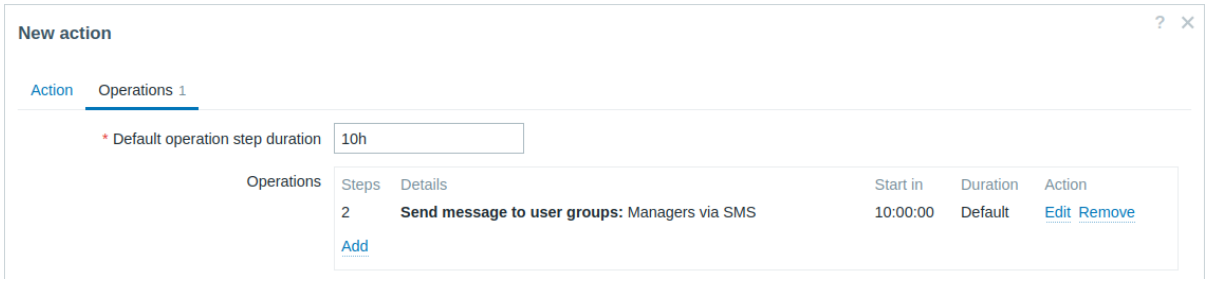
Se il problema viene risolto ed è configurato un messaggio di ripristino, esso verrà inviato a coloro che hanno ricevuto almeno un messaggio di problema all'interno di questo scenario di escalation.

Note:
Se il trigger che ha generato un'escalation attiva viene disabilitato, Zabbix invia un messaggio informativo al riguardo a tutti coloro che hanno già ricevuto notifiche.

Esempio 2

Invio di una notifica ritardata relativa a un problema persistente. Per configurare:

- Nella scheda *Operations*, impostare *Default operation step duration* su "10h" (10 ore).
- Impostare gli *Steps* dell'escalation da "2" a "2".



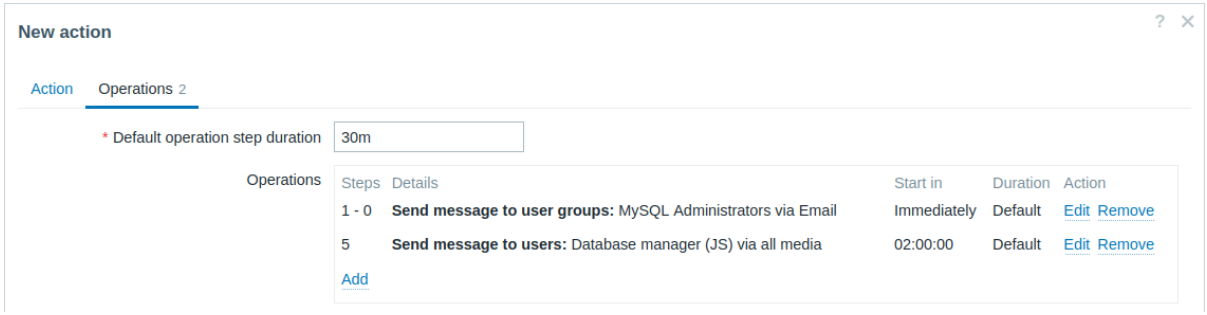
Una notifica verrà inviata solo allo Step 2 dello scenario di escalation, ovvero 10 ore dopo l'inizio del problema.

È possibile personalizzare il testo del messaggio con qualcosa come "Il problema dura da più di 10 ore".

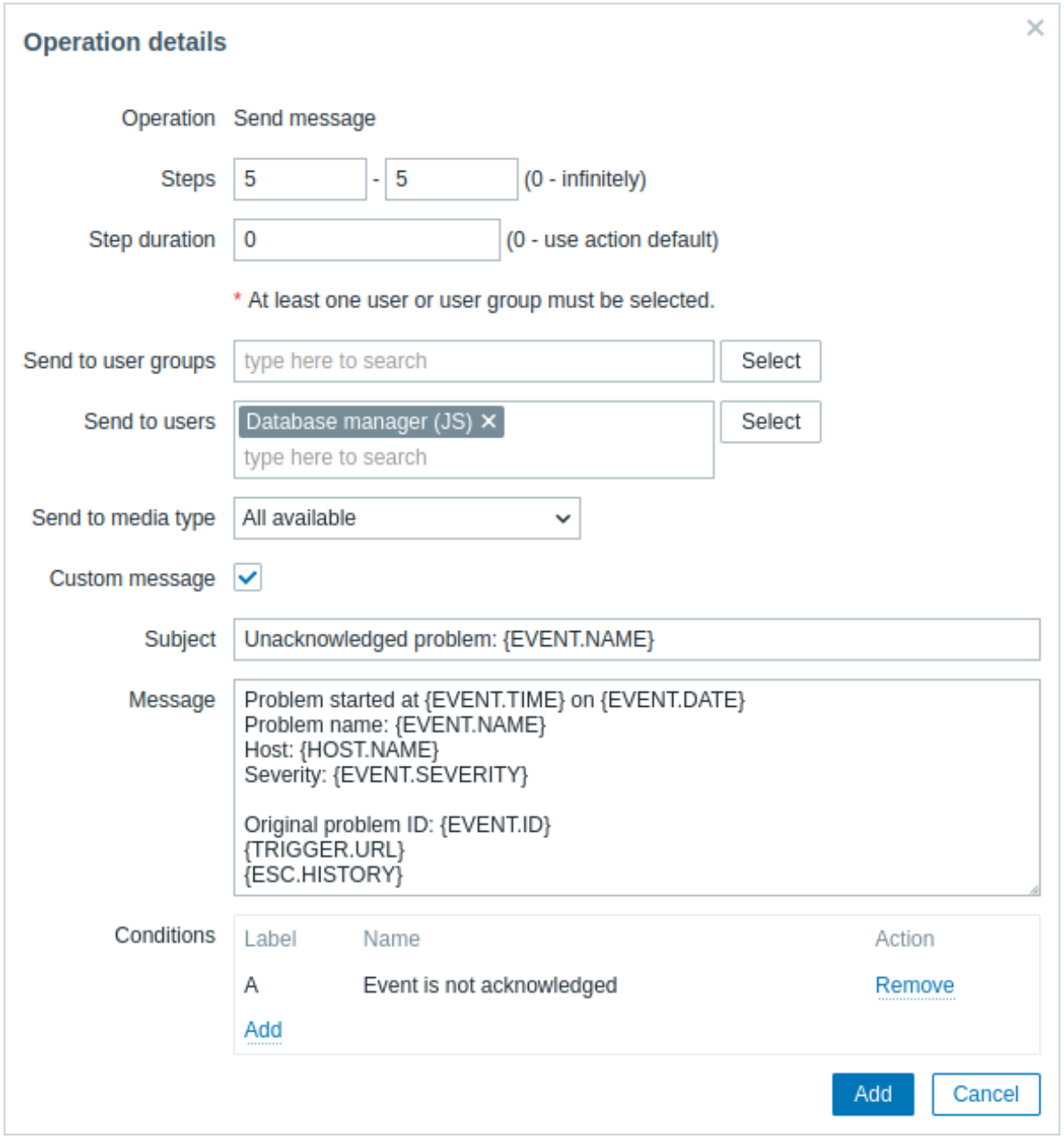
Esempio 3

Escalation del problema al responsabile.

Nel primo esempio sopra abbiamo configurato l'invio periodico di messaggi agli amministratori MySQL. In questo caso, gli amministratori riceveranno quattro messaggi prima che il problema venga escalato al responsabile del database. Si noti che il responsabile riceverà un messaggio solo nel caso in cui il problema non sia ancora stato confermato, presumibilmente perché nessuno ci sta ancora lavorando.



Dettagli dell'operazione 2:



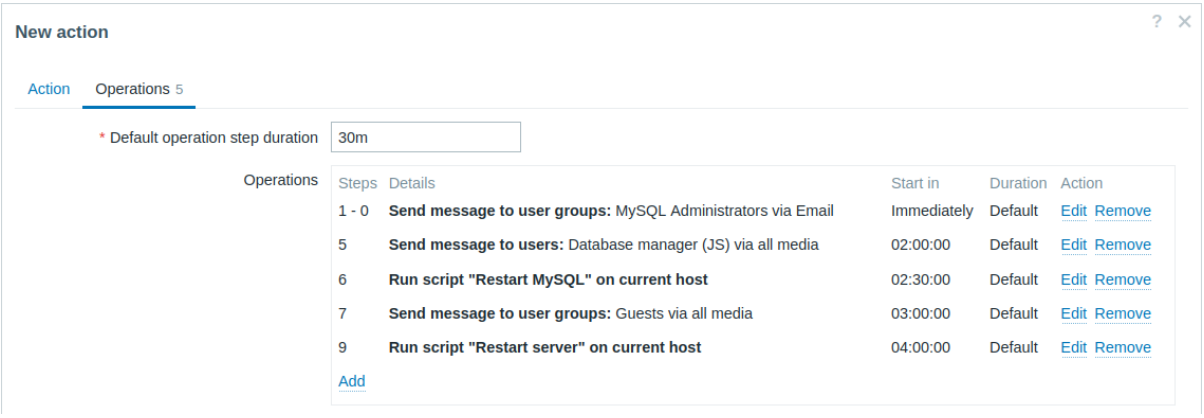
Si noti l'uso della macro {ESC.HISTORY} nel messaggio personalizzato. La macro conterrà informazioni su tutti i passaggi eseguiti in precedenza in questa escalation, come le notifiche inviate e i comandi eseguiti.

Esempio 4

Uno scenario più complesso. Dopo più messaggi agli amministratori MySQL e l'escalation al manager, Zabbix proverà a riavviare il database MySQL. Ciò avverrà se il problema esiste da 2:30 ore e non è stato riconosciuto.

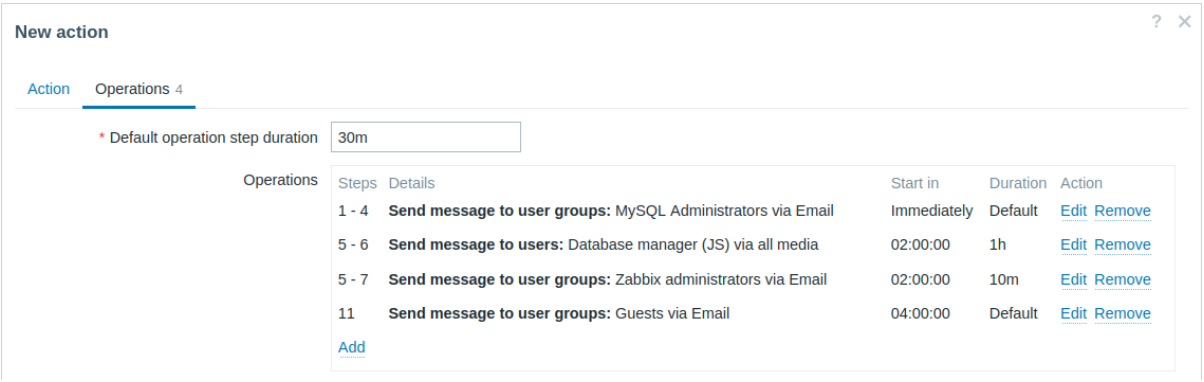
Se il problema esiste ancora, dopo altri 30 minuti Zabbix invierà un messaggio a tutti gli utenti guest.

Se questo non aiuta, dopo un'altra ora Zabbix riavvierà il server con il database MySQL (secondo comando remoto) utilizzando comandi IPMI.



Esempio 5

Un'escalation con diverse operazioni che hanno intervalli di step sovrapposti e intervalli personalizzati. La durata predefinita dello step dell'operazione è di 30 minuti.



Le notifiche verranno inviate come segue:

- Agli amministratori MySQL a 0:00, 0:30, 1:00 e 1:30 dopo l'inizio del problema.
- Al responsabile del database a 2:00 e 2:10 (la durata personalizzata dello step più breve di 10 minuti definita nell'operazione successiva prevale sulla durata dello step più lunga di 1 ora configurata qui, come descritto in Dettagli dell'operazione per *Durata dello step* quando gli step si sovrappongono).
- Agli amministratori Zabbix a 2:00, 2:10 e 2:20 dopo l'inizio del problema (viene applicata la durata personalizzata dello step di 10 minuti).
- Agli utenti guest a 4:00 dopo l'inizio del problema (la durata predefinita dello step di 30 minuti entra in vigore tra gli step 8 e 11).

Ricezione di notifiche su item non supportati

Panoramica

È possibile ricevere notifiche sugli item non supportati in Zabbix.

Fa parte del concetto di eventi interni in Zabbix, che consente agli utenti di essere notificati in queste occasioni. Gli eventi interni riflettono un cambiamento di stato:

- quando gli item passano da "normale" a "non supportato" (e viceversa);
- quando i trigger passano da "normale" a "sconosciuto" (e viceversa);
- quando le regole di discovery a basso livello passano da "normale" a "non supportato" (e viceversa).

Questa sezione presenta una guida pratica per **ricevere una notifica** quando un item diventa non supportato.

Configurazione

Nel complesso, il processo di impostazione della notifica dovrebbe sembrare familiare a coloro che hanno già impostato avvisi in Zabbix.

Passaggio 1

Configurare un supporto di notifica, ad esempio email, SMS o script, da utilizzare per le notifiche. Fare riferimento alle sezioni corrispondenti del manuale per eseguire questa operazione.

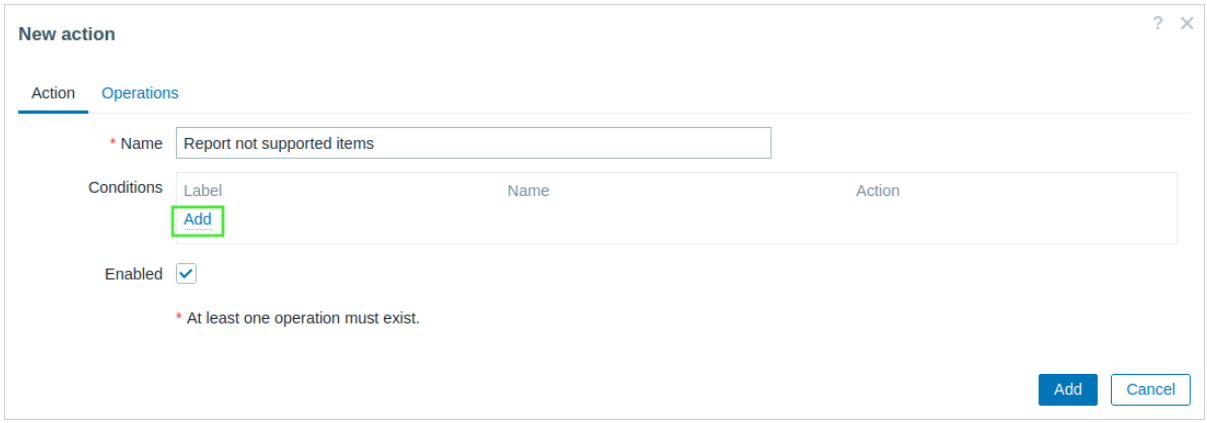
Attention:
Per le notifiche sugli eventi interni viene utilizzata la gravità predefinita (‘Not classified’), quindi lasciarla selezionata durante la configurazione dei supporti utente se si desidera ricevere notifiche per gli eventi interni.

Passaggio 2

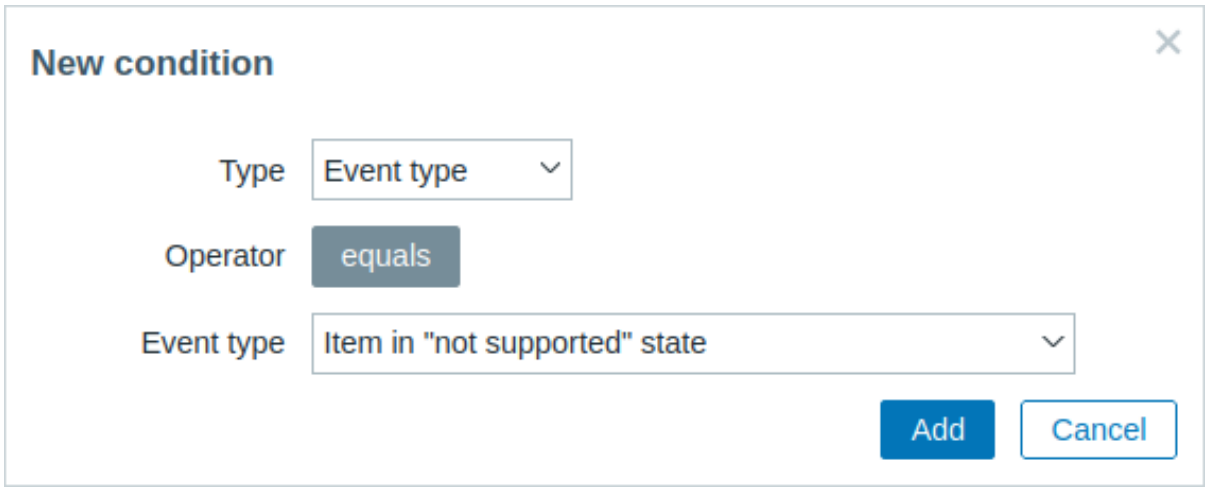
1. Vai a *Alerts* > *Actions* > *Internal actions*.
2. Fai clic su *Create action* nell’angolo in alto a destra della pagina per aprire un modulo di configurazione dell’azione.

Passaggio 3

1. Nella scheda *Action*, inserire un nome per l’azione.
2. Fare clic su *Add* nel blocco *Conditions* per aggiungere una nuova condizione.



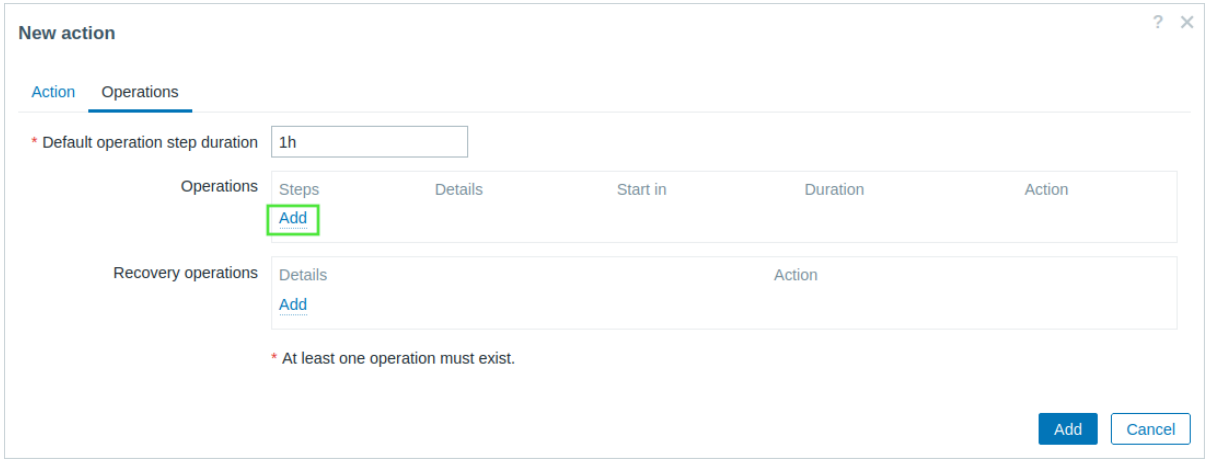
3. Nella finestra pop-up *New condition*, selezionare ”Event type” come tipo di condizione, quindi selezionare ”Item in ’not supported’ state” come tipo di evento.



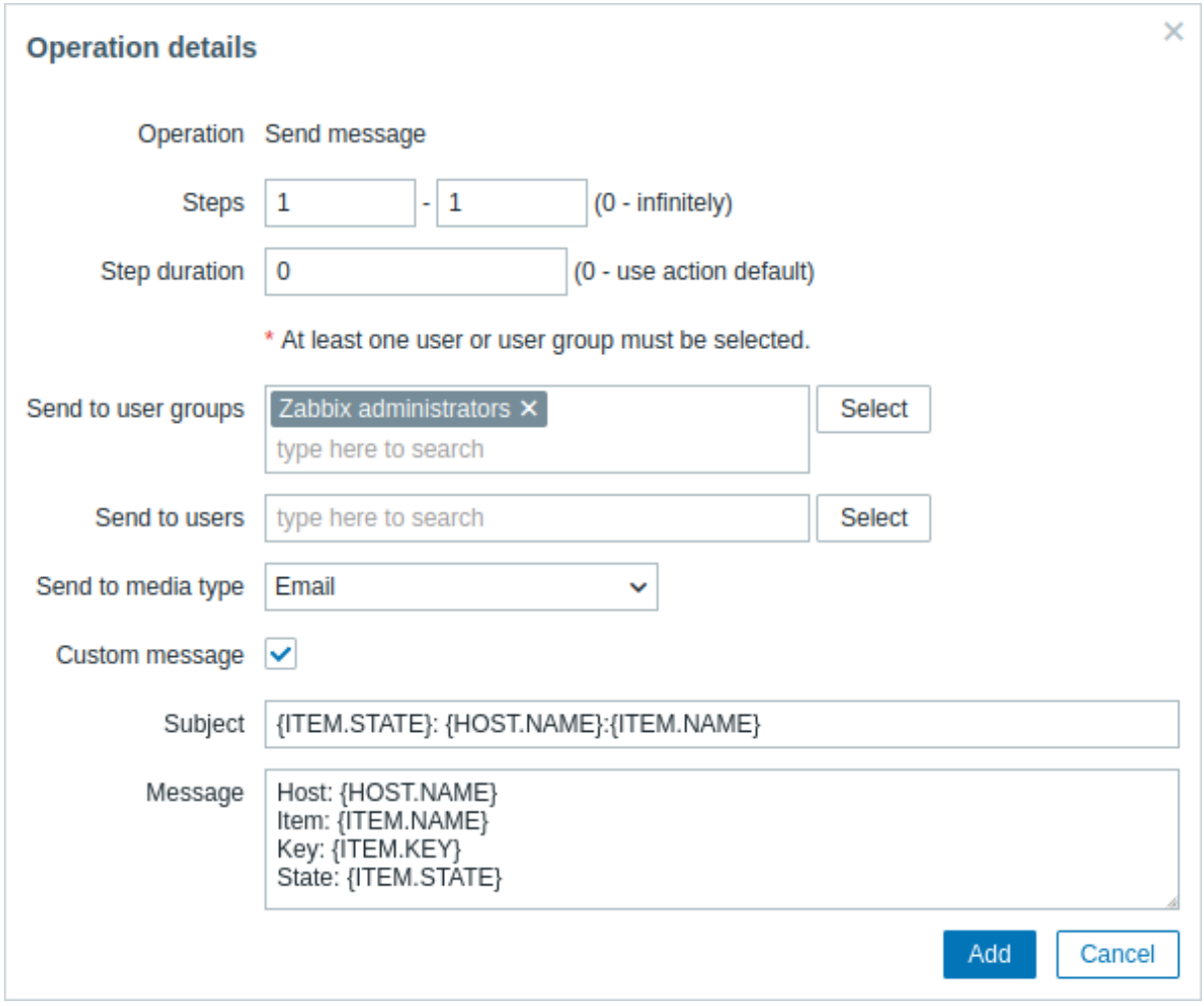
4. Fare clic su *Add* per elencare effettivamente la condizione nel blocco *Conditions*.

Passaggio 4

1. Nella scheda *Operations*, fare clic su *Add* nel blocco *Operations* per aggiungere una nuova operazione.



2. Selezionare alcuni destinatari del messaggio (gruppi di utenti/utenti) e il tipo di supporto (oppure ”All”) da utilizzare per la consegna.
3. Selezionare la casella di controllo *Custom message* se si desidera inserire l’oggetto e il contenuto personalizzati del messaggio relativo al problema.



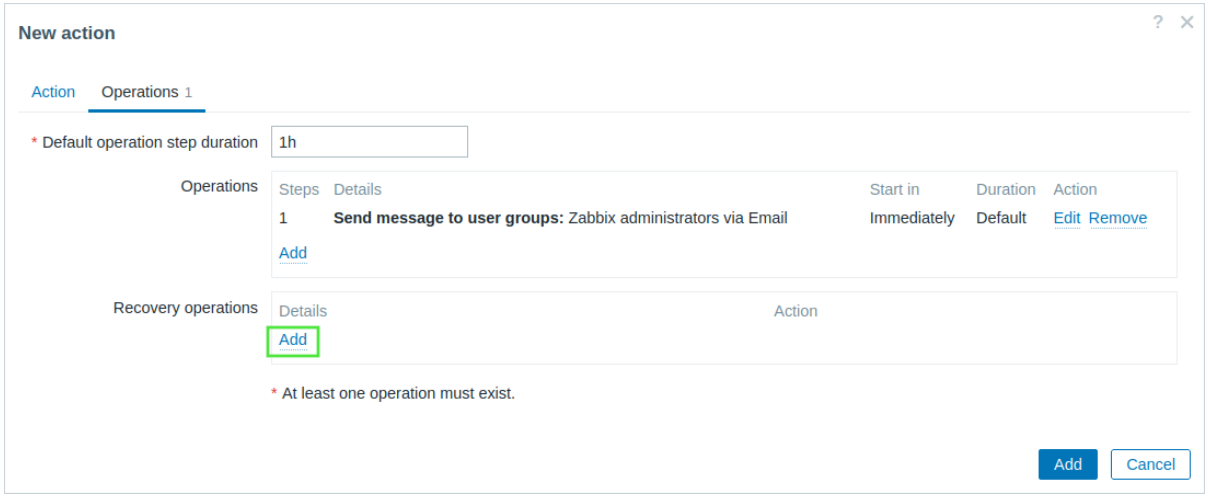
4. Fare clic su *Add* per aggiungere effettivamente l’operazione al blocco *Operations*.

Se si desidera ricevere più di una notifica, impostare la durata del passaggio dell’operazione (intervallo tra i messaggi inviati) e aggiungere un altro passaggio.

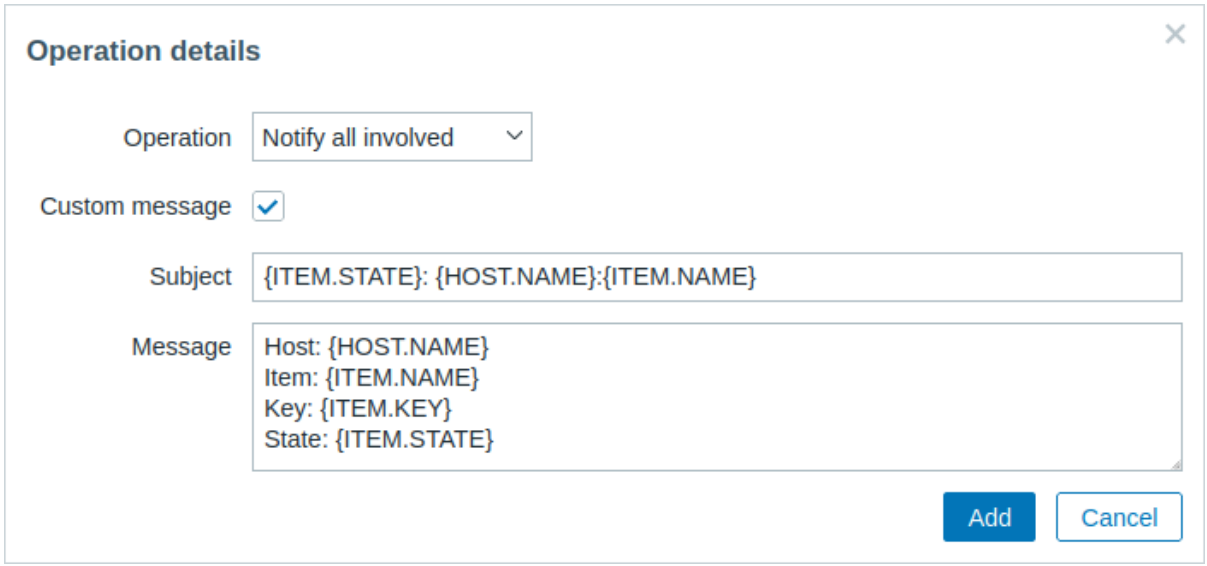
Passaggio 5

Il blocco *Operazioni di ripristino* consente di configurare una notifica di ripristino quando un item torna allo stato normale.

1. Fare clic su *Aggiungi* nel blocco *Operazioni di ripristino* per aggiungere una nuova operazione di ripristino.



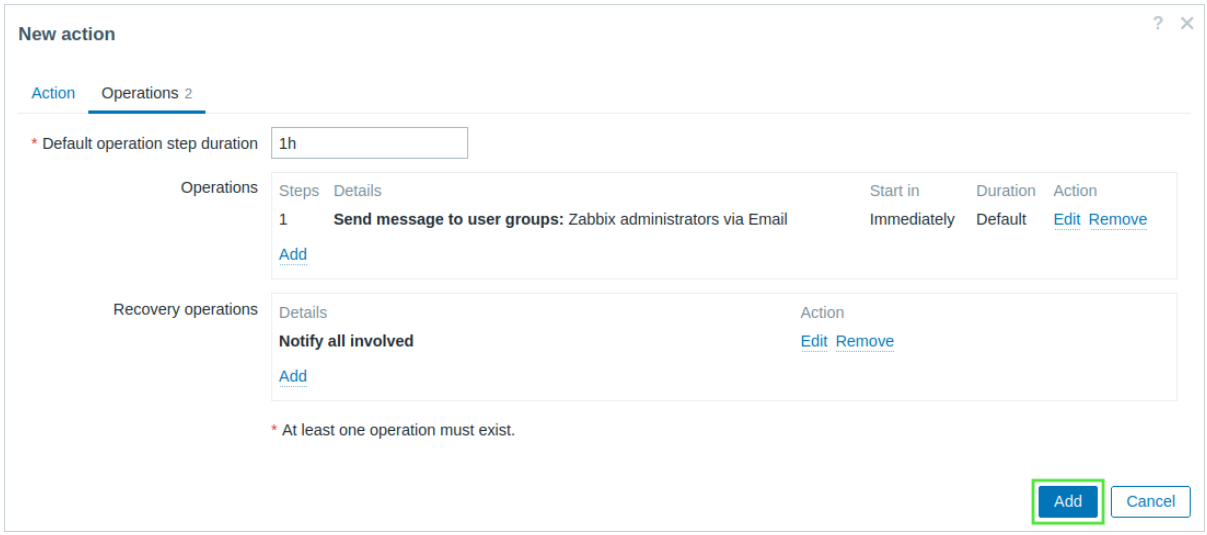
2. Selezionare il tipo di operazione ”Notifica a tutti gli interessati”.
3. Selezionare la casella di controllo *Messaggio personalizzato* se si desidera inserire l’oggetto e il contenuto personalizzati del messaggio di ripristino.



4. Fare clic su *Aggiungi* nella finestra pop-up *Dettagli dell’operazione* per elencare effettivamente l’operazione nel blocco *Operazioni di ripristino*.

Passaggio 6

Al termine, fai clic sul pulsante *Add* in fondo al modulo.



Ecco fatto, hai finito! Ora puoi aspettarti di ricevere la tua prima notifica da Zabbix se qualche item diventa non supportato.

Macro

Panoramica

Zabbix supporta numerose macro integrate che possono essere utilizzate in varie situazioni. Queste macro sono variabili, identificate da una sintassi specifica:

`{MACRO}`

Le macro vengono risolte in un valore specifico a seconda del contesto.

L'uso efficace delle macro consente di risparmiare tempo e rendere la configurazione di Zabbix più trasparente.

In uno degli utilizzi tipici, una macro può essere usata in un template. Ad esempio, un trigger in un template può essere denominato "Il carico del processore è troppo elevato su {HOST.NAME}". Quando il template viene applicato all'host, ad esempio al server Zabbix, il nome verrà risolto in "Il carico del processore è troppo elevato su Zabbix server" quando il trigger viene visualizzato nella sezione Monitoraggio.

Le macro possono essere utilizzate nei parametri delle chiavi degli item. Una macro può essere utilizzata solo per una parte del parametro, ad esempio `item.key[server_{HOST.HOST}_local]`. Non è necessario racchiudere il parametro tra virgolette doppie, poiché Zabbix gestirà eventuali simboli speciali ambigui, se presenti nella macro risolta.

In Zabbix sono disponibili anche altri tipi di macro.

Zabbix supporta le seguenti macro:

- `{MACRO}` - macro integrata (vedere elenco completo)
- `{<macro>.<func>(<params>)}` - funzioni delle macro
- `{$MACRO}` - macro definita dall'utente, facoltativamente con contesto
- `{#MACRO}` - macro per la low-level discovery
- `{?EXPRESSION}` - macro di espressione

Funzioni macro

Panoramica

Le funzioni macro offrono la possibilità di personalizzare i valori delle macro (ad esempio, accorciando o estraendo sottostringhe specifiche), rendendole più facili da usare.

La sintassi di una funzione macro è:

```
{macro.func(params)}
```

dove

- **macro** - la macro da personalizzare;
- **func** - la funzione da applicare (vedi funzioni supportate);
- **params** - un elenco di parametri della funzione separati da virgole, che devono essere racchiusi tra **doppi apici** se:
  - iniziano con uno spazio o con doppi apici;
  - contengono parentesi tonde di chiusura o una virgola.

Per esempio:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

Le funzioni macro sono supportate per

- macro integrate
- macro utente
- macro di discovery a basso livello
- macro di espressione

Le funzioni macro possono essere usate in tutte le posizioni che supportano le macro elencate. Questo vale salvo che sia esplicitamente indicato che è prevista solo una macro (ad esempio, durante la configurazione delle macro host o dei filtri delle regole di discovery a basso livello).

È supportata una sola funzione per macro; non sono supportate più funzioni macro in catena.

Note:
Consultare gli esempi di escaping per i casi in cui le funzioni macro vengono usate all'interno di altri contesti (funzione, chiave item, un'altra macro, ecc).

Funzioni supportate

Le funzioni sono elencate senza informazioni aggiuntive.
Fare clic sulla funzione per visualizzare tutti i dettagli.

Funzione	Descrizione
btoa	Codifica del valore della macro in formato Base64.
fmtnum	Formattazione dei numeri per controllare il numero di cifre visualizzate dopo il separatore decimale.
fmttime	Formattazione dell'ora.
htmldecode	Decodifica del valore della macro dalla codifica HTML.
htmlencode	Codifica del valore della macro nella codifica HTML.
iregsub	Estrazione di una sottostringa tramite una corrispondenza con un'espressione regolare (senza distinzione tra maiuscole e minuscole).
jsonpath	Restituzione del risultato JSONPath.
lowercase	Conversione dei caratteri del valore della macro in minuscolo.
regrepl	Sostituzione di un carattere o di una sottostringa nel valore della macro.
regsub	Estrazione di una sottostringa tramite una corrispondenza con un'espressione regolare (con distinzione tra maiuscole e minuscole).
tr	Traslitterazione dei caratteri del valore della macro.
uppercase	Conversione dei caratteri del valore della macro in maiuscolo.
urldecode	Decodifica del valore della macro dalla codifica URL.
urlencode	Codifica del valore della macro nella codifica URL.
xmlxpath	Restituzione del risultato XML XPath.

Dettagli della funzione

I parametri opzionali della funzione sono indicati da < >.

btoa

Codifica di un valore macro in codifica Base64. La codifica Base64 è un metodo per rappresentare dati binari come testo, utile per l'archiviazione e la trasmissione sicura di contenuti binari tramite protocolli basati su testo.

Esempio:

```
{{ITEM.VALUE}.btoa()} - codificherà in Base64 un valore come "zabbix" in "emFiYml4"
```

fmtnum(digits)

Formattazione dei numeri per controllare il numero di cifre stampate dopo il separatore decimale.

Parametri:

- **digits** - il numero di cifre dopo il separatore decimale. Intervallo valido: 0-20. Non verranno prodotti zeri finali.

Esempi:

```
{{ITEM.VALUE}.fmtnum(2)} - restituirà "24.35" da un valore ricevuto di "24.3483523"
{{ITEM.VALUE}.fmtnum(0)} - restituirà "24" da un valore ricevuto di "24.3483523"
```

fmttime(format,<time_shift>)

Formattazione dell'ora.
 Si noti che questa funzione può essere usata con macro che risolvono in un valore in uno dei seguenti formati di ora:

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (standard ISO8601)
- timestamp UNIX

Parametri:

- **format** - stringa di formato obbligatoria, compatibile con la formattazione della funzione `strftime`;
- **time_shift** (opzionale) - lo spostamento temporale applicato all'ora prima della formattazione; deve iniziare con `-<N><time_unit>` o `+<N><time_unit>`, dove:

- `N` - il numero di unità di tempo da aggiungere o sottrarre;
- `time_unit` - h (ora), d (giorno), w (settimana), M (mese) o y (anno).

Commenti:

- Il parametro `time_shift` supporta operazioni temporali multistep e può includere `/<time_unit>` per spostarsi all'inizio dell'unità di tempo (`/d` - mezzanotte, `/w` - primo giorno della settimana (lunedì), `/M` - primo giorno del mese, ecc.). Esempi: `-1w` - esattamente 7 giorni indietro; `-1w/w` - lunedì della settimana precedente; `-1w/w+1d` - martedì della settimana precedente.
- Le operazioni temporali vengono calcolate da sinistra a destra senza priorità. Ad esempio, `-1M/d+1h/w` verrà interpretato come `((-1M/d)+1h)/w`.

Esempi:

```
{{TIMESTAMP}.fmttime(%B)} - restituirà "October" da un valore ricevuto di "1633098961"
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} - restituirà "1 September" da un valore ricevuto di "1633098961"
{{EVENT.TIMESTAMP}.fmttime("%x %X")} - restituirà la rappresentazione di data e ora specifica della locale
{{EVENT.TIME}.fmttime("%x %X")} - restituirà la rappresentazione di data e ora specifica della locale; ad
```

htmldecode

Decodifica di un valore macro dalla codifica HTML.

Sono supportati i seguenti caratteri:

Valore	Valore decodificato
&	&
<	<
>	>
"	"
'	'
'	'

Esempio:

```
{{ITEM.VALUE}.htmldecode()} - decodificherà in HTML un valore come "&lt;" in "<"
```

htmlencode

Codifica di un valore macro nella codifica HTML.

Sono supportati i seguenti caratteri:

Valore	Valore codificato
&	&
<	<
>	>
"	"
'	'

Esempio:

```
{{ITEM.VALUE}.htmlencode()} - codificherà in HTML un carattere come "<" in "&lt;"
```

iregsub(pattern,output)

Estrazione di una sottostringa tramite corrispondenza con un'espressione regolare (senza distinzione tra maiuscole e minuscole).

Parametri:

- **pattern** - l'espressione regolare da confrontare;
- **output** - le opzioni di output. I segnaposto **\1 - \9** sono supportati per i gruppi di cattura. **\0** restituisce il testo corrispondente.

Commenti:

- Se non vi è alcuna corrispondenza per l'espressione regolare, la funzione restituisce una stringa vuota.
- Se il pattern della funzione è un'espressione regolare non valida, allora la macro viene valutata come 'UNKNOWN' (tranne per le macro di low-level discovery, nel qual caso la funzione verrà ignorata e la macro rimarrà irrisolta).
- I riferimenti a gruppi di cattura inesistenti nella stringa di sostituzione vengono sostituiti con una stringa vuota.

Esempio:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - verrà risolta in "ERROR" se vengono ricevute
```

jsonpath(pattern,<default>)

Restituisce il risultato JSONPath.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **pattern** - il pattern obbligatorio;
- **default** - il valore di fallback facoltativo se la query JSONPath non restituisce dati. In caso di altri errori viene restituito un errore.

Commenti:

- La funzione restituisce un errore se:
  - Il JSON non è valido.
  - La query JSONPath non riesce.
  - Il risultato JSONPath non contiene dati e il parametro `default` non è specificato.

Esempio:

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

minuscolo

Trasformazione di tutti i caratteri del valore della macro in minuscolo.
Funziona con set di caratteri a byte singolo (come ASCII) e non supporta UTF-8.

Esempio:

```
{{ITEM.VALUE}.lowercase()} - trasformerà un valore come "Zabbix SERVER" in "zabbix server" (minuscolo)
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

Sostituzione di carattere/sottostringa nel valore della macro.

Parametri:

- **pattern** - l'espressione regolare da confrontare;
- **replacement** - la stringa di sostituzione. I segnaposto **\1 - \9** sono supportati nelle stringhe di sostituzione per catturare i gruppi.

Commenti:

- Tutte le occorrenze che corrispondono al pattern vengono sostituite.
- I pattern e le sostituzioni vengono elaborati in sequenza, con ogni coppia successiva applicata in base al risultato della sostituzione precedente.
- I riferimenti a gruppi di cattura inesistenti nella stringa di sostituzione vengono sostituiti con una stringa vuota.

Esempi:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - sostituirà "oldParam" con "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - tutti i caratteri non alfabetici verranno escape con una barra
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - rimuoverà una barra rovesciata finale (ad esempio, per
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - sostituirà più parti nel valore dell'item
```

regsub(pattern,output)

Estrazione di sottostringhe tramite una corrispondenza con espressione regolare (con distinzione tra maiuscole e minuscole).

Parametri:

- **pattern** - l'espressione regolare da confrontare;
- **output** - le opzioni di output. I segnaposto **\1 - \9** sono supportati per i gruppi di cattura. **\0** restituisce il testo corrispondente.

Commenti:

- Se non vi è alcuna corrispondenza per l'espressione regolare, la funzione restituisce una stringa vuota.
- Se il pattern della funzione è un'espressione regolare non corretta, allora la macro viene valutata come 'UNKNOWN' (tranne per le macro di low-level discovery, nel qual caso la funzione verrà ignorata e la macro rimarrà irrisolta).
- I riferimenti a gruppi di cattura inesistenti nella stringa di sostituzione vengono sostituiti con una stringa vuota.

Esempi:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - verrà risolto in "Problem ID: 123" se viene ricevuto
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - verrà risolto in "ERROR" se vengono ricevute l
```

Vedi altri esempi.

tr(characters,replacement)

Traslitterazione dei caratteri del valore della macro.

- **characters** - l'insieme dei caratteri da sostituire;
- **replacement** - l'insieme dei caratteri sostitutivi corrispondenti per posizione.

Esempi:

```
{{ITEM.VALUE}.tr(abc, xyz)} - sostituirà tutte le occorrenze di "a" con "x", "b" con "y", "c" con "z"
{{ITEM.VALUE}.tr(abc, xyzq)} - sostituirà tutte le occorrenze di "a" con "x", "b" con "y", "c" con "z" ("q
{{ITEM.VALUE}.tr(abcde, xyz)} - sostituirà tutte le occorrenze di "a" con "x", "b" con "y", "c" con "z", "
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - sostituirà tutte le occorrenze della barra rovesciata con la barra, e
{{ITEM.VALUE}.tr(A-Z,a-z)} - convertirà tutte le lettere in minuscolo
{{ITEM.VALUE}.tr(0-9a-z,*)} - sostituirà tutti i numeri e le lettere minuscole con "*"
{{ITEM.VALUE}.tr(0-9,ab)} - sostituirà tutte le occorrenze di 0 con "a" e tutte le occorrenze di 1, 2, 3,
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - sostituirà tutti i numeri, i caratteri "abc" e l'intervallo A-L con "*"
{{ITEM.VALUE}.tr("\n","*")} - sostituirà le occorrenze di fine riga con *
{{ITEM.VALUE}.tr("e", "\n")} - sostituirà tutte le "e" con un fine riga
```

Per includere caratteri letterali:

```
barra rovesciata - deve essere preceduta da escape come \\
apice singolo - deve essere preceduto da escape come \'
doppio apice - deve essere preceduto da escape come \"
```

Sequenze di escape supportate con barra rovesciata:

```
\\\\ => \\ - doppia barra rovesciata in barra rovesciata singola
\\a  => \a - avviso
\\b  => \b - backspace
\\f  => \f - avanzamento pagina
\\n  => \n - nuova riga
\\r  => \r - ritorno carrello
\\t  => \t - tabulazione orizzontale
\\v  => \v - tabulazione verticale
```

maiuscolo

Trasformazione di tutti i caratteri del valore della macro in maiuscolo.
Funziona con set di caratteri a byte singolo (come ASCII) e non supporta UTF-8.

Esempio:

```
{{ITEM.VALUE}.uppercase()} - trasformerà un valore come "Zabbix Server" in "ZABBIX SERVER" (maiuscolo)
```

urldecode

Decodifica di un valore macro dalla codifica URL.

Esempio:

```
{{ITEM.VALUE}.urldecode()} - decodificherà tramite URL un valore come "%2F" in "/"
```

urlencode

Codifica di un valore macro nel formato URL encoding.

Esempio:

```
{{ITEM.VALUE}.urlencode()} - codificherà in URL un carattere come "/" in "%2F"
```

xmlxpath(pattern,<default>)

Restituisce il risultato XPath XML.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **pattern** - il pattern obbligatorio;

- **default** - il valore di fallback facoltativo se la query XPath non restituisce dati. In caso di altri errori viene restituito un errore.

Commenti:

- La funzione restituisce un errore se:
  - L'XML non è valido.
  - La query XPath non riesce.
  - Il risultato XPath non contiene dati e il parametro `default` non è specificato.
  - Zabbix è stato compilato senza il supporto XML.

Esempio:

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

Esempi aggiuntivi

La tabella seguente mostra altri esempi di utilizzo delle funzioni macro.

`{#IFALIAS}` è una macro LLD ed è definita solo nei contesti di low-level discovery (regole di discovery, prototipi e item/trigger creati da essi). Se utilizzata al di fuori di LLD, il token rimarrà non espanso.

Funzione macro	Valore ricevuto	Output
`{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}`	123Log line	Problem
`{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}`	123 Log line	Problem
`{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}`	Log line	Problem ID:
`{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}`	MySQL crashed errno 123	Problem ID: MySQL_123
`{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}`	123 Log line	*UNKNOWN* (espressione regolare non valida)
`{{#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)} (espressione regolare non valida)
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}`	customername_1	{$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)}"} (espressione regolare non valida)
`"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"`	customername_1	"{$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)}\"}" (espressione regolare non valida)

Visualizzazione dei valori completi degli item

I valori lunghi delle macro risolte {ITEM.VALUE} e {ITEM.LASTVALUE} per gli item di tipo testo/log vengono troncati a 20 caratteri in alcune posizioni del frontend. Per visualizzare i valori completi di queste macro è possibile utilizzare le funzioni macro, ad esempio:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

Vedi anche: dettagli delle macro {ITEM.VALUE} e {ITEM.LASTVALUE} macro details.

Macro utente

Panoramica

Le macro utente sono supportate in Zabbix per una maggiore flessibilità, oltre alle macro supportate immediatamente disponibili.

Le macro utente possono essere definite a livello globale, di template e di host. Queste macro hanno una sintassi speciale:

```
{$MACRO}
```

Zabbix risolve le macro secondo la seguente precedenza:

1. Macro a livello di host (controllate per prime).
2. Macro definite per i template di primo livello dell'host (ovvero, i template collegati direttamente all'host), ordinate per ID del template.
3. Macro definite per i template di secondo livello dell'host, ordinate per ID del template.
4. Macro definite per i template di terzo livello dell'host, ordinate per ID del template, ecc.
5. Macro globali (controllate per ultime).

In altre parole, se una macro non esiste per un host, Zabbix proverà a trovarla nei template dell'host a profondità crescente. Se non viene ancora trovata, verrà utilizzata una macro globale, se esistente.

Warning:
Se una macro con lo **stesso nome** esiste in più template collegati dello stesso livello, verrà utilizzata la macro del template con l'ID più basso. Pertanto, avere macro con lo stesso nome in più template rappresenta un rischio di configurazione.

Se Zabbix non riesce a trovare una macro, la macro non verrà risolta.

Attention:
Le macro, incluse le macro utente, rimangono non risolte nella sezione Configurazione (ad esempio, nell'elenco dei trigger) per progettazione, al fine di rendere più trasparente la configurazione complessa.

Le macro utente possono essere utilizzate in:

- nome dell'item
- parametro della chiave dell'item
- intervalli di aggiornamento dell'item e intervalli flessibili
- nome e descrizione del trigger
- parametri e costanti dell'espressione del trigger (vedere gli esempi)
- molte altre posizioni - vedere l'elenco completo

Casi d'uso comuni delle macro globali e del host

- utilizzare una macro globale in più posizioni; quindi modificare il valore della macro e applicare le modifiche di configurazione a tutte le posizioni con un clic
- sfruttare i template con attributi specifici del host: password, numeri di porta, nomi di file, espressioni regolari, ecc.

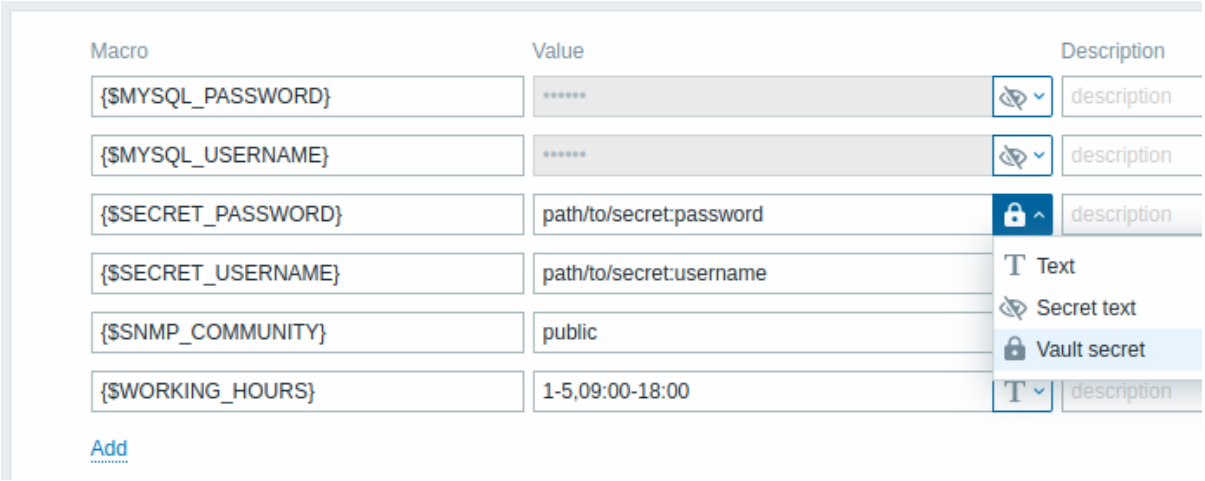
Note:
Si consiglia di usare le macro dell'host invece delle macro globali, perché l'aggiunta, l'aggiornamento o l'eliminazione delle macro globali impone un aggiornamento incrementale della configurazione per tutti gli host. Per ulteriori informazioni, vedere Passive and active agent checks.

Configurazione

Per definire le macro utente, vai alla posizione corrispondente nel frontend:

- Per le macro globali, visita *Amministrazione* > *Macro*.
- Per le macro a livello di host e template, apri le proprietà dell'host o del template e cerca la scheda *Macro*.

Una macro utente ha i seguenti attributi:



Parameter	Description
Macro	Nome della macro. Il nome deve essere racchiuso tra parentesi graffe e iniziare con il simbolo del dollaro. Esempio: `{$FRONTEND_URL}`. Nei nomi delle macro sono consentiti i seguenti caratteri: **A-Z** (solo maiuscole), **0-9**, _, **.**
Value	Valore della macro. Sono supportati tre tipi di valore: **Text** (predefinito) - valore in testo semplice **Secret text** - il valore viene mascherato con asterischi **Vault secret** - il valore contiene un percorso/query a un vault secret. Per cambiare il tipo di valore, fai clic sul pulsante alla fine del campo di input del valore. La lunghezza massima del valore di una macro utente è di 2048 caratteri.
Description	Campo di testo usato per fornire ulteriori informazioni su questa macro.

Quando configuri le macro utente, **nota** i seguenti comportamenti specifici del contesto:

- se le macro utente vengono usate negli item o nei trigger del template, considera di aggiungere tali macro anche al template (anche se sono definite globalmente); in questo modo, le macro di tipo *Text* funzioneranno come previsto dopo l'esportazione del template in XML e l'importazione in un altro sistema (i valori delle macro segrete non vengono esportati)
- se le macro utente vengono usate nelle espressioni di trigger, tali macro verranno risolte solo se fanno riferimento a un parametro o a una costante; NON verranno risolte se fanno riferimento a un host, a una chiave di item, a una funzione, a un operatore o a un'altra espressione di trigger (le macro segrete non possono essere usate nelle espressioni di trigger)
- se le macro utente vengono usate su un host che ha una regola di discovery a basso livello con prototipi di host, gli host scoperti erediteranno tutte le macro utente definite su quell'host

Esempi

Esempio 1

Utilizzo della macro a livello di host nella chiave dell'item "Status of SSH daemon":

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Questo item può essere assegnato a più host, a condizione che il valore di `{$SSH_PORT}` sia definito su tali host.

Esempio 2

Utilizzo di una macro a livello di host nel trigger "Il carico della CPU è troppo elevato":

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

Un trigger di questo tipo verrebbe creato nel template, non modificato nei singoli host.

> **Note:**
> Se si desidera utilizzare il numero di valori come parametro della funzione (ad esempio, `max(/host/key,#3)`), includere il simbolo cancelletto nella definizione della macro, come segue: `SOME_PERIOD => #3`.

Esempio 3

Utilizzo di due macro nel trigger "Il carico della CPU è troppo elevato":

`min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}`

Si noti che una macro può essere utilizzata come parametro della funzione del trigger, in questo esempio la funzione `min()`.

Esempio 4

Sincronizzare la condizione di indisponibilità dell'agent con l'intervallo di aggiornamento dell'item:

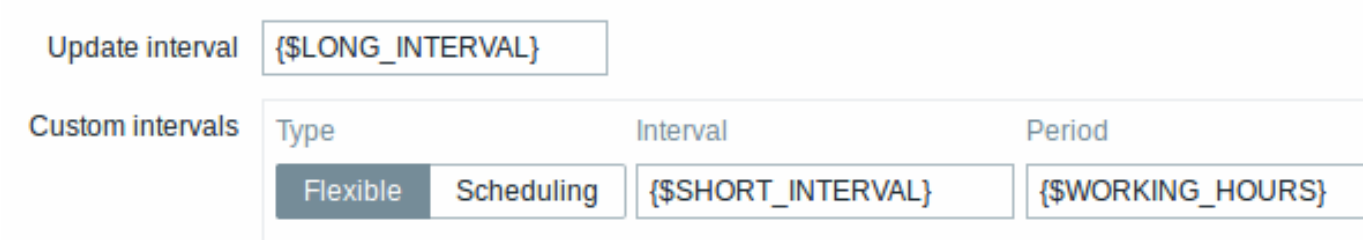
- definire la macro `{$INTERVAL}` e utilizzarla nell'intervallo di aggiornamento dell'item;
- utilizzare `{$INTERVAL}` come parametro del trigger di indisponibilità dell'agent:

`nodata(/ca_001/agent.ping,{$INTERVAL})=1`

Esempio 5

Centralizzare la configurazione dell'orario di lavoro:

- creare una macro globale `{$WORKING_HOURS}` uguale a `1-5,09:00-18:00`;
- utilizzarla nel campo *Orario di lavoro* in *Amministrazione* > *Generale* > *Interfaccia utente*;
- utilizzarla nel campo *Quando attivo* in *Utenti* > *Utenti*, scheda *Supporti* di un utente;
- utilizzarla per configurare un polling più frequente degli item durante l'orario di lavoro:

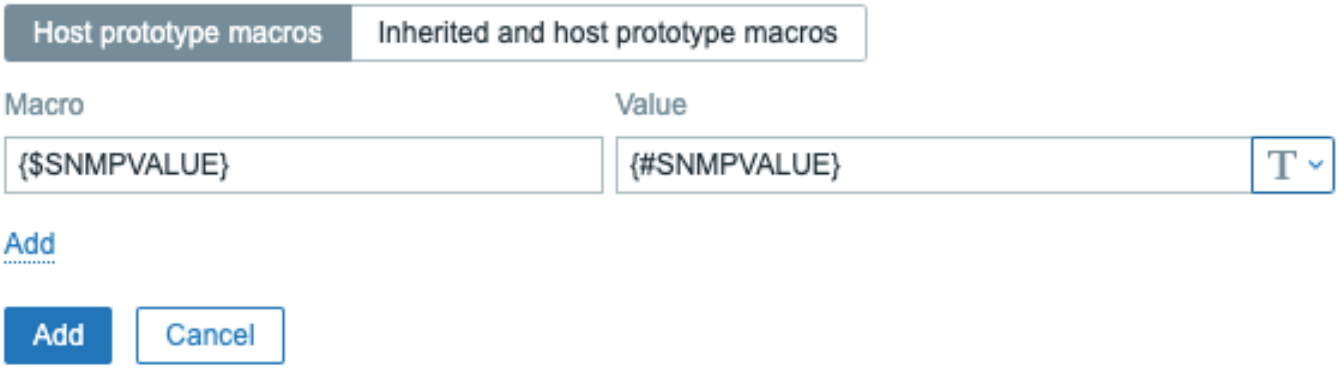


- utilizzarla nella condizione dell'azione *Periodo di tempo*;
- modificare l'orario di lavoro in *Amministrazione* > *Macro*, se necessario.

Esempio 6

Utilizzare una macro del prototipo host per configurare gli item per gli host rilevati:

1. In un prototipo host, definire la macro utente `{$SNMPVALUE}` con la macro `{#SNMPVALUE}` di low-level discovery come valore:



2. Assegnare il template *Generic SNMPv2* al prototipo host.
3. Utilizzare `{$SNMPVALUE}` nel campo *SNMP OID* degli item del template *Generic SNMPv2*.

Contesto delle macro utente

Vedere macro utente con contesto.

Macro utente con contesto

Panoramica

È possibile utilizzare un contesto facoltativo nelle macro utente, consentendo di sostituire il valore predefinito con uno specifico per il contesto.

Il contesto viene aggiunto al nome della macro; la sintassi dipende dal fatto che il contesto sia un valore di testo statico:

`{$MACRO:"testo statico"}`

oppure un'espressione regolare:

`{$MACRO:regex:"espressione regolare"}`

Si noti che una macro con contesto basato su un'espressione regolare può essere definita solo nella configurazione delle macro utente. Se il prefisso `regex:` viene utilizzato altrove come contesto di una macro utente, ad esempio in un'espressione di trigger, verrà trattato come contesto statico.

Le virgolette del contesto sono facoltative (vedere anche le note importanti).

Esempi di contesti delle macro:

Esempio	Descrizione
`{$LOW_SPACE_LIMIT}`	Macro utente senza contesto.
`{$LOW_SPACE_LIMIT:/tmp}`	Macro utente con contesto (stringa statica).
`{$LOW_SPACE_LIMIT:regex:"^/tmp$"}`	Macro utente con contesto (espressione regolare). Equivale a `{$LOW_SPACE_LIMIT:/tmp}`.
`{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}`	Macro utente con contesto (espressione regolare). Corrisponde a tutte le stringhe precedute da /var/log/.

Casi d'uso

Le macro utente con contesto possono essere definite per ottenere soglie più flessibili nelle espressioni di trigger (in base ai valori recuperati dalla discovery a basso livello). Ad esempio, è possibile definire le seguenti macro:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:"^/[a-z]+$"} = 30

Quindi una macro di discovery a basso livello può essere usata come contesto della macro in un trigger prototype per la discovery dei file system montati:

`last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}`

Dopo la discovery, verranno applicate soglie diverse per lo spazio libero nei trigger, a seconda dei punti di mount o dei tipi di file system rilevati. Gli eventi di problema verranno generati se:

- la cartella /home ha meno del 20% di spazio libero su disco
- le cartelle che corrispondono al pattern regexp (come /etc, /tmp o /var) hanno meno del 30% di spazio libero su disco
- le cartelle che non corrispondono al pattern regexp e non sono /home hanno meno del 10% di spazio libero su disco

Note importanti

- Se esiste più di una macro utente con contesto, Zabbix proverà prima a far corrispondere le macro con contesto semplice e poi le macro con contesto basato su espressioni regolari in un ordine non definito.

Warning:
Non creare macro di contesto diverse che corrispondano alla stessa stringa, per evitare un comportamento non definito.

- Se una macro con il relativo contesto non viene trovata su host, nei template collegati o a livello globale, allora viene cercata la macro senza contesto.
- Nel contesto sono supportate solo le macro di low-level discovery. Qualsiasi altra macro viene ignorata e trattata come testo normale.

Tecnicamente, il contesto della macro viene specificato usando regole simili ai parametri della chiave item, tranne per il fatto che il contesto della macro non viene analizzato come più parametri se è presente un carattere `,`:

- Il contesto della macro deve essere racchiuso tra virgolette " se il contesto contiene un carattere } o inizia con un carattere ". Le virgolette all'interno di un contesto tra virgolette devono essere precedute dal carattere di escape \.
- Il carattere \ stesso non viene sottoposto a escape, il che significa che è impossibile avere un contesto tra virgolette che termini con il carattere \ - la macro {$MACRO:"a:\b\c\"} non è valida.
- Gli spazi iniziali nel contesto vengono ignorati, quelli finali no:
  - Ad esempio {$MACRO:A} è uguale a {$MACRO: A}, ma non a {$MACRO:A }.
- Tutti gli spazi prima delle virgolette iniziali e dopo quelle finali vengono ignorati, ma tutti gli spazi all'interno delle virgolette no:
  - Le macro {$MACRO:"A"}, {$MACRO: "A"}, {$MACRO:"A" } e {$MACRO: "A" } sono uguali, ma le macro {$MACRO:"A"} e {$MACRO:" A "} non lo sono.

Le seguenti macro sono tutte equivalenti, perché hanno lo stesso contesto: {$MACRO:A}, {$MACRO: A} e {$MACRO:"A"}. Questo è in contrasto con le chiavi item, dove 'key[a]', 'key[a]' e 'key["a"]' hanno lo stesso significato semantico, ma sono diverse ai fini dell'univocità.

Macro utente segrete

Panoramica

Zabbix offre due opzioni per proteggere le informazioni sensibili nei valori delle macro utente:

- Testo segreto
- Segreto del vault

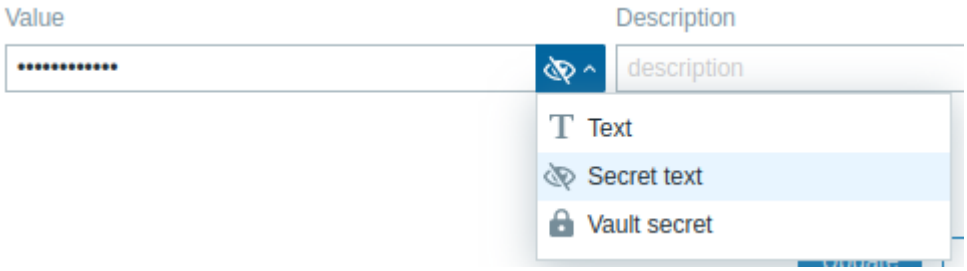
Note:
Sebbene il valore di una macro segreta sia nascosto, può essere rivelato tramite l'uso negli item. Ad esempio, in uno script esterno, un'istruzione `echo` che fa riferimento a una macro segreta può essere usata per rivelare il valore della macro nel frontend, perché Zabbix server ha accesso al valore reale della macro. Vedere le posizioni in cui i valori delle macro segrete vengono mostrati in chiaro.

Le macro segrete non possono essere utilizzate nelle espressioni dei trigger.

Testo segreto

Con le macro di testo segreto, il valore della macro viene mascherato con asterischi.

Per rendere segreto un valore di macro, fai clic sul pulsante alla fine del campo *Value* e seleziona l'opzione *Secret text*:



Una volta salvata la configurazione, non sarà più possibile visualizzare il valore.

Per modificare il valore della macro, passa il mouse sul campo *Value* e fai clic sul pulsante *Set new value* (che appare al passaggio del mouse):



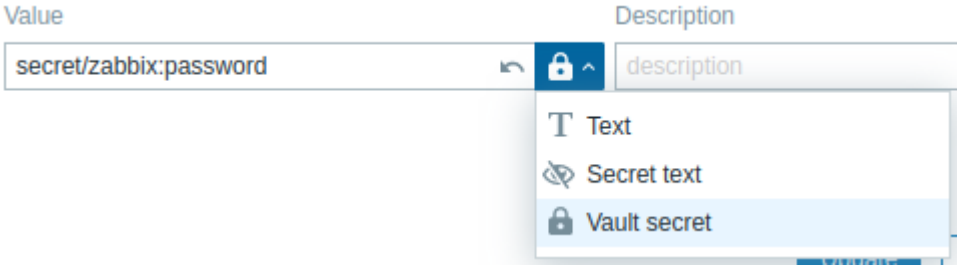
Quando fai clic sul pulsante *Set new value* (o cambi il tipo di valore della macro), il valore corrente verrà cancellato. Puoi ripristinare il valore originale facendo clic sulla freccia alla fine del campo *Value* (disponibile solo prima di salvare la nuova configurazione). Nota che il ripristino del valore originale non lo renderà visibile.

Note:
Gli URL che contengono una macro segreta non funzioneranno, poiché la macro in essi verrà risolta come "******".

Segreto Vault

Con le macro segreto Vault, il valore della macro viene memorizzato in un software esterno di gestione dei segreti (vault).

Per configurare una macro segreto Vault, fai clic sul pulsante alla fine del campo *Value* e seleziona l'opzione *Vault secret*:



Il valore della macro deve puntare a un segreto del vault. Il formato di input dipende dal provider del vault. Per esempi di configurazione specifici per provider, consulta:

- HashiCorp
- CyberArk

I valori delle macro segrete del vault vengono recuperati dal vault da Zabbix server (e da Zabbix proxy, se *Resolve secret vault macros by* è impostato su *Zabbix server and proxy*) a ogni aggiornamento dei dati di configurazione e quindi memorizzati nella cache di configurazione. Zabbix server e Zabbix proxy possono utilizzare vault diversi.

Se *Resolve secret vault macros by* è impostato su *Zabbix server*, allora i segreti del vault vengono recuperati solo dal server e Zabbix proxy riceve i valori delle macro segrete del Vault da Zabbix server a ogni sincronizzazione della configurazione e li memorizza nella propria cache di configurazione. Ciò significa che uno Zabbix proxy non può avviare la raccolta dei dati dopo un riavvio finché non riceve l'aggiornamento della configurazione da Zabbix server.

Per aggiornare manualmente i valori segreti dal vault, utilizzare l'opzione `secrets_reload` di runtime control (solo server).

La crittografia deve essere abilitata tra Zabbix server e proxy; in caso contrario, viene registrato un messaggio di avviso del server.

Warning:
Se il valore di una macro non può essere recuperato correttamente, il corrispondente item che utilizza tale valore diventerà non supportato.

Posizioni non mascherate

Questo elenco fornisce le posizioni dei parametri in cui i valori delle macro segrete non sono mascherati.

Note:
I valori delle macro segrete rimarranno mascherati nelle posizioni seguenti se referenziati indirettamente. Ad esempio, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} macro integrate usate nei tipi di media (parametri Script o Webhook) verranno risolte in chiavi item contenenti macro segrete mascherate, ad esempio `net.tcp.port[******,******]` invece di `net.tcp.port[192.0.2.1,80]`.

Contesto		Parametro
Items, item prototypes, LLD rules		
	Item	*Parametri della chiave dell'item*
	Item prototype	*Parametri della chiave dell'item prototype*
	Low-level discovery rule	*Parametri della chiave dell'item di discovery*
	SNMP agent	*Community SNMP*
		Nome del contesto (SNMPv3)
		Nome di sicurezza (SNMPv3)
		Passphrase di autenticazione (SNMPv3)
		Passphrase di privacy (SNMPv3)
	HTTP agent	*URL*
		Campi di query
		Corpo della richiesta
		Intestazioni
		Nome utente
		Password
		Password della chiave SSL
	Script	*Parametri*
		Script
	Browser	*Parametri*
		Script
	Database monitor	*Query SQL*
	TELNET agent	*Script*
		Nome utente
		Password
	SSH agent	*Script*
		Nome utente
		Password
	Simple check	*Nome utente*
		Password
	JMX agent	*Nome utente*
		Password
Item value preprocessing		
	JavaScript preprocessing step	*Script*
Web scenarios		
	Web scenario	*Valore della variabile*

Contesto		Parametro
		Valore dell'intestazione
		URL
		Valore del campo di query
		Valore del campo post
		Post grezzo
	Web scenario authentication	*Utente*
		Password
		Password della chiave SSL
Connectors		
	Connector	*URL*
		Nome utente
		Password
		Token
		Proxy HTTP
		File del certificato SSL
		File della chiave SSL
		Password della chiave SSL
Network discovery		
	SNMP	*Community SNMP*
		Nome del contesto (SNMPv3)
		Nome di sicurezza (SNMPv3)
		Passphrase di autenticazione (SNMPv3)
		Passphrase di privacy (SNMPv3)
Global scripts		
	Webhook	*Script JavaScript*
		Valore del parametro dello script JavaScript
	Telnet	*Nome utente*
		Password
	SSH	*Nome utente*
		Password
	Script	*Script*
Media types		
	Script	*Parametri dello script*
	Webhook	*Parametri*
IPMI management		
	Host	*Nome utente*
		Password

Macro di discovery a basso livello

Panoramica

Esiste un tipo di macro utilizzato nella funzione low-level discovery (LLD):

```
{#MACRO}
```

Si tratta di una macro utilizzata in una regola LLD che restituisce valori reali del nome del file system, dell'interfaccia di rete, dell'OID SNMP, ecc.

Queste macro possono essere utilizzate per creare prototipi di entità (prototipi di item, trigger, grafici, regole LLD, host e gruppi host). Durante il rilevamento di file system, interfacce di rete, macchine virtuali, ecc. reali, queste macro vengono sostituite con valori reali e costituiscono la base per la creazione di entità reali.

Alcune macro di low-level discovery sono fornite "preconfezionate" con la funzione LLD in Zabbix: {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. Tuttavia, non è obbligatorio attenersi a questi nomi quando si crea una regola di low-level discovery personalizzata. In tal caso, è possibile utilizzare qualsiasi altro nome di macro LLD e fare riferimento a quel nome.

Tipi di dati supportati

Quando si definiscono regole di discovery personalizzate, i valori delle proprietà restituiti negli oggetti JSON per le macro LLD devono essere uno dei seguenti tipi primitivi:

- string

- number
- boolean

Array, oggetti e valori null non sono supportati. Qualsiasi macro LLD che faccia riferimento a un tale valore rimarrà non espansa e apparirà letteralmente (ad esempio `'{#MY_MACRO}'`) durante il preprocessing e la creazione degli item.

Posizioni supportate

Le macro LLD possono essere utilizzate:

- nel filtro della regola di low-level discovery
- nelle regole di low-level discovery annidate, in
  - parametri di preprocessing JSONPath
  - campo JSONPath per macro LLD personalizzate
- per i prototipi di item e i prototipi di discovery in
  - nome
  - parametri della chiave
  - unità
  - intervallo di aggiornamento[1]
  - timeout[1]
  - periodo di conservazione dello storico[1]
  - periodo di conservazione dei trend[1]
  - passaggi di preprocessing del valore dell'item
  - OID SNMP
  - campo del sensore IPMI
  - espressione dell'item calcolato/aggregato, in:
    - costanti dell'espressione e parametri di funzione
    - parametri della chiave dell'item
  - condizioni di filtro dell'item aggregato (nome del gruppo host e nome del tag)
  - script SSH e script Telnet
  - query SQL di monitoraggio database
  - campo endpoint dell'item JMX
  - descrizione
  - campo URL dell'agent HTTP
  - campo dei campi di query HTTP dell'agent HTTP
  - campo del corpo della richiesta dell'agent HTTP
  - campo dei codici di stato richiesti dell'agent HTTP
  - chiave e valore del campo header dell'agent HTTP
  - campo nome utente per l'autenticazione HTTP dell'agent HTTP
  - campo password per l'autenticazione HTTP dell'agent HTTP
  - campo proxy HTTP dell'agent HTTP
  - campo file del certificato SSL HTTP dell'agent HTTP
  - campo file della chiave SSL HTTP dell'agent HTTP
  - campo password della chiave SSL HTTP dell'agent HTTP
  - tag
- per i prototipi di trigger in
  - nome
  - dati operativi
  - espressione (solo nelle costanti e nei parametri di funzione)
  - URL
  - descrizione
  - tag
- per i prototipi di grafico in
  - nome
- per i prototipi di host in
  - nome
  - nome visibile
  - campi personalizzati dell'interfaccia: IP, DNS, porta, community SNMP v1/v2, nome del contesto SNMP v3, nome di sicurezza SNMP v3, passphrase di autenticazione SNMP v3, passphrase di privacy SNMP v3
  - nome del prototipo di gruppo host
  - valore del tag host
  - valore della macro host
  - (vedere l'elenco completo)

In tutti questi punti, eccetto il filtro della regola di low-level discovery, le macro LLD possono essere utilizzate all'interno del contesto

macro utente statico.

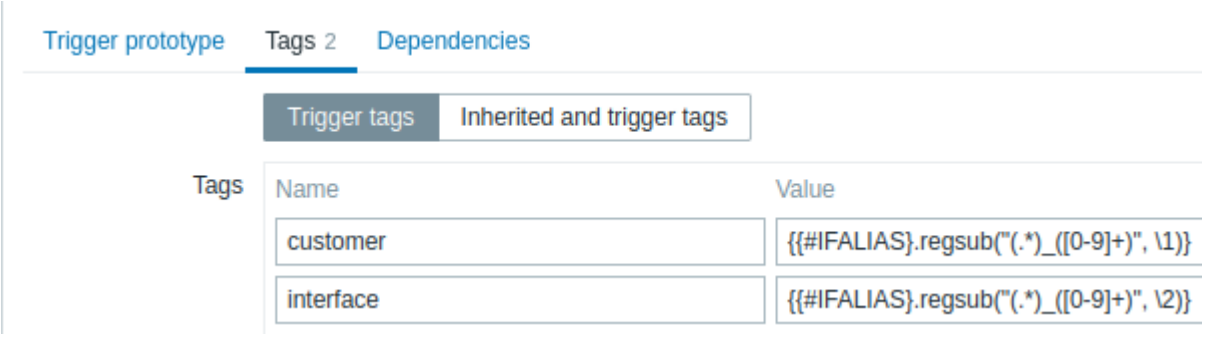
Utilizzo delle funzioni macro

Le funzioni macro sono supportate con le macro di low-level discovery (ad eccezione del filtro della regola di low-level discovery) e consentono di estrarre una determinata parte del valore della macro utilizzando un'espressione regolare.

Ad esempio, è possibile estrarre il nome del cliente e il numero dell'interfaccia dalla seguente macro LLD per utilizzarli nei tag degli eventi:

```
{#IFALIAS}=customername_1
```

A tale scopo, è possibile utilizzare la funzione macro `regsub` con la macro nel campo del valore del tag dell'evento di un prototipo di trigger:



Si noti che le virgole non sono consentite nei parametri della chiave di un item non racchiusi tra virgolette; pertanto, il parametro contenente una funzione macro deve essere racchiuso tra virgolette. Il carattere backslash (\) deve essere utilizzato per eseguire l'escape delle virgolette doppie all'interno del parametro. Esempio:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Per ulteriori informazioni sulla sintassi delle funzioni macro, vedere: Funzioni macro

Le funzioni macro sono supportate nelle macro di low-level discovery a partire da Zabbix 4.0.

Prototipi di item senza parametri della chiave

Quando i parametri della chiave dell'item non vengono utilizzati, inserire la macro LLD tra le parentesi dei parametri della chiave dell'item `[...]` in modo che venga trattata come un parametro e sostituita durante il rilevamento:

```
v_[{#MACRO}]
```

Note a piè di pagina

[1] Nei campi contrassegnati con [1] una singola macro deve occupare l'intero campo. Macro multiple in un campo o macro mescolate al testo non sono supportate.

Macro di espressione

Panoramica

Le macro di espressione consentono di eseguire calcoli nei campi.

Il loro valore viene calcolato prima risolvendo eventuali macro interne e poi valutando l'espressione risultante.

Sintassi:

```
{?EXPRESSION}
```

`EXPRESSION` usa la stessa sintassi e supporta le stesse funzioni delle espressioni trigger.

Esempio:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Per le posizioni che supportano le macro di espressione, cerca "{?EXPRESSION}" nella tabella delle macro supportate.

Note sull'uso:

- Usa le macro {FUNCTION.*} per fare riferimento ai valori di funzione delle espressioni trigger/espressioni di recupero.
- Usa le macro {HOST.HOST<1-9>} e {ITEM.KEY<1-9>} per fare riferimento a host e item.

- Nei template, usa le macro {HOST.HOST<1-9>} oppure ometti del tutto l'host per il primo host, ad esempio `{?avg(//item1,1h)}`, invece dei nomi dei template, poiché i nomi dei template non vengono sostituiti con i nomi degli host durante il collegamento dei template.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Vedi anche Esempi di espressioni trigger per un esempio di utilizzo di una macro di espressione in un nome evento.

Utenti e gruppi di utenti

Panoramica

Tutti gli utenti in Zabbix accedono all'applicazione Zabbix tramite il frontend basato sul web. A ciascun utente viene assegnato un nome di accesso univoco e una password.

Tutte le password degli utenti sono crittografate e archiviate nel database di Zabbix. Gli utenti non possono usare il proprio ID utente e la password per accedere direttamente al server UNIX, a meno che non siano stati configurati anche in modo appropriato in UNIX. La comunicazione tra il web server e il browser dell'utente può essere protetta tramite SSL.

Con un flessibile schema dei permessi utente, è possibile limitare e differenziare i diritti per:

- Accedere alle funzioni amministrative del frontend di Zabbix.
- Eseguire determinate azioni nel frontend.
- Accedere agli host monitorati nei gruppi di host.
- Usare metodi API specifici.

Configurazione di un utente

Panoramica

L'installazione iniziale di Zabbix include due utenti predefiniti:

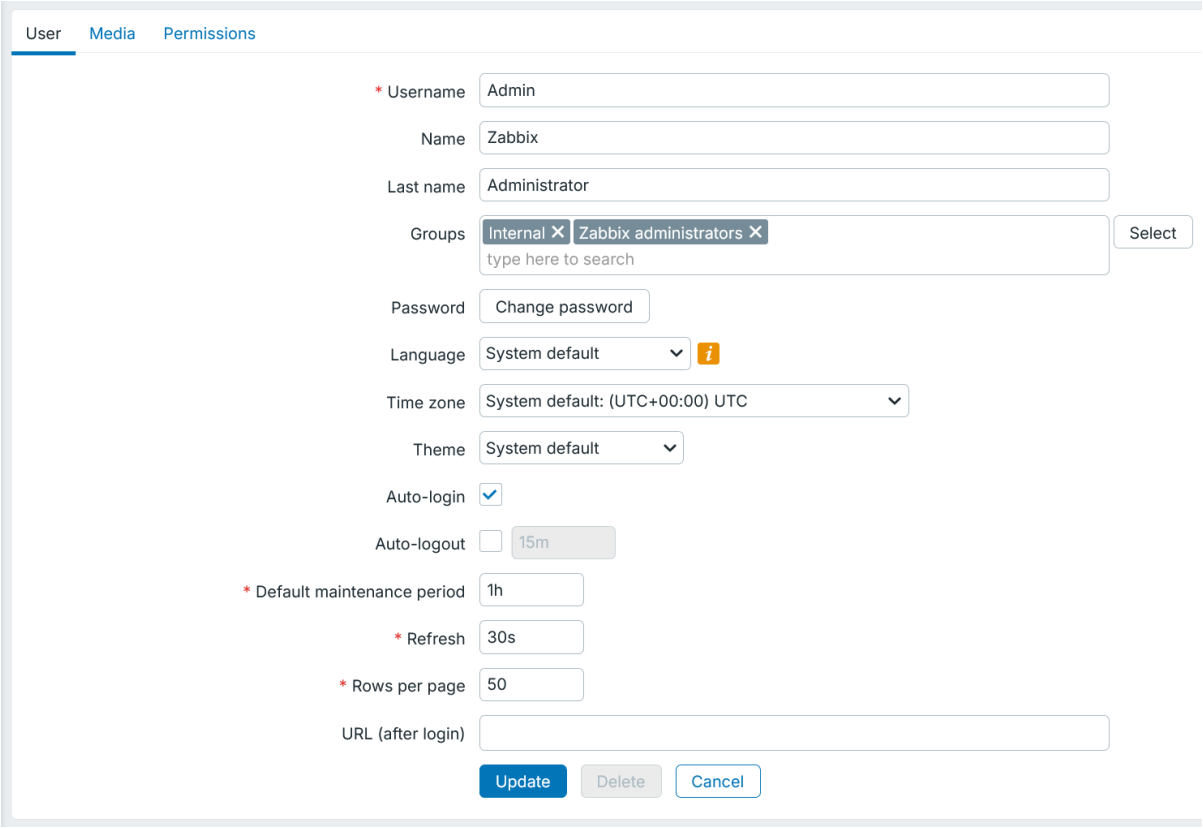
- *Admin* - un superuser di Zabbix con autorizzazioni complete.
- *guest* - uno speciale utente di Zabbix. L'utente 'guest' è disabilitato per impostazione predefinita. Per abilitare l'utente, rimuovere il gruppo predefinito *Disabled* dai gruppi di utenti. È possibile accedere con questo utente e visualizzare le pagine di monitoraggio in Zabbix. Per impostazione predefinita, 'guest' non dispone di autorizzazioni sugli oggetti Zabbix.

Per configurare un utente:

1. Andare a *Users> Users*.
2. Fare clic su *Create user* (oppure sul nome di un utente per modificare un utente esistente).
3. Modificare gli attributi dell'utente nel modulo.

Attributi generali

La scheda *Utente* contiene gli attributi generali dell'utente:



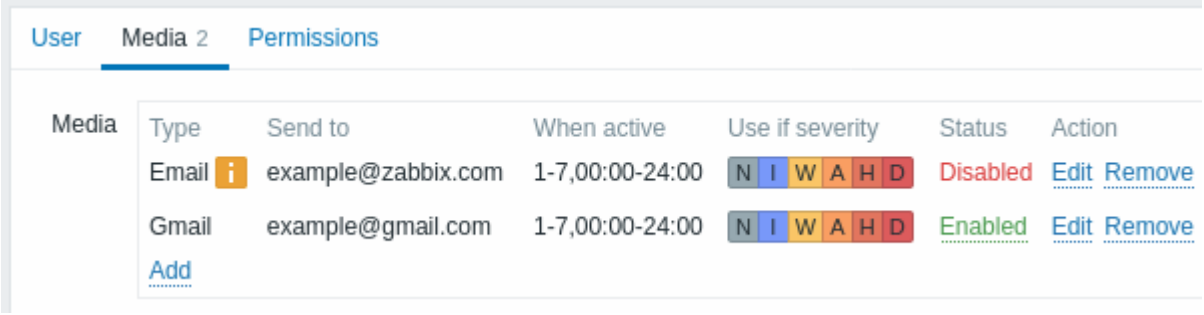
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Nome utente	Nome utente univoco, utilizzato come nome di accesso.
Nome	Nome dell'utente (facoltativo). Se non è vuoto, è visibile nelle informazioni di riconoscimento e nelle informazioni sul destinatario della notifica.
Cognome	Cognome dell'utente (facoltativo). Se non è vuoto, è visibile nelle informazioni di riconoscimento e nelle informazioni sul destinatario della notifica.
Gruppi	Selezionare i gruppi di utenti a cui appartiene l'utente. Questo campo supporta il completamento automatico: iniziando a digitare il nome di un gruppo di utenti, viene visualizzato un elenco a discesa dei gruppi corrispondenti. Scorrere verso il basso per selezionare. In alternativa, fare clic su *Seleziona* per aggiungere gruppi. Fare clic su "x" per rimuovere il gruppo selezionato. L'appartenenza ai gruppi di utenti determina a quali gruppi di host e host l'utente ha accesso.
Password	Due campi per inserire la password dell'utente oppure un pulsante *Cambia password* se l'utente esiste già. Facendo clic sul pulsante *Cambia password* vengono aperti due campi per inserire una nuova password. Per l'utente con il *ruolo di super amministratore* che cambia la propria password, facendo clic sul pulsante *Cambia password* viene aperto un campo aggiuntivo per inserire la password attuale (precedente). Dopo una modifica della password eseguita correttamente, l'utente per il quale è stata modificata la password verrà disconnesso da tutte le sessioni attive. La password può essere modificata solo per gli utenti che utilizzano l'autenticazione interna di Zabbix.
Lingua	Lingua del frontend Zabbix. Per il funzionamento delle traduzioni è necessaria l'estensione php gettext.
Fuso orario	Selezionare il fuso orario per sostituire il fuso orario globale a livello utente oppure selezionare **Predefinito di sistema** per utilizzare le impostazioni del fuso orario globale.

Parametro	Descrizione
Tema	Definisce l'aspetto del frontend: **Predefinito di sistema** - utilizza le impostazioni predefinite del sistema **Blu** - tema blu standard **Blu (classico)** - tema blu standard con caratteri legacy **Scuro** - tema scuro alternativo **Blu scuro** - tema blu scuro **Scuro (classico)** - tema scuro alternativo con caratteri legacy **Chiaro ad alto contrasto** - tema chiaro ad alto contrasto **Scuro ad alto contrasto** - tema scuro ad alto contrasto
Accesso automatico	Selezionare questa casella per fare in modo che Zabbix ricordi l'utente e lo faccia accedere automaticamente per 30 giorni. Quando si accede con *Ricordami per 30 giorni*: - La disconnessione automatica viene reimpostata (la sessione persiste per 30 giorni). - L'accesso automatico viene abilitato per una riautenticazione trasparente. Quando si accede senza *Ricordami per 30 giorni*: - L'accesso automatico viene abilitato per una riautenticazione trasparente. - La disconnessione automatica continua a essere regolata dall'impostazione standard di timeout. A questo scopo vengono utilizzati i cookie del browser.
Disconnessione automatica	Selezionando questa casella, l'utente verrà disconnesso automaticamente dopo il numero di secondi impostato (minimo 90 secondi, massimo 1 giorno). Questa impostazione viene ignorata se *Ricordami per 30 giorni* è abilitato, poiché la sessione viene estesa per l'intero periodo. Sono supportati i suffissi temporali, ad esempio 90s, 5m, 2h, 1d. Questa opzione non funzionerà: * Se l'opzione di configurazione globale "Mostra un avviso se Zabbix server è inattivo" è abilitata e il frontend Zabbix viene mantenuto aperto. * Quando le pagine del menu Monitoraggio eseguono aggiornamenti delle informazioni in background. * Se si accede con l'opzione *Ricordami per 30 giorni* selezionata.
Periodo di manutenzione predefinito	Impostare il periodo di manutenzione predefinito per la manutenzione ad hoc.
Aggiornamento	Impostare la frequenza di aggiornamento utilizzata per i grafici, i dati in testo semplice e così via. È possibile impostare 0 per disabilitarla. Sono supportati i suffissi temporali, ad esempio 90s, 5m, 1h.
Righe per pagina	È possibile determinare quante righe visualizzare per pagina negli elenchi.
URL (dopo l'accesso)	È possibile fare in modo che Zabbix trasferisca l'utente a un URL specifico dopo un accesso riuscito, ad esempio alla pagina *Problemi*.

Media dell'utente

La scheda *Media* contiene un elenco di tutti i media definiti per l'utente. I media vengono utilizzati per inviare notifiche.



Fare clic su *Aggiungi* per assegnare un media all'utente.

Se il tipo di media è stato disabilitato:

- Dopo il nome viene visualizzata un'icona informativa gialla.
- Nella colonna Stato viene visualizzato *Disabilitato*.

Note:
Le autorizzazioni degli utenti per modificare i dettagli dei propri media possono essere concesse o revocate in base al loro ruolo utente (vedere l'opzione *Creare e modificare i propri media*). Anche le autorizzazioni dei super admin per modificare i dettagli dei media di altri utenti possono essere concesse o revocate in base al loro ruolo utente (vedere l'opzione *Creare e modificare i media degli utenti*).

Per informazioni dettagliate sulla configurazione dei media dell'utente, vedere la sezione Tipi di media.

Permessi

La scheda *Permessi* contiene informazioni sui seguenti elementi:

- Ruolo utente (obbligatorio per qualsiasi utente appena creato), che può essere modificato solo da un utente di tipo *Super admin*.

Attention:
Non è possibile creare utenti senza un ruolo utente (ad eccezione dell'API User di Zabbix). Gli utenti creati in precedenza che non dispongono di un ruolo possono comunque essere modificati senza assegnare loro un ruolo. Tuttavia, una volta assegnato un ruolo, questo può essere modificato, ma non rimosso.

 Gli utenti senza ruolo possono accedere a Zabbix solo utilizzando l'autenticazione LDAP o SAML, a condizione che le loro informazioni LDAP/SAML corrispondano alle mappature dei gruppi utente configurate in Zabbix.

- Tipo di utente (*User*, *Admin*, *Super admin*) definito nella configurazione del ruolo utente.
- Gruppi di host e template a cui l'utente ha accesso.
  - Per impostazione predefinita, gli utenti di tipo *User* e *Admin* non hanno accesso ad alcun gruppo, template o host. Per concedere tale accesso, gli utenti devono essere inclusi in gruppi utente configurati con autorizzazioni per le entità pertinenti.
- Diritti di accesso alle sezioni e agli elementi del frontend di Zabbix, ai moduli e ai metodi API.
  - Gli elementi a cui è consentito l'accesso vengono visualizzati in verde, mentre quelli a cui l'accesso è negato vengono visualizzati in grigio chiaro.
  - Se l'accesso all'API è abilitato, i metodi API del ruolo utente sono elencati in *Allowed methods* o *Denied methods*, a seconda che nella configurazione del ruolo utente sia selezionato *Allow list* o *Deny list*. Se non sono specificati metodi API, viene visualizzato *None*. *None* in *Allowed methods* significa che non è consentito alcun metodo API, mentre *None* in *Denied methods* significa che sono consentiti tutti i metodi API disponibili per il tipo di utente.
- Diritti di eseguire azioni specifiche.
  - Le azioni che l'utente può eseguire vengono visualizzate in verde, mentre quelle negate vengono visualizzate in grigio chiaro.

Per ulteriori informazioni, consultare la pagina Permessi.

Permessi

Panoramica

Le autorizzazioni in Zabbix dipendono dal tipo di utente, dai ruoli utente personalizzati e dall'accesso agli host, che viene specificato in base al gruppo utenti.

Tipi di utenti

Le autorizzazioni in Zabbix dipendono principalmente dal tipo di utente:

- *Utente* - dispone di diritti di accesso limitati alle sezioni del menu (vedere sotto) e, per impostazione predefinita, non ha accesso ad alcuna risorsa. Qualsiasi autorizzazione ai gruppi di host o template deve essere assegnata esplicitamente.
- *Amministratore* - dispone di diritti di accesso incompleti alle sezioni del menu (vedere sotto). Per impostazione predefinita, l'utente non ha accesso ad alcun gruppo di host. Qualsiasi autorizzazione ai gruppi di host o template deve essere concessa esplicitamente.
- *Super amministratore* - ha accesso a tutte le sezioni del menu. L'utente dispone dell'accesso in lettura e scrittura a tutti i gruppi di host e template. Le autorizzazioni non possono essere revocate negando l'accesso a gruppi specifici.

Accesso al menu

Note:
L'accesso limitato ad alcuni elementi dell'interfaccia utente impedisce solo l'apertura della pagina; non rimuove la possibilità di accedere ai dati sottostanti in altre parti dell'interfaccia.

La tabella seguente illustra l'accesso alle sezioni del menu di Zabbix per tipo di utente:

Sezione del menu		Utente	Amministratore	Super amministratore
Dashboard		+	+	+
Monitoraggio		+	+	+
	Problemi	+	+	+
	Host	+	+	+
	Ultimi dati	+	+	+
	Mappe	+	+	+
	Discovery		+	+
APM		+	+	+
	Tracce	+	+	+
	Metriche	+	+	+
	Log	+	+	+
Servizi		+	+	+
	Servizi	+	+	+
	SLA		+	+
	Report SLA	+	+	+
Inventario		+	+	+
	Panoramica	+	+	+
	Host	+	+	+
Report		+	+	+
	Informazioni di sistema		+	+
	Report pianificati		+	+
	Report sulla disponibilità	+	+	+
	Primi 100 trigger	+	+	+
	Registro di audit			+
	Registro delle azioni			+
	Notifiche		+	+
Raccolta dati			+	+
	Gruppi di template		+	+
	Gruppi di host		+	+
	Template		+	+
	Host		+	+
	Manutenzione		+	+
	Correlazione degli eventi			+
	Discovery		+	+
Avvisi			+	+
	Azioni dei trigger		+	+
	Azioni dei servizi		+	+
	Azioni di discovery		+	+
	Azioni di autoregistrazione		+	+
	Azioni interne		+	+
	Tipi di supporto			+
	Script			+
Utenti				+
	Gruppi di utenti			+
	Ruoli utente			+
	Utenti			+
	Token API			+
	Autenticazione			+
	Dispositivi			+
Amministrazione				+
	Origine dati			+
	Generale			+
	Registro di audit			+
	Housekeeping			+
	Gruppi di proxy			+
	Proxy			+
	Macro			+
	Coda			+

Ruoli utente

I ruoli utente consentono di apportare modifiche personalizzate alle autorizzazioni definite dal tipo di utente. Non è possibile aggiungere autorizzazioni (poiché ciò supererebbe quelle del tipo di utente), ma alcune autorizzazioni possono essere revocate.

Inoltre, un ruolo utente determina l'accesso non solo alle sezioni del menu, ma anche ai servizi, ai moduli, ai metodi API e a varie azioni nel frontend.

I ruoli utente sono configurati nella sezione *Utenti* > *Ruoli utente* dagli utenti Super admin.

I ruoli utente vengono assegnati agli utenti nella modulo di configurazione dell'utente, nella scheda *Autorizzazioni*, dagli utenti Super admin.

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

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

Add | Cancel

Accesso agli host

L'accesso a qualsiasi dato di host e template in Zabbix viene concesso ai gruppi di utenti solo a livello di gruppo host/template.

Ciò significa che a un singolo utente non può essere concesso direttamente l'accesso a un host (o a un gruppo host). L'accesso a un host può essere concesso solo facendo parte di un gruppo di utenti a cui è stato concesso l'accesso al gruppo host che contiene l'host.

Allo stesso modo, a un utente può essere concesso l'accesso a un template solo facendo parte di un gruppo di utenti a cui è stato concesso l'accesso al gruppo di template che contiene il template.

Gruppi di utenti

Panoramica

I gruppi di utenti consentono di raggruppare gli utenti sia per motivi organizzativi sia per assegnare autorizzazioni ai dati. Le autorizzazioni per visualizzare e configurare i dati dei gruppi di host e dei gruppi di template vengono assegnate ai gruppi di utenti, non ai singoli utenti.

Spesso può essere utile separare le informazioni disponibili per un gruppo di utenti da quelle disponibili per un altro. È possibile farlo raggruppando gli utenti e assegnando quindi autorizzazioni diverse ai gruppi di host e ai gruppi di template.

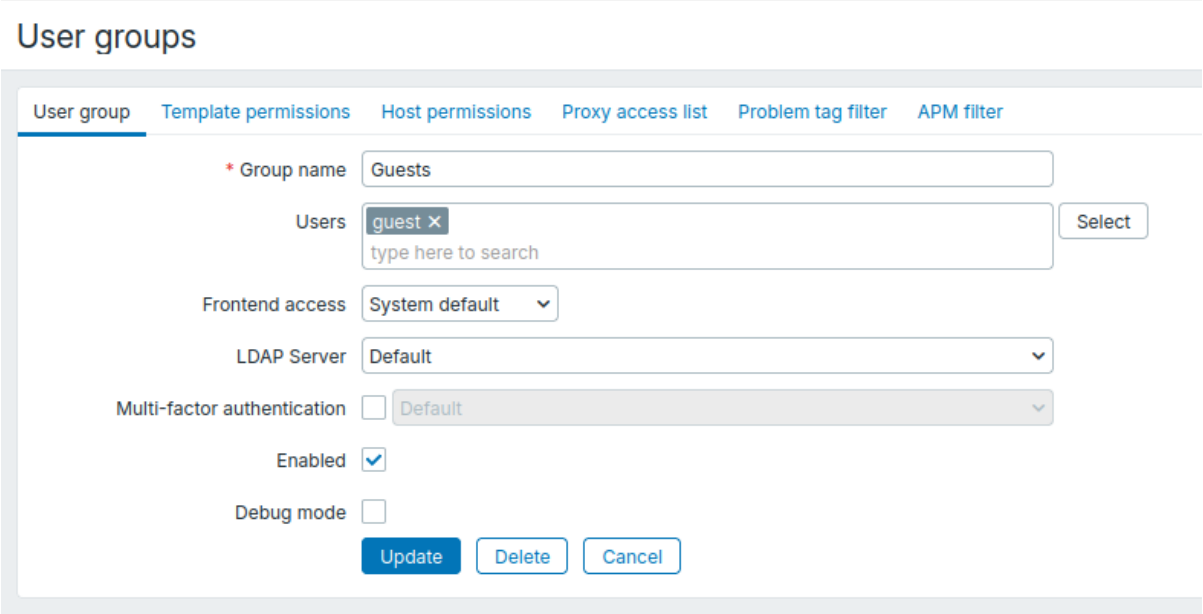
Un utente può appartenere a un numero qualsiasi di gruppi.

Configurazione

Per configurare un gruppo di utenti:

1. Vai a *Utenti* > *Gruppi di utenti*.
2. Fai clic su *Crea gruppo di utenti* (oppure sul nome del gruppo per modificare un gruppo esistente).
3. Modifica gli attributi del gruppo nel modulo.

La scheda **Gruppo di utenti** contiene gli attributi generali del gruppo:

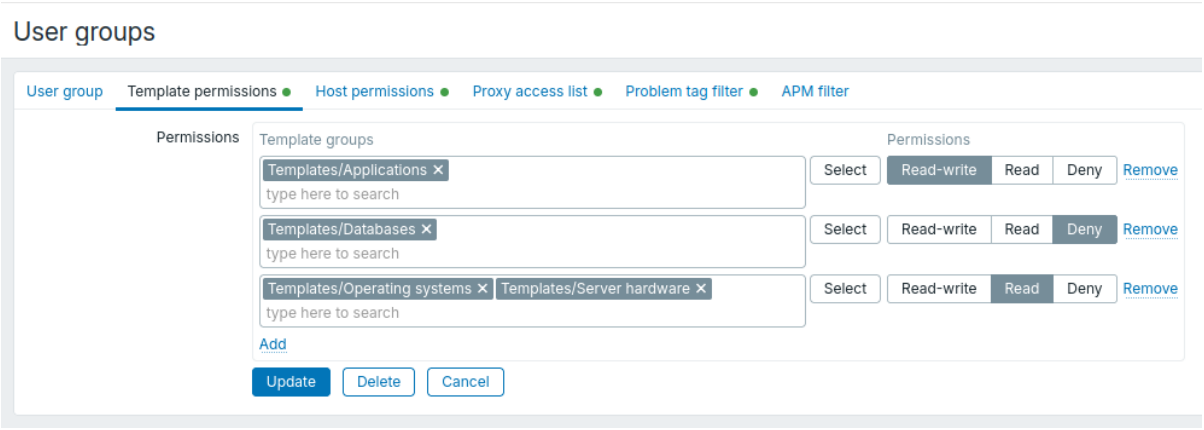


Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

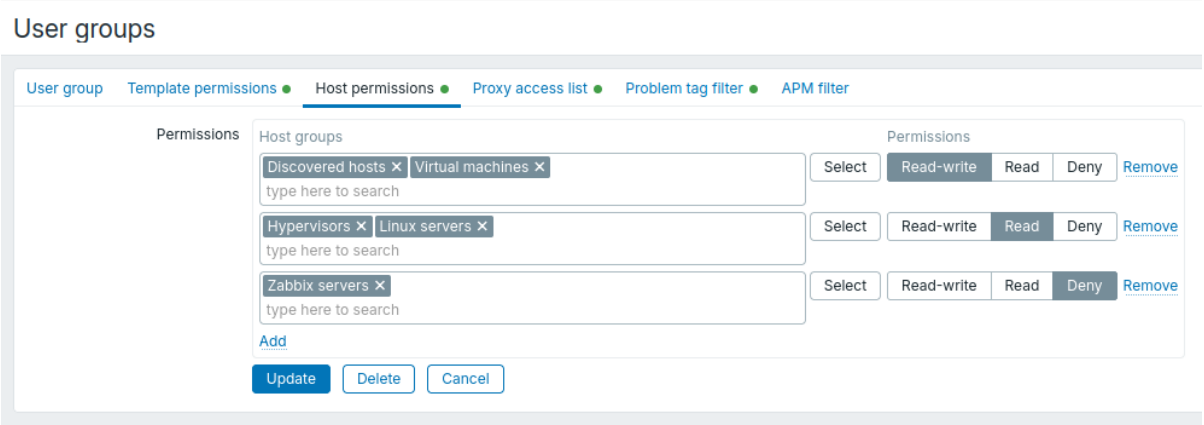
Parametro	Descrizione
Nome del gruppo	Nome univoco del gruppo.
Utenti	Per aggiungere utenti al gruppo, inizia a digitare il nome di un utente esistente. Quando viene visualizzato l'elenco a discesa con i nomi degli utenti corrispondenti, scorri verso il basso per selezionare. In alternativa, puoi fare clic sul pulsante *Seleziona* per selezionare gli utenti in una finestra popup.

Parametro	Descrizione
Accesso al frontend	Modalità di autenticazione degli utenti del gruppo. **Predefinito di sistema** - utilizza il metodo di autenticazione predefinito (impostato globalmente) **Interno** - utilizza l'autenticazione interna di Zabbix (anche se l'autenticazione LDAP è utilizzata globalmente). Ignorato se l'autenticazione HTTP è il metodo predefinito globale. **LDAP** - utilizza l'autenticazione LDAP (anche se l'autenticazione interna è utilizzata globalmente). Ignorato se l'autenticazione HTTP è il metodo predefinito globale. **Disabilitato** - l'accesso al frontend di Zabbix è vietato per questo gruppo
server LDAP	Seleziona quale server LDAP utilizzare per autenticare l'utente. Questo campo è abilitato solo se *Accesso al frontend* è impostato su LDAP o Predefinito di sistema.
Autenticazione a più fattori	Seleziona quale metodo di autenticazione a più fattori utilizzare per autenticare l'utente: **Predefinito** - utilizza il metodo impostato come predefinito nella configurazione MFA; questa opzione è selezionata per impostazione predefinita per i nuovi gruppi di utenti se MFA è abilitata; **<Nome del metodo>** - utilizza il metodo selezionato (ad esempio, "Zabbix TOTP"); **Disabilitato** - MFA è disabilitata per questo gruppo; questa opzione è selezionata per impostazione predefinita per i nuovi gruppi di utenti se MFA è disabilitata. Se un utente appartiene a più gruppi di utenti con MFA abilitata (o almeno un gruppo ha MFA abilitata), si applicano le seguenti regole di autenticazione: se un gruppo utilizza il metodo MFA "Predefinito", questo metodo autenticherà l'utente; in caso contrario, per l'autenticazione verrà utilizzato il primo metodo (in ordine alfabetico).
Abilitato	Stato del gruppo di utenti e dei membri del gruppo. *Selezionato* - il gruppo di utenti e gli utenti sono abilitati *Non selezionato* - il gruppo di utenti e gli utenti sono disabilitati
Modalità di debug	Seleziona questa casella per attivare la modalità di debug per gli utenti.

La scheda **Autorizzazioni template** consente di specificare l'accesso del gruppo di utenti ai gruppi di template e, tramite questi, ai dati dei template:



La scheda **Autorizzazioni host** consente di specificare l'accesso del gruppo di utenti ai gruppi di host e, tramite questi, ai dati degli host:



Fai clic su Add per scegliere i gruppi di template/host (sia un gruppo principale sia un gruppo nidificato) e assegnare loro le autorizzazioni. Inizia a digitare il nome del gruppo (verrà visualizzato un elenco a discesa dei gruppi corrispondenti) oppure fai clic su *Seleziona* per aprire una finestra popup contenente tutti i gruppi.

Utilizza quindi i pulsanti delle opzioni per assegnare le autorizzazioni ai gruppi scelti. Le autorizzazioni disponibili sono:

- **Lettura-scrittura** - accesso in lettura e scrittura a un gruppo
- **Lettura** - accesso in sola lettura a un gruppo
- **Nega** - accesso negato a un gruppo

Se lo stesso gruppo di template/host viene aggiunto in più righe con autorizzazioni diverse, verrà applicata l'autorizzazione più restrittiva.

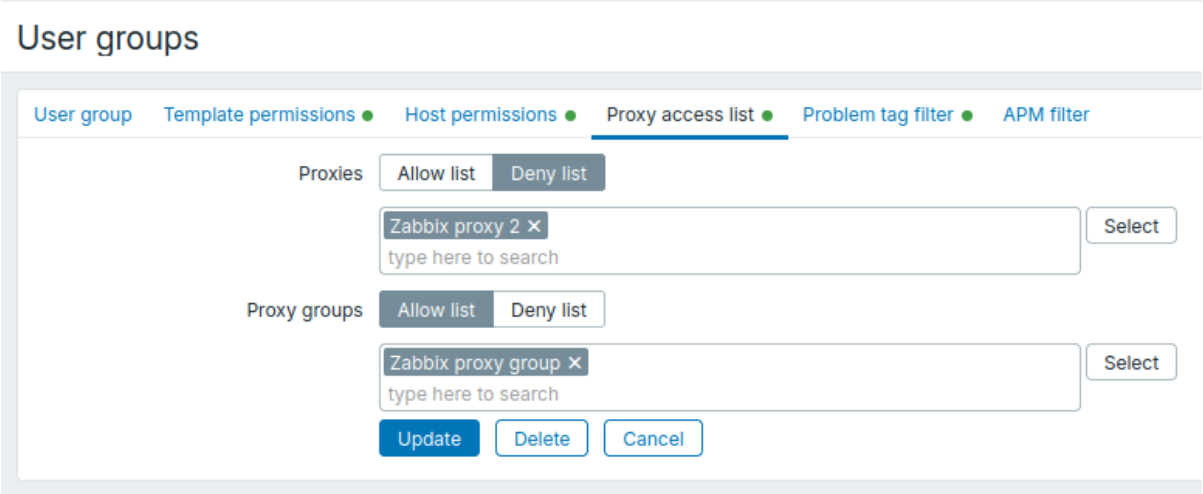
Un utente *Super amministratore* può imporre ai gruppi nidificati di avere lo stesso livello di autorizzazioni del gruppo principale; questa operazione può essere eseguita nel modulo di configurazione del gruppo host/template.

Le schede **Autorizzazioni template** e **Autorizzazioni host** supportano lo stesso insieme di parametri.

Le autorizzazioni attuali per i gruppi vengono visualizzate nel blocco *Autorizzazioni* e possono essere modificate o rimosse.

Note:
Se un gruppo di utenti dispone di autorizzazioni **Lettura-scrittura** per un host e di autorizzazioni **Nega** o di nessuna autorizzazione per un template collegato a questo host, gli utenti di tale gruppo non potranno modificare gli item basati su template nell'host e il nome del template verrà visualizzato come *Template inaccessibile*.

La scheda **Elenco di accesso ai proxy** consente di specificare l'accesso del gruppo di utenti ai proxy, singolarmente o tramite gruppi di proxy:



Utilizza il campo *Gruppi di proxy* per applicare l'autorizzazione ai gruppi selezionati. Un proxy appartenente a un gruppo può essere autorizzato o negato solo tramite quel gruppo. Utilizza il campo *Proxy* per aggiungere singoli proxy che non appartengono ad alcun gruppo di proxy.

Proxy e *Gruppi di proxy* supportano lo stesso insieme di parametri.

Fai clic sul pulsante *Seleziona* accanto al campo *Proxy* o *Gruppi di proxy* per selezionare rispettivamente proxy o gruppi di proxy. In alternativa, inizia a digitare il nome di un proxy o di un gruppo di proxy nel campo corrispondente. Viene visualizzato un elenco a discesa dei proxy o dei gruppi di proxy corrispondenti.

Seleziona *Elenco consentiti* per consentire al gruppo di utenti di accedere solo ai proxy e ai gruppi di proxy selezionati.
 Se la selezione *Proxy* è vuota per l'*Elenco consentiti*, il gruppo di utenti non avrà accesso ad alcun proxy che non appartiene a un gruppo di proxy. Se la selezione *Gruppi di proxy* è vuota per l'*Elenco consentiti*, il gruppo di utenti non avrà accesso ad alcun gruppo di proxy.

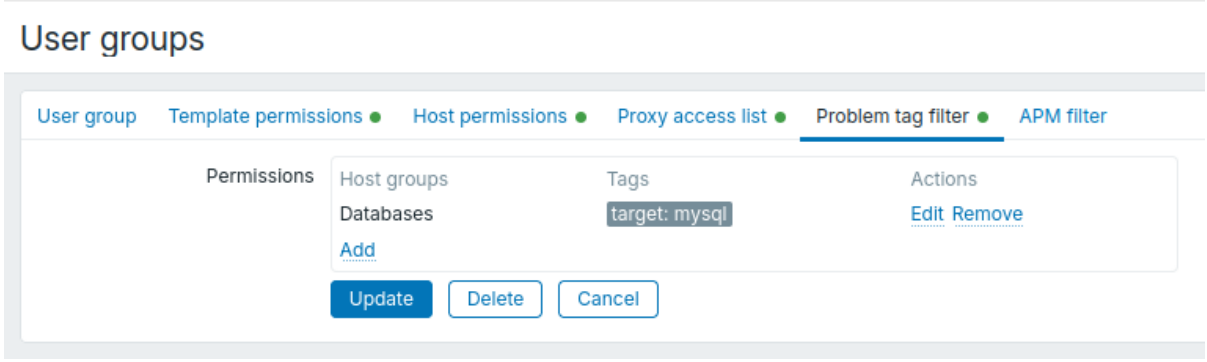
Seleziona *Elenco negati* per negare al gruppo di utenti l'accesso ai proxy e ai gruppi di proxy selezionati. L'accesso a tutti gli altri proxy e gruppi di proxy è consentito.
 Se la selezione *Proxy* è vuota per l'*Elenco negati*, il gruppo di utenti avrà accesso completo a tutti i proxy che non appartengono a un gruppo di proxy. Se la selezione *Gruppi di proxy* è vuota per l'*Elenco negati*, il gruppo di utenti avrà accesso completo a tutti i gruppi di proxy.

Regole di precedenza delle autorizzazioni per gli utenti appartenenti a più gruppi di utenti:

- L'*Elenco negati* prevale sull'*Elenco consentiti*. Se un gruppo nega l'accesso a un proxy/gruppo di proxy, l'utente non potrà accedervi.

- Le autorizzazioni dell'*Elenco consentiti* sono cumulative. Se più gruppi consentono l'accesso, l'utente potrà accedere a tutti i proxy consentiti da tali gruppi.

La scheda **Filtro dei tag dei problemi** consente di impostare autorizzazioni basate sui tag per i gruppi di utenti, in modo da visualizzare i problemi filtrati per nome e valore del tag:

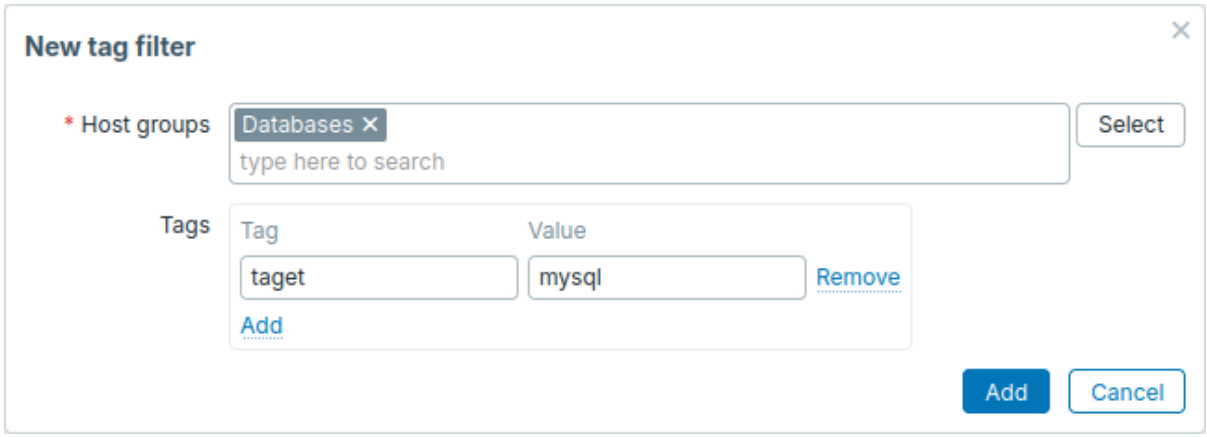


Fai clic su Add per scegliere i gruppi di host. Per selezionare un gruppo di host a cui applicare un filtro dei tag, fai clic su *Seleziona* per ottenere l'elenco completo dei gruppi di host esistenti oppure inizia a digitare il nome di un gruppo di host per visualizzare un elenco a discesa dei gruppi corrispondenti. Verranno visualizzati solo i gruppi di host, perché il filtro dei tag dei problemi non può essere applicato ai gruppi di template.

È possibile aggiungere nomi di tag senza valori, ma non valori senza nomi. Nel blocco *Autorizzazioni* vengono visualizzati solo i primi tre tag (con i relativi valori, se presenti); se sono presenti altri tag, è possibile visualizzarli facendo clic sull'icona ••• o passandovi sopra con il mouse.

Il filtro dei tag consente di separare l'accesso al gruppo di host dalla possibilità di visualizzare i problemi.

Ad esempio, se un amministratore di database deve visualizzare solo i problemi del database "MySQL", è necessario creare prima un gruppo di utenti per gli amministratori di database, quindi specificare il nome del tag "target" e il valore "mysql".



Se viene specificato il nome del tag "target" e il campo del valore viene lasciato vuoto, il gruppo di utenti visualizzerà tutti i problemi con il nome del tag "target" per il gruppo di host selezionato.

Assicurati che il nome e il valore del tag siano specificati correttamente; in caso contrario, il gruppo di utenti non visualizzerà alcun problema.

Esaminiamo un esempio in cui un utente è membro di diversi gruppi di utenti selezionati. In questo caso, il filtraggio utilizzerà una condizione OR per i tag.

Gruppo di utenti A			**Gruppo di utenti B**			**Risultato visibile per un utente (membro) di entrambi i gruppi**
Filtro dei tag						
Gruppo di host	*Nome del tag*	*Valore del tag*	*Gruppo di host*	*Nome del tag*	*Valore del tag*	
Database	target	mysql	Database	target	oracle	problemi target:mysql o target:oracle visibili

Non configurato nel **Filtro dei tag dei problemi**	Database	target	oracle	problemi target:oracle visibili

Accesso da diversi gruppi di utenti

Un utente può appartenere a un numero qualsiasi di gruppi di utenti. Questi gruppi possono avere permessi di accesso diversi a host o template.

Pertanto, è importante sapere a quali entità un utente senza privilegi potrà accedere di conseguenza. Nell'esempio seguente, considera come l'accesso all'host **X** (nel Gruppo host 1) sarà influenzato in varie situazioni per un utente che appartiene ai gruppi di utenti A e B.

- Se il Gruppo A ha solo accesso in *Lettura* al Gruppo host 1, ma il Gruppo B ha accesso in *Lettura-scrittura* al Gruppo host 1, l'utente otterrà accesso in **Lettura-scrittura** a 'X'.

> **Attention:**
> I permessi di "Lettura-scrittura" hanno la precedenza sui permessi di "Lettura".

- Nello stesso scenario di cui sopra, se 'X' si trova contemporaneamente anche nel Gruppo host 2 che è **negato** al Gruppo A o B, l'accesso a 'X' sarà **non disponibile**, nonostante l'accesso in *Lettura-scrittura* al Gruppo host 1.
- Se il Gruppo A non ha permessi definiti e il Gruppo B ha accesso in *Lettura-scrittura* al Gruppo host 1, l'utente otterrà accesso in **Lettura-scrittura** a 'X'.
- Se il Gruppo A ha accesso di *Negazione* al Gruppo host 1 e il Gruppo B ha accesso in *Lettura-scrittura* al Gruppo host 1, all'utente sarà **negato** l'accesso a 'X'.

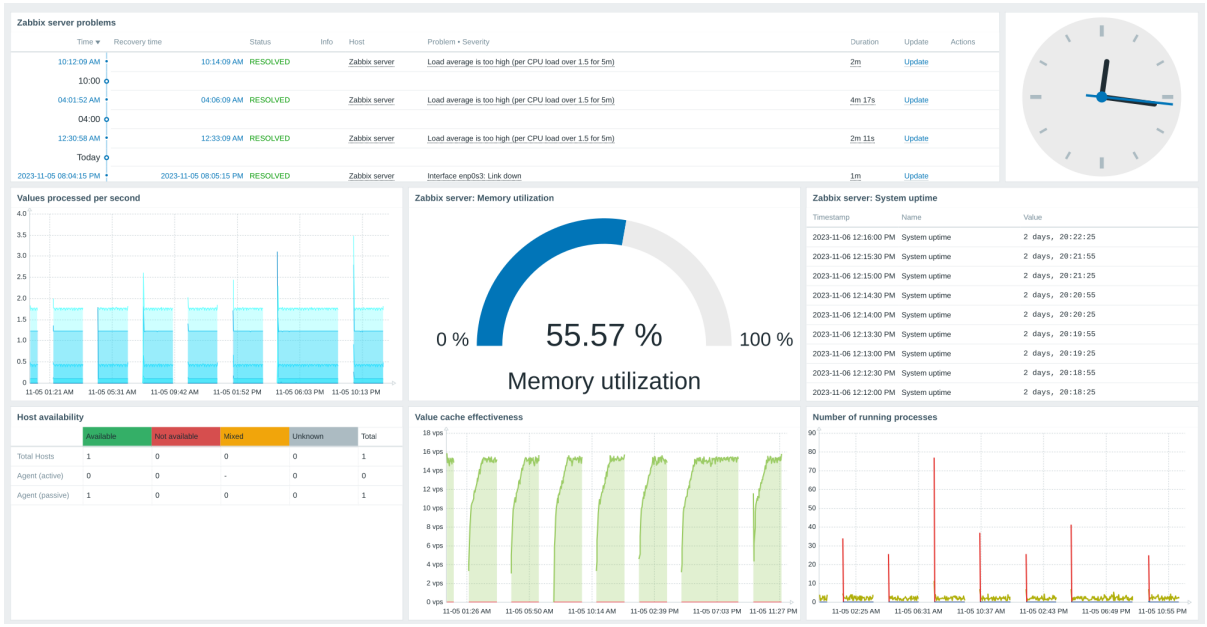
Altri dettagli

- Un utente di livello Admin con accesso in *lettura-scrittura* a un host non potrà collegare/scollegare template, se non ha accesso al gruppo di template a cui essi appartengono. Con accesso in *sola lettura* al gruppo di template potrà collegare/scollegare template all'host, tuttavia non vedrà alcun template nell'elenco dei template e non potrà operare con i template in altri punti.
- Un utente di livello Admin con accesso in *sola lettura* a un host non vedrà l'host nell'elenco degli host nella sezione di configurazione; tuttavia, i trigger dell'host saranno accessibili nella configurazione dei servizi IT.
- Qualsiasi utente che non sia Super Admin (incluso 'guest') può vedere le mappe di rete purché la mappa sia vuota o contenga solo immagini. Quando alla mappa vengono aggiunti host, gruppi di host o trigger, i permessi vengono rispettati.
- Zabbix server non invierà notifiche agli utenti definiti come destinatari delle operazioni di un'azione se l'accesso all'host interessato è esplicitamente "negato".

Report pianificati

Panoramica

Con la funzionalità *Report pianificati*, è possibile configurare l'invio della versione PDF di una determinata dashboard ai destinatari specificati a intervalli ricorrenti.



Prerequisiti:

- Il servizio web di Zabbix deve essere installato e configurato correttamente per abilitare la generazione di report pianificati. Per le istruzioni, vedere Configurazione dei report pianificati.
- Un utente deve avere un ruolo utente di tipo *Admin* o *Super admin* con le seguenti autorizzazioni:
  - *Report pianificati* nel blocco *Accesso agli elementi dell'interfaccia utente* (per visualizzare le impostazioni dei report)
  - *Gestione dei report pianificati* nel blocco *Accesso alle azioni* (per creare/modificare i report)

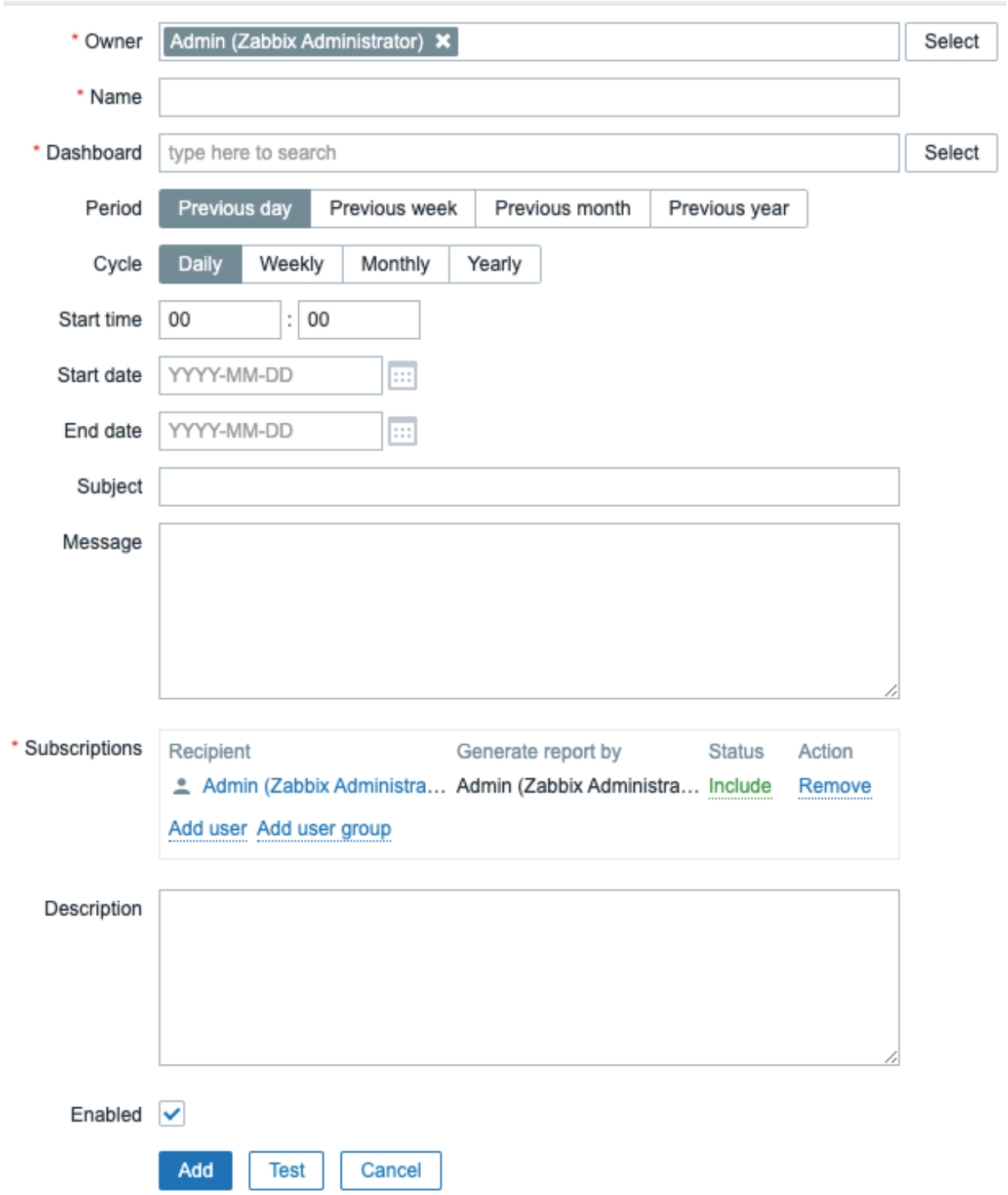
Per creare un report pianificato nel frontend di Zabbix, procedere come segue:

1. Accedere a *Report* > *Report pianificati*.
2. Fare clic su *Crea report* nell'angolo superiore destro della schermata.
3. Inserire i parametri del report nel modulo.

In alternativa, è possibile creare un report aprendo un report esistente, facendo clic sul pulsante *Clona* e salvandolo con un nome diverso.

Configurazione

La scheda *Report pianificati* contiene gli attributi generali del report.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Proprietario	Utente che crea un report. Gli utenti con livello *Super admin* possono modificare il proprietario. Per gli utenti con livello *Admin*, questo campo è di sola lettura.
Nome	Nome del report; deve essere univoco.
Dashboard	Dashboard sulla quale si basa il report; è possibile selezionare una sola dashboard alla volta. Per selezionare una dashboard, iniziare a digitare il nome: verrà visualizzato un elenco delle dashboard corrispondenti; scorrere verso il basso per selezionarla. In alternativa, fare clic su *Seleziona* accanto al campo e selezionare una dashboard dall'elenco visualizzato.
Periodo	Periodo di tempo per il quale verrà preparato il report. Selezionare il giorno, la settimana, il mese o l'anno precedente.
Ciclo	Frequenza di generazione del report. I report possono essere inviati giornalmente, settimanalmente, mensilmente o annualmente. La modalità "Settimanale" consente di selezionare i giorni della settimana in cui il report verrà inviato.
Ora di inizio	Ora del giorno nel formato hh:mm in cui verrà preparato il report. Si noti che verrà utilizzato il fuso orario del server Zabbix.

Parametro	Descrizione
Ripeti il	Giorni della settimana in cui verrà inviato il report. Questo campo è disponibile solo se *Ciclo* è impostato su ”Settimanale”.
Data di inizio	Data in cui deve iniziare la generazione regolare dei report.
Data di fine	Data in cui deve terminare la generazione regolare dei report.
Oggetto	Oggetto dell’email del report. Macro supportate: {TIME}, {TIMESTAMP}.
Messaggio	Corpo dell’email del report. Macro supportate: {TIME}, {TIMESTAMP}.
Sottoscrizioni	Elenco dei destinatari del report. Per impostazione predefinita, include solo il proprietario del report. Qualsiasi utente Zabbix con un media email configurato può essere specificato come destinatario del report. Fare clic su *Aggiungi utente* o *Aggiungi gruppo di utenti* per aggiungere altri destinatari. Fare clic sul nome utente per modificare le impostazioni: *Genera report in base a* - indica se i dati del report devono essere generati in base ai permessi della dashboard dell’utente corrente o del destinatario. *Stato* - selezionare ”Includi” per inviare il report all’utente oppure ”Escludi” per impedire l’invio del report a questo utente. Almeno un utente deve avere lo stato ”Includi”. Lo stato ”Escludi” può essere utilizzato per escludere utenti specifici da un gruppo di utenti incluso. Si noti che gli utenti con permessi insufficienti (ovvero gli utenti con un ruolo basato sul tipo di utente *Admin* che non sono membri dello stesso gruppo di utenti del destinatario o del proprietario del report) visualizzeranno ”Utente non accessibile” o ”Gruppo di utenti non accessibile” invece dei nomi effettivi nei campi *Destinatario* e *Genera report in base a*; i campi *Stato* e *Azione* saranno visualizzati in sola lettura.
Abilitato	Stato del report. Deselezionando questa casella di controllo, il report verrà disabilitato.
Descrizione	Descrizione facoltativa del report. Questa descrizione è destinata all’uso interno e non verrà inviata ai destinatari del report.

Pulsanti del modulo

I pulsanti nella parte inferiore del modulo consentono di eseguire diverse operazioni.

Add	Aggiunge un report. Questo pulsante è disponibile solo per i nuovi report.
Update	Aggiorna le proprietà di un report.
Clone	Crea un altro report basato sulle proprietà del report corrente.
Test	Verifica che la configurazione del report sia corretta inviando un report all’utente corrente.
Delete	Elimina il report.
Cancel	Annulla la modifica delle proprietà del report.

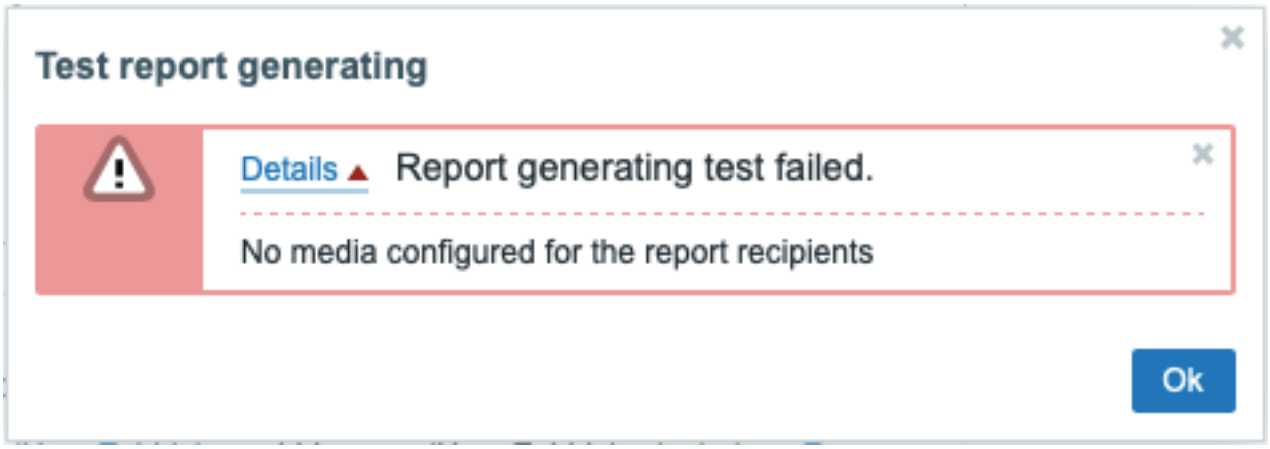
Test

Per testare un report, fare clic sul pulsante *Test* nella parte inferiore del modulo di configurazione del report.

Note:
Il pulsante *Test* non è disponibile se il modulo di configurazione del report è stato aperto dal menu delle azioni del dashboard.

Se la configurazione è corretta, il report di test viene inviato immediatamente all’utente corrente. Per i report di test, le impostazioni degli iscritti e dell’utente *Genera report per* vengono ignorate.

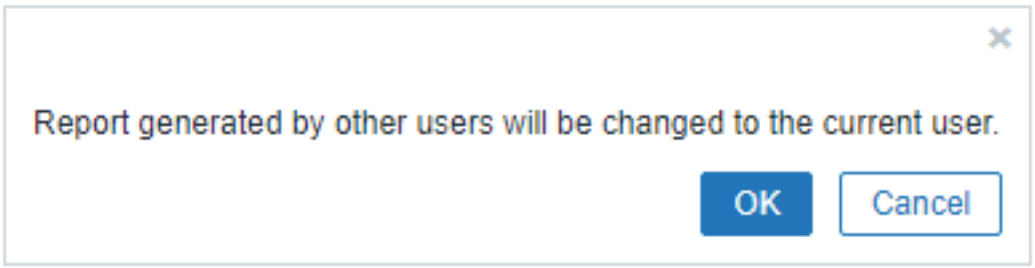
Se la configurazione non è corretta, viene visualizzato un messaggio di errore che descrive la possibile causa.



Aggiornamento di un report

Per aggiornare un report esistente, fare clic sul nome del report, apportare le modifiche di configurazione necessarie e quindi fare clic sul pulsante *Update*.

Se un report esistente viene aggiornato da un altro utente e questo utente modifica la Dashboard, facendo clic sul pulsante *Update* verrà visualizzato un messaggio di avviso: "Il report generato da altri utenti verrà modificato in base all'utente corrente".



Facendo clic su *OK* in questo passaggio si verificheranno le seguenti modifiche:

- Le impostazioni di *Generate report by* verranno aggiornate per mostrare l'utente che ha modificato il report per ultimo (a meno che *Generate report by* non sia impostato sul destinatario).
- Gli utenti visualizzati come "Inaccessible user" o "Inaccessible user group" verranno eliminati dall'elenco degli iscritti al report.

Facendo clic su *Cancel* verrà chiuso il modulo di configurazione e l'aggiornamento del report verrà annullato.

Clonazione di un report

Per clonare rapidamente un report esistente, fare clic sul pulsante *Clone* nella parte inferiore di un modulo di configurazione di un report esistente. Quando si clona un report creato da un altro utente, l'utente corrente diventa il proprietario del nuovo report.

Le impostazioni del report verranno copiate nel nuovo modulo di configurazione del report nel rispetto dei permessi utente:

- Se l'utente che clona un report non dispone dei permessi per una dashboard, il campo *Dashboard* verrà svuotato.
- Se l'utente che clona un report non dispone dei permessi per alcuni utenti o gruppi di utenti nell'elenco *Subscriptions*, i destinatari non accessibili non verranno clonati.
- Le impostazioni di *Generate report by* verranno aggiornate per visualizzare l'utente corrente (a meno che *Generate report by* non sia impostato sul destinatario).

Modificare le impostazioni richieste e il nome del report, quindi fare clic su *Add*.

Esportazione dei dati

Panoramica

Zabbix supporta l'esportazione dei dati in tempo reale in due modi:

- esportazione in file
- streaming verso sistemi esterni

È possibile esportare le seguenti entità:

- eventi dei trigger

- valori degli item
- trend (solo esportazione in file)

Esportazione su file

Panoramica

È possibile configurare l'esportazione in tempo reale degli eventi di trigger, dei valori degli item e dei trend in un formato JSON delimitato da newline.

L'esportazione viene effettuata in file, in cui ogni riga del file di esportazione è un oggetto JSON. Le mappature dei valori non vengono applicate.

In caso di errori (i dati non possono essere scritti nel file di esportazione oppure il file di esportazione non può essere rinominato o non può esserne creato uno nuovo dopo la rinomina), il dato viene scartato e non viene mai scritto nel file di esportazione. Viene scritto solo nel database di Zabbix. La scrittura dei dati nel file di esportazione riprende quando il problema di scrittura viene risolto.

Il file di esportazione viene creato con permessi di lettura e scrittura solo per il proprietario del file. Inoltre, il file è leggibile dal gruppo del proprietario. Tutti gli altri permessi sono negati.

Si noti che host/item può non avere metadati (gruppi host, nome host, nome item) se l'host/item è stato rimosso dopo la ricezione dei dati, ma prima che il server esportasse i dati.

Configurazione

L'esportazione in tempo reale degli eventi dei trigger, dei valori degli item e dei trend viene configurata specificando una directory per i file di esportazione. Consultare il parametro ExportDir nella configurazione del server.

Sono disponibili altri due parametri:

- `ExportFileSize` può essere utilizzato per impostare la dimensione massima consentita di un singolo file di esportazione. Quando un processo deve scrivere in un file, ne verifica prima la dimensione. Se supera il limite configurato, il file viene rinominato aggiungendo `.old` al nome e viene creato un nuovo file con il nome originale.

Attention:
Verrà creato un file per ogni processo che scriverà i dati (ovvero circa 4-30 file). Poiché la dimensione predefinita per ogni file di esportazione è 1G, la conservazione di file di esportazione di grandi dimensioni può esaurire rapidamente lo spazio su disco.

- `ExportType` consente di specificare quali tipi di entità (eventi, cronologia, trend) verranno esportati.

Protocollo di esportazione JSON delimitato da newline

Questa sezione presenta i dettagli del protocollo di esportazione in un formato JSON delimitato da newline, utilizzato in:

- esportazione dei dati su file
- streaming verso sistemi esterni

Tutti i file hanno estensione `.ndjson`. Ogni riga del file di esportazione è un oggetto JSON.

Esportazione dell'evento trigger

Le seguenti informazioni vengono esportate per un evento di problema:

Field	Type	Description
clock	number	Numero di secondi trascorsi dall'Epoch al momento in cui il problema è stato rilevato (parte intera).
ns	number	Numero di nanosecondi da aggiungere a `clock` per ottenere un orario preciso di rilevamento del problema.
value	number	1 (sempre).
eventid	number	ID dell'evento di problema.
name	string	Nome dell'evento di problema.
severity	number	Gravità dell'evento di problema (0 - Non classificato, 1 - Informazione, 2 - Avviso, 3 - Medio, 4 - Alto, 5 - Disastro).
hosts	array	Elenco degli host coinvolti nell'espressione del trigger; deve esserci almeno un elemento nell'array.
-	object	
host	string	Nome dell'host.

Field		Type	Description
	name	string	Nome visibile dell'host.
groups		array	Elenco dei gruppi di host di tutti gli host coinvolti nell'espressione del trigger; deve esserci almeno un elemento nell'array.
-		string	Nome del gruppo di host.
tags		array	Elenco dei tag del problema (può essere vuoto).
-		object	
	tag	string	Nome del tag.
	value	string	Valore del tag (può essere vuoto).

Le seguenti informazioni vengono esportate per un evento di ripristino:

Field	Type	Description
clock	number	Numero di secondi trascorsi dall'Epoch al momento in cui il problema è stato risolto (parte intera).
ns	number	Numero di nanosecondi da aggiungere a `clock` per ottenere un orario preciso di risoluzione del problema.
value	number	0 (sempre).
eventid	number	ID dell'evento di ripristino.
p_eventid	number	ID dell'evento di problema.

Esempi di esportazione degli eventi trigger

Problema:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

Ripristino:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problema (generazione di eventi problema multipli):

```
{"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
```

Ripristino:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}
```

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Esportazione del valore dell'item

Le seguenti informazioni vengono esportate per un valore di item raccolto:

Field		Type	Description
host		object	Nome host dell'host dell'item.
	host	string	Nome host.
	name	string	Nome host visibile.
groups		array	Elenco dei gruppi host dell'host dell'item; nell'array deve essere presente almeno un elemento.
	-	string	Nome del gruppo host.
item_tags		array	Elenco dei tag dell'item (può essere vuoto).
	-	object	
	tag	string	Nome del tag.
	value	string	Valore del tag (può essere vuoto).
itemid		number	ID dell'item.
name		string	Nome visibile dell'item.
clock		number	Numero di secondi trascorsi dall'Epoch al momento in cui il valore è stato raccolto (parte intera).

Field	Type	Description
ns	number	Numero di nanosecondi da aggiungere a `clock` per ottenere un tempo di raccolta del valore preciso.
timestamp (*Log* only)	number	0 se non disponibile.
source (*Log* only)	string	Stringa vuota se non disponibile.
severity (*Log* only)	number	0 se non disponibile.
eventid (*Log* only)	number	0 se non disponibile.
value	number (for numeric items) or string (for text items)	Valore dell'item raccolto.
type	number	Tipo di valore raccolto: 0 - numerico float, 1 - carattere, 2 - log, 3 - numerico unsigned, 4 - testo, 5 - binario, 6 - JSON

Esempio di esportazione del valore dell'item

Valore numerico (senza segno):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valore numerico (float):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valore carattere, testo:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valore di log:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Esportazione delle tendenze

Le seguenti informazioni vengono esportate per un valore di tendenza calcolato:

Field		Type	Description
host		object	Nome host dell'host dell'item.
	host	string	Nome host.
	name	string	Nome host visibile.
groups		array	Elenco dei gruppi host dell'host dell'item; nell'array deve essere presente almeno un elemento.
	-	string	Nome del gruppo host.
item_tags		array	Elenco dei tag dell'item (può essere vuoto).
	-	object	
	tag	string	Nome del tag.
	value	string	Valore del tag (può essere vuoto).
itemid		number	ID dell'item.
name		string	Nome visibile dell'item.
clock		number	Numero di secondi dall'Epoch al momento in cui il valore è stato raccolto (parte intera).
count		number	Numero di valori raccolti per una determinata ora.
min		number	Valore minimo dell'item per una determinata ora.
avg		number	Valore medio dell'item per una determinata ora.
max		number	Valore massimo dell'item per una determinata ora.

Field	Type	Description
type	number	Tipo di valore: 0 - float numerico, 3 - unsigned numerico

Esempi di esportazione delle tendenze

Valore numerico (unsigned):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valore numerico (float):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Streaming verso sistemi esterni

Panoramica

È possibile trasmettere i valori degli item e gli eventi da Zabbix a sistemi esterni tramite HTTP (vedere i dettagli del protocollo).

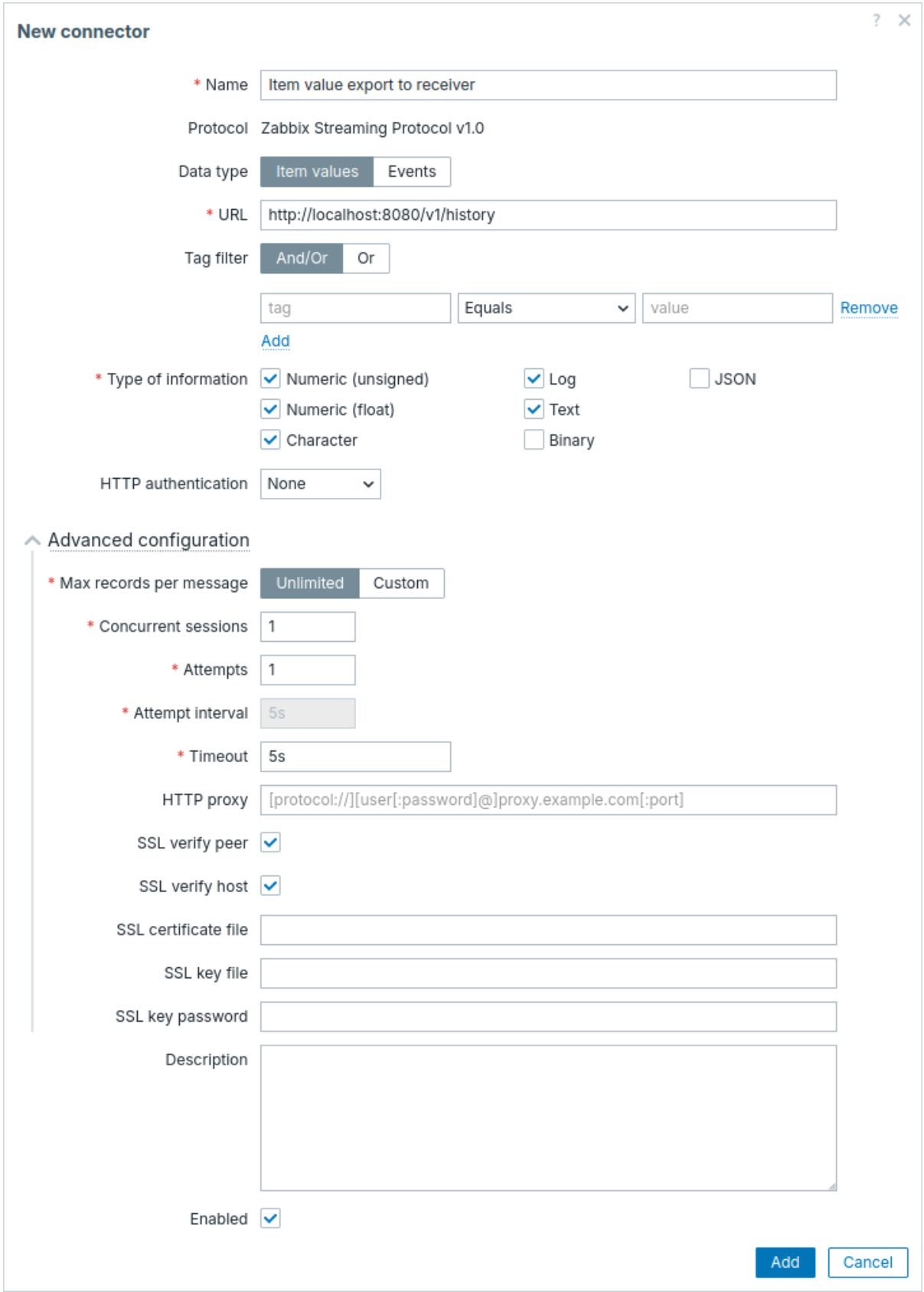
Il filtro per tag può essere utilizzato per trasmettere sottoinsiemi di valori degli item o di eventi.

Due tipi di processi del server Zabbix sono responsabili della trasmissione dei dati: `connector manager` e `connector worker`. Un item interno di Zabbix, `zabbix[connector_queue]`, consente di monitorare il numero di valori accodati nella coda del connector.

Configurazione

Per configurare lo streaming dei dati verso un sistema esterno, sono necessari i seguenti passaggi:

1. Configurare un sistema remoto per ricevere i dati da Zabbix. A questo scopo, sono disponibili i seguenti strumenti:
   - Un esempio di semplice receiver che registra le informazioni ricevute nei file `events.ndjson` e `history.ndjson`.
   - Kafka connector for Zabbix server - un server leggero scritto in Go, progettato per inoltrare i valori degli item e gli eventi da un server Zabbix a un broker Kafka.
2. Impostare in Zabbix il numero richiesto di worker del connector modificando il parametro `StartConnectors` in `zabbix_server.conf`. Il numero di worker del connector deve corrispondere al numero di connector configurato nel frontend di Zabbix (o essere superiore se il numero di sessioni simultanee è maggiore di 1). Quindi, riavviare il server Zabbix.
3. Configurare un nuovo connector nel frontend di Zabbix (*Administration* > *General* > *Connectors*) e ricaricare la cache del server con il comando `zabbix_server -R config_cache_reload`.



I campi obbligatori sono contrassegnati da un asterisco.

Parametro	Descrizione
Name	Inserire il nome del connector.
Data type	Selezionare il tipo di dati da trasmettere: **Item values** - trasmette i valori degli item da Zabbix ai sistemi esterni; **Events** - trasmette gli eventi da Zabbix ai sistemi esterni.
URL	Inserire l'URL del receiver. Sono supportate le macro utente.

Parametro	Descrizione
Tag filter	Esporta solo i valori degli item o gli eventi che corrispondono al filtro dei tag. Se non impostato, esporta tutto. È possibile includere o escludere tag e valori dei tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag vengono raggruppate con la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione.
Type of information	Selezionare il tipo di informazione (numerica (senza segno), numerica (float), carattere, ecc.) in base al quale filtrare i valori degli item che il connector deve trasmettere. Questo campo è disponibile se *Data type* è impostato su ”Item values”.
HTTP authentication	Selezionare l’opzione di autenticazione: **None** - non viene utilizzata alcuna autenticazione; **Basic** - viene utilizzata l’autenticazione di base; **NTLM** - viene utilizzata l’autenticazione NTLM (Windows NT LAN Manager); **Kerberos** - viene utilizzata l’autenticazione Kerberos (vedere anche: Configurazione di Kerberos con Zabbix); **Digest** - viene utilizzata l’autenticazione Digest; **Bearer** - viene utilizzata l’autenticazione Bearer.
Username	Inserire il nome utente (fino a 255 caratteri). Sono supportate le macro utente. Questo campo è disponibile se *HTTP authentication* è impostato su ”Basic”, ”NTLM”, ”Kerberos” o ”Digest”.
Password	Inserire la password dell’utente (fino a 255 caratteri). Sono supportate le macro utente. Questo campo è disponibile se *HTTP authentication* è impostato su ”Basic”, ”NTLM”, ”Kerberos” o ”Digest”.
Bearer token	Inserire il token Bearer. Sono supportate le macro utente. Questo campo è disponibile ed è obbligatorio se *HTTP authentication* è impostato su ”Bearer”.
Advanced configuration	Fare clic sull’intestazione *Advanced configuration* per visualizzare le opzioni di configurazione avanzate (vedere di seguito).
Max records per message	Specificare il numero massimo di valori o eventi che possono essere trasmessi in un singolo messaggio.
Concurrent sessions	Selezionare il numero di processi sender da eseguire per questo connector. È possibile specificare fino a 100 sessioni; il valore predefinito è ”1”.
Attempts	Numero di tentativi per la trasmissione dei dati. È possibile specificare fino a 5 tentativi; il valore predefinito è ”1”.
Attempt interval	Specificare per quanto tempo il connector deve attendere dopo un tentativo non riuscito di trasmettere i dati. È possibile specificare fino a 10s; il valore predefinito è ”5s”. Questo campo è disponibile se *Attempts* è impostato su ”2” o più. I tentativi non riusciti sono quelli in cui non è stato possibile stabilire una connessione oppure in cui il codice di risposta HTTP non è 200, 201, 202, 203 o 204. I nuovi tentativi vengono **attivati** in caso di errori di comunicazione o quando il codice di risposta HTTP non è 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415 o 422. I reindirizzamenti vengono seguiti, quindi 302 -> 200 è una risposta positiva, mentre 302 -> 503 attiverà un nuovo tentativo.
Timeout	Specificare il timeout del messaggio (1-60 secondi, valore predefinito - 5 secondi). Sono supportati i suffissi temporali (ad esempio, 30s, 1m). Sono supportate le macro utente.

Parametro	Descrizione
HTTP proxy	È possibile specificare un proxy HTTP nel seguente formato: `[protocol://][username[:password]@]proxy.example.com[:port]` Sono supportate le macro utente. Il prefisso facoltativo `protocol://` può essere utilizzato per specificare protocolli proxy alternativi (il supporto per il prefisso del protocollo è stato aggiunto in cURL 7.21.7). Se non viene specificato alcun protocollo, il proxy verrà trattato come proxy HTTP. Per impostazione predefinita, verrà utilizzata la porta 1080. Se *HTTP proxy* è specificato, il proxy sovrascriverà le variabili d'ambiente relative al proxy, come `http_proxy` e `HTTPS_PROXY`. Se non è specificato, il proxy non sovrascriverà le variabili d'ambiente relative al proxy. Il valore inserito viene passato così com'è, senza alcun controllo di validità. È inoltre possibile inserire l'indirizzo di un proxy SOCKS. Se si specifica il protocollo errato, il connector non riuscirà a trasmettere i valori degli item o gli eventi da Zabbix. Si noti che con il proxy HTTP è supportata solo l'autenticazione semplice.
SSL verify peer	Selezionare la casella per verificare il certificato SSL del web server. Il certificato del server viene verificato rispetto ai certificati dell'autorità di certificazione (CA) attendibili presenti nel percorso CA a livello di sistema. È possibile sovrascrivere il percorso dei file CA utilizzando il parametro di configurazione del server Zabbix o del proxy `SSLCALocation`.
SSL verify host	Selezionare la casella per verificare che il campo *Common Name* o il campo *Subject Alternate Name* del certificato del web server corrisponda. Questa opzione imposta l'opzione cURL `CURLOPT_SSL_VERIFYHOST`.
SSL certificate file	Nome del file del certificato SSL utilizzato per l'autenticazione del client. Il file del certificato deve essere in formato PEM[1]. Sono supportate le macro utente. Se il file del certificato contiene anche la chiave privata, lasciare vuoto il campo *SSL key file*. Se la chiave è crittografata, specificare la password nel campo *SSL key password*. La directory contenente questo file è specificata dal parametro di configurazione del server Zabbix o del proxy `SSLCertLocation`.
SSL key file	Nome del file della chiave privata SSL utilizzato per l'autenticazione del client. Il file della chiave privata deve essere in formato PEM[1]. Sono supportate le macro utente. La directory contenente questo file è specificata dal parametro di configurazione del server Zabbix o del proxy `SSLKeyLocation`.
SSL key password	Password del file della chiave privata SSL. Sono supportate le macro utente.
Description	Inserire la descrizione del connector.
Enabled	Selezionare la casella per abilitare il connector.

Quando il connector Kafka è configurato con un elenco separato da virgole di indirizzi dei broker bootstrap (ad esempio, `Kafka.URL=kafka1.example.com:9093,kafka2.example.com:9093`), il client Kafka si connette al broker o ai broker che rispondono per primi e utilizza i metadati del cluster. Se l'elenco contiene indirizzi appartenenti a cluster Kafka diversi, verrà utilizzato solo il cluster che risponde più velocemente e gli altri indirizzi verranno registrati come non disponibili; di conseguenza, potrebbero comparire avvisi all'avvio come il seguente anche se il connector è connesso:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

In alcuni ambienti (reti private, reti di container o configurazioni DNS/hosts non standard), i nomi host o gli indirizzi IP possono risolversi in indirizzi di loopback (ad esempio, `127.0.0.1/localhost`) oppure essere normalizzati dal client, rendendo fuorvianti tali avvisi. Per ridurre la confusione, assicurarsi che tutti gli indirizzi `Kafka.URL` appartengano allo stesso cluster Kafka, verificare la risoluzione DNS dall'host del connector e i valori `advertised.listeners` dei broker e preferire gli indirizzi che si risolvono nell'indirizzo pubblicizzato dal broker.

Protocollo

La comunicazione tra il server e il receiver avviene tramite HTTP usando REST API, NDJSON, "Content-Type: application/x-ndjson".

Per ulteriori dettagli, vedere Newline-delimited JSON export protocol.

Richiesta del server

Esempio di streaming dei valori degli item:

```
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
```

Esempio di streaming degli eventi:

```
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}
```

Risposta del receiver

La risposta è composta dal codice di stato della risposta HTTP e dal payload JSON. Il codice di stato della risposta HTTP deve essere ”200”, ”201”, ”202”, ”203” o ”204” per le richieste gestite correttamente, altri per le richieste non riuscite.

Esempio di successo:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

Esempio con errori:

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

Gateway SNMP

Panoramica

Zabbix SNMP gateway è un’estensione AgentX per snmpd che supporta sia il polling SNMP sia il trapping.

Con Zabbix SNMP gateway è possibile utilizzare il protocollo SNMP per recuperare:

- dati dei trigger;
- dati dei trigger di problema;
- stato dei gruppi di host (conteggio dei trigger per stato del trigger per gruppo)

I dati vengono recuperati tramite l’OID, che è una combinazione di una base comune e di un suffisso specifico. La **base** comune viene impostata nel file di configurazione di SNMP gateway, ad esempio:

- `BaseOID=1.3.6.1.4.1.3043.7.55` - per qualsiasi dato dei trigger;
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - per i dati dei trigger di problema;
- `BaseOID=1.3.6.1.4.1.3046.7.55` - per lo stato dei gruppi di host.

Il **suffisso** dell’OID viene impostato nella configurazione dei trigger dell’host come **tag** (ad esempio, `OIDSuffix:3`) nel frontend.

In questo caso tutte le informazioni per il trigger saranno disponibili in `OID=1.3.6.1.4.1.3043.7.55.X.3`. Qui "X" sarà il numero dei campi dati del trigger (cioè 1 - suffisso, 2 - ID, 3 - espressione, 4 - descrizione, ecc.).

Per una descrizione più dettagliata e un esempio del file di configurazione, vedere il file readme di SNMP gateway.

Installazione e configurazione

Per istruzioni su quanto segue, consultare il file readme di SNMP gateway:

- installazione e configurazione di snmpd;
- abilitazione del supporto AgentX;
- configurazione di Zabbix SNMP gateway;
- configurazione delle trap SNMP per le modifiche di stato dei trigger.

Recupero dei dati

Una volta configurato tutto correttamente, è possibile utilizzare i comandi `snmpwalk` e `snmpget` per recuperare i dati:

```
[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
...
```

Opzioni di filtro

È possibile limitare le informazioni sui trigger di problema nella configurazione dell'SNMP gateway:

- per gravità (per impostazione predefinita `ProblemMinSeverity=-1`)
- nascondendo i problemi confermati (per impostazione predefinita `ProblemHideAck=false`)

È possibile limitare il conteggio dei problemi per gruppo di host nella configurazione dell'SNMP gateway:

- per trigger con stato sconosciuto (per impostazione predefinita `CountUnknown=false`)
- per trigger con problemi confermati/non confermati/tutti (per impostazione predefinita `CountAcknowledgeStatus=all`)

Esportazione/importazione della configurazione

Panoramica

La funzionalità di esportazione/importazione di Zabbix consente di scambiare varie entità di configurazione tra un sistema Zabbix e un altro.

Casi d'uso tipici di questa funzionalità:

- condividere template o mappe di rete - gli utenti Zabbix possono condividere i propri parametri di configurazione
- caricare un template su Zabbix Community templates. In seguito, altri possono scaricare il template e importare il file in Zabbix.
- integrarsi con strumenti di terze parti - i formati universali YAML, XML e JSON rendono possibile l'integrazione e l'importazione/esportazione dei dati con strumenti e applicazioni di terze parti

Cosa può essere esportato/importato

Gli oggetti che possono essere esportati/importati sono:

- Host groups (*solo tramite Zabbix API*)
- Template groups (*solo tramite Zabbix API*)
- template

- host
- Mappe di rete
- Tipi di media
- Immagini
- Dashboard
- Espressioni regolari

Formato di esportazione

I dati possono essere esportati tramite il frontend web di Zabbix o Zabbix API. I formati di esportazione supportati sono YAML, XML e JSON.

Dettagli sull'esportazione

- Tutti gli elementi supportati vengono esportati in un unico file.
- Le entità di host e template (item, trigger, grafici, regole di discovery) ereditate dai template collegati non vengono esportate. Eventuali modifiche apportate a tali entità a livello di host (ad esempio, intervallo dell'item modificato, espressione regolare modificata o prototipi aggiunti alla regola di discovery a basso livello) andranno perse durante l'esportazione; durante l'importazione, tutte le entità dei template collegati vengono ricreate come nel template collegato originale.
- Le entità create dalla discovery a basso livello e tutte le entità che dipendono da esse non vengono esportate. Ad esempio, un trigger creato per un item generato da una regola LLD non verrà esportato.
- Quando l'host/template esportato contiene entità che supportano i timeout, i valori di timeout verranno esportati se tali entità hanno timeout propri configurati.

Dettagli sull'importazione

- L'importazione si interrompe al primo errore.
- Durante l'aggiornamento delle immagini esistenti nell'importazione delle immagini, il campo "imagetype" viene ignorato, cioè non è possibile modificare il tipo di immagine tramite importazione.
- Quando si importano host/template usando l'opzione "Delete missing", le macro di host/template non presenti nel file di importazione verranno eliminate dall'host/template dopo l'importazione.
- I tag vuoti per item, trigger, graph, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes non hanno significato, cioè è come se fossero mancanti.
- Se le entità dell'host/template importato hanno timeout propri configurati, verranno applicati; in caso contrario, verranno applicati i timeout del proxy/globali.
- L'importazione supporta YAML, XML e JSON; il file di importazione deve avere un'estensione corretta: .yaml e .yml per YAML, .xml per XML e .json per JSON. Vedere le informazioni sulla compatibilità per le versioni XML supportate.
- L'importazione supporta i file di configurazione solo in codifica UTF-8 (con o senza BOM); altre codifiche (UTF16LE, UTF16BE, UTF32LE, UTF32BE, ecc.) comporteranno un errore di conversione dell'importazione.

Formato base YAML

Il formato di esportazione YAML contiene i seguenti nodi:

- Nodo radice per l'esportazione YAML di Zabbix
- Versione di esportazione

```
zabbix_export:
  version: '8.0'
```

Gli altri nodi dipendono dagli oggetti esportati.

Formato XML

Il formato di esportazione XML contiene i seguenti tag:

- Intestazione predefinita per i documenti XML
- Tag radice per l'esportazione XML di Zabbix
- Versione di esportazione

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>8.0</version>
</zabbix_export>
```

Gli altri tag dipendono dagli oggetti esportati.

Formato JSON

Il formato di esportazione JSON contiene i seguenti oggetti:

- Oggetto radice per l'esportazione JSON di Zabbix

- Versione di esportazione

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Gli altri oggetti dipendono dagli oggetti esportati.

Gruppi di template

Panoramica

Nel frontend, i gruppi di template possono essere esportati solo con l'esportazione del template. Quando un template viene esportato, tutti i gruppi a cui appartiene vengono esportati automaticamente insieme ad esso.

L'API consente di esportare i gruppi di template indipendentemente dai template.

Formato di esportazione

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Network devices'
```

Elementi esportati

Element	Type	Description
uuid	string	Identificatore univoco per questo gruppo di template.
name	string	Nome del gruppo.

Gruppi host

Panoramica

Nel frontend, i gruppi host possono essere esportati solo con l'esportazione dell'host. Quando un host viene esportato, tutti i gruppi a cui appartiene vengono esportati automaticamente insieme ad esso.

L'API consente di esportare i gruppi host indipendentemente dagli host.

Formato di esportazione

```
   host_groups:
    - uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Zabbix servers'
```

Elementi esportati

Element	Type	Description
uuid	string	Identificatore univoco per questo gruppo di host.
name	string	Nome del gruppo.

Template

Panoramica

I template vengono esportati con molti oggetti correlati e relazioni tra oggetti.

L'esportazione del template contiene:

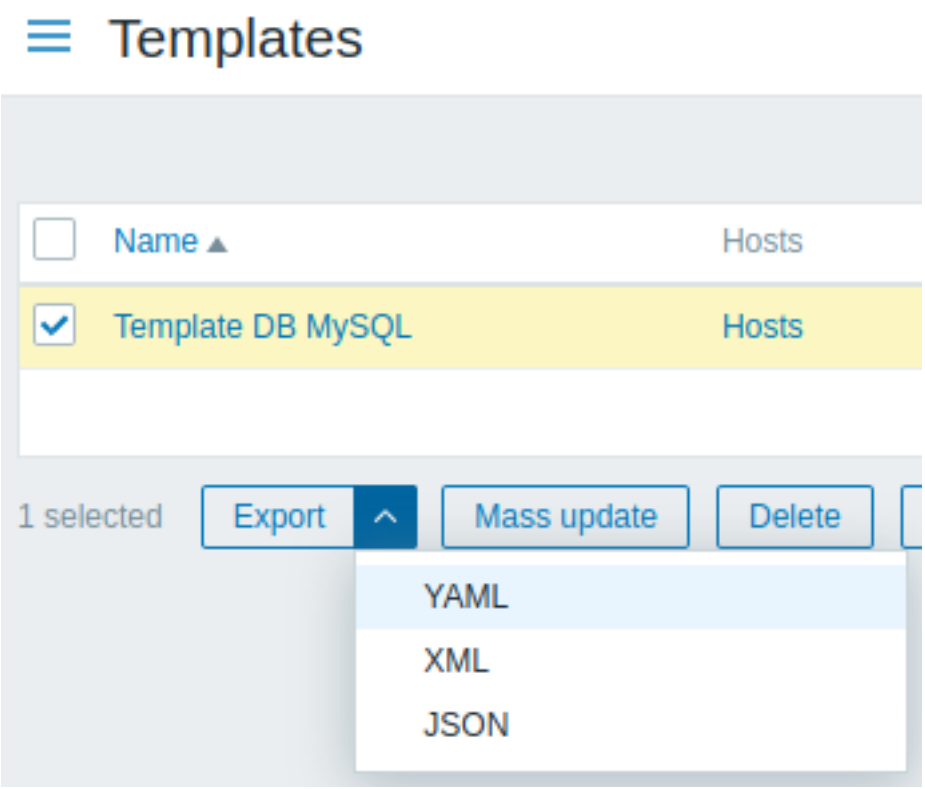
- Gruppi di template collegati
- Gruppi di host collegati (se utilizzati nella configurazione del host prototype)
- Dati del template
- Collegamento ad altri template
- Collegamento ai gruppi di template
- item collegati direttamente

- trigger collegati direttamente
- grafici collegati direttamente
- dashboard collegati direttamente
- regole di discovery collegate direttamente con tutti i prototipi
- scenari web collegati direttamente
- mappe dei valori

Esportazione

Per esportare i template, procedere come segue:

1. Andare su *Data collection* > *Templates*.
2. Selezionare le caselle di controllo dei template da esportare.
3. Fare clic su *Export* sotto l'elenco per esportare nel formato YAML predefinito, oppure fare clic sulla freccia per selezionare un formato di esportazione diverso.



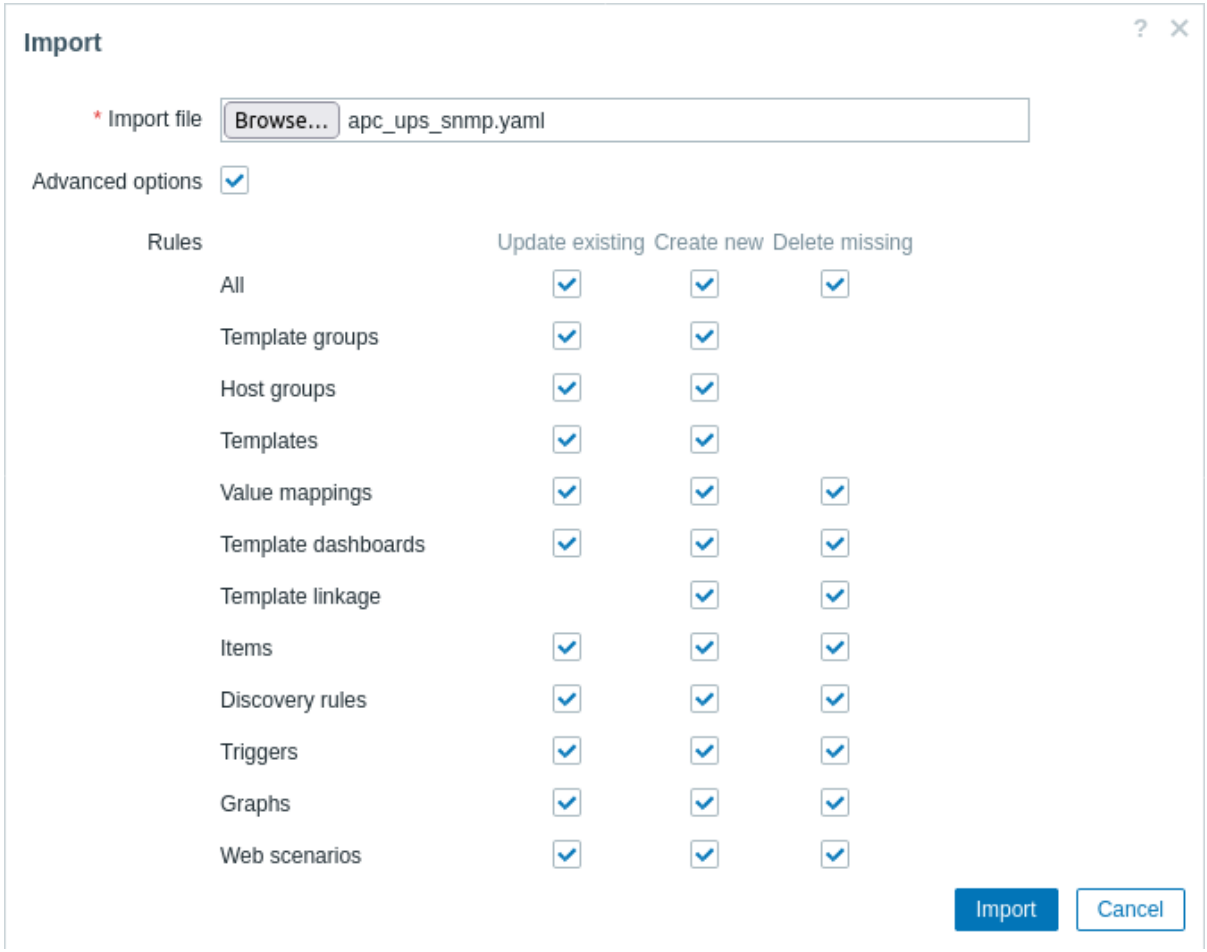
A seconda del formato selezionato, i template vengono esportati in un file locale con un nome predefinito:

- `zabbix_export_templates.yaml` - nell'esportazione YAML;
- `zabbix_export_templates.xml` - nell'esportazione XML;
- `zabbix_export_templates.json` - nell'esportazione JSON.

Importazione

Per importare i template, procedere come segue:

1. Andare su *Raccolta dati* > *Template*.
2. Fare clic su *Importa* nell'angolo superiore destro.
3. Selezionare il file di importazione.
4. Selezionare le caselle di controllo delle regole di importazione da applicare (vedere di seguito).
5. Fare clic su *Importa* nell'angolo inferiore destro del modulo di configurazione.



Se si seleziona la casella di controllo *Opzioni avanzate*, verrà visualizzato un elenco dettagliato di tutti gli elementi importabili; selezionare o deselezionare ogni regola di importazione secondo necessità.

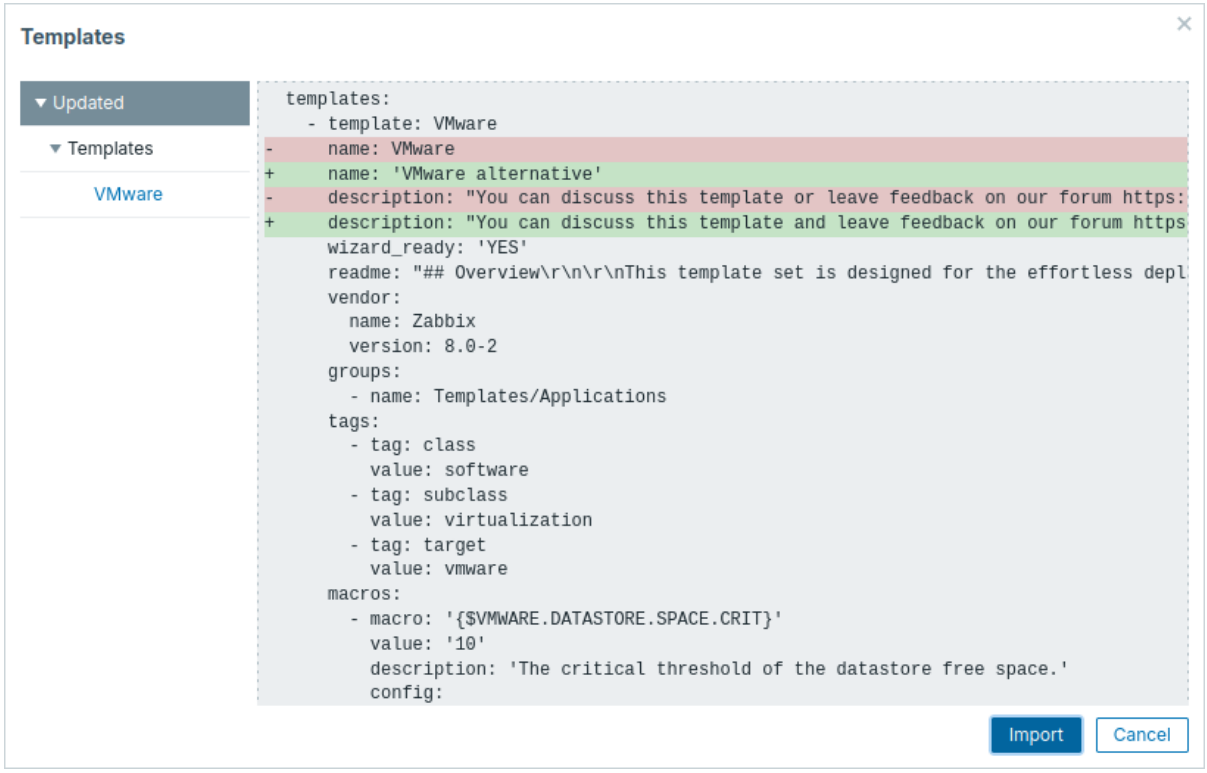
Se si fa clic sulla casella di controllo nella riga *Tutti*, tutti gli elementi sottostanti verranno selezionati o deselezionati.

Regole di importazione:

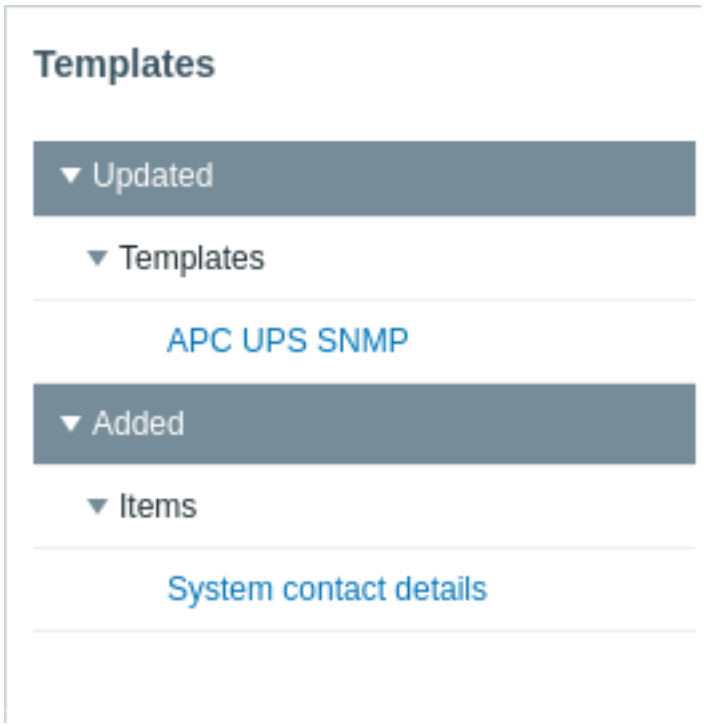
Regola	Descrizione
Aggiorna esistenti	Gli elementi esistenti verranno aggiornati utilizzando i dati del file di importazione. In caso contrario, non verranno aggiornati.
Crea nuovi	I nuovi elementi verranno creati utilizzando i dati del file di importazione. In caso contrario, non verranno creati.
Elimina mancanti	Gli elementi esistenti non presenti nel file di importazione verranno rimossi. In caso contrario, non verranno rimossi. Se *Elimina mancanti* è selezionato per *Collegamento template*, i collegamenti al template attuali non presenti nel file di importazione verranno rimossi. Le entità (item, trigger, grafici ecc.) ereditate dai template scollegati non verranno rimosse, a meno che l'opzione *Elimina mancanti* non sia selezionata anche per ciascuna entità.

Attention:
Se esistono già template con gli stessi nomi, durante l'importazione è necessario selezionare le opzioni *Elimina mancanti* per ottenere un'importazione pulita. In questo modo, i vecchi item che non sono più presenti nel template aggiornato verranno rimossi (si noti che ciò comporterà la perdita della cronologia di questi vecchi item).

Nella schermata successiva sarà possibile visualizzare il contenuto del template da importare. Se si tratta di un nuovo template, tutti gli elementi saranno elencati in verde. Se si sta aggiornando un template esistente, i nuovi elementi del template saranno evidenziati in verde; gli elementi rimossi saranno evidenziati in rosso; gli elementi che non sono cambiati saranno elencati su uno sfondo grigio.



Il menu a sinistra può essere utilizzato per spostarsi nell'elenco delle modifiche. La sezione *Aggiornati* evidenzia tutte le modifiche apportate agli elementi esistenti del template. La sezione *Aggiunti* elenca i nuovi elementi del template. Gli elementi di ciascuna sezione sono raggruppati per tipo di elemento; fare clic sulla freccia grigia per espandere o comprimere il gruppo di elementi.



Esaminare le modifiche al template, quindi fare clic su *Importa* per eseguire l'importazione del template. Nel frontend verrà visualizzato un messaggio che indica se l'importazione è riuscita o non è riuscita.

Formato di esportazione Quando un template viene esportato, Zabbix genera un formato YAML, JSON o XML strutturato. L'esportazione include gli elementi del template, come i metadati del template, gli item, le macro, i trigger, i dashboard e altro ancora.

Ogni elemento ha uno scopo specifico e può contenere elementi nidificati.

Le sezioni seguenti descrivono ciascun elemento nel formato di esportazione. Gli esempi usano il template Linux by Zabbix agent.

Un'ellissi `(...)` indica elementi omessi per brevità. La nota `(see table below)` viene usata quando l'elemento è spiegato in modo più dettagliato nelle sezioni successive.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (see table below)
  triggers: (see table below)
  graphs: (see table below)
```

Metadati del template

Element	Type	Descrizione
version	string	(obbligatorio) Versione di esportazione.
template_groups		(obbligatorio) Elemento radice per i gruppi di template.
uuid	string	(obbligatorio) Identificatore univoco per questo gruppo di template.
name	string	(obbligatorio) Nome del gruppo di template.
host_groups		Elemento radice per i gruppi di host utilizzati dai prototipi di host.
uuid	string	(obbligatorio) Identificatore univoco per questo gruppo di host.
name	string	(obbligatorio) Nome del gruppo di host.
templates		Elemento radice per i template.
triggers		Elemento radice per gli elementi trigger del template, che sono gli stessi dei trigger degli item del template.
graphs		Elemento radice per gli elementi grafico del template, che sono gli stessi dei grafici dell'host.

Template

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        Questo è un template ufficiale per Linux. Richiede Zabbix agent 8.0 o versioni successive. (...)
      wizard_ready: 'YES'
      readme: |
        ## Panoramica

        Questo è un template ufficiale per Linux. Richiede Zabbix agent 8.0 o versioni successive. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Sistemi operativi'
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Timeout dopo il quale l'agent è considerato non disponibile. Funziona solo per gli
          config: (see table below)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Soglia critica dell'utilizzo della CPU espressa in %.'
          config: (see table below)
```

```
      (...)
    dashboards: (see table below)
    valuemaps: (see table below)
  (...)
```

Elemento	Tipo	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo template.
template	string	(obbligatorio) Nome univoco del template.
name	string	Nome visibile del template.
description	text	Descrizione del template.
wizard_ready	text	Indica se il template è disponibile per la selezione nella Procedura guidata host. Valori possibili:[1] NO (0, predefinito), YES (1). Vedi anche: Oggetto template (`wizard_ready`).
readme	text	Istruzioni di configurazione specifiche del template da visualizzare nella Procedura guidata host. Supporta la formattazione Markdown.
vendor		Elemento radice per il vendor del template (presente se il template esportato contiene dati del vendor).
name	string	(obbligatorio) Nome del vendor del template.
version	string	(obbligatorio) Versione del template. Per i template pronti all'uso, la versione viene visualizzata come segue: versione principale di Zabbix, delimitatore ("-"), numero di revisione (incrementato a ogni nuova versione del template e azzerato a ogni versione principale di Zabbix). Ad esempio, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Elemento radice per i template collegati.
name	string	(obbligatorio) Nome del template.
groups		Elemento radice per i gruppi di template.
name	string	(obbligatorio) Nome del gruppo di template.
items		Elemento radice per gli item del template.
discovery_rules		Elemento radice per le regole di discovery low-level del template.
httptests		Elemento radice per gli scenari web del template.
tags		Elemento radice per i tag del template.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
macros		Elemento radice per le macro utente del template.
macro	string	(obbligatorio) Nome della macro utente.
type	string	Tipo di macro utente. Valori possibili:[1] TEXT (0, predefinito), SECRET_TEXT (1), VAULT (2). Vedi anche: Oggetto macro utente (`type`).
value	string	Valore della macro utente.
description	string	Descrizione della macro utente.
config		Elemento radice per la configurazione della macro del template, responsabile di come la macro viene visualizzata nella Procedura guidata host.
dashboards		Elemento radice per i dashboard del template.
valuemaps		Elemento radice per le mappe dei valori del template.

Item del template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Ping dell''agent Zabbix'
          key: agent.ping
          description: 'L''agent restituisce sempre "1" per questo item. Può essere utilizzato in combinaz
          valuemap:
            name: 'Stato del ping dell''agent Zabbix'
          tags:
            - tag: component
              value: system
        (...)
```

```
      - uuid: 58818005e76d46dda14d6592f601ab00
        name: 'Numero di pacchetti installati'
        key: system.sw.packages.get
        delay: 1h
        preprocessing: (vedere la tabella seguente)
        tags:
          - tag: component
            value: os
        triggers: (vedere la tabella seguente)
      - uuid: 403cebed115441369e94d35d070ca7b8
        name: 'Utilizzo della memoria'
        type: DEPENDENT
        key: vm.memory.utilization
        value_type: FLOAT
        units: '%'
        description: 'La percentuale di memoria utilizzata viene calcolata come `100-pavailable`.'
        preprocessing: (vedere la tabella seguente)
        master_item:
          key: 'vm.memory.size[pavailable]'
        tags:
          - tag: component
            value: memory
        triggers: (vedere la tabella seguente)
      (...)
```

Elemento	Tipo	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo item.
name	string	(obbligatorio) Nome dell'item.
type	string	Tipo di item. Valori possibili:[1] ZABBIX_PASSIVE (0, predefinito), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), TELEMETRY_QUERY (24).
snmp_oid	string	(obbligatorio per gli item SNMP_AGENT) ID dell'oggetto SNMP.
key	string	(obbligatorio) Chiave dell'item.
delay	string	Intervallo di aggiornamento dell'item. Predefinito: 1m. Il valore sarà sempre 0 per gli item TRAP.
history	string	Periodo di tempo (utilizzando un suffisso temporale, una macro utente o una macro LLD) per il quale devono essere conservati i dati della cronologia. Predefinito: 31d.
trends	string	Periodo di tempo (utilizzando un suffisso temporale, una macro utente o una macro LLD) per il quale devono essere conservati i dati dei trend. Predefinito: 365d.
status	string	Stato dell'item. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
value_type	string	Tipo di valore ricevuto. Valori possibili:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, predefinito), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Elenco di indirizzi IP, separati da virgole, degli host autorizzati a inviare dati per l'item. Supportato per gli item TRAP e HTTP_AGENT.
units	string	Unità del valore ricevuto (bps, B, ecc.).
params	text	Parametri aggiuntivi a seconda del tipo di item (script eseguito per gli item SSH e TELNET; query SQL per gli item ODBC; formula per gli item CALCULATED; script per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER).
ipmi_sensor	string	Sensore IPMI. Supportato per gli item IPMI.
authtype	string	Tipo di autenticazione. Supportato per gli item SSH e HTTP_AGENT. Valori possibili per gli item SSH:[1] PASSWORD (0, predefinito), PUBLIC_KEY (1). Valori possibili per gli item HTTP_AGENT:[1] NONE (0, predefinito), BASIC (1), NTLM (2), Kerberos (3) o Digest (4).

Elemento	Tipo	Descrizione
username	string	(obbligatorio per gli item SSH e TELNET) Nome utente per l'autenticazione. Supportato per gli item SIMPLE, ODBC, JMX e HTTP_AGENT. Quando viene utilizzato per gli item JMX, è necessario specificare anche `password` (vedere sotto) oppure lasciare vuoti entrambi gli elementi.
password	string	(obbligatorio per gli item SSH e TELNET) Password per l'autenticazione. Supportato per gli item SIMPLE, ODBC, JMX e HTTP_AGENT. Quando viene utilizzato per gli item JMX, è necessario specificare anche `username` (vedere sopra) oppure lasciare vuoti entrambi gli elementi.
publickey	string	(obbligatorio per gli item SSH) Nome del file della chiave pubblica.
privatekey	string	(obbligatorio per gli item SSH) Nome del file della chiave privata.
description	text	Descrizione dell'item.
inventory_link	string	Campo dell'inventario dell'host popolato dall'item. Valori possibili:[1] NONE (0), ALIAS (4), ecc. (vedere Inventario dell'host per i campi supportati).
valuemap		Elemento radice per le mappe dei valori dell'item.
name	string	(obbligatorio) Nome della mappa dei valori da utilizzare per l'item.
logtimefmt	string	Formato dell'ora nelle voci di log. Supportato per gli item di tipo valore LOG.
preprocessing		Elemento radice per la pre-elaborazione dei valori dell'item.
step		Elemento radice per i passaggi di pre-elaborazione dei valori degli item del template.
jmx_endpoint	string	Endpoint JMX. Supportato per gli item JMX.
master_item		(obbligatorio per gli item DEPENDENT) Elemento radice per il master item dell'item dipendente.
key	string	(obbligatorio) Chiave del master item dell'item dipendente.
timeout	string	Timeout della richiesta di polling dei dati dell'item. Supportato per l'elenco Timeout dei tipi di item.
url	string	(obbligatorio per gli item HTTP_AGENT) Stringa URL.
query_fields		Elemento radice per i parametri della query. Supportato per gli item HTTP_AGENT.
name	string	(obbligatorio per gli item HTTP_AGENT) Nome del parametro della query.
value	string	Valore del parametro della query. Supportato per gli item HTTP_AGENT.
parameters		Elemento radice per i parametri definiti dall'utente. Supportato per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
name	string	(obbligatorio per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER) Nome del parametro definito dall'utente.
value	string	Valore del parametro definito dall'utente. Supportato per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
posts	string	Dati del corpo della richiesta HTTP(S). Supportato per gli item HTTP_AGENT.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole. Supportato per gli item HTTP_AGENT.
follow_redirects	string	Segue i reindirizzamenti della risposta durante il polling dei dati. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0), YES (1, predefinito).
post_type	string	Tipo di corpo dei dati post. Supportato per gli item HTTP_AGENT. Valori possibili:[1] RAW (0, predefinito), JSON (2), XML (3).
http_proxy	string	Stringa di connessione al proxy HTTP(S). Supportato per gli item HTTP_AGENT.
headers		Elemento radice per le intestazioni della richiesta HTTP(S). Supportato per gli item HTTP_AGENT.
name	string	(obbligatorio per gli item HTTP_AGENT) Nome dell'intestazione.
value	string	(obbligatorio per gli item HTTP_AGENT) Valore dell'intestazione.
retrieve_mode	string	Parte della risposta da memorizzare. Supportato per gli item HTTP_AGENT. Valori possibili:[1] BODY (0, predefinito), HEADERS (1), BOTH (2).
request_method	string	Tipo di metodo della richiesta. Supportato per gli item HTTP_AGENT. Valori possibili:[1] GET (0, predefinito), POST (1), PUT (2), HEAD (3).

Elemento	Tipo	Descrizione
output_format	string	Modalità di elaborazione della risposta. Supportato per gli item HTTP_AGENT. Valori possibili:[1] RAW (0, predefinito), JSON (1).
allow_traps	string	Consente di popolare il valore in modo analogo all'item trapper. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Supportato per gli item HTTP_AGENT.
ssl_key_file	string	Percorso del file della chiave SSL privata. Supportato per gli item HTTP_AGENT.
ssl_key_password	string	Password del file della chiave SSL. Supportato per gli item HTTP_AGENT.
verify_peer	string	Indica se verificare che il certificato dell'host sia autentico. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
verify_host	string	Indica se verificare che il nome host della connessione corrisponda a quello nel certificato dell'host. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
time_shift	string	Quanto indietro nel tempo, rispetto all'ora corrente, termina l'intervallo temporale interrogato dall'item. Supportato per gli item TELEMETRY_QUERY.
lookback_limit	string	Quanto indietro nel tempo, rispetto all'ora corrente, l'item cerca i dati quando non esiste una query precedente da cui continuare (ad esempio, alla prima esecuzione dell'item) o quando la query precedente è più vecchia di `delay` (ad esempio perché il server Zabbix era inattivo). Supportato per gli item TELEMETRY_QUERY.
granularity	string	Durata di una porzione in cui viene suddiviso l'intero intervallo temporale della query. Supportato per gli item TELEMETRY_QUERY.
query		(obbligatorio per gli item TELEMETRY_QUERY) Elemento radice per la query di telemetria. Definisce cosa interroga l'item dall'origine dati, come filtra i dati e come aggrega il risultato.
signal_type	string	(obbligatorio) Tipo di segnale interrogato dall'item. Supportato per gli item TELEMETRY_QUERY. Valori possibili:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obbligatorio per il tipo di segnale METRICS) Tipo di punto metrica interrogato dall'item. Supportato per gli item TELEMETRY_QUERY. Valori possibili:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obbligatorio) Elemento radice per le colonne dei dati di telemetria interrogate dall'item. Questo elemento contiene i seguenti elementi: `column (string)` - (obbligatorio) Nome della colonna dei dati di telemetria interrogata dall'item. `attribute_key (string)` - (obbligatorio per le colonne che contengono coppie chiave-valore) Nome della chiave interrogata dall'item, all'interno della colonna che contiene coppie chiave-valore.
aggregated_columns		(obbligatorio) Elemento radice per le colonne aggregate dei dati di telemetria interrogate dall'item. Questo elemento contiene i seguenti elementi: `column (string)` - (obbligatorio per le funzioni MIN, MAX, AVG, SUM e PERCENTILE) Nome della colonna dei dati di telemetria interrogata dall'item. `function (string)` - (obbligatorio) Funzione applicata dall'item ai record della colonna. Valori possibili:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters (array)` - (obbligatorio per le funzioni PERCENTILE) Percentile calcolato dall'item sui record della colonna. `alias (string)` - (obbligatorio) Nome univoco del valore calcolato, utilizzato per identificarlo nel valore dell'item.

Elemento	Tipo	Descrizione
filter		(obbligatorio) Elemento radice per il filtro della query di telemetria applicato ai record delle colonne nella query. Questo elemento contiene i seguenti elementi: `evaltype` (`string`) - (obbligatorio) Metodo di valutazione della condizione del filtro. Valori possibili:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obbligatorio per i metodi di valutazione FORMULA) Espressione definita dall'utente utilizzata per valutare le condizioni dei filtri con un'espressione personalizzata. `conditions` (`string`) - (obbligatorio) Elemento radice per le condizioni del filtro dei dati di telemetria che un record in una colonna di dati di telemetria deve soddisfare. L'elemento `conditions` contiene i seguenti elementi: `column` (`string`) - (obbligatorio) Nome della colonna dei dati di telemetria a cui si applicano le condizioni del filtro. `attribute_key` (`string`) - (obbligatorio per le colonne che contengono coppie chiave-valore) Nome della chiave a cui si applica la condizione, all'interno della colonna che contiene coppie chiave-valore. `operator` (`string`) - (obbligatorio) Operatore della condizione del filtro. Valori possibili per le colonne a valore singolo:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Valori possibili per le colonne che contengono coppie chiave-valore:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obbligatorio per i metodi di valutazione FORMULA) ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. `value` (`string`) - (obbligatorio per le operazioni EQUAL, NOT_EQUAL, LIKE e NOT LIKE) Valore con cui effettuare il confronto.
tags		Elemento radice per i tag dell'item.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
triggers		Elemento radice per i trigger degli item del template.

Note:
Vedere anche: Oggetto item (la proprietà pertinente con un nome corrispondente).

Passi di preprocessing del valore dell'item del template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Element	Type	Description
type	string	(obbligatorio) Il tipo di passo di preprocessing del valore dell'item. Valori possibili:[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, calcolato: valore ricevuto - valore precedente), CHANGE_PER_SECOND (10, calcolato: (valore ricevuto - valore precedente)/(tempo attuale - tempo dell'ultimo controllo)), 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		(obbligatorio) Elemento radice per i parametri del passo di preprocessing del valore dell'item.
parameter	string	Parametro individuale del passo di preprocessing del valore dell'item.
error_handler	string	Tipo di azione usato in caso di errore del passo di preprocessing. Valori possibili:[1] ORIGINAL_ERROR (0, predefinito), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parametri del gestore degli errori.

Note:
Vedi anche: Oggetto di preprocessing dell'item (fare riferimento alla proprietà pertinente con lo stesso nome).

Trigger degli item del template

```
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	Description
uuid	string	(required) Identificatore univoco per questo trigger.
expression	string	(required) Espressione del trigger.
recovery_mode	string	Base per la generazione degli eventi OK. Valori possibili:[1] EXPRESSION (0, predefinito), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Espressione di recupero del trigger.
correlation_mode	string	Modalità di correlazione (nessuna correlazione degli eventi oppure correlazione degli eventi per tag). Valori possibili:[1] DISABLED (0, predefinito), TAG_VALUE (1).
correlation_tag	string	Nome del tag da usare per la correlazione degli eventi.
name	string	(required) Nome del trigger.
event_name	string	Nome dell'evento.
opdata	string	Dati operativi.
url_name	string	Etichetta per l'URL associato al trigger.
url	string	URL associato al trigger.
status	string	Stato del trigger. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
priority	string	Gravità del trigger. Valori possibili:[1] NOT_CLASSIFIED (0, predefinito), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Descrizione del trigger.
type	string	Tipo di generazione dell'evento (singolo evento di problema o più eventi di problema). Valori possibili:[1] SINGLE (0, predefinito), MULTIPLE (1).
manual_close	string	Chiusura manuale degli eventi di problema. Valori possibili:[1] NO (0, predefinito), YES (1).
dependencies		Elemento radice per le dipendenze.
name	string	(required) Nome del trigger di dipendenza.
expression	string	(required) Espressione del trigger di dipendenza.
recovery_expression	string	Espressione di recupero del trigger di dipendenza.
tags		Elemento radice per i tag del trigger.
tag	string	(required) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Trigger object (fare riferimento alla proprietà pertinente con lo stesso nome).

Regole di discovery a basso livello del template

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'The discovery of network interfaces.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
La maggior parte degli elementi delle regole di discovery a basso livello del template è la stessa degli item del template.
La tabella seguente descrive gli elementi che differiscono dagli item del template.

Element	Type	Description
type	string	Tipo di item. Valori possibili:[1] ZABBIX_PASSIVE (0, predefinito), 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	(obbligatorio) La chiave della regola di discovery a basso livello.
filter		Elemento radice per i filtri delle regole di discovery a basso livello del template.
lifetime	string	Periodo di tempo (usando secondi, suffisso di tempo, o macro utente), dopo il quale le risorse non più scoperte verranno eliminate. Predefinito: 7d.
lifetime_type	string	Scenario per l'eliminazione delle risorse LLD perse. Valori possibili: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Periodo di tempo (usando secondi, suffisso di tempo, o macro utente), dopo il quale le risorse non più scoperte verranno disabilitate.
enabled_lifetime_type	string	Scenario per la disabilitazione delle risorse LLD perse. Valori possibili: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Elemento radice per gli elementi prototype del template, che sono gli stessi degli item del template.
trigger_prototypes		Elemento radice per gli elementi prototype dei trigger del template, che sono gli stessi dei trigger degli item del template.
graph_prototypes		Elemento radice per gli elementi prototype dei grafici del template, che sono gli stessi dei grafici dell'host.
host_prototypes		Elemento radice per gli elementi prototype degli host del template, che sono gli stessi degli host.
parent_discovery_rule		Elemento radice per la regola di discovery a basso livello padre (o rule prototype) del prototype della regola di discovery a basso livello. Questa proprietà indica che si tratta di un prototype di regola LLD, figlio diretto della regola/rule prototype referenziata.
key	string	(obbligatorio) La chiave della regola di discovery a basso livello padre (o rule prototype).
master_item	string	(obbligatorio per le regole DEPENDENT) Elemento radice per il master item della regola dipendente.
lld_macro_paths		Elemento radice per i percorsi delle macro della regola di discovery a basso livello.
lld_macro	string	(obbligatorio) Nome della macro della regola di discovery a basso livello.
path	string	(obbligatorio) Selettore per il valore, che verrà assegnato alla macro corrispondente.
preprocessing		Elemento radice per il preprocessing del valore della regola di discovery a basso livello.
step		Elemento radice per gli elementi step del preprocessing del valore della regola di discovery a basso livello, che sono gli stessi degli step di preprocessing del valore dell'item del template, ma con meno valori possibili. Vedi anche: oggetto preprocessing della regola LLD.
overrides		Elemento radice per le regole di override della regola di discovery a basso livello.
name	string	(obbligatorio) Nome univoco dell'override.
step	string	(obbligatorio) Numero d'ordine univoco dell'override.
stop	string	Interrompi l'elaborazione degli override successivi se corrisponde.
filter		Elemento radice per gli elementi del filtro della regola di override della regola di discovery a basso livello del template, che sono gli stessi dei filtri delle regole di discovery a basso livello del template.
operations		Elemento radice per le operazioni di override della regola di discovery a basso livello del template.

Note:
Vedi anche: oggetto regola LLD (fare riferimento alla proprietà pertinente con il nome corrispondente).

Filtri delle regole di discovery low-level del template

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
```

```
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
        - uuid: 2bbdc79f082d4c618e01bec625e9c90a
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.MATCHES}'
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
              - macro: '{#DEVTYPE}'
                value: disk
          (...)
```

Element	Type	Description
evaltype	string	Sovrascrive il metodo di valutazione delle condizioni del filtro. Valori possibili:[1] AND_OR (0, predefinito), AND (1), OR (2), FORMULA (3).
formula	string	Formula di calcolo personalizzata per le condizioni del filtro.
conditions		Elemento radice per le condizioni del filtro.
macro	string	(obbligatorio) Il nome della macro di discovery low-level su cui eseguire il controllo.
value	string	Valore da confrontare.
operator	string	Operatore della condizione. Valori possibili:[1] MATCHES_REGEX (8, predefinito), NOT_MATCHES_REGEX (9).
formulaid	string	(obbligatorio) ID univoco arbitrario usato per fare riferimento a una condizione dall'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente quando modifica le condizioni del filtro, ma verrà generato nuovamente nelle richieste successive.

Note:
Vedi anche: Oggetto filtro della regola LLD (fare riferimento alla proprietà pertinente con lo stesso nome).

Operazioni di override della regola di discovery low-level del template

Element	Type	Description
operationobject	string	Oggetto a cui viene applicata l'operazione. Valori possibili:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	Operatore della condizione di override. Valori possibili:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Un'espressione regolare o una stringa per l'operatore della condizione di override.
status	string	Stato dell'oggetto dopo l'operazione di override.
discover	string	Indica se l'oggetto viene aggiunto come risultato della discovery.
delay	string	Intervallo di aggiornamento impostato per il prototipo dell'item dopo l'operazione di override.
history	string	Periodo di conservazione della history impostato per il prototipo dell'item dopo l'operazione di override.
trends	string	Periodo di conservazione dei trend impostato per il prototipo dell'item dopo l'operazione di override.
severity	string	Gravità del prototipo del trigger impostata dopo l'operazione di override.
tags		Elemento radice per i tag impostati per l'oggetto dopo l'operazione di override.
tag	string	(obbligatorio) Nome del tag.

Element	Type	Description
value	string	Valore del tag.
templates		Elemento radice per i template collegati al prototipo dell'host dopo l'operazione di override.
name	string	(obbligatorio) Nome del template.
inventory_mode	string	Modalità di inventory del prototipo dell'host impostata dopo l'operazione di override.

Note:
Vedi anche: LLD rule override operation object (fare riferimento alla proprietà pertinente con lo stesso nome).

Scenari web del template

Element	Type	Description
uuid	string	(required) Identificatore univoco per questo scenario web.
name	string	(required) Nome dello scenario web.
delay	string	Frequenza (in secondi, time suffix, o user macro) di esecuzione dello scenario web. Predefinito: 1m.
attempts	integer	Numero di tentativi per l'esecuzione dei passaggi dello scenario web. Valori possibili:[1] 1-10 (predefinito: 1).
agent	string	Client agent. Zabbix si comporterà come il browser selezionato. Questo è utile quando un sito restituisce contenuti diversi per browser diversi. Predefinito: Zabbix.
http_proxy	string	Proxy che verrà usato dallo scenario web, specificato come: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento radice per le variabili dello scenario web che possono essere usate nei passaggi dello scenario.
name	string	(required) Nome della variabile.
value	text	(required) Valore della variabile.
headers		Elemento radice per le intestazioni HTTP da inviare durante l'esecuzione di una richiesta. Le intestazioni devono essere elencate usando la stessa sintassi con cui apparirebbero nel protocollo HTTP.
name	string	(required) Nome dell'intestazione.
value	text	(required) Valore dell'intestazione.
status	string	Stato dello scenario web. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
authentication	string	Metodo di autenticazione. Valori possibili:[1] NONE (0, predefinito), BASIC (1), NTLM (2), Kerberos (3), o Digest (4).
http_user	string	Nome utente usato per l'autenticazione BASIC (HTTP), NTLM, Kerberos o Digest.
http_password	string	Password usata per l'autenticazione BASIC (HTTP), NTLM, Kerberos o Digest.
verify_peer	string	Verifica il certificato SSL del server web. Valori possibili:[1] NO (0, predefinito), YES (1).
verify_host	string	Verifica che il campo *Common Name* o il campo *Subject Alternate Name* del certificato del server web corrisponda. Valori possibili:[1] NO (0, predefinito), YES (1).
ssl_cert_file	string	Nome del file del certificato SSL usato per l'autenticazione del client (deve essere in formato PEM).
ssl_key_file	string	Nome del file della chiave privata SSL usato per l'autenticazione del client (deve essere in formato PEM).
ssl_key_password	string	Password del file della chiave privata SSL.
steps		Elemento radice obbligatorio per i passaggi dello scenario web del template.
tags		Elemento radice per i tag dello scenario web.
tag	string	(required) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Oggetto scenario web (fare riferimento alla proprietà pertinente con lo stesso nome).

Passi dello scenario web del template

Element	Type	Description
name	string	(required) Nome del passo dello scenario web.
url	string	(required) URL da monitorare.
query_fields		Elemento radice per i parametri di query (un array di campi HTTP da aggiungere all'URL durante l'esecuzione di una richiesta).
name	string	(required) Nome del parametro di query.
value	string	Valore del parametro di query.
posts		Elemento radice per le variabili HTTP POST (una stringa (raw post data) o un array di campi HTTP (dati dei campi del modulo)).
name	string	(required) Nome del campo POST.
value	string	(required) Valore del campo POST.
variables		Elemento radice delle variabili a livello di passo (macro) che devono essere applicate dopo questo passo. Se il valore della variabile ha un prefisso 'regex:', il suo valore viene estratto dai dati restituiti da questo passo in base al pattern di espressione regolare che segue il prefisso 'regex:'
name	string	(required) Nome della variabile.
value	text	(required) Valore della variabile.
headers		Elemento radice per le intestazioni HTTP da inviare durante l'esecuzione di una richiesta.
name	string	(required) Nome dell'intestazione.
value	text	(required) Valore dell'intestazione.
follow_redirects	string	Segui i redirect HTTP. Valori possibili:[1] NO (0), YES (1, predefinito).
retrieve_mode	string	Modalità di recupero della risposta HTTP. Valori possibili:[1] BODY (0, predefinito), HEADERS (1), BOTH (2).
timeout	string	Timeout (in secondi, time suffix, o user macro) dell'esecuzione del passo. Predefinito: 15s.
required	string	Testo che deve essere presente nella risposta (ignorato se vuoto).
status_codes	string	Un elenco delimitato da virgole dei codici di stato HTTP accettati (ad esempio, 200-201,210-299; ignorato se vuoto).

Note:
Vedi anche: Web scenario step object (fare riferimento alla proprietà pertinente con il nome corrispondente).

Configurazione delle macro del template

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Secondi dall'ultimo agent Zabbix rilevato'
            description: 'Timeout dopo il quale l'agent viene considerato non disponibile.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'Soglia di utilizzo della CPU espressa'
            description: 'Soglia critica di utilizzo della CPU espressa in %. Nell'intervallo da 0 a 100 i
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
```

```
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Controllo delle interfacce'
            description: 'Genera un trigger quando lo stato operativo dell'interfaccia cambia in "Link dow
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Elemento	Tipo	Descrizione
type	string	(obbligatorio) Tipo del campo di input della macro. Valori possibili:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	Posizione della macro nell'elenco delle macro.
section_name	string	Etichetta della sezione comprimibile in cui la macro è raggruppata.
label	string	(obbligatorio per le macro TEXT, LIST, CHECKBOX) Etichetta della macro.
description	text	Testo di aiuto della macro. Supporta la formattazione Markdown.
required	string	Indica che la macro è obbligatoria. Valori possibili:[1] NO (0), YES (1). Supportato per le macro TEXT e LIST.
regex	string	Espressione regolare per convalidare l'input dell'utente in un campo di testo. Supportato per le macro TEXT.
options		Elemento radice per gli elementi LIST o i valori CHECKBOX.
value	string	(obbligatorio per le macro LIST) Valore dell'elemento LIST.
text	string	(obbligatorio per le macro LIST) Testo dell'elemento LIST.
checked	string	(obbligatorio per le macro CHECKBOX) Valore che rappresenta uno stato selezionato. Valori possibili:[1] 0 (false), 1 (true). Supportato per le macro CHECKBOX.
unchecked	string	(obbligatorio per le macro CHECKBOX) Valore che rappresenta uno stato non selezionato. Valori possibili:[1] 0 (false), 1 (true). Supportato per le macro CHECKBOX.

Note:
Vedi anche: Oggetto di configurazione delle macro (fare riferimento alla proprietà pertinente con il nome corrispondente).

Dashboard del template

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Elemento	Tipo	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo dashboard.
name	string	(obbligatorio) Nome del dashboard del template.
display pe-riod	integer	Periodo di visualizzazione delle pagine del dashboard.
auto_start	string	Avvio automatico della presentazione. Valori possibili:[1] NO (0), YES (1, predefinito).
pages		Elemento radice per le pagine del dashboard del template.
name	string	Nome della pagina.

Elemento	Tipo	Descrizione
display period	integer	Intervallo della presentazione.
sortorder	integer	Ordine di ordinamento della pagina.
widgets		Elemento radice per i widget del dashboard del template.

Note:
Vedi anche: Oggetto dashboard del template (fare riferimento alla proprietà pertinente con lo stesso nome).

Widget della dashboard del template

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                    - type: STRING
                      name: reference
                      value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Description
type	string	(required) Tipo di widget.
name	string	Nome del widget.
x	integer	Posizione orizzontale dal lato sinistro della dashboard del template. Valori possibili:[1] 0-71.
y	integer	Posizione verticale dalla parte superiore della dashboard del template. Valori possibili:[1] 0-63.

Element	Type	Description
width	integer	Larghezza del widget. Valori possibili:[1] 1-72.
height	integer	Altezza del widget. Valori possibili:[1] 1-64.
hide_header	string	Nascondi l'intestazione del widget. Valori possibili:[1] NO (0, predefinito), YES (1).
fields		Elemento radice per i campi del widget della dashboard del template.
type	string	(required) Tipo di campo del widget. Valori possibili:[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	(required) Nome del campo del widget.
value	mixed	(required) Valore del campo del widget, a seconda del tipo di campo.

Note:
Vedi anche: Oggetto widget della dashboard del template (fare riferimento alla proprietà pertinente con lo stesso nome).

Mappe valori del template

```
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
        (...)
```

Elemento	Tipo	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questa mappa valori.
name	string	(obbligatorio) Nome della mappa valori.
mapping		Elemento radice per le mappature.
type	string	Tipo di corrispondenza della mappatura. Valori possibili:[1] EQUAL (0, predefinito), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Valore originale.
newvalue	string	(obbligatorio) Valore a cui viene mappato il valore originale.

Note:
Vedi anche: Oggetto value map (fare riferimento alla proprietà pertinente con lo stesso nome).

Note a piè di pagina [1] I valori interi dell'API tra parentesi, ad esempio ENABLED (0), sono menzionati solo come riferimento. Per ulteriori informazioni, vedere la pagina dell'oggetto API collegata nella voce della tabella o alla fine di ciascuna sezione.

Host

Panoramica

Gli host vengono esportati con molti oggetti correlati e relazioni tra oggetti.

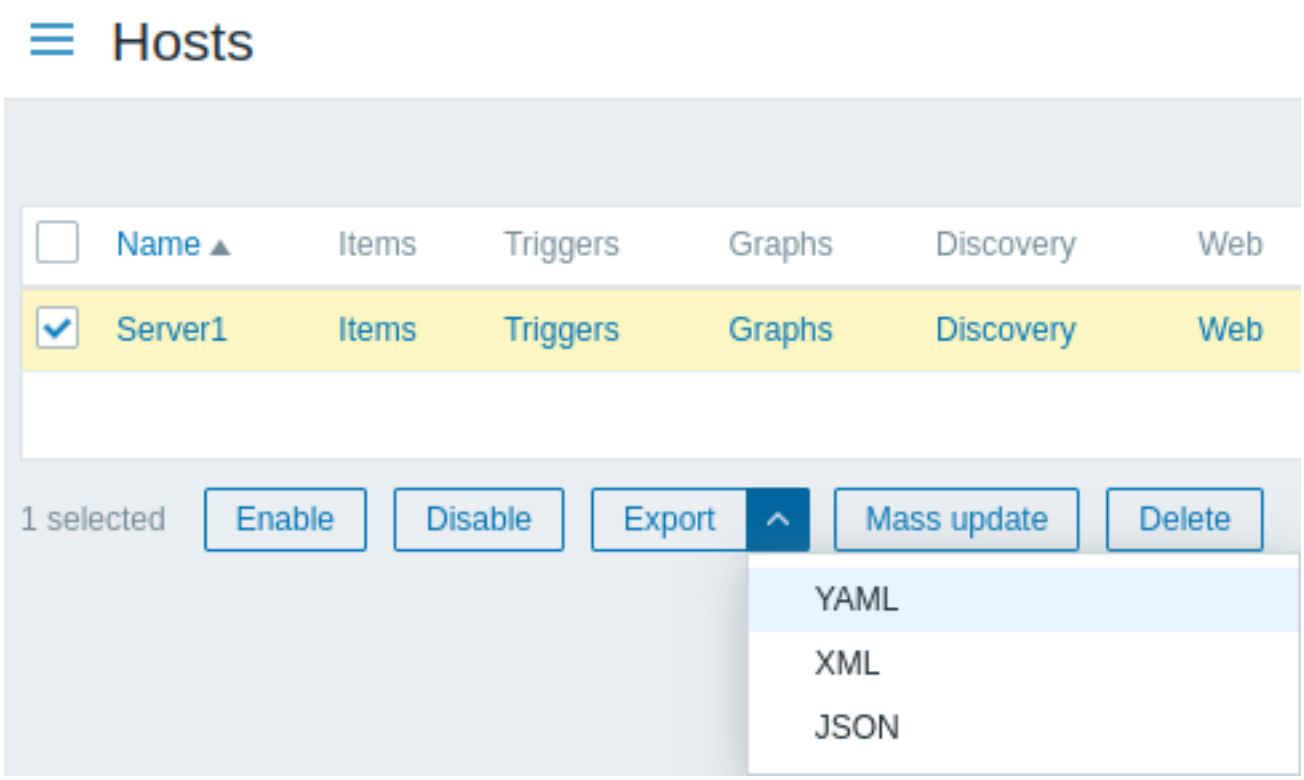
L'esportazione dell'host contiene:

- [host group] collegati (/manual/config/xml_export_import/hostgroups)
- Dati dell'host
- Collegamento ai template
- Collegamento ai gruppi di host
- Interfacce dell'host
- item collegati direttamente
- trigger collegati direttamente
- Regole di discovery collegate direttamente con tutti i prototipi
- Scenari web collegati direttamente
- Macro dell'host
- Dati di inventario dell'host
- Mappe dei valori
- grafici collegati (/manual/config/xml_export_import/hosts#host-graphs)

Esportazione

Per esportare gli host, procedere come segue:

1. Andare in *Data collection* > *Hosts*.
2. Selezionare le caselle di controllo degli host da esportare.
3. Fare clic su *Export* sotto l'elenco per esportare nel formato YAML predefinito, oppure fare clic sulla freccia per selezionare un formato di esportazione diverso.



A seconda del formato selezionato, gli host vengono esportati in un file locale con un nome predefinito:

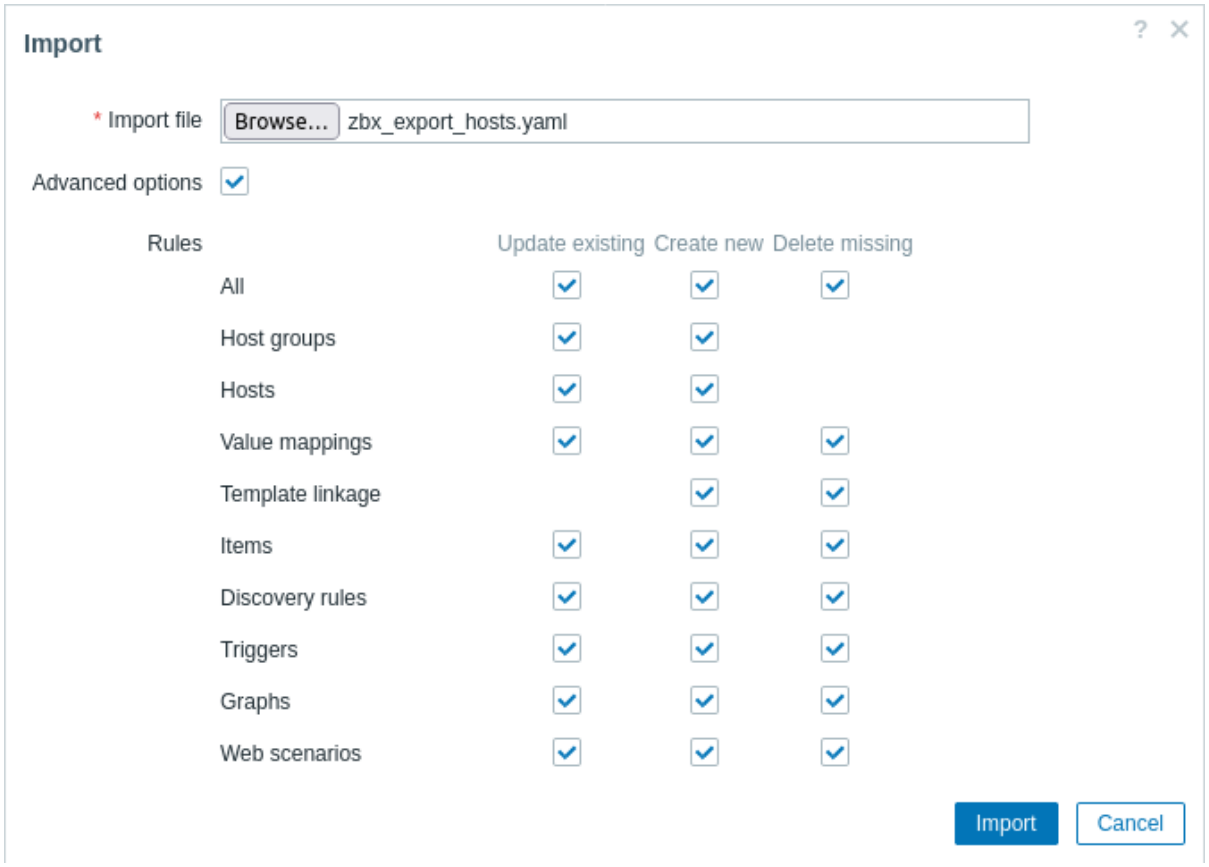
- `zabbix_export_hosts.yaml` - nell'esportazione YAML;
- `zabbix_export_hosts.xml` - nell'esportazione XML;
- `zabbix_export_hosts.json` - nell'esportazione JSON.

Importazione

Per importare gli host, procedere come segue:

1. Andare a *Raccolta dati* > *Host*.
2. Fare clic su *Importa* nell'angolo in alto a destra.

3. Selezionare il file di importazione.
4. Selezionare le caselle di controllo delle regole di importazione da applicare (vedere di seguito).
5. Fare clic su *Importa* nell'angolo in basso a destra del modulo di configurazione.



Se si seleziona la casella di controllo *Opzioni avanzate*, verrà visualizzato un elenco dettagliato di tutti gli elementi importabili: selezionare o deselezionare ogni regola di importazione secondo necessità.

Se si fa clic sulla casella di controllo nella riga *Tutti*, tutti gli elementi sottostanti verranno selezionati o deselezionati.

Regole di importazione:

Regola	Descrizione
Aggiorna esistenti	Gli elementi esistenti verranno aggiornati utilizzando i dati del file di importazione. In caso contrario, non verranno aggiornati.
Crea nuovi	I nuovi elementi verranno creati utilizzando i dati del file di importazione. In caso contrario, non verranno creati.
Elimina mancanti	Gli elementi esistenti non presenti nel file di importazione verranno rimossi. In caso contrario, non verranno rimossi. Se *Elimina mancanti* è selezionato per *Collegamento template*, i collegamenti ai template attuali non presenti nel file di importazione verranno rimossi. Le entità (item, trigger, grafici, ecc.) ereditate dai template scollegati non verranno rimosse, a meno che l'opzione *Elimina mancanti* non sia selezionata anche per ciascuna entità.

Non è possibile esaminare le modifiche agli host durante l'importazione. Nel frontend verrà visualizzato un messaggio che indica se l'importazione è riuscita o non riuscita.

Formato di esportazione

Quando un host viene esportato, Zabbix genera un formato strutturato YAML, JSON o XML. L'esportazione include elementi dell'host, come i metadati dell'host, gli item, le macro, i trigger e altro ancora.

Ogni elemento svolge uno scopo specifico e può contenere elementi nidificati.

Le sezioni seguenti descrivono ciascun elemento nel formato di esportazione. Gli esempi usano un host con il template MySQL by Zabbix agent 2. A scopo dimostrativo degli elementi, il template viene scollegato dopo la creazione dell'host. Lo scollegamento mantiene tutti gli elementi nella configurazione (a differenza di scollegare e cancellare, che li rimuove).

Un'ellissi `(...)` indica elementi omessi per brevità. La nota `(see table below)` viene usata quando l'elemento è spiegato in modo più dettagliato nelle sezioni successive.

```
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (see table below)
  graphs: (see table below)
```

Metadati dell'host

Element		Type	Description
version		string	(required) Versione di Zabbix.
host_groups			Elemento radice per i gruppi di host.
	uuid	string	(required) Identificatore univoco per questo gruppo di host.
	name	string	(required) Nome del gruppo di host.
hosts			Elemento radice per hosts.
graphs			Elemento radice per grafici dell'host.

Host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'Password dell'utente MySQL.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(required) Nome host univoco.
name	string	Nome host visibile.
description	text	Descrizione dell'host.
monitored_by	string	Come viene monitorato l'host. Valori possibili:[1] SERVER (0, predefinito), PROXY (1) o PROXY_GROUP (2).
proxy		Elemento radice per il proxy.
name	string	(required) Nome del proxy (se presente) che monitora l'host.
proxy_group		Elemento radice per il gruppo di proxy.
name	string	(required) Nome del gruppo di proxy (se presente) utilizzato per monitorare l'host.
status	string	Stato dell'host. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
ipmi_authtype	string	Tipo di autenticazione della sessione IPMI. Valori possibili:[1] DEFAULT (-1, predefinito), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).

Element	Type	Description
ipmi_privilege	string	Livello di privilegio della sessione IPMI. Valori possibili:[1] CALLBACK (1), USER (2, predefinito), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Nome utente per i controlli IPMI.
ipmi_password	string	Password per i controlli IPMI.
templates		Elemento radice per i template collegati.
name	string	(required) Nome del template.
groups		Elemento radice per i gruppi host a cui appartiene l'host.
name	string	(required) Nome del gruppo host.
interfaces		Elemento radice per le interfacce host.
items		Elemento radice per gli item host.
discovery_rules		Elemento radice per le regole di discovery low-level dell'host.
http_tests		Elemento radice per gli scenari web dell'host.
tags		Elemento radice per i tag dell'host.
tag	string	(required) Nome del tag.
value	string	Valore del tag.
macros		Elemento radice per le macro dell'host.
macro	string	(required) Nome della macro utente.
type	string	Tipo di macro utente. Valori possibili:[1] TEXT (0, predefinito), SECRET_TEXT (1), VAULT (2).
value	string	Valore della macro utente.
description	string	Descrizione della macro utente.
inventory		Elemento radice per l'inventario dell'host.
<inventory_property>	string	Proprietà dell'inventario. Tutte le proprietà hanno il rispettivo elemento (`type`, `name`, `os`, ecc.; ad esempio, vedere Formato di esportazione).
inventory_mode	string	Modalità inventario. Valori possibili:[1] DISABLED (-1), MANUAL (0, predefinito), AUTOMATIC (1).
valuemaps		Elemento radice per le mappe dei valori dell'host.

Note:
Vedi anche: Oggetto Host (fare riferimento alla proprietà pertinente con il nome corrispondente).

Interfacce host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	Description
default	string	Indica se questa è l'interfaccia host primaria. Nota che può esserci una sola interfaccia primaria di un tipo su un host. Valori possibili:[1] NO (0), YES (1, predefinito).
type	string	Tipo di interfaccia. Valori possibili:[1] ZABBIX (1, predefinito), SNMP (2), IPMI (3), JMX (4).
useip	string	Indica se usare l'IP come interfaccia per la connessione all'host (in caso contrario, verrà usato DNS). Valori possibili:[1] NO (0), YES (1, predefinito).
ip	string	(obbligatorio per connessioni IP) Indirizzo IP (IPv4 o IPv6).
dns	string	(obbligatorio per connessioni DNS) Nome DNS.
port	string	Numero di porta.
details		Elemento radice per i dettagli dell'interfaccia.
version	string	Usa questa versione SNMP. Valori possibili:[1] SNMPV1 (1), SNMP_V2C (2, predefinito), SNMP_V3 (3).
community	string	(obbligatorio per item SNMPv1 e SNMPv2) Community SNMP.

Element	Type	Description
max_repetitions	string	Valore massimo di ripetizione per le richieste bulk SNMP native (PDU GetBulkRequest). Supportato per item SNMPv2 e SNMPv3 (`discovery[]` e `walk[]`). Predefinito: 10.
contextname	string	Nome del contesto SNMPv3. Supportato per item SNMPv3.
securityname	string	Nome di sicurezza SNMPv3. Supportato per item SNMPv3.
securitylevel	string	Livello di sicurezza SNMPv3. Supportato per item SNMPv3. Valori possibili:[1] NOAUTHNOPRIV (0, predefinito), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	Protocollo di autenticazione SNMPv3. Supportato per item SNMPv3. Valori possibili:[1] MD5 (0, predefinito), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	Passphrase di autenticazione SNMPv3. Supportato per item SNMPv3.
privprotocol	string	Protocollo di privacy SNMPv3. Supportato per item SNMPv3. Valori possibili:[1] DES (0, predefinito), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	Passphrase di privacy SNMPv3. Supportato per item SNMPv3.
bulk	string	Usa richieste bulk per SNMP. Valori possibili:[1] NO (0), YES (1, predefinito).
interface_ref	string	Nome di riferimento dell'interfaccia da usare negli item (formato: `if<N>`).

Note:
Vedi anche: Host interface object (fare riferimento alla proprietà pertinente con lo stesso nome).

Item dell'host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Utilizzo del disco della cache binlog'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Numero di transazioni che hanno utilizzato una cache temporanea su disco perché no
          preprocessing: (vedere la tabella seguente)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Utilizzo del buffer pool'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
          params: |
            ( last(//mysql.innodb_buffer_pool_pages_total) -
            last(//mysql.innodb_buffer_pool_pages_free) ) /
            ( last(//mysql.innodb_buffer_pool_pages_total) +
            ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
            ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
```

```
      description: 'Rapporto tra le pagine utilizzate e il totale delle pagine nel buffer pool.'
      tags:
        - tag: component
          value: memory
      triggers: (vedere la tabella seguente)
    (...)
    - name: Uptime
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'Numero di secondi durante i quali il server è stato attivo.'
      preprocessing: (vedere la tabella seguente)
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: (vedere la tabella seguente)
    (...)
```

Elemento	Tipo	Descrizione
name	string	(obbligatorio) Nome dell'item.
type	string	Tipo di item. Valori possibili:[1] ZABBIX_PASSIVE (0, predefinito), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23), TELEMETRY_QUERY (24).
snmp_oid	string	(obbligatorio per gli item SNMP_AGENT) ID dell'oggetto SNMP.
key	string	(obbligatorio) Chiave dell'item.
delay	string	Intervallo di aggiornamento dell'item. Predefinito: 1m. Il valore sarà sempre 0 per gli item TRAP.
history	string	Periodo di tempo (utilizzando un suffisso temporale, una macro utente o una macro LLD) per il quale devono essere conservati i dati della cronologia. Predefinito: 31d.
trends	string	Periodo di tempo (utilizzando un suffisso temporale, una macro utente o una macro LLD) per il quale devono essere conservati i dati dei trend. Predefinito: 365d.
status	string	Stato dell'item. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
value_type	string	Tipo di valore ricevuto. Valori possibili:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, predefinito), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Elenco di indirizzi IP, separati da virgole, degli host autorizzati a inviare dati per l'item. Supportato per gli item TRAP e HTTP_AGENT.
units	string	Unità del valore ricevuto (bps, B, ecc.).
params	text	Parametri aggiuntivi in base al tipo di item (script eseguito per gli item SSH e TELNET; query SQL per gli item ODBC; formula per gli item CALCULATED; script per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER).
ipmi_sensor	string	Sensore IPMI. Supportato per gli item IPMI.
authtype	string	Tipo di autenticazione. Supportato per gli item SSH e HTTP_AGENT. Valori possibili per gli item SSH:[1] PASSWORD (0, predefinito), PUBLIC_KEY (1). Valori possibili per gli item HTTP_AGENT:[1] NONE (0, predefinito), BASIC (1), NTLM (2), Kerberos (3) o Digest (4).
username	string	(obbligatorio per gli item SSH e TELNET) Nome utente per l'autenticazione. Supportato per gli item SIMPLE, ODBC, JMX e HTTP_AGENT. Quando viene utilizzato per gli item JMX, è necessario specificare anche `password` (vedere sotto) oppure lasciare vuoti entrambi gli elementi.

Elemento	Tipo	Descrizione
password	string	(obbligatorio per gli item SSH e TELNET) Password per l'autenticazione. Supportato per gli item SIMPLE, ODBC, JMX e HTTP_AGENT. Quando viene utilizzato per gli item JMX, è necessario specificare anche `username` (vedere sopra) oppure lasciare vuoti entrambi gli elementi.
publickey	string	(obbligatorio per gli item SSH) Nome del file della chiave pubblica.
privatekey	string	(obbligatorio per gli item SSH) Nome del file della chiave privata.
description	text	Descrizione dell'item.
inventory_link	string	Campo dell'inventario dell'host popolato dall'item. Valori possibili:[1] NONE (0), ALIAS (4), ecc. (vedere Inventario dell'host per i campi supportati).
valuemap		Elemento radice per le mappe dei valori dell'item.
name	string	(obbligatorio) Nome della mappa dei valori da utilizzare per l'item.
logtimefmt	string	Formato dell'ora nelle voci di log. Supportato per gli item con tipo di valore LOG.
preprocessing		Elemento radice per la pre-elaborazione dei valori dell'item.
step		Elemento radice per i passaggi di pre-elaborazione dei valori degli item dell'host.
interface_ref	string	Riferimento all'interfaccia dell'host (formato: `if<N>`).
jmx_endpoint	string	Endpoint JMX. Supportato per gli item JMX.
master_item		(obbligatorio per gli item DEPENDENT) Elemento radice per l'item master dell'item dipendente.
key	string	(obbligatorio) Chiave dell'item master dell'item dipendente.
timeout	string	Timeout della richiesta di polling dei dati dell'item. Supportato per l'elenco Timeout dei tipi di item.
url	string	(obbligatorio per gli item HTTP_AGENT) Stringa URL.
query_fields		Elemento radice per i parametri della query. Supportato per gli item HTTP_AGENT.
name	string	(obbligatorio per gli item HTTP_AGENT) Nome del parametro della query.
value	string	Valore del parametro della query. Supportato per gli item HTTP_AGENT.
parameters		Elemento radice per i parametri definiti dall'utente. Supportato per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
name	string	(obbligatorio per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER) Nome del parametro definito dall'utente.
value	string	Valore del parametro definito dall'utente. Supportato per gli item ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
posts	string	Dati del corpo della richiesta HTTP(S). Supportato per gli item HTTP_AGENT.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole. Supportato per gli item HTTP_AGENT.
follow_redirects	string	Segue i reindirizzamenti della risposta durante il polling dei dati. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0), YES (1, predefinito).
post_type	string	Tipo di corpo dei dati post. Supportato per gli item HTTP_AGENT. Valori possibili:[1] RAW (0, predefinito), JSON (2), XML (3).
http_proxy	string	Stringa di connessione al proxy HTTP(S). Supportato per gli item HTTP_AGENT.
headers		Elemento radice per le intestazioni della richiesta HTTP(S). Supportato per gli item HTTP_AGENT.
name	string	(obbligatorio per gli item HTTP_AGENT) Nome dell'intestazione.
value	string	(obbligatorio per gli item HTTP_AGENT) Valore dell'intestazione.
retrieve_mode	string	Parte della risposta da memorizzare. Supportato per gli item HTTP_AGENT. Valori possibili:[1] BODY (0, predefinito), HEADERS (1), BOTH (2).
request_method	string	Tipo di metodo della richiesta. Supportato per gli item HTTP_AGENT. Valori possibili:[1] GET (0, predefinito), POST (1), PUT (2), HEAD (3).
output_format	string	Modalità di elaborazione della risposta. Supportato per gli item HTTP_AGENT. Valori possibili:[1] RAW (0, predefinito), JSON (1).

Elemento	Tipo	Descrizione
allow_traps	string	Consente di popolare il valore nello stesso modo di un item trapper. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Supportato per gli item HTTP_AGENT.
ssl_key_file	string	Percorso del file della chiave SSL privata. Supportato per gli item HTTP_AGENT.
ssl_key_password	string	Password del file della chiave SSL. Supportato per gli item HTTP_AGENT.
verify_peer	string	Indica se verificare che il certificato dell'host sia autentico. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
verify_host	string	Indica se verificare che il nome dell'host della connessione corrisponda a quello presente nel certificato dell'host. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
time_shift	string	Quanto indietro nel tempo, rispetto all'ora corrente, termina l'intervallo temporale interrogato dall'item. Supportato per gli item TELEMETRY_QUERY.
lookback_limit	string	Quanto indietro nel tempo, rispetto all'ora corrente, l'item cerca i dati quando non esiste una query precedente da cui continuare (ad esempio, alla prima esecuzione dell'item) oppure quando la query precedente è più vecchia di `delay` (ad esempio perché Zabbix server era inattivo). Supportato per gli item TELEMETRY_QUERY.
granularity	string	Durata di una porzione in cui viene suddiviso l'intero intervallo temporale della query. Supportato per gli item TELEMETRY_QUERY.
query		(obbligatorio per gli item TELEMETRY_QUERY) Elemento radice per la query di telemetria. Definisce cosa interroga l'item dall'origine dati, come filtra i dati e come aggrega il risultato.
signal_type	string	(obbligatorio) Tipo di segnale interrogato dall'item. Supportato per gli item TELEMETRY_QUERY. Valori possibili:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obbligatorio per il tipo di segnale METRICS) Tipo di punto metrica interrogato dall'item. Supportato per gli item TELEMETRY_QUERY. Valori possibili:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obbligatorio) Elemento radice per le colonne dei dati di telemetria interrogate dall'item. Questo elemento contiene i seguenti elementi: `column (string)` - (obbligatorio) Nome della colonna dei dati di telemetria interrogata dall'item. `attribute_key (string)` - (obbligatorio per le colonne che contengono coppie chiave-valore) Nome della chiave interrogata dall'item all'interno della colonna che contiene coppie chiave-valore.
aggregated_columns		(obbligatorio) Elemento radice per le colonne aggregate dei dati di telemetria interrogate dall'item. Questo elemento contiene i seguenti elementi: `column (string)` - (obbligatorio per le funzioni MIN, MAX, AVG, SUM e PERCENTILE) Nome della colonna dei dati di telemetria interrogata dall'item. `function (string)` - (obbligatorio) Funzione applicata dall'item ai record della colonna. Valori possibili:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters (array)` - (obbligatorio per le funzioni PERCENTILE) Percentile calcolato dall'item sui record della colonna. `alias (string)` - (obbligatorio) Nome univoco del valore calcolato, utilizzato per identificarlo nel valore dell'item.

Elemento	Tipo	Descrizione
filter		(obbligatorio) Elemento radice per il filtro della query di telemetria applicato ai record delle colonne nella query. Questo elemento contiene i seguenti elementi: `evaltype` (`string`) - (obbligatorio) Metodo di valutazione della condizione del filtro. Valori possibili:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obbligatorio per i metodi di valutazione FORMULA) Espressione definita dall'utente utilizzata per valutare le condizioni dei filtri con un'espressione personalizzata. `conditions` (`string`) - (obbligatorio) Elemento radice per le condizioni del filtro dei dati di telemetria che un record in una colonna di dati di telemetria deve soddisfare. L'elemento `conditions` contiene i seguenti elementi: `column` (`string`) - (obbligatorio) Nome della colonna dei dati di telemetria a cui si applicano le condizioni del filtro. `attribute_key` (`string`) - (obbligatorio per le colonne che contengono coppie chiave-valore) Nome della chiave a cui si applica la condizione, all'interno della colonna che contiene coppie chiave-valore. `operator` (`string`) - (obbligatorio) Operatore della condizione del filtro. Valori possibili per le colonne a valore singolo:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Valori possibili per le colonne che contengono coppie chiave-valore:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obbligatorio per i metodi di valutazione FORMULA) ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. `value` (`string`) - (obbligatorio per le operazioni EQUAL, NOT_EQUAL, LIKE e NOT LIKE) Valore con cui effettuare il confronto.
tags		Elemento radice per i tag dell'item.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
triggers		Elemento radice per i trigger degli item dell'host.

Note:
Vedere anche: Oggetto item (fare riferimento alla proprietà corrispondente con lo stesso nome).

Passaggi di preprocessing del valore dell'item dell'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	Description
type	string	(obbligatorio) Il tipo di passaggio di preprocessing del valore dell'item. Valori possibili:[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, calcolato: valore ricevuto - valore precedente), CHANGE_PER_SECOND (10, calcolato: (valore ricevuto - valore precedente)/(tempo attuale - tempo dell'ultimo controllo)), 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		(obbligatorio) Elemento radice per i parametri del passaggio di preprocessing del valore dell'item.
parameter	string	Parametro individuale del passaggio di preprocessing del valore dell'item.
error_handler	string	Tipo di azione usato in caso di errore del passaggio di preprocessing. Valori possibili:[1] ORIGINAL_ERROR (0, predefinito), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parametri del gestore degli errori.

Note:
Vedi anche: Oggetto di preprocessing dell'item (fare riferimento alla proprietà pertinente con lo stesso nome).

Trigger dell'item host

```
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 non ha ricevuto alcun dato per gli item negli ultimi 30 minuti.'
              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: 'L'uptime di MySQL è inferiore a 10 minuti.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Element	Type	Description
uuid	string	(required) Identificatore univoco per questo trigger.
expression	string	(required) Espressione del trigger.

Element	Type	Description
recovery_mode	string	Base per la generazione degli eventi OK. Valori possibili:[1] EXPRESSION (0, predefinito), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Espressione di recupero del trigger.
correlation_mode	string	Modalità di correlazione (nessuna correlazione degli eventi oppure correlazione degli eventi per tag). Valori possibili:[1] DISABLED (0, predefinito), TAG_VALUE (1).
correlation_tag	string	Nome del tag da usare per la correlazione degli eventi.
name	string	(required) Nome del trigger.
event_name	string	Nome dell'evento.
opdata	string	Dati operativi.
url_name	string	Etichetta per l'URL associato al trigger.
url	string	URL associato al trigger.
status	string	Stato del trigger. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
priority	string	Gravità del trigger. Valori possibili:[1] NOT_CLASSIFIED (0, predefinito), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Descrizione del trigger.
type	string	Tipo di generazione dell'evento (singolo evento di problema o più eventi di problema). Valori possibili:[1] SINGLE (0, predefinito), MULTIPLE (1).
manual_close	string	Chiusura manuale degli eventi di problema. Valori possibili:[1] NO (0, predefinito), YES (1).
dependencies		Elemento radice per le dipendenze.
name	string	(required) Nome del trigger dipendente.
expression	string	(required) Espressione del trigger dipendente.
recovery_expression	string	Espressione di recupero del trigger dipendente.
tags		Elemento radice per i tag del trigger.
tag	string	(required) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Trigger object (fare riferimento alla proprietà pertinente con lo stesso nome).

Regole di discovery low-level dell'host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Database size.'
              preprocessing:
                - type: DISCARD_UNCHANGED_HEARTBEAT
                  parameters:
                    - 1h
              interface_ref: if1
              tags:
                - tag: component
                  value: storage
```

```
            - tag: database
              value: '{#DATABASE}'
        lld_macro_paths:
          - lld_macro: '{#DATABASE}'
            path: $.Database
        preprocessing:
          - type: DISCARD_UNCHANGED_HEARTBEAT
            parameters:
              - 1d
    (...)
```

Attention:
La maggior parte degli elementi delle regole di discovery low-level dell'host è la stessa degli item dell'host. La tabella seguente descrive gli elementi che differiscono dagli item dell'host.

Element	Type	Description
type	string	Tipo di item. Valori possibili:[1] ZABBIX_PASSIVE (0, predefinito), 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	(obbligatorio) La chiave della regola di discovery low-level.
filter		Elemento radice per i filtri delle regole di discovery low-level dell'host.
lifetime	string	Periodo di tempo (usando secondi, suffisso di tempo o macro utente), dopo il quale le risorse non più scoperte verranno eliminate. Predefinito: 7d.
lifetime_type	string	Scenario per l'eliminazione delle risorse LLD perse. Valori possibili: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Periodo di tempo (usando secondi, suffisso di tempo o macro utente), dopo il quale le risorse non più scoperte verranno disabilitate.
enabled_lifetime_type	string	Scenario per la disabilitazione delle risorse LLD perse. Valori possibili: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Elemento radice per gli elementi item prototype dell'host, che sono gli stessi degli item dell'host.
trigger_prototypes		Elemento radice per i trigger prototype dell'host, che sono gli stessi dei trigger degli item dell'host.
graph_prototypes		Elemento radice per i graph prototype dell'host, che sono gli stessi dei graph dell'host.
host_prototypes		Elemento radice per i host prototype, che sono gli stessi degli host.
parent_discovery_rule		Elemento radice per la regola di discovery low-level padre (o rule prototype) del rule prototype di discovery low-level. Questa proprietà indica che si tratta di un rule prototype LLD, figlio diretto della regola/rule prototype referenziata.
key	string	(obbligatorio) La chiave della regola di discovery low-level padre (o rule prototype).
master_item	string	(obbligatorio per le regole DEPENDENT) Elemento radice per il master item della regola dipendente.
lld_macro_paths		Elemento radice per i percorsi delle macro della regola di discovery low-level.
lld_macro	string	(obbligatorio) Nome della macro della regola di discovery low-level.
path	string	(obbligatorio) Selettore per il valore che verrà assegnato alla macro corrispondente.
preprocessing		Elemento radice per il preprocessing del valore della regola di discovery low-level.
step		Elemento radice per gli elementi del passo di preprocessing del valore della regola di discovery low-level, che sono gli stessi dei passi di preprocessing del valore dell'item dell'host, ma con meno valori possibili. Vedi anche: oggetto di preprocessing della regola LLD.
overrides		Elemento radice per le regole di override della regola di discovery low-level.
name	string	(obbligatorio) Nome univoco dell'override.
step	string	(obbligatorio) Numero d'ordine univoco dell'override.
stop	string	Interrompe l'elaborazione degli override successivi se corrisponde.
filter		Elemento radice per gli elementi del filtro della regola di override della regola di discovery low-level, che sono gli stessi dei filtri delle regole di discovery low-level dell'host.
operations		Elemento radice per le operazioni di override delle regole di discovery low-level dell'host.

Note:
Vedi anche: oggetto della regola LLD (fare riferimento alla proprietà pertinente con il nome corrispondente).

Filtri della regola di discovery low-level dell'host

```
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
          (...)
```

Elemento	Tipo	Descrizione
evaltype	string	Sostituisce il metodo di valutazione delle condizioni del filtro. Valori possibili:[1] AND_OR (0, predefinito), AND (1), OR (2), FORMULA (3).
formula	string	Formula di calcolo personalizzata per le condizioni del filtro.
conditions		Elemento radice per le condizioni del filtro.
macro	string	(obbligatorio) Il nome della macro di discovery low-level su cui eseguire il controllo.
value	string	Valore con cui confrontare.
operator	string	Operatore della condizione. Valori possibili:[1] MATCHES_REGEX (8, predefinito), NOT_MATCHES_REGEX (9).
formulaid	string	(obbligatorio) ID univoco arbitrario usato per fare riferimento a una condizione dall'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente quando modifica le condizioni del filtro, ma verrà generato nuovamente nelle richieste successive.

Note:
Vedi anche: LLD rule filter object (fare riferimento alla proprietà pertinente con lo stesso nome).

Operazioni di override della regola di discovery low-level dell'host

Element	Type	Description
operationobject	string	Oggetto a cui viene applicata l'operazione. Valori possibili:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4).
operator	string	Operatore della condizione di override. Valori possibili:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Un'espressione regolare o una stringa per l'operatore della condizione di override.
status	string	Stato dell'oggetto dopo l'operazione di override.
discover	string	Indica se l'oggetto viene aggiunto come risultato della discovery.
delay	string	Intervallo di aggiornamento impostato per il prototipo di item dopo l'operazione di override.
history	string	Periodo di conservazione della cronologia impostato per il prototipo di item dopo l'operazione di override.
trends	string	Periodo di conservazione dei trend impostato per il prototipo di item dopo l'operazione di override.
severity	string	Gravità del prototipo di trigger impostata dopo l'operazione di override.
tags		Elemento radice per i tag impostati per l'oggetto dopo l'operazione di override.

Element	Type	Description
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
templates		Elemento radice per i template collegati al prototipo di host dopo l'operazione di override.
name	string	(obbligatorio) Nome del template.
inventory_mode	string	Modalità di inventario del prototipo di host impostata dopo l'operazione di override.

Note:
Vedi anche: LLD rule override operation object (fare riferimento alla proprietà pertinente con lo stesso nome).

Scenari web dell'host

Element	Type	Description
uuid	string	(required) Identificatore univoco per questo scenario web.
name	string	(required) Nome dello scenario web.
delay	string	Frequenza (usando secondi, time suffix, o user macro) di esecuzione dello scenario web. Predefinito: 1m.
attempts	integer	Numero di tentativi per l'esecuzione dei passi dello scenario web. Valori possibili: 1-10 (predefinito: 1).
agent	string	Client agent. Zabbix si comporterà come il browser selezionato. Questo è utile quando un sito web restituisce contenuti diversi per browser diversi. Predefinito: Zabbix.
http_proxy	string	Proxy che verrà usato dallo scenario web, indicato come: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento radice per le variabili dello scenario web che possono essere usate nei passi dello scenario.
name	string	(required) Nome della variabile.
value	text	(required) Valore della variabile.
headers		Elemento radice per le intestazioni HTTP da inviare durante l'esecuzione di una richiesta. Le intestazioni devono essere elencate usando la stessa sintassi con cui apparirebbero nel protocollo HTTP.
name	string	(required) Nome dell'intestazione.
value	text	(required) Valore dell'intestazione.
status	string	Stato dello scenario web. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1).
authentication	string	Metodo di autenticazione. Valori possibili:[1] NONE (0, predefinito), BASIC (1), NTLM (2), Kerberos (3), o Digest (4).
http_user	string	Nome utente usato per l'autenticazione BASIC (HTTP), NTLM, Kerberos, o Digest.
http_password	string	Password usata per l'autenticazione BASIC (HTTP), NTLM, Kerberos, o Digest.
verify_peer	string	Verifica il certificato SSL del server web. Valori possibili:[1] NO (0, predefinito), YES (1).
verify_host	string	Verifica che il campo *Common Name* o il campo *Subject Alternate Name* del certificato del server web corrisponda. Valori possibili:[1] NO (0, predefinito), YES (1).
ssl_cert_file	string	Nome del file del certificato SSL usato per l'autenticazione del client (deve essere in formato PEM).
ssl_key_file	string	Nome del file della chiave privata SSL usata per l'autenticazione del client (deve essere in formato PEM).
ssl_key_password	string	Password del file della chiave privata SSL.
steps		(required) Elemento radice per i passi dello scenario web dell'host.
tags		Elemento radice per i tag dello scenario web.
tag	string	(required) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Oggetto scenario web (fare riferimento alla proprietà pertinente con lo stesso nome).

Passi dello scenario web dell'host

Element	Type	Description
name	string	(required) Nome del passo dello scenario web.
url	string	(required) URL da monitorare.
query_fields		Elemento radice per i parametri di query (un array di campi HTTP da aggiungere all'URL durante l'esecuzione di una richiesta).
name	string	(required) Nome del parametro di query.
value	string	Valore del parametro di query.
posts		Elemento radice per le variabili HTTP POST (una stringa (raw post data) o un array di campi HTTP (dati dei campi del modulo)).
name	string	(required) Nome del campo POST.
value	string	(required) Valore del campo POST.
variables		Elemento radice delle variabili a livello di passo (macro) che devono essere applicate dopo questo passo. Se il valore della variabile ha un prefisso 'regex:', il suo valore viene estratto dai dati restituiti da questo passo in base al pattern di espressione regolare che segue il prefisso 'regex:'
name	string	(required) Nome della variabile.
value	text	(required) Valore della variabile.
headers		Elemento radice per le intestazioni HTTP da inviare durante l'esecuzione di una richiesta.
name	string	(required) Nome dell'intestazione.
value	text	(required) Valore dell'intestazione.
follow_redirects	string	Segui i reindirizzamenti HTTP. Valori possibili:[1] NO (0), YES (1, predefinito).
retrieve_mode	string	Modalità di recupero della risposta HTTP. Valori possibili:[1] BODY (0, predefinito), HEADERS (1), BOTH (2).
timeout	string	Timeout (in secondi, time suffix, o user macro) dell'esecuzione del passo. Predefinito: 15s.
required	string	Testo che deve essere presente nella risposta (ignorato se vuoto).
status_codes	string	Elenco separato da virgole dei codici di stato HTTP accettati (ad esempio, 200-201,210-299; ignorato se vuoto).

Note:
Vedi anche: Oggetto del passo dello scenario web (fare riferimento alla proprietà pertinente con il nome corrispondente).

Grafici dell'host

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (see table below)
    (...)
```

Element	Type	Description
uuid	string	Identificatore univoco del grafico.
name	string	(obbligatorio) Nome del grafico.
width	integer	Larghezza del grafico, in pixel. Usato per l'anteprima e per i grafici a torta/esplosi. Valori possibili:[1] 20-65535 (predefinito: 900).
height	integer	Altezza del grafico, in pixel. Usato per l'anteprima e per i grafici a torta/esplosi. Valori possibili:[1] 20-65535 (predefinito: 900).
yaxismin	double	Valore minimo dell'asse Y. Supportato per il valore minimo FIXED dell'asse Y. Predefinito: 0.
yaxismax	double	Valore massimo dell'asse Y. Supportato per il valore minimo FIXED dell'asse X. Predefinito: 0.

Element	Type	Description
show_work_period	string	Evidenzia le ore non lavorative. Supportato per i grafici NORMAL e STACKED. Valori possibili:[1] NO (0), YES (1, predefinito).
show_triggers	string	Visualizza i valori dei trigger semplici come una linea. Supportato per i grafici NORMAL e STACKED. Valori possibili:[1] NO (0), YES (1, predefinito).
type	string	Tipo di grafico. Valori possibili:[1] NORMAL (0, predefinito), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	Visualizza la legenda del grafico. Valori possibili:[1] NO (0), YES (1, predefinito).
show_3d	string	Abilita lo stile 3D. Supportato per i grafici NORMAL e STACKED. Valori possibili:[1] NO (0, predefinito), YES (1).
percent_left	double	Mostra la linea del percentile per l'asse sinistro. Supportato per i grafici NORMAL. Predefinito: 0.
percent_right	double	Mostra la linea del percentile per l'asse destro. Supportato per i grafici NORMAL. Predefinito: 0.
ymin_type_1	string	Valore minimo dell'asse Y. Supportato per i grafici NORMAL e STACKED. Valori possibili:[1] CALCULATED (0, predefinito), FIXED (1), ITEM (2).
ymin_item_1		(obbligatorio se `ymin_type_1` è impostato su `ITEM`) Elemento radice per i dettagli del singolo item.
host	string	(obbligatorio) Host dell'item.
key	string	(obbligatorio) Chiave dell'item.
ymax_type_1	string	Valore massimo dell'asse Y. Supportato per i grafici NORMAL e STACKED. Valori possibili:[1] CALCULATED (0, predefinito), FIXED (1), ITEM (2).
ymax_item_1		(obbligatorio se `ymax_type_1` è impostato su `ITEM`) Elemento radice per i dettagli del singolo item.
host	string	(obbligatorio) Host dell'item.
key	string	(obbligatorio) Chiave dell'item.
graph_items		(obbligatorio) Elemento radice per i host graph items.

Note:
Vedi anche: Graph object (fare riferimento alla proprietà pertinente con lo stesso nome).

Elementi del grafico dell'host

```
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
    (...)
```

Elemento	Tipo	Descrizione
sortorder	integer	Ordine di disegno. Il valore più piccolo viene disegnato per primo. Può essere usato per disegnare linee o aree dietro (o davanti a) un'altra.
drawtype	string	Stile di disegno dell'elemento del grafico. Supportato per i grafici NORMAL. Valori possibili:[1] SINGLE_LINE (0, predefinito), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color	string	Colore dell'elemento (6 simboli, esadecimale).
yaxisside	string	Lato del grafico in cui verrà disegnata la scala Y dell'elemento del grafico. Supportato per i grafici NORMAL e STACKED. Valori possibili:[1] LEFT (0, predefinito), RIGHT (1).
calc_fnc	string	Dati da disegnare se esiste più di un valore per un item. Valori possibili:[1] MIN (1), AVG (2, predefinito), MAX (4), ALL (7; minimo, medio e massimo; supportato per i grafici semplici), LAST (9, supportato per i grafici a torta/esplosi).
type	string	Tipo di elemento del grafico. Valori possibili:[1] SIMPLE (0, predefinito), GRAPH_SUM (2; il valore dell'item rappresenta l'intera torta; supportato per i grafici a torta/esplosi).
item		(obbligatorio) Item individuale.
host	string	(obbligatorio) Host dell'item.
key	string	(obbligatorio) Chiave dell'item.

Note:
Vedi anche: Graph item object (fare riferimento alla proprietà pertinente con lo stesso nome).

Mappe dei valori dell'host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element	Type	Description
uuid	string	(obbligatorio) Identificatore univoco per questa mappa dei valori.
name	string	(obbligatorio) Nome della mappa dei valori.
mapping		Elemento radice per le mappature.
type	string	Tipo di corrispondenza della mappatura. Valori possibili:[1] EQUAL (0, predefinito), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Valore originale.
newvalue	string	(obbligatorio) Valore a cui viene mappato il valore originale.

Note:
Vedi anche: Oggetto value map (fare riferimento alla proprietà corrispondente con lo stesso nome).

Note a piè di pagina [1] I valori interi dell'API tra parentesi, ad esempio ENABLED (0), sono menzionati solo come riferimento. Per ulteriori informazioni, vedere la pagina dell'oggetto API collegata nella voce della tabella o alla fine di ogni sezione.

Mappe di rete

Panoramica

La mappa di rete export contiene:

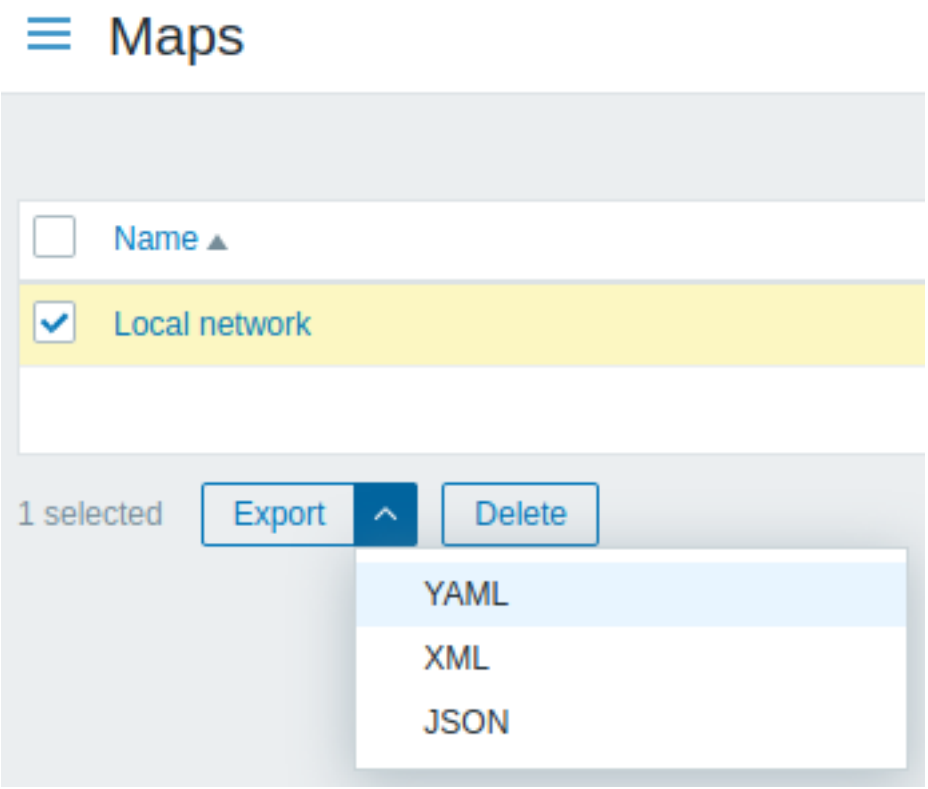
- Tutte le immagini correlate
- La struttura della mappa (tutte le impostazioni della mappa, tutti gli elementi contenuti con le relative impostazioni, i collegamenti della mappa e gli indicatori di stato dei collegamenti della mappa)

Warning:
Eventuali gruppi di host, host, trigger, altre mappe o altri elementi che potrebbero essere correlati alla mappa esportata non vengono esportati. Pertanto, se manca almeno uno degli elementi a cui la mappa fa riferimento, l'importazione non andrà a buon fine.

Esportazione

Per esportare le mappe di rete, procedere come segue:

1. Andare su *Monitoring* > *Maps*.
2. Selezionare le caselle di controllo delle mappe di rete da esportare.
3. Fare clic su *Export* sotto l'elenco per esportare nel formato YAML predefinito, oppure fare clic sulla freccia per selezionare un formato di esportazione diverso.



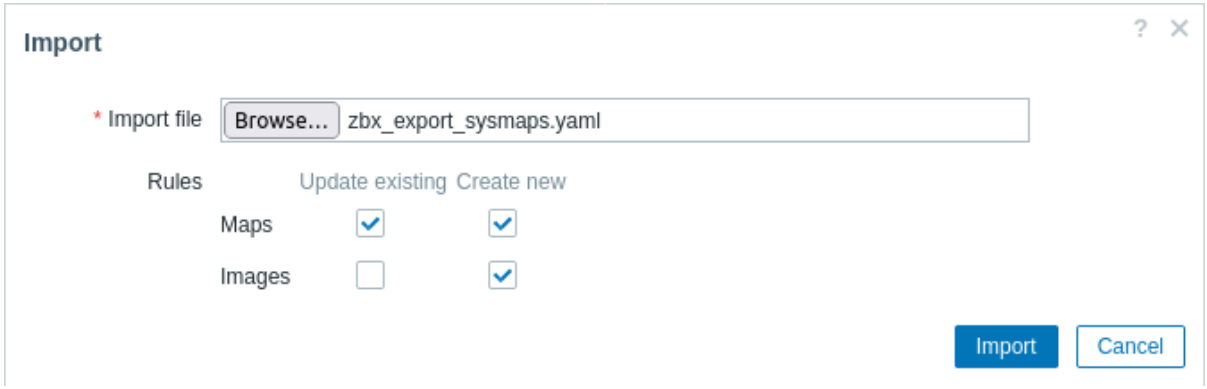
A seconda del formato selezionato, le mappe vengono esportate in un file locale con un nome predefinito:

- `zabbix_export_maps.yaml` - nell'esportazione YAML;
- `zabbix_export_maps.xml` - nell'esportazione XML;
- `zabbix_export_maps.json` - nell'esportazione JSON.

Importazione

Per importare le mappe di rete, procedere come segue:

1. Andare a *Monitoring* > *Maps*.
2. Fare clic su *Import* nell'angolo superiore destro.
3. Selezionare il file di importazione.
4. Selezionare le caselle di controllo delle regole di importazione che si desidera applicare (vedere di seguito).
5. Fare clic su *Import* nell'angolo inferiore destro del modulo di configurazione.



Regole di importazione:

Regola	Descrizione
Update existing	Le mappe esistenti verranno aggiornate utilizzando i dati del file di importazione. In caso contrario, non verranno aggiornate.
Create new	Verranno create nuove mappe utilizzando i dati del file di importazione. In caso contrario, non verranno create.

Se si deselezionano entrambe le opzioni relative alle mappe e si selezionano le rispettive opzioni per le immagini, verranno importate solo le immagini. L'importazione delle immagini è disponibile solo per gli utenti *Super admin*.

Non è possibile esaminare le modifiche alle mappe di rete durante l'importazione. Nel frontend verrà visualizzato un messaggio che indica se l'importazione è riuscita o meno.

Warning:
Se si sostituisce un'immagine esistente, ciò influirà su tutte le mappe che utilizzano questa immagine.

Formato di esportazione

Esporta in YAML:

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
```

```
    label_string_trigger: ''
    label_string_map: ''
    label_string_image: ''
    expand_macros: '1'
    background: {  }
    iconmap: {  }
    urls: {  }
    selements:
      - elementtype: '0'
        elements:
          - host: 'Zabbix server'
        label: |
          {HOST.NAME}
          {HOST.CONN}
        label_location: '0'
        x: '111'
        'y': '61'
        elementsubtype: '0'
        areatype: '0'
        width: '200'
        height: '200'
        viewtype: '0'
        use_iconmap: '0'
        selementid: '1'
        icon_off:
          name: Zabbix_server_3D_(128)
        icon_on: {  }
        icon_disabled: {  }
        icon_maintenance: {  }
        urls: {  }
        evaltype: '0'
    shapes:
      - type: '0'
        x: '0'
        'y': '0'
        width: '680'
        height: '15'
        text: '{MAP.NAME}'
        font: '9'
        font_size: '11'
        font_color: '000000'
        text_halign: '0'
        text_valign: '0'
        border_type: '0'
        border_width: '0'
        border_color: '000000'
        background_color: ''
        zindex: '0'
    lines: {  }
    links: {  }
```

Elementi esportati

Gli elementi esportati sono spiegati nella tabella seguente.

Element	Type	Description
images		Elemento radice per le immagini.
name	string	Nome univoco dell'immagine.
imagetype	integer	Tipo di immagine. Valori possibili: 1 - Image; 2 - Background.
encodedImage	string	Immagine codificata in Base64.

Element	Type	Description
maps		Elemento radice per maps.

Maps

Element	Type	Description
name	string	Nome univoco della mappa.
width	integer	Larghezza della mappa, in pixel.
height	integer	Altezza della mappa, in pixel.
label_type	integer	Tipo di etichetta dell'elemento della mappa. Valori possibili: 0 - Etichetta; 1 - Indirizzo IP dell'host; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Nessuno.
label_location	integer	Posizione predefinita dell'etichetta dell'elemento della mappa. Valori possibili: 0 - In basso; 1 - A sinistra; 2 - A destra; 3 - In alto.
highlight	integer	Abilita l'evidenziazione delle icone per i trigger attivi e gli stati degli host. Valori possibili: 0 - No; 1 - Sì.
expandproblem	integer	Mostra il trigger di problema per gli elementi con un singolo problema. Valori possibili: 0 - No; 1 - Sì.
markelements	integer	Evidenzia gli elementi della mappa che hanno recentemente cambiato stato. Valori possibili: 0 - No; 1 - Sì.
show_unack	integer	Visualizzazione dei problemi. Valori possibili: 0 - Conteggio di tutti i problemi; 1 - Conteggio dei problemi non riconosciuti; 2 - Conteggio separato dei problemi riconosciuti e non riconosciuti.
severity_min	integer	Gravità minima del trigger da mostrare sulla mappa per impostazione predefinita. Valori possibili: 0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro.
show_suppressed	integer	Mostra i problemi che altrimenti sarebbero soppressi (non visualizzati) a causa della manutenzione dell'host. Valori possibili: 0 - No; 1 - Sì.
grid_size	integer	Dimensione della cella della griglia della mappa in pixel. Supportato se `grid_show` è impostato su 0. Valori possibili: 20, 40, 50, 75 o 100.
grid_show	integer	Mostra una griglia nella configurazione della mappa. Valori possibili: 0 - Sì; 1 - No.

Element	Type	Description
grid_align	integer	Allinea automaticamente le icone nella configurazione della mappa. Valori possibili: 0 - Sì; 1 - No.
label_format	integer	Usa la configurazione avanzata delle etichette. Valori possibili: 0 - No; 1 - Sì.
label_type_host	integer	Mostra l'etichetta come etichetta dell'host. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 1 - Indirizzo IP dell'host; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Nessuno; 5 - Etichetta personalizzata.
label_type_hostgroup	integer	Mostra l'etichetta come etichetta del gruppo di host. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Nessuno; 5 - Etichetta personalizzata.
label_type_trigger	integer	Mostra l'etichetta come etichetta del trigger. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Nessuno; 5 - Etichetta personalizzata.
label_type_map	integer	Mostra l'etichetta come etichetta della mappa. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Nessuno; 5 - Etichetta personalizzata.
label_type_image	integer	Mostra l'etichetta come etichetta dell'immagine. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 2 - Nome dell'elemento; 4 - Nessuno; 5 - Etichetta personalizzata.
label_string_host	string	Etichetta personalizzata per gli elementi host. Supportato se `label_type_host` è impostato su 5.
label_string_hostgroup	string	Etichetta personalizzata per gli elementi del gruppo di host. Supportato se `label_type_hostgroup` è impostato su 5.
label_string_trigger	string	Etichetta personalizzata per gli elementi trigger. Supportato se `label_type_trigger` è impostato su 5.
label_string_map	string	Etichetta personalizzata per gli elementi della mappa. Supportato se `label_type_map` è impostato su 5.
label_string_image	string	Etichetta personalizzata per gli elementi immagine. Supportato se `label_type_image` è impostato su 5.
expand_macros	integer	Espande i macro nelle etichette nella configurazione della mappa. Valori possibili: 0 - No; 1 - Sì.

Element		Type	Description
background			Elemento radice per l'immagine di sfondo (se presente). Supportato se `imagetype` è impostato su 2.
	name	string	Nome dell'immagine di sfondo.
iconmap			Elemento radice per la mappatura delle icone (se presente).
	name	string	Nome della mappatura delle icone.
urls			Elemento radice per gli URL usati dalle mappe o da ciascun elemento della mappa.
	name	string	Nome del collegamento.
	url	string	URL del collegamento.
	elementtype	integer	Tipo di item della mappa a cui appartiene il collegamento. Valori possibili: 0 - Host; 1 - Mappa; 2 - Trigger; 3 - Gruppo di host; 4 - Immagine.
selements			Elemento radice per i selements della mappa.
shapes			Elemento radice per le forme della mappa.
	type	integer	Tipo di forma. Valori possibili: 0 - Rettangolo; 1 - Ellisse.
	x	integer	Coordinate X della forma in pixel.
	y	integer	Coordinate Y della forma in pixel.
	width	integer	Larghezza della forma.
	height	integer	Altezza della forma.
	text	string	Testo all'interno della forma.
	font	integer	Stile del carattere del testo. Valori possibili: 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	Dimensione del carattere in pixel.
	font_color	string	Colore del carattere rappresentato in codice esadecimale.
	text_halign	integer	Allineamento orizzontale del testo. Valori possibili: 0 - Centro; 1 - Sinistra; 2 - Destra.
	text_valign	integer	Allineamento verticale del testo. Valori possibili: 0 - Centro; 1 - In alto; 2 - In basso.
	border_type	integer	Tipo di bordo per la forma. Valori possibili: 0 - Nessuno; 1 - Linea spessa; 2 - Linea puntinata; 3 - Linea tratteggiata.
	border_width	integer	Larghezza del bordo in pixel.
	border_color	string	Colore del bordo rappresentato in codice esadecimale.

Element		Type	Description
	background_color	string	Colore di sfondo (riempimento) rappresentato in codice esadecimale.
	zindex	integer	Valore per l'ordinamento di tutte le forme e le linee (z-index).
lines			Elemento radice per le linee della mappa.
	x1	integer	Coordinate X del punto 1 della linea in pixel.
	y1	integer	Coordinate Y del punto 1 della linea in pixel.
	x2	integer	Coordinate X del punto 2 della linea in pixel.
	y2	integer	Coordinate Y del punto 2 della linea in pixel.
	line_type	integer	Tipo di linea. Valori possibili: 0 - Nessuno; 1 - Linea spessa; 2 - Linea puntinata; 3 - Linea tratteggiata.
	line_width	integer	Larghezza della linea in pixel.
	line_color	string	Colore della linea rappresentato in codice esadecimale.
	zindex	integer	Valore per l'ordinamento di tutte le forme e le linee (z-index).
links			Elemento radice per i collegamenti tra gli elementi della mappa.
	drawtype	integer	Stile del collegamento. Valori possibili: 0 - Linea; 2 - Linea spessa; 3 - Linea puntinata; 4 - Linea tratteggiata.
	color	string	Colore del collegamento (6 simboli, esadecimale).
	label	string	Etichetta del collegamento.
	selementid1	id	ID di un elemento da collegare.
	selementid2	id	ID dell'altro elemento da collegare.
linktriggers			Elemento radice per gli indicatori di stato dei collegamenti.

Note:
Vedi anche: Oggetto Map (fare riferimento alla proprietà pertinente con lo stesso nome).

Elementi della mappa

Element	Type	Description
elementtype	integer	Tipo di elemento della mappa. Valori possibili: 0 - Host; 1 - Mappa; 2 - Trigger; 3 - Gruppo host; 4 - Immagine.
elements		Elemento radice per le entità Zabbix (host, gruppo host, mappa, ecc.) rappresentate sulla mappa. Tutte le entità hanno il rispettivo element (`host`, ecc.; vedere, ad esempio, Formato di esportazione).
label	string	Etichetta dell'icona.
label_location	integer	Posizione dell'etichetta. Valori possibili: -1 - Usa il valore predefinito della mappa; 0 - In basso; 1 - A sinistra; 2 - A destra; 3 - In alto.
x	integer	Posizione sull'asse X.
y	integer	Posizione sull'asse Y.

Element	Type	Description
elementsubtype	integer	Sottotipo dell'elemento. Supportato se `elementtype` è impostato su 3. Valori possibili: 0 - Singolo gruppo host; 1 - Tutti i gruppi host.
areatype	integer	Dimensione dell'area. Supportato se `elementtype` è impostato su 1. Valori possibili: 0 - Uguale all'intera mappa; 1 - Dimensione personalizzata.
width	integer	Larghezza dell'area. Supportato se `areatype` è impostato su 1.
height	integer	Altezza dell'area. Supportato se `areatype` è impostato su 1.
viewtype	integer	Algoritmo di posizionamento dell'area. Supportato se `elementsubtype` è impostato su 1. Valori possibili: 0 - Posiziona in modo uniforme nell'area.
use_iconmap	integer	Usa la mappatura delle icone per questo elemento. Rilevante solo se la mappatura delle icone è attivata a livello di mappa. Valori possibili: 0 - No; 1 - Sì.
selementid	id	ID univoco del record dell'elemento.
icon_off		Elemento radice per l'immagine da usare quando l'elemento è nello stato `OK`.
name	string	Nome univoco dell'immagine.
icon_on		Elemento radice per l'immagine da usare quando l'elemento è nello stato `Problem`.
name	string	Nome univoco dell'immagine.
icon_disabled		Elemento radice per l'immagine da usare quando l'elemento è disabilitato.
name	string	Nome univoco dell'immagine.
icon_maintenance		Elemento radice per l'immagine da usare quando l'elemento è in manutenzione.
name	string	Nome univoco dell'immagine.
urls		Elemento radice per gli URL usati dalle mappe o da ciascun elemento della mappa.
name	string	Nome del collegamento.
url	string	URL del collegamento.
evaltype	integer	Tipo di valutazione per i tag.
tags		Elemento radice per i tag dei problemi (per gli elementi host e gruppo host). Se vengono specificati dei tag, sulla mappa verranno visualizzati solo i problemi con questi tag.
tag	string	Nome del tag.
value	string	Valore del tag.
operator	integer	Operatore.

Note:
Vedi anche: Oggetto elemento della mappa (fare riferimento alla proprietà pertinente con lo stesso nome).

Indicatori di stato del collegamento della mappa

Elemento	Tipo	Descrizione
drawtype	integer	Stile del collegamento quando il trigger è nello stato 'Problem'. Valori possibili: 0 - Linea; 2 - Linea in grassetto; 3 - Linea puntinata; 4 - Linea tratteggiata.
color	string	Colore del collegamento (6 simboli, esadecimale) quando il trigger è nello stato 'Problem'.
trigger		Elemento radice per il trigger usato per indicare lo stato del collegamento.
description	string	Nome del trigger.

Elemento	Tipo	Descrizione
expression	string	Espressione del trigger.
recovery_expression	string	Espressione di recupero del trigger.

Note:
Vedi anche: Map link trigger object (fare riferimento alla proprietà pertinente con lo stesso nome).

Tipi di media

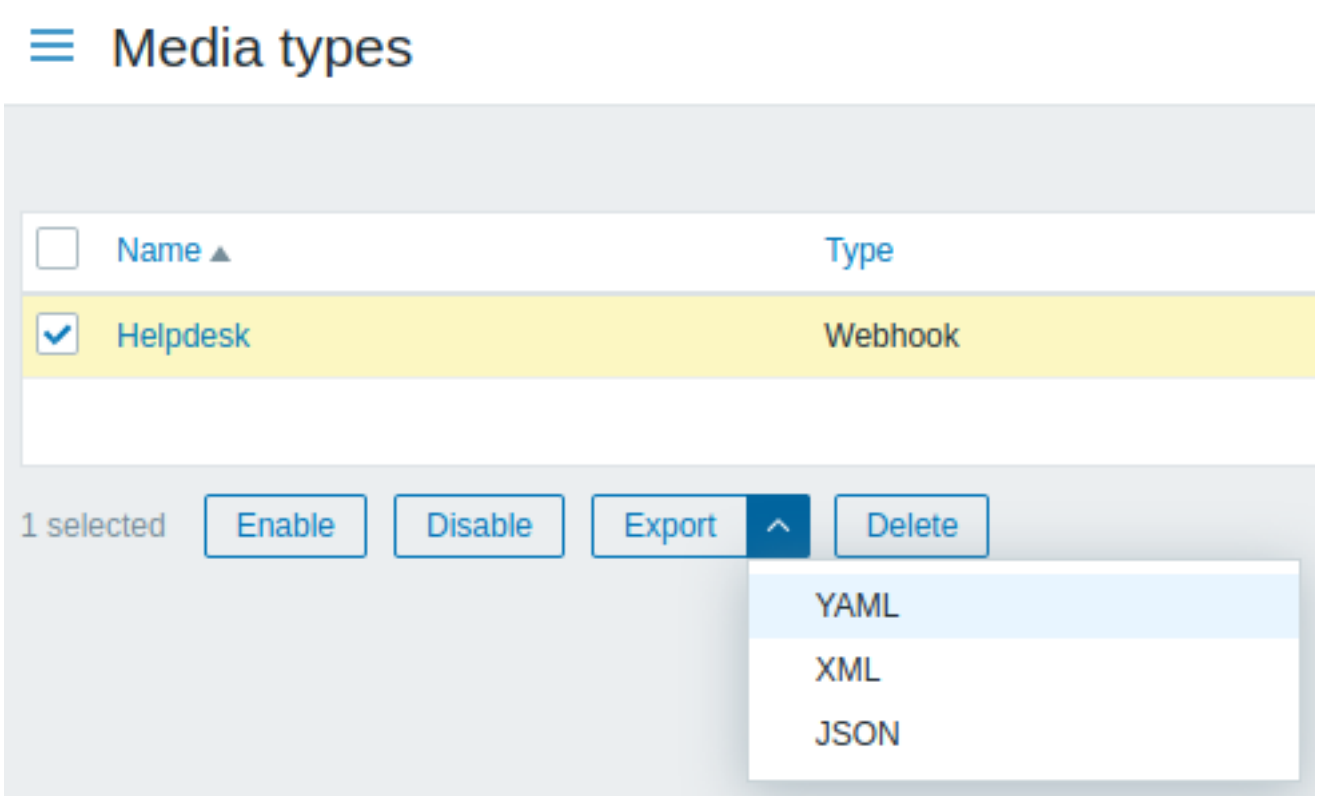
Panoramica

I tipi di media vengono esportati con tutti gli oggetti correlati e le relazioni tra gli oggetti.

Esportazione

Per esportare i tipi di media, procedere come segue:

1. Andare su *Alerts* > *Media types*.
2. Selezionare le caselle di controllo dei tipi di media da esportare.
3. Fare clic su *Export* sotto l'elenco per esportare nel formato YAML predefinito, oppure fare clic sulla freccia per selezionare un formato di esportazione diverso.



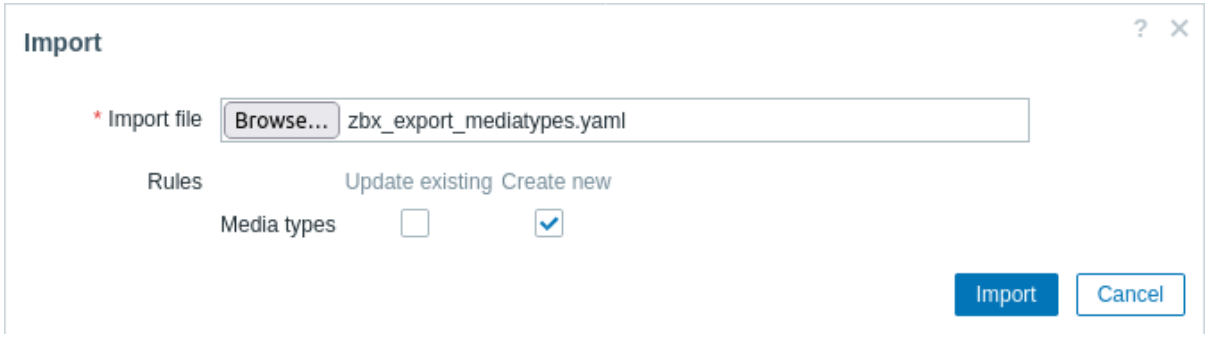
A seconda del formato selezionato, i tipi di media vengono esportati in un file locale con un nome predefinito:

- `zabbix_export_mediatypes.yaml` - nell'esportazione YAML;
- `zabbix_export_mediatypes.xml` - nell'esportazione XML;
- `zabbix_export_mediatypes.json` - nell'esportazione JSON.

Importazione

Per importare i tipi di media, procedere come segue:

1. Andare a *Avvisi* > *Tipi di media*.
2. Fare clic su *Importa* nell'angolo superiore destro.
3. Selezionare il file di importazione.
4. Selezionare le caselle di controllo delle regole di importazione da applicare (vedere di seguito).
5. Fare clic su *Importa* nell'angolo inferiore destro del modulo di configurazione.



Regole di importazione:

Regola	Descrizione
Aggiorna esistenti	Selezionare questa casella di controllo per aggiornare i tipi di media esistenti utilizzando i dati del file di importazione.
Crea nuovi	Selezionare questa casella di controllo per creare nuovi tipi di media utilizzando i dati del file di importazione. Se il file di importazione contiene un tipo di media già esistente (ad esempio, *Email*), non ne verrà creato uno nuovo e il tipo di media esistente non verrà aggiornato.

Non è possibile esaminare le modifiche ai tipi di media durante l'importazione. Nel frontend verrà visualizzato un messaggio che indica l'esito positivo o negativo dell'importazione.

Formato di esportazione

Esporta in 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 'Parametro "event_source" non corretto specificato: "' + params.event_source + '".\n
        }

        if (params.event_value !== '0' && params.event_value !== '1'
            && (params.event_source === '0' || params.event_source === '3')) {
            throw 'Parametro "event_value" non corretto specificato: ' + params.event_value + '\nDeve
        }

        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" deve essere compreso tra -2 e 2';
        }

        if (params.event_source === '0' && isNaN(params.triggerid)) {
            throw 'il campo "triggerid" non è un numero';
```

```
        }

        if (isNaN(params.eventid)) {
            throw 'il campo "eventid" non è un numero';
        }

        if (typeof params.message !== 'string' || params.message.trim() === '') {
            throw 'il campo "message" non può essere vuoto';
        }

        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 'il campo "retry" deve essere un numero con valore almeno 30 se "priority" è imp
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'il campo "expire" deve essere un numero con valore al massimo 10800 se "priorit
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Invio richiesta: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Ricevuta risposta con codice di stato ' + request.getStatu

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Impossibile analizzare la risposta ricevuta da Pus
                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 'Errore sconosciuto. Controlla il log di debug per ulteriori informazioni.';
            }
        }
```

```
        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Notifica Pushover non riuscita: ' + error);
        throw 'Notifica Pushover non riuscita: ' + error;
    }
description: |
  Fare riferimento alla guida di configurazione qui: https://git.zabbix.com/projects/ZBX/repos/zabbi

  Imposta il parametro token con la chiave della tua applicazione Pushover.
  Quando assegni il media Pushover all'utente Zabbix, aggiungi la chiave utente nel campo send to.
message_templates:
  - event_source: TRIGGERS
    operation_mode: PROBLEM
    subject: 'Problema: {EVENT.NAME}'
    message: |
      Il problema è iniziato alle {EVENT.TIME} il {EVENT.DATE}
      Nome del problema: {EVENT.NAME}
      Host: {HOST.NAME}
      Gravità: {EVENT.SEVERITY}
      Dati operativi: {EVENT.OPDATA}
      ID problema originale: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: RECOVERY
    subject: 'Risolto in {EVENT.DURATION}: {EVENT.NAME}'
    message: |
      Il problema è stato risolto alle {EVENT.RECOVERY.TIME} il {EVENT.RECOVERY.DATE}
      Nome del problema: {EVENT.NAME}
      Durata del problema: {EVENT.DURATION}
      Host: {HOST.NAME}
      Gravità: {EVENT.SEVERITY}
      ID problema originale: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: UPDATE
    subject: 'Problema aggiornato in {EVENT.AGE}: {EVENT.NAME}'
    message: |
      {USER.FULLNAME} {EVENT.UPDATE.ACTION} il problema il {EVENT.UPDATE.DATE} alle {EVENT.UPDATE.TI
      {EVENT.UPDATE.MESSAGE}

      Lo stato attuale del problema è {EVENT.STATUS}, l'età è {EVENT.AGE}, confermato: {EVENT.ACK.ST
  - event_source: DISCOVERY
    operation_mode: PROBLEM
    subject: 'Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
    message: |
      Regola di discovery: {DISCOVERY.RULE.NAME}

      IP del dispositivo: {DISCOVERY.DEVICE.IPADDRESS}
      DNS del dispositivo: {DISCOVERY.DEVICE.DNS}
      Stato del dispositivo: {DISCOVERY.DEVICE.STATUS}
      Uptime del dispositivo: {DISCOVERY.DEVICE.UPTIME}

      Nome del servizio del dispositivo: {DISCOVERY.SERVICE.NAME}
      Porta del servizio del dispositivo: {DISCOVERY.SERVICE.PORT}
      Stato del servizio del dispositivo: {DISCOVERY.SERVICE.STATUS}
      Uptime del servizio del dispositivo: {DISCOVERY.SERVICE.UPTIME}
  - event_source: AUTOREGISTRATION
    operation_mode: PROBLEM
    subject: 'Autoregistrazione: {HOST.HOST}'
    message: |
      Nome host: {HOST.HOST}
```

```
            IP host: {HOST.IP}
            Porta agent: {HOST.PORT}
```

Elementi esportati

Gli elementi esportati sono spiegati nella tabella seguente.

Element	Type	Description
name	string	(obbligatorio) Nome del media type.
type	string	(obbligatorio) Trasporto utilizzato dal media type. Valori possibili:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Indica se il media type è abilitato. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1)
max_sessions	integer	Il numero massimo di alert che possono essere elaborati in parallelo. Valori possibili per SMS:[1] 1 (predefinito). Valori possibili per gli altri media type:[1] 0-100 (dove 0 - illimitato).
attempts	integer	Il numero massimo di tentativi per inviare un alert. Valori possibili:[1] 1-10 (predefinito: 3).
attempt_interval	string	L'intervallo (usando secondi o time suffix) tra i tentativi di ripetizione. Valori possibili:[1] 0-60s (predefinito: 10s).
description	string	Descrizione del media type.
message_templates		Elemento radice per i template dei messaggi del media type.
event_source	string	(obbligatorio) Origine dell'evento. Valori possibili:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).
operation_mode	string	Modalità operativa. Valori possibili:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Oggetto del messaggio.
message	string	Corpo del messaggio.

Note:
Vedi anche: Media type object (fare riferimento alla proprietà pertinente con lo stesso nome).

Email

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di media *Email*.

Element	Type	Description
provider	string	Provider email.
smtp_server	string	Server SMTP.
smtp_port	integer	Porta del server SMTP a cui connettersi. Predefinito: 25.
smtp_helo	string	SMTP helo.
smtp_email	string	Indirizzo email da cui verranno inviate le notifiche.
smtp_security	string	Livello di sicurezza della connessione SMTP da utilizzare. Valori possibili:[1] NONE (0, predefinito), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	Verifica SSL dell'host per SMTP. Valori possibili:[1] NO (0, predefinito), YES (1).
smtp_verify_peer	string	Verifica SSL del peer per SMTP. Valori possibili:[1] NO (0, predefinito), YES (1).
smtp_authentication	string	Metodo di autenticazione SMTP da utilizzare. Valori possibili:[1] NONE (0, predefinito), PASSWORD (1), OAUTH (2).
username	string	Nome utente.
password	string	Password di autenticazione.
redirection_url	string	URL del frontend Zabbix a cui reindirizzare il ritorno dell'autorizzazione OAuth.
client_id	string	Identificatore del client registrato nel server di autorizzazione OAuth.
authorization_url	string	URL OAuth, con parametri, per ottenere i token di accesso e di aggiornamento.
token_url	string	URL OAuth per scambiare il token di autorizzazione con i token di accesso e di aggiornamento.

Element	Type	Description
message_format	string	Formato del messaggio. Valori possibili:[1] TEXT (0), HTML (1, predefinito).

Note:
Vedi anche: Media type object (fare riferimento alla proprietà pertinente con il nome corrispondente).

SMS

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di media *SMS*.

Element	Type	Description
gsm_modem	string	(required) Nome del dispositivo seriale del modem GSM.

Note:
Vedi anche: Media type object (fare riferimento alla proprietà pertinente con lo stesso nome).

Script

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di media *Script*.

Element	Type	Description
script name	string	(required) Nome dello script.
parameters		Elemento radice per i parametri dello script.
sortorder	string	(required) Ordine dei parametri dello script passati allo script come argomenti della riga di comando.
value	string	Valore del parametro dello script.

Note:
Vedi anche: Media type object (per la proprietà corrispondente con lo stesso nome).

Webhook

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di media *Webhook*.

Elemento	Tipo	Descrizione
script	string	Script.
timeout	string	Intervallo di timeout della richiesta HTTP dello script Javascript. Valori possibili:[1] 1-60s (predefinito: 30s).
process_tags	string	Indica se elaborare i tag restituiti. Valori possibili:[1] NO (0, predefinito), YES (1).
show_event_menu	string	Indica la presenza di una voce nel menu evento se la macro {EVENT.TAGS.*} è stata risolta correttamente nei campi `event_menu_url` e `event_menu_name`. Valori possibili:[1] NO (0, predefinito), YES (1).
event_menu_url	string	URL della voce del menu evento. Supporta la macro {EVENT.TAGS.*}.
event_menu_name	string	Nome della voce del menu evento. Supporta la macro {EVENT.TAGS.*}.
parameters		Elemento radice per i parametri del tipo di media webhook.
name	string	(obbligatorio) Nome del parametro webhook.
value	string	Valore del parametro webhook.

Note:
Vedi anche: Media type object (fare riferimento alla proprietà pertinente con lo stesso nome).

Note a piè di pagina

[1] I valori interi dell'API tra parentesi, ad esempio, ENABLED (0), sono menzionati solo a scopo di riferimento. Per ulteriori informazioni, vedere la pagina dell'oggetto API collegata nella voce della tabella o alla fine di ciascuna sezione.

Dashboard

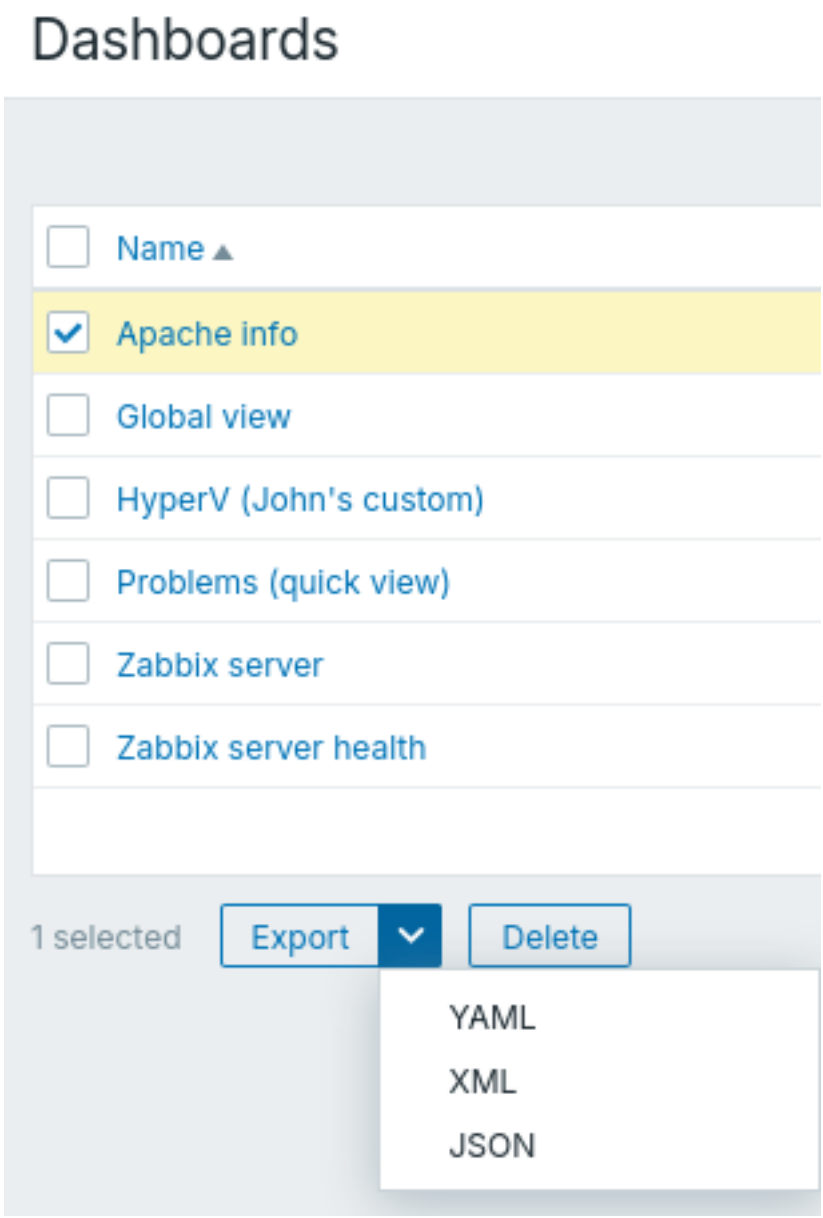
Panoramica

Le dashboard vengono esportate con tutti gli oggetti correlati e le relazioni tra gli oggetti.

Esportazione

Per esportare i dashboard, procedere come segue:

1. Andare su *Dashboards* > *All dashboards*.
2. Selezionare le caselle di controllo dei dashboard da esportare.
3. Fare clic su *Export* sotto l'elenco per esportare nel formato YAML predefinito, oppure fare clic sulla freccia per selezionare un formato di esportazione diverso.



A seconda del formato selezionato, i dashboard vengono esportati in un file locale con un nome predefinito:

- `zbx_export_dashboards.yaml` - nell'esportazione YAML;
- `zbx_export_dashboards.xml` - nell'esportazione XML;
- `zbx_export_dashboards.json` - nell'esportazione JSON.

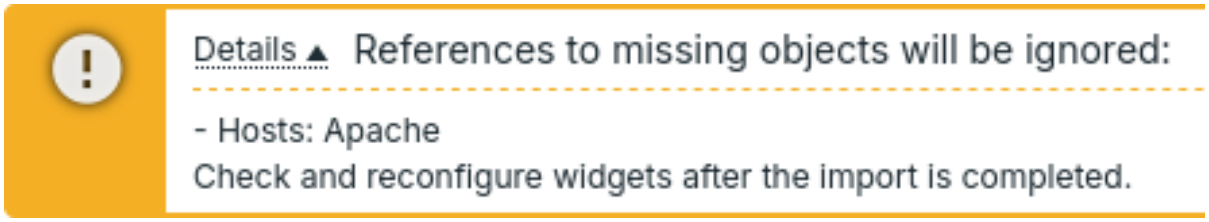
Importazione

Per importare le dashboard, procedere come segue:

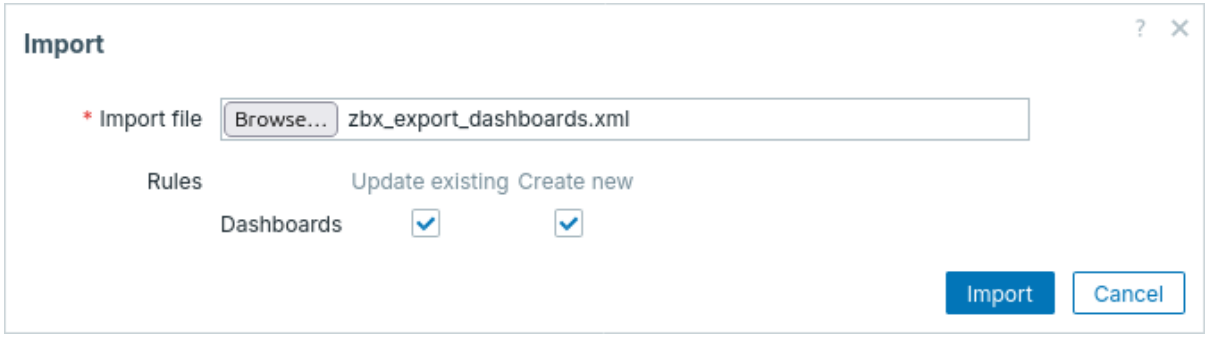
1. Andare a *Dashboard* > *Tutte le dashboard*.
2. Fare clic su *Importa* nell'angolo superiore destro. L'opzione è disabilitata se l'utente non dispone dell'autorizzazione *Creare e modificare le dashboard*.

3. Selezionare il file di importazione.
4. Selezionare le caselle di controllo delle regole di importazione da applicare (vedere di seguito).
5. Fare clic su *Importa* nell'angolo inferiore destro del modulo di configurazione.

Se il file di importazione fa riferimento a oggetti che non esistono nel sistema di destinazione, viene visualizzato un messaggio di avviso. Questo comportamento è previsto per i widget che fanno riferimento a host, item o altri oggetti specifici. In questi casi, Zabbix consente di importare la dashboard, cancella i riferimenti mancanti e contrassegna il widget come non completamente configurato quando i dati sono obbligatori oppure rimuove i criteri di filtro quando i campi vengono utilizzati solo per il filtraggio.



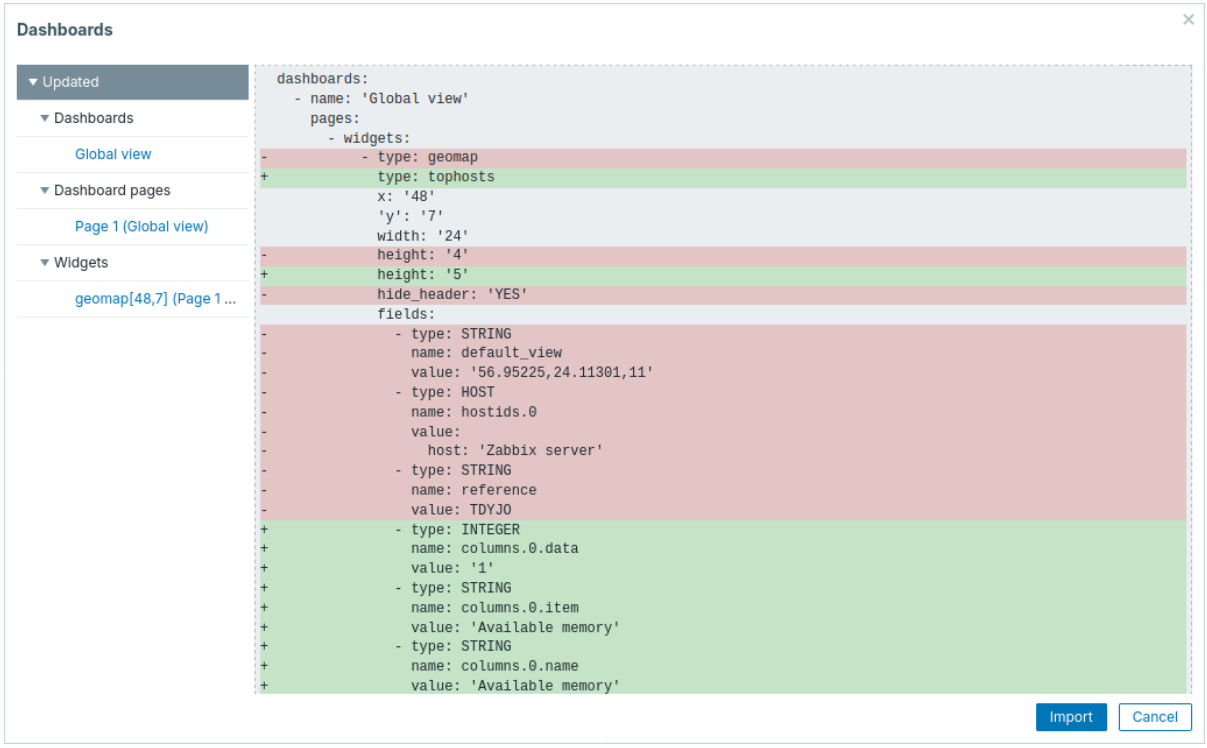
Facendo nuovamente clic sul pulsante *Importa*, la dashboard verrà aggiunta e/o aggiornata.



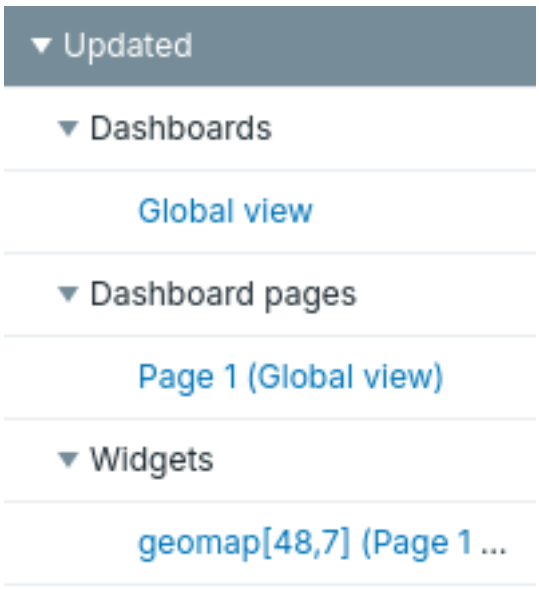
Regole di importazione:

Regola	Descrizione
Aggiorna esistenti	Gli elementi esistenti verranno aggiornati utilizzando i dati del file di importazione. In caso contrario, non verranno aggiornati.
Crea nuovi	I nuovi elementi verranno creati utilizzando i dati del file di importazione. In caso contrario, non verranno creati.

Nella schermata successiva sarà possibile visualizzare il contenuto delle dashboard importate. Se si tratta di una nuova dashboard, tutti gli elementi saranno elencati in verde. Se si aggiorna una dashboard esistente, i nuovi elementi della dashboard saranno evidenziati in verde; gli elementi rimossi saranno evidenziati in rosso; gli elementi che non sono cambiati saranno elencati su uno sfondo grigio.



Il menu a sinistra può essere utilizzato per spostarsi nell'elenco delle modifiche. La sezione *Aggiornati* evidenzia tutte le modifiche apportate agli elementi esistenti della dashboard. La sezione *Aggiunti* elenca i nuovi elementi della dashboard. Gli elementi di ciascuna sezione sono raggruppati per tipo di elemento; fare clic sulla freccia grigia per espandere o comprimere il gruppo di elementi.



Esaminare le modifiche alla dashboard, quindi fare clic su *Importa* per eseguire l'importazione della dashboard. Nel frontend verrà visualizzato un messaggio che indica il successo o il fallimento dell'importazione.

Formato di esportazione

Esporta in YAML:

```
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
```

```
            name: 'Top hosts per utilizzo CPU'
            width: '25'
            height: '4'
            fields:
              - type: INTEGER
                name: column
                value: '1'
              - type: INTEGER
                name: columns.0.data
                value: '2'
              - type: STRING
                name: columns.0.item
                value: 'System name'
              - type: STRING
                name: columns.0.name
                value: 'Host name'
              - type: STRING
                name: columns.1.base_color
                value: 4CAF50
              - type: INTEGER
                name: columns.1.data
                value: '1'
              - type: INTEGER
                name: columns.1.display
                value: '3'
              - type: STRING
                name: columns.1.item
                value: 'CPU utilization'
              - type: STRING
                name: columns.1.max
                value: '100'
              - type: STRING
                name: columns.1.min
                value: '0'
              - type: STRING
                name: columns.1.name
                value: Utilization
              - type: INTEGER
                name: columns.2.data
                value: '1'
              - type: STRING
                name: columns.2.item
                value: 'Load average (1m avg)'
              - type: STRING
                name: columns.2.name
                value: '1m avg'
              - type: INTEGER
                name: columns.3.data
                value: '1'
              - type: STRING
                name: columns.3.item
                value: 'Load average (5m avg)'
              - type: STRING
                name: columns.3.name
                value: '5m avg'
              - type: INTEGER
                name: columns.4.data
                value: '1'
              - type: STRING
                name: columns.4.item
                value: 'Load average (15m avg)'
              - type: STRING
```

```
            name: columns.4.name
            value: '15m avg'
          - type: INTEGER
            name: columns.5.data
            value: '1'
          - type: INTEGER
            name: columns.5.decimal_places
            value: '0'
          - type: STRING
            name: columns.5.item
            value: 'Number of processes'
          - type: STRING
            name: columns.5.name
            value: Processes
          - type: STRING
            name: columnsthresholds.1.color.0
            value: FFFF00
          - type: STRING
            name: columnsthresholds.1.color.1
            value: FF8000
          - type: STRING
            name: columnsthresholds.1.color.2
            value: FF4000
          - type: STRING
            name: columnsthresholds.1.threshold.0
            value: '50'
          - type: STRING
            name: columnsthresholds.1.threshold.1
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
      - type: hostavail
        'y': '4'
        width: '25'
        fields:
          - type: INTEGER
            name: interface_type.0
            value: '1'
      - type: problems
        name: 'Problemi attuali'
        'y': '6'
        width: '48'
        height: '5'
        fields:
          - type: STRING
            name: reference
            value: BRUQE
          - type: INTEGER
            name: show
            value: '3'
          - type: INTEGER
            name: show_tags
            value: '3'
      - type: systeminfo
        x: '25'
        width: '23'
        height: '4'
      - type: problemsbysv
```

```
      name: 'Problemi per gravità'
      x: '25'
      'y': '4'
      width: '23'
      fields:
        - type: STRING
          name: reference
          value: LBSVB
        - type: INTEGER
          name: show_type
          value: '1'
    - type: gauge
      name: 'Zabbix server'
      x: '48'
      width: '12'
      height: '4'
      hide_header: 'YES'
      fields:
        - type: INTEGER
          name: angle
          value: '270'
        - type: INTEGER
          name: desc_size
          value: '8'
        - type: ITEM
          name: itemid.0
          value:
            host: 'Zabbix server'
            key: vm.memory.utilization
        - type: STRING
          name: max
          value: '100'
        - type: STRING
          name: min
          value: '0'
        - type: INTEGER
          name: scale_size
          value: '11'
        - type: INTEGER
          name: show.0
          value: '1'
        - type: INTEGER
          name: show.1
          value: '2'
        - type: INTEGER
          name: show.2
          value: '3'
        - type: INTEGER
          name: show.3
          value: '4'
        - type: INTEGER
          name: show.4
          value: '5'
        - type: STRING
          name: thresholds.0.color
          value: 42A5F5
        - type: STRING
          name: thresholds.0.threshold
          value: '0'
        - type: STRING
          name: thresholds.1.color
          value: 9CCC65
```

```
        - type: STRING
          name: thresholds.1.threshold
          value: '20'
        - type: STRING
          name: thresholds.2.color
          value: FFEE58
        - type: STRING
          name: thresholds.2.threshold
          value: '40'
        - type: STRING
          name: thresholds.3.color
          value: FFA726
        - type: STRING
          name: thresholds.3.threshold
          value: '60'
        - type: STRING
          name: thresholds.4.color
          value: FF7043
        - type: STRING
          name: thresholds.4.threshold
          value: '80'
        - type: INTEGER
          name: th_arc_size
          value: '9'
        - type: INTEGER
          name: th_show_arc
          value: '1'
        - type: INTEGER
          name: th_show_labels
          value: '1'
        - type: INTEGER
          name: units_bold
          value: '1'
        - type: INTEGER
          name: units_size
          value: '24'
        - type: INTEGER
          name: value_arc_size
          value: '30'
        - type: INTEGER
          name: value_bold
          value: '1'
        - type: INTEGER
          name: value_size
          value: '24'
    - type: svggraph
      x: '48'
      'y': '4'
      width: '24'
      height: '3'
      hide_header: 'YES'
      fields:
        - type: INTEGER
          name: axisx
          value: '0'
        - type: STRING
          name: ds.0.color
          value: AB47BC
        - type: INTEGER
          name: ds.0.fill
          value: '0'
        - type: STRING
```

```
            name: ds.0.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.0.items.0
            value: 'Available memory'
          - type: INTEGER
            name: ds.0.transparency
            value: '2'
          - type: INTEGER
            name: ds.0.type
            value: '2'
          - type: INTEGER
            name: ds.0.width
            value: '2'
          - type: INTEGER
            name: ds.1.axisy
            value: '1'
          - type: STRING
            name: ds.1.color
            value: 42A5F5
          - type: INTEGER
            name: ds.1.fill
            value: '0'
          - type: STRING
            name: ds.1.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.1.items.0
            value: 'CPU utilization'
          - type: INTEGER
            name: ds.1.transparency
            value: '0'
          - type: INTEGER
            name: ds.1.type
            value: '2'
          - type: INTEGER
            name: ds.1.width
            value: '2'
          - type: STRING
            name: reference
            value: TOYGS
      - type: geomap
        x: '48'
        'y': '7'
        width: '24'
        height: '4'
        hide_header: 'YES'
        fields:
          - type: STRING
            name: default_view
            value: '56.95225,24.11301,11'
          - type: HOST
            name: hostids.0
            value:
              host: 'Zabbix server'
          - type: STRING
            name: reference
            value: TDYJO
      - type: clock
        name: 'Ora locale'
        x: '60'
        width: '12'
```

```
            hide_header: 'YES'
            fields:
              - type: INTEGER
                name: clock_type
                value: '1'
              - type: INTEGER
                name: show.0
                value: '1'
              - type: INTEGER
                name: show.1
                value: '2'
              - type: INTEGER
                name: show.2
                value: '3'
              - type: INTEGER
                name: time_sec
                value: '0'
              - type: INTEGER
                name: tzone_format
                value: '1'
          - type: item
            name: 'Prestazioni del server'
            x: '60'
            'y': '2'
            width: '12'
            hide_header: 'YES'
            fields:
              - type: INTEGER
                name: aggregate_function
                value: '2'
              - type: INTEGER
                name: decimal_size
                value: '32'
              - type: STRING
                name: description
                value: 'Values per second'
              - type: INTEGER
                name: desc_h_pos
                value: '0'
              - type: INTEGER
                name: desc_v_pos
                value: '0'
              - type: ITEM
                name: itemid.0
                value:
                  host: 'Zabbix server'
                  key: 'zabbix[wcache,values]'
              - type: INTEGER
                name: show.0
                value: '1'
              - type: INTEGER
                name: show.1
                value: '2'
              - type: INTEGER
                name: show.2
                value: '5'
              - type: INTEGER
                name: sparkline.fill
                value: '2'
              - type: INTEGER
                name: sparkline.width
                value: '2'
```

```
            - type: STRING
              name: thresholds.0.color
              value: DCEDC8
            - type: STRING
              name: thresholds.0.threshold
              value: '20'
            - type: STRING
              name: thresholds.1.color
              value: FFF9C4
            - type: STRING
              name: thresholds.1.threshold
              value: '40'
            - type: STRING
              name: thresholds.2.color
              value: FFCCBC
            - type: STRING
              name: thresholds.2.threshold
              value: '60'
            - type: INTEGER
              name: units_bold
              value: '0'
            - type: INTEGER
              name: units_show
              value: '0'
            - type: INTEGER
              name: value_h_pos
              value: '0'
            - type: INTEGER
              name: value_size
              value: '32'
```

Elementi esportati

Gli elementi esportati sono spiegati nella tabella seguente.

Element	Type	Description
name	string	(required) Nome della dashboard.
pages		Elemento radice per le pagine della dashboard
widgets		Elemento radice per le pagine della dashboard

Note:
Vedi anche: Dashboard object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Widget della dashboard

Element	Type	Description
type	string	(obbligatorio) Tipo del widget della dashboard. Valori possibili: actionlog - Registro azioni; clock - Orologio; discovery - Stato della discovery; favgraphs - Grafici preferiti; favmaps - Mappe preferite; gauge - Indicatore; geomap - Geomap; graph - Grafico (classico); graphprototype - Prototipo di grafico; honeycomb - Honeycomb; hostavail - Disponibilità dell'host; hostcard - Scheda host; hostnavigator - Navigatore host; itemcard - Scheda item; itemhistory - Cronologia item; itemnavigator - Navigatore item; item - Valore dell'item; map - Mappa; navtree - Albero di navigazione della mappa; piechart - Grafico a torta; problemhosts - Host con problemi; problems - Problemi; problemsbysv - Problemi per gravità; scatterplot - Grafico a dispersione; slareport - Report SLA; svggraph - Grafico; systeminfo - Informazioni di sistema; tophosts - Host principali; topitems - Item principali; toptriggers - Trigger principali; trigover - Panoramica dei trigger; url - URL; web - Monitoraggio web.
x	integer	Una posizione orizzontale dal lato sinistro della dashboard. Valori possibili:[1] 0-71.
y	integer	Una posizione verticale dalla parte superiore della dashboard. Valori possibili:[1] 0-63.
name	string	Nome personalizzato del widget.
width	integer	La larghezza del widget. Valori possibili:[1] 1-72.
height	integer	L'altezza del widget. Valori possibili:[1] 1-64.
hide_header	string	Nasconde l'intestazione del widget. Valori possibili:[1] NO (0, predefinito), YES (1).
fields		Elemento radice per i campi del widget della dashboard.

Note:
Vedi anche: Widget della dashboard (fare riferimento alla proprietà pertinente con lo stesso nome).

Campo widget della dashboard

Elemento	Tipo	Descrizione
type	string	(obbligatorio) Tipo di campo del widget. Valori possibili:[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).

Elemento	Tipo	Descrizione
name	string	(obbligatorio) Nome del campo del widget.
value	mixed	(obbligatorio) Valore del campo del widget, a seconda del tipo di campo.
host	string	Host dell'item.
key	string	Chiave dell'item.

Note:
Vedi anche: Campo widget della dashboard (fare riferimento alla proprietà pertinente con lo stesso nome).

Note a piè di pagina

[1] I valori interi dell'API tra parentesi, ad esempio ENABLED (0), sono menzionati solo a titolo di riferimento. Per ulteriori informazioni, vedere la pagina dell'oggetto API collegata nella voce della tabella o alla fine di ogni sezione.

Espressioni regolari

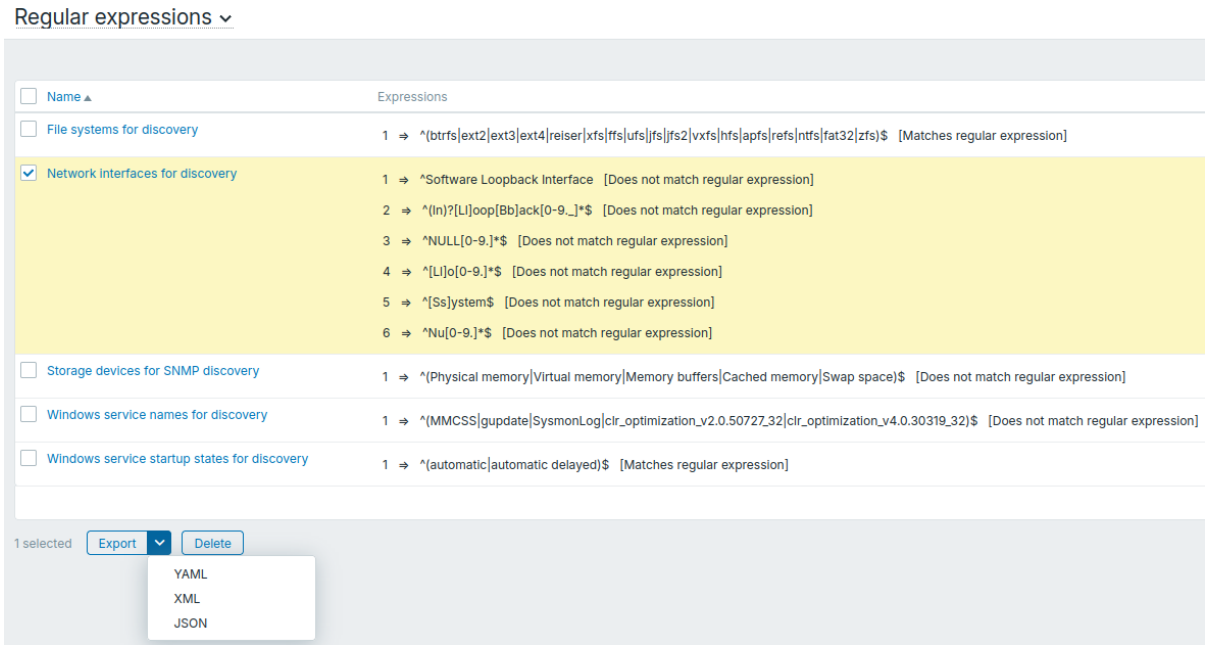
Panoramica

Le espressioni regolari vengono esportate con tutti gli oggetti correlati e le relazioni tra gli oggetti.

Esportazione

Per esportare le espressioni regolari, procedere come segue:

1. Andare in *Administration* > *General* > *Regular expressions*.
2. Selezionare le caselle di controllo delle espressioni regolari da esportare.
3. Fare clic su *Export* sotto l'elenco per esportare nel formato YAML predefinito, oppure fare clic sulla freccia per selezionare un formato di esportazione diverso.



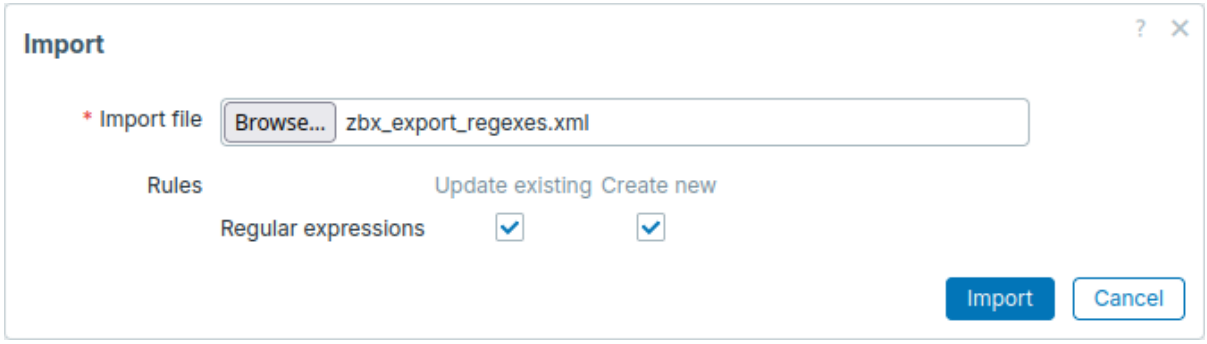
A seconda del formato selezionato, le espressioni regolari vengono esportate in un file locale con un nome predefinito:

- `zbx_export_regexes.yaml` - nell'esportazione YAML;
- `zbx_export_regexes.xml` - nell'esportazione XML;
- `zbx_export_regexes.json` - nell'esportazione JSON.

Importazione

Per importare le espressioni regolari, procedere come segue:

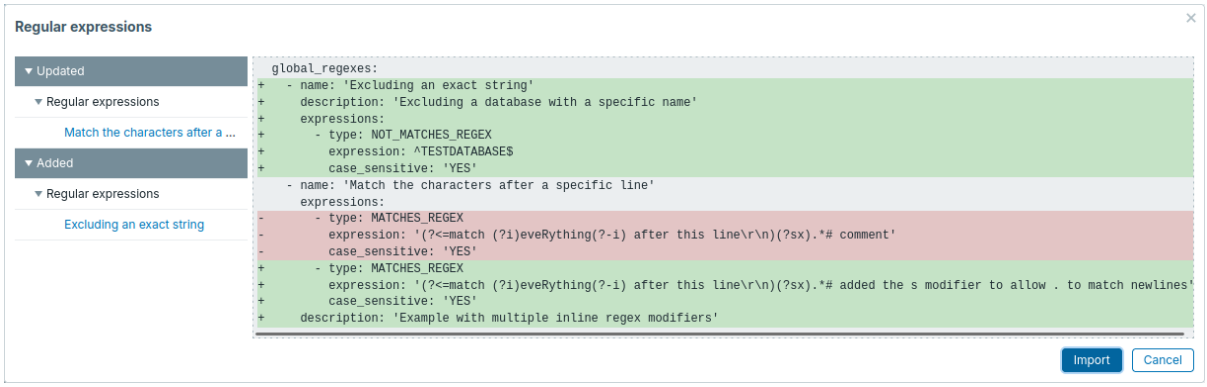
1. Andare in *Administration* > *General* > *Regular expressions*.
2. Fare clic su *Import* nell'angolo superiore destro.
3. Selezionare il file di importazione.
4. Selezionare le caselle di controllo delle regole di importazione che si desidera applicare (vedere di seguito).
5. Fare clic su *Import* nell'angolo inferiore destro del modulo di configurazione.



Regole di importazione:

Rule	Description
Update existing	Le espressioni regolari esistenti verranno aggiornate utilizzando i dati del file di importazione, sovrascrivendo completamente le espressioni esistenti con lo stesso nome. In caso contrario, non verranno aggiornate.
Create new	Verranno create nuove espressioni regolari utilizzando i dati del file di importazione. In caso contrario, non verranno create.

Nella schermata successiva sarà possibile visualizzare il contenuto delle espressioni regolari importate. Se si tratta di una nuova espressione regolare, tutti i dati saranno elencati in verde. Se si aggiorna un'espressione regolare esistente, i nuovi dati dell'espressione regolare saranno evidenziati in verde; i dati rimossi saranno evidenziati in rosso; i dati invariati saranno visualizzati su uno sfondo grigio.



Il menu a sinistra può essere utilizzato per spostarsi nell'elenco delle modifiche. La sezione *Updated* evidenzia tutte le modifiche apportate alle espressioni regolari esistenti. La sezione *Added* elenca le nuove espressioni regolari. Fare clic sulla freccia per espandere o comprimere un gruppo di espressioni regolari.

Esaminare le modifiche alle espressioni regolari, quindi fare clic su *Import* per eseguire l'importazione delle espressioni regolari. Nel frontend verrà visualizzato un messaggio che indica l'esito positivo o negativo dell'importazione.

Formato di esportazione

Esporta in YAML:

```yaml
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'File systems for discovery'
      description: 'Espressione regolare globale usata per individuare e identificare i file system sull'h
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'Storage devices for SNMP discovery'
      expressions:
        - type: NOT_MATCHES_REGEX
          expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
          case_sensitive: 'YES'
```

Elementi esportati

Elementi esportati delle espressioni regolari:

Elemento	Tipo	Descrizione
global_regexes	object	Elemento radice per le espressioni regolari.
name	string	(obbligatorio) Nome dell'espressione regolare.
description	text	Descrizione dell'espressione regolare.
expressions	object	Elemento radice per le espressioni regolari.
type	string	Tipo dell'espressione regolare.
expression	string	Espressione regolare.
exp_delimiter	string	Delimitatore dell'espressione regolare; presente quando il tipo è *Contains any substring from list*.
case_sensitive	string	Opzione di distinzione tra maiuscole e minuscole per l'espressione regolare.

Note:
Vedi anche: Oggetto espressione regolare (fare riferimento alla proprietà pertinente con lo stesso nome).

Espressioni regolari

Panoramica Perl Compatible Regular Expressions (PCRE, PCRE2) sono supportate in Zabbix.

Esistono due modi per utilizzare le espressioni regolari in Zabbix:

- Inserire manualmente un'espressione regolare
- Utilizzare un'espressione regolare globale creata in Zabbix

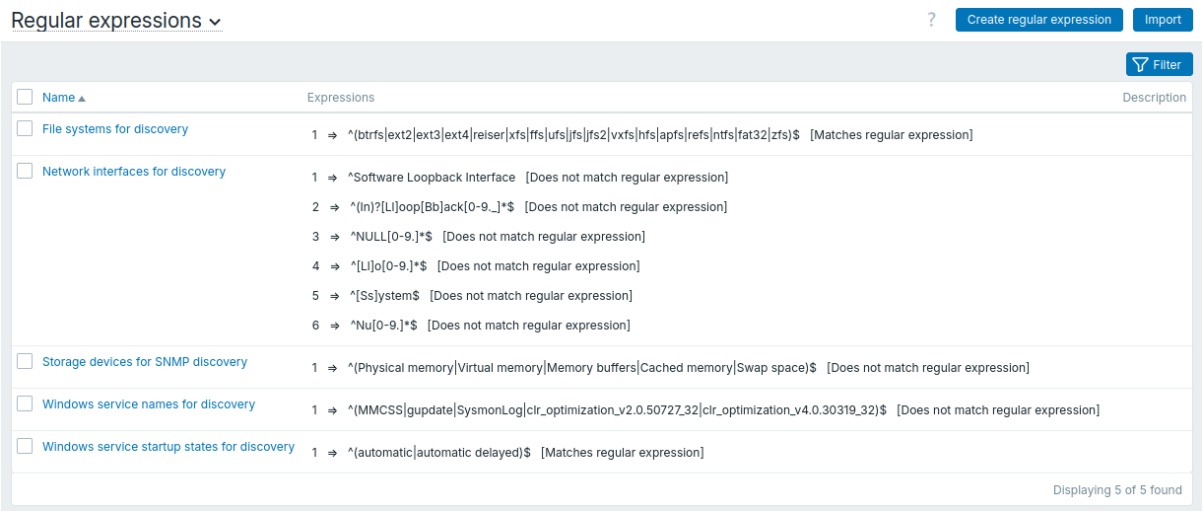
Espressioni regolari È possibile inserire manualmente un'espressione regolare nei punti supportati.
 Si noti che l'espressione non deve iniziare con @, poiché questo simbolo viene utilizzato in Zabbix per fare riferimento alle espressioni regolari globali.

Attention:
È possibile esaurire la memoria dello stack quando si utilizzano espressioni regolari. Per ulteriori informazioni, vedere PCRE PERFORMANCE.

Si noti che, nella corrispondenza multilinea, gli ancoraggi ^ e $ corrispondono rispettivamente all'inizio e alla fine di ogni riga, invece che all'inizio e alla fine dell'intera stringa.

Vedere anche gli esempi di escaping corretto in vari contesti.

Espressioni regolari globali Zabbix fornisce un editor avanzato nel frontend per creare e testare espressioni regolari complesse.
 Per accedervi, vai su *Administration* > *General* > *Regular expressions*.
 Zabbix include un certo numero di espressioni regolari globali predefinite nel proprio set di dati predefinito.

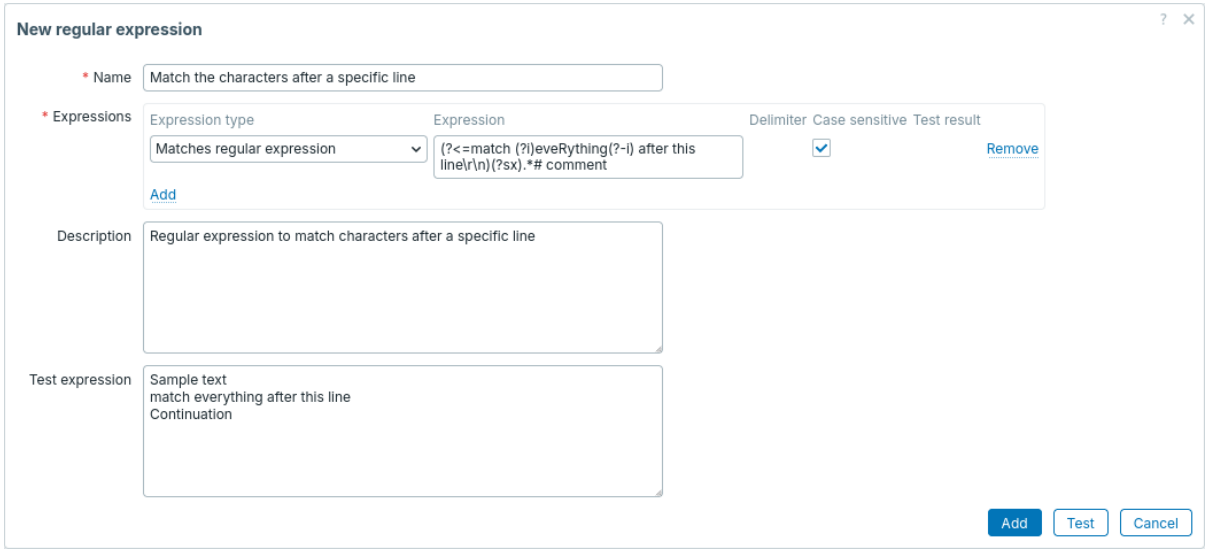


Puoi filtrare l'elenco delle espressioni regolari per nome e descrizione per trovare le espressioni di cui hai bisogno.

Configurazione

Per creare un'espressione regolare globale:

1. Fare clic su *Crea espressione regolare*.
2. Inserire i parametri dell'espressione regolare nel modulo.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Nome	Impostare il nome dell'espressione regolare. Sono consentiti tutti i caratteri Unicode.
Espressioni	Fare clic su *Aggiungi* nella sezione *Espressioni* per aggiungere una nuova sottoespressione.
Tipo di espressione	Selezionare il tipo di espressione: **Contiene la stringa** - corrisponde alla sottostringa. **Contiene una qualsiasi sottostringa dell'elenco** - corrisponde a una qualsiasi sottostringa di un elenco delimitato. L'elenco delimitato include una virgola `,`, un punto `.` o una barra `/`. **Non contiene la stringa** - corrisponde a qualsiasi stringa tranne la sottostringa. **Corrisponde all'espressione regolare** - corrisponde all'espressione regolare. **Non corrisponde all'espressione regolare** - non corrisponde all'espressione regolare.
Espressione	Inserire la sottostringa/espressione regolare. Lunghezza massima: 2048 caratteri.
Delimitatore	Una virgola `,`, un punto `.` o una barra `/` per separare le stringhe di testo in un'espressione regolare. Questo parametro è attivo solo quando è selezionato il tipo di espressione *Contiene una qualsiasi sottostringa dell'elenco*.
Maiuscole/minuscole	Una casella di controllo per specificare se un'espressione regolare distingue tra maiuscole e minuscole.
Descrizione	Inserire una descrizione per l'espressione regolare.
Testa espressione	Un valore da utilizzare per testare l'espressione regolare.

Una barra `/` nell'espressione viene trattata letteralmente e non come delimitatore. Ciò consente di salvare senza errori le espressioni contenenti una barra.

Attention:
Il nome di un'espressione regolare personalizzata in Zabbix può contenere virgole, spazi, ecc. Nei casi in cui ciò potrebbe causare un'interpretazione errata durante il riferimento (ad esempio, una virgola nel parametro di una chiave item), l'intero riferimento può essere racchiuso tra virgolette, come in: *"@My custom regexp for purpose1, purpose2"*.
 I nomi delle espressioni regolari non devono essere racchiusi tra virgolette in altre posizioni (ad esempio, nelle proprietà delle regole LLD).

Dopo aver creato un'espressione regolare, è possibile utilizzarla in diversi punti del frontend facendo riferimento al suo nome preceduto da `@`, ad esempio `@mycustomregexp`.

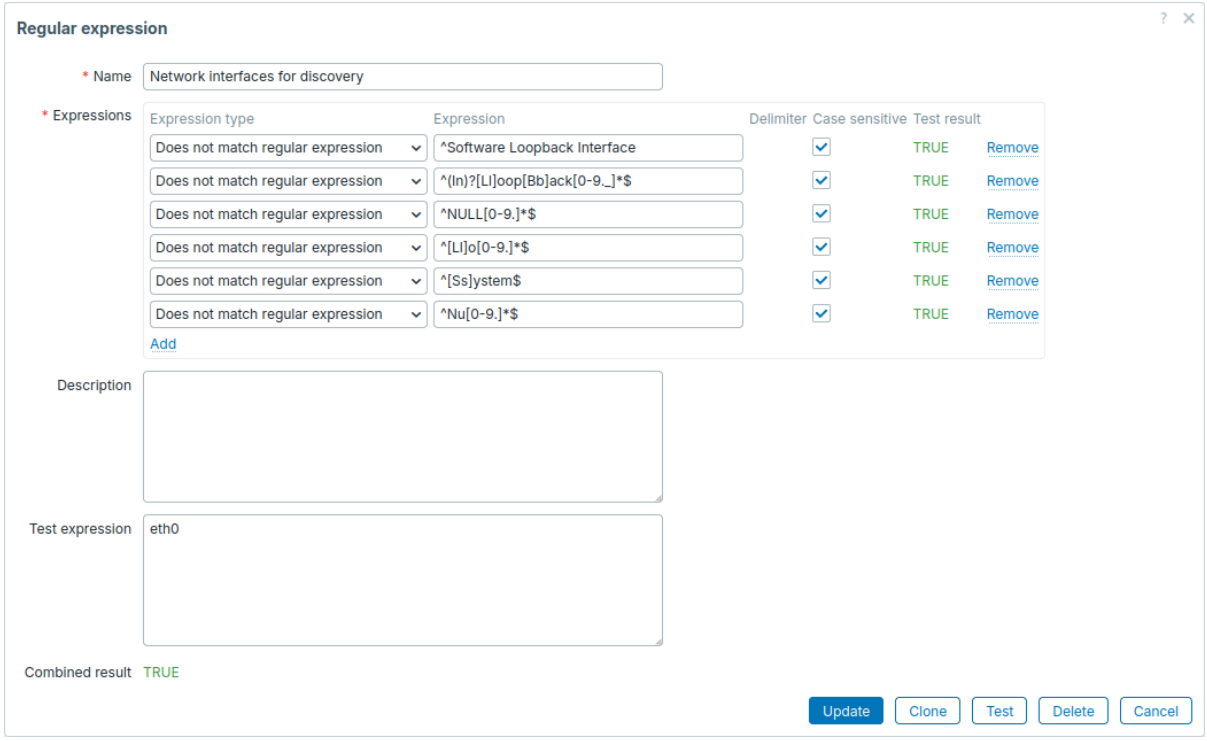
Pulsanti del modulo

I pulsanti nella parte inferiore del modulo consentono di eseguire varie operazioni:

Button	Description
Add	Aggiungi un'espressione regolare. Questo pulsante è disponibile solo per le nuove espressioni regolari.
Update	Aggiorna le proprietà di un'espressione regolare.
Clone	Crea un'altra espressione regolare basata sulle proprietà dell'espressione regolare corrente.
Test	Verifica un'espressione regolare e le sue sottoespressioni rispetto a un valore fornito.
Delete	Elimina l'espressione regolare.
Cancel	Annulla la modifica delle proprietà dell'espressione regolare.

Test di un'espressione regolare

È possibile testare l'espressione regolare e le sue sottoespressioni inserendo un valore nel campo *Test expression* e facendo clic sul pulsante *Test*.



Nella colonna *Test result* viene visualizzato lo stato di ciascuna sottoespressione. Il campo *Combined result* visualizza lo stato complessivo dell'espressione regolare. Se sono definite più sottoespressioni, Zabbix usa l'operatore logico AND per calcolare il *Combined result*; cioè, se almeno un risultato è `False`, anche il *Combined result* è `False`.

Espressioni regolari globali predefinite

Il dataset predefinito include le seguenti espressioni regolari globali predefinite:

Nome	Espressione	Corrisponde a
File systems for discovery	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	"btrfs" o "ext2" o "ext3" o "ext4" o "reiser" o "xfs" o "ffs" o "ufs" o "jfs" o "jfs2" o "vxfs" o "hfs" o "apfs" o "refs" o "ntfs" o "fat32" o "zfs".
Network interfaces for discovery	`^Software Loopback Interface`	Stringhe che iniziano con "Software Loopback Interface".
	`^lo$`	"lo".
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Stringhe che iniziano facoltativamente con "In", seguite da "L" o "l", poi da "oop", poi da "B" o "b", poi da "ack", e facoltativamente seguite da un numero qualsiasi di cifre, punti o underscore.
	`^NULL[0-9.]*$`	Stringhe che iniziano con "NULL" e sono facoltativamente seguite da un numero qualsiasi di cifre o punti.

Nome	Espressione	Corrisponde a
	`^[Ll]o[0-9.]*$`	Stringhe che iniziano con ”Lo” o ”lo” e sono facoltativamente seguite da un numero qualsiasi di cifre o punti.
	`^[Ss]ystem$`	”System” o ”system”.
	`^Nu[0-9.]*$`	Stringhe che iniziano con ”Nu” e sono facoltativamente seguite da un numero qualsiasi di cifre o punti.
Storage devices for SNMP discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” o ”Virtual memory” o ”Memory buffers” o ”Cached memory” o ”Swap space”.
Windows service names for discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS” o ”gupdate” o ”SysmonLog” o stringhe corrispondenti a ”clr_optimization_v2.0.50727_32” o ”clr_optimization_v4.0.30319_32”, dove ogni punto può essere sostituito da qualsiasi carattere tranne un ritorno a capo.
Windows service startup states for discovery	`^(automatic|automatic delayed)$`	”automatic” o ”automatic delayed”.

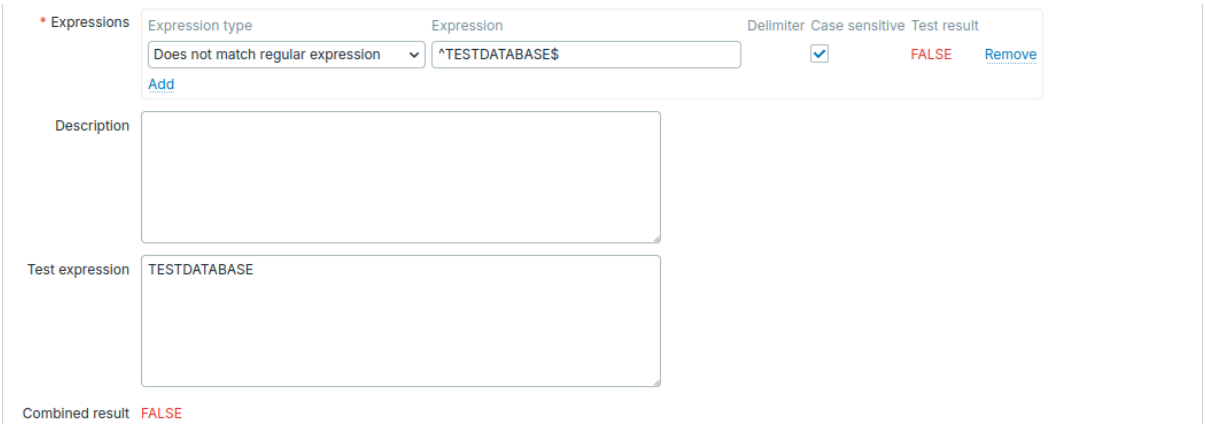
Importazione/esportazione di un’espressione regolare

È possibile importare o esportare espressioni regolari globali come descritto nella pagina di configurazione di esportazione/importazione Espressioni regolari.

Esempi Esempio di esclusione di una stringa esatta

Usa la seguente espressione nella discovery a basso livello per individuare i database, tranne un database con un nome specifico:

```
^TESTDATABASE$
```

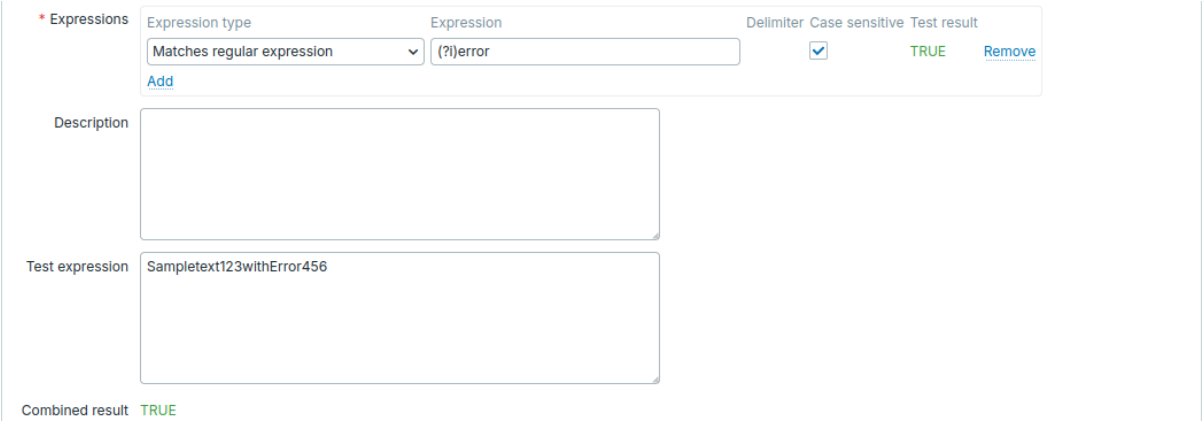


Sia *Test result* sia *Combined result* mostrano `FALSE`: il nome corrisponde a ”TESTDATABASE”, quindi la condizione *Does not match regular expression* non è soddisfatta.

Esempio con un modificatore regex inline per una corrispondenza senza distinzione tra maiuscole e minuscole

Usa la seguente espressione regolare, che include il modificatore inline `(?i)` per eseguire una corrispondenza senza distinzione tra maiuscole e minuscole dei caratteri ”error”:

```
(?i)error
```



Sia *Test result* sia *Combined result* mostrano TRUE: i caratteri "error" vengono trovati anche con *Case sensitive* selezionato, perché il modificatore inline `(?i)` rende la corrispondenza senza distinzione tra maiuscole e minuscole.

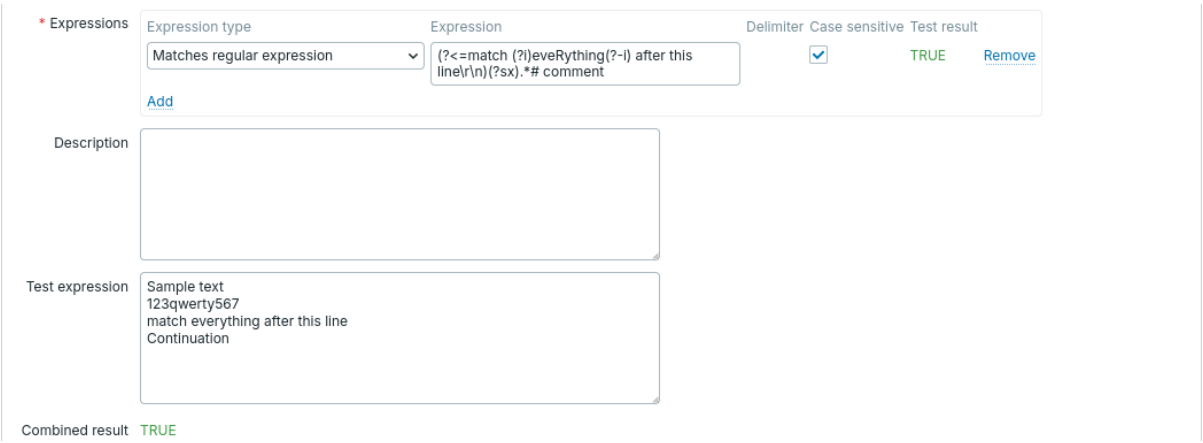
Esempio con più modificatori regex inline

Usa la seguente espressione regolare, che include più modificatori inline, per corrispondere ai caratteri dopo una riga specifica:

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

dove:

- `(?i)` - attiva la corrispondenza senza distinzione tra maiuscole e minuscole;
- `(?-i)` - disattiva la corrispondenza senza distinzione tra maiuscole e minuscole;
- `(?s)` - abilita la modalità dotall, consentendo a `.` di corrispondere ai caratteri di nuova riga;
- `(?x)` - abilita la modalità estesa, consentendo di ignorare spazi bianchi e commenti nell'espressione regolare.



Sia *Test result* sia *Combined result* mostrano TRUE: i caratteri che seguono la riga specifica vengono corrisposti.

Attention:
Il modificatore g non può essere specificato inline. L'elenco dei modificatori disponibili è disponibile nella pagina man di pcresyntax. Per ulteriori informazioni sulla sintassi PCRE, consulta la documentazione HTML di PCRE.

Supporto delle espressioni regolari per posizione

Posizione		Espressione regolare	Espressione regolare globale	Corrispondenza multilinea	Commenti
Item dell'agent					
	eventlog[]	Sì	Sì	Sì	parametri `regexp`, `severity`, `source`, `eventid`
	eventlog.count[]				parametri `regexp`, `severity`, `source`, `eventid`
	log[]				parametro `regexp`
	log.count[]				

Posizione		Espressione regolare	Espressione regolare globale	Corrispondenza multilinea	Commenti
	logrt[]		Sì/No		il parametro `regexp` supporta entrambi, il parametro `file_regexp` supporta solo espressioni non globali
	logrt.count[]				
	proc.cpu.util[] proc.get[] proc.mem[] proc.num[]		No	No	parametro `cmdline`
	sensor[]				parametri `device` e `sensor` su Linux 2.4
	system.hw.macaddr[]				parametro `interface`
	system.sw.packages[]				parametro `regexp`
	system.sw.packages.get[]				parametro `regexp`
	vfs.dir.count[]				parametri `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.dir.get[]				parametri `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.dir.size[]				parametri `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.file.regexp[]			Sì	parametro `regexp`
	vfs.file.regmatch[]				
	web.page.regexp[]				
SNMP traps					
	snmptrap[]	Sì	Sì	No	parametro `regexp`
Preprocessing del valore dell'item		Sì	No	No	campo `pattern`
Funzioni per trigger/item calcolati					
	count()	Sì	Sì	Sì	parametro `pattern` se il parametro `operator` è *regexp* o *iregexp*
	countunique()	Sì	Sì		
	find()	Sì	Sì		
	logeventid()	Sì	Sì	No	parametro `pattern`
	logsource()				
Low-level discovery					
	Filtri	Sì	Sì	No	campo *Espressione regolare*
	Override	Sì	No		nelle opzioni *corrisponde*, *non corrisponde* per le condizioni *Operazione*
Condizioni delle azioni		Sì	No	No	nelle opzioni *corrisponde*, *non corrisponde* per le condizioni di autoregistrazione *Nome host* e *Metadati host*
Script		Sì	Sì	No	campo *Regola di convalida dell'input*
Monitoraggio web		Sì	No	Sì	*Variabili* con prefisso **regex:** campo *Stringa richiesta*

Posizione		Espressione regolare	Espressione regolare globale	Corrispondenza multilinea	Commenti
Contesto delle macro utente Funzioni macro		Sì	No	No	nel contesto della macro con prefisso **regex:**
	regsub() iregsub()	Sì	No	No	parametro `pattern`
Indicatori di collega-mento nelle mappe		Sì	No	No	campo *Pattern* (per gli item di testo)
Mappatura delle icone		Sì	Sì	No	campo *Espressione*
Mappatura dei valori		Sì	No	No	campo *Valore* se il tipo di mappatura è `regexp`
Parametri di configu-razione dell'agent					
	AllowKeyRegexp DenyKeyRegexp	Sì	No	No	Pattern

Eventi e risoluzione

Panoramica In Zabbix vengono generati diversi tipi di eventi:

- eventi di trigger - ogni volta che un trigger cambia stato (*OK→PROBLEM→OK*)
- eventi di servizio - ogni volta che un servizio cambia stato (*OK→PROBLEM→OK*)
- eventi di discovery - quando vengono rilevati host o servizi
- eventi di autoregistrazione - quando gli agent attivi vengono autoregistrati dal server
- eventi interni - quando un item/regola di discovery a basso livello diventa non supportato o un trigger entra in uno stato sconosciuto

Gli eventi sono associati a un timestamp e possono essere la base per azioni come l'invio di email di notifica, la creazione di ticket o l'esecuzione di comandi remoti.

Per visualizzare i dettagli di un evento nel frontend, vai su *Monitoring > Problems*. Lì puoi fare clic sulla data e sull'ora dell'evento per aprire la pagina dei dettagli dell'evento, dove puoi vedere informazioni aggiuntive sull'evento, incluso se è stato soppresso a causa della manutenzione o soppresso manualmente da un utente.

Risoluzione dei problemi Gli eventi di problema creati dai trigger vengono risolti automaticamente, se la condizione che ha causato il problema non esiste più.

Per aggiungere flessibilità e praticità alla gestione degli eventi di problema, sono disponibili le seguenti opzioni:

- Correlazione degli eventi
- Conferma e soppressione degli eventi
- Chiusura manuale dei problemi
- Periodi di manutenzione - definiscono i periodi in cui i problemi vengono soppressi
- Tagging - etichettare i problemi in base a una caratteristica comune

Eventi trigger La modifica dello stato del trigger è la fonte di eventi più frequente e più importante. Ogni volta che il trigger cambia stato, viene generato un evento. L'evento contiene i dettagli della modifica dello stato del trigger: quando è avvenuta e qual è il nuovo stato.

I trigger generano due tipi di eventi: `Problem` e `OK`.

Eventi di problema

Un evento di problema viene creato:

- quando un'espressione di trigger viene valutata come TRUE se il trigger è in stato OK;
- ogni volta che un'espressione di trigger viene valutata come TRUE se la generazione di più eventi di problema è abilitata per il trigger.

Eventi OK

Un evento OK chiude il o gli eventi di problema correlati e può essere creato da 3 componenti:

- trigger - in base alle impostazioni "OK event generation" e "OK event closes";
- correlazione degli eventi
- task manager - quando un evento viene chiuso manualmente

I trigger hanno un'impostazione "OK event generation" che controlla come vengono generati gli eventi OK:

- *Expression* - un evento OK viene generato per un trigger nello stato di problem quando la sua espressione viene valutata come FALSE. Questa è l'impostazione più semplice, abilitata per impostazione predefinita.
- *Recovery expression* - un evento OK viene generato per un trigger nello stato di problem quando la sua espressione viene valutata come FALSE e la recovery expression viene valutata come TRUE. Questa opzione può essere usata se i criteri di ripristino del trigger sono diversi dai criteri di problem.
- *None* - un evento OK non viene mai generato. Questa opzione può essere usata insieme alla generazione di più eventi di problem per inviare semplicemente una notifica quando accade qualcosa.

Inoltre, i trigger hanno un'impostazione "OK event closes" che controlla quali eventi di problem vengono chiusi:

- *All problems* - un evento OK chiuderà tutti i problem aperti creati dal trigger
- *All problems if tag values match* - un evento OK chiuderà i problem aperti creati dal trigger che hanno almeno un valore di tag corrispondente. Il tag è definito dall'impostazione del trigger "Tag for matching". Se non ci sono eventi di problem da chiudere, l'evento OK non viene generato. Questo è spesso chiamato correlazione degli eventi a livello di trigger.

Correlazione degli eventi

La correlazione degli eventi (chiamata anche correlazione globale degli eventi) è un modo per configurare regole personalizzate per la chiusura degli eventi (con conseguente generazione di eventi OK).

Le regole definiscono come associare i nuovi eventi problema agli eventi problema esistenti e consentono di chiudere il nuovo evento o gli eventi associati generando i corrispondenti eventi OK.

Tuttavia, la correlazione degli eventi deve essere configurata con molta attenzione, poiché può influire negativamente sulle prestazioni dell'elaborazione degli eventi o, se configurata in modo errato, chiudere più eventi del previsto (nel peggiore dei casi, potrebbero essere chiusi tutti gli eventi problema). Alcuni suggerimenti per la configurazione:

1. ridurre sempre l'ambito della correlazione impostando un tag univoco per l'evento di controllo (l'evento associato agli eventi precedenti) e utilizzare la condizione di correlazione "new event tag"
2. non dimenticare di aggiungere una condizione basata sull'evento precedente quando si utilizza l'operazione "close old event", altrimenti tutti i problemi esistenti potrebbero essere chiusi
3. evitare di utilizzare nomi di tag comuni impiegati da diverse configurazioni di correlazione

Task manager

Se l'impostazione 'Allow manual close' è abilitata per il trigger, è possibile chiudere manualmente gli eventi di problema generati dal trigger. Questa operazione viene eseguita nel frontend quando si aggiorna un problema. L'evento non viene chiuso direttamente: viene invece creato un task 'close event', che viene gestito a breve dal task manager. Il task manager genererà un evento OK corrispondente e l'evento di problema verrà chiuso.

Eventi del servizio Gli eventi del servizio vengono generati solo se le azioni del servizio per questi eventi sono abilitate. In questo caso, ogni modifica dello stato del servizio crea un nuovo evento:

- Evento di problema - quando lo stato del servizio cambia da `OK` a `Problem`
- Evento OK - quando lo stato del servizio cambia da `Problem` a `OK`

L'evento contiene i dettagli della modifica dello stato del servizio: quando è avvenuta e qual è il nuovo stato.

Eventi di discovery Zabbix esegue periodicamente la scansione degli intervalli IP definiti nelle regole di discovery di rete. La frequenza del controllo è configurabile singolarmente per ciascuna regola. Una volta che un host o un servizio viene scoperto, viene generato un evento di discovery (o più eventi).

Zabbix genera i seguenti eventi:

Evento	Quando viene generato
Service Up	Ogni volta che Zabbix rileva un servizio attivo.
Service Down	Ogni volta che Zabbix non riesce a rilevare il servizio.
Host Up	Se almeno uno dei servizi è UP per l'IP.
Host Down	Se tutti i servizi non rispondono.
Service Discovered	Se il servizio torna disponibile dopo un periodo di inattività o viene scoperto per la prima volta.
Service Lost	Se il servizio viene perso dopo essere stato UP.
Host Discovered	Se l'host torna disponibile dopo un periodo di inattività o viene scoperto per la prima volta.
Host Lost	Se l'host viene perso dopo essere stato UP.

Eventi di autoregistrazione dell'agent attivo L'autoregistrazione dell'agent attivo crea eventi in Zabbix.

Se configurata, viene creato un evento di autoregistrazione dell'agent attivo quando un agent attivo precedentemente sconosciuto richiede controlli oppure se i metadati dell'host sono cambiati. Il server aggiunge un nuovo host autoregistrato, usando l'indirizzo IP e la porta dell'agent ricevuti.

Per ulteriori informazioni, vedere la pagina autoregistrazione dell'agent attivo.

Eventi interni Gli eventi interni si verificano quando:

- un item cambia stato da 'normal' a 'unsupported'
- un item cambia stato da 'unsupported' a 'normal'
- una regola di discovery a basso livello cambia stato da 'normal' a 'unsupported'
- una regola di discovery a basso livello cambia stato da 'unsupported' a 'normal'
- un trigger cambia stato da 'normal' a 'unknown'
- un trigger cambia stato da 'unknown' a 'normal'

L'obiettivo dell'introduzione degli eventi interni è consentire agli utenti di essere avvisati quando si verifica un qualsiasi evento interno, ad esempio quando un item diventa unsupported e smette di raccogliere dati.

Gli eventi interni vengono creati solo quando le azioni interne per questi eventi sono abilitate. Per interrompere la generazione di eventi interni (ad esempio, per item che diventano unsupported), disabilita tutte le azioni per gli eventi interni in *Alerts* > *Actions* > *Internal actions*.

> **Note:**
> Se le azioni interne sono disabilitate, mentre un oggetto si trova nello stato 'unsupported', verrà comunque creato un evento di recovery per questo oggetto.
> Se le azioni interne sono abilitate, mentre un oggetto si trova nello stato 'unsupported', verrà creato un evento di recovery per questo oggetto, anche se per l'oggetto non è stato creato alcun 'problem event'.

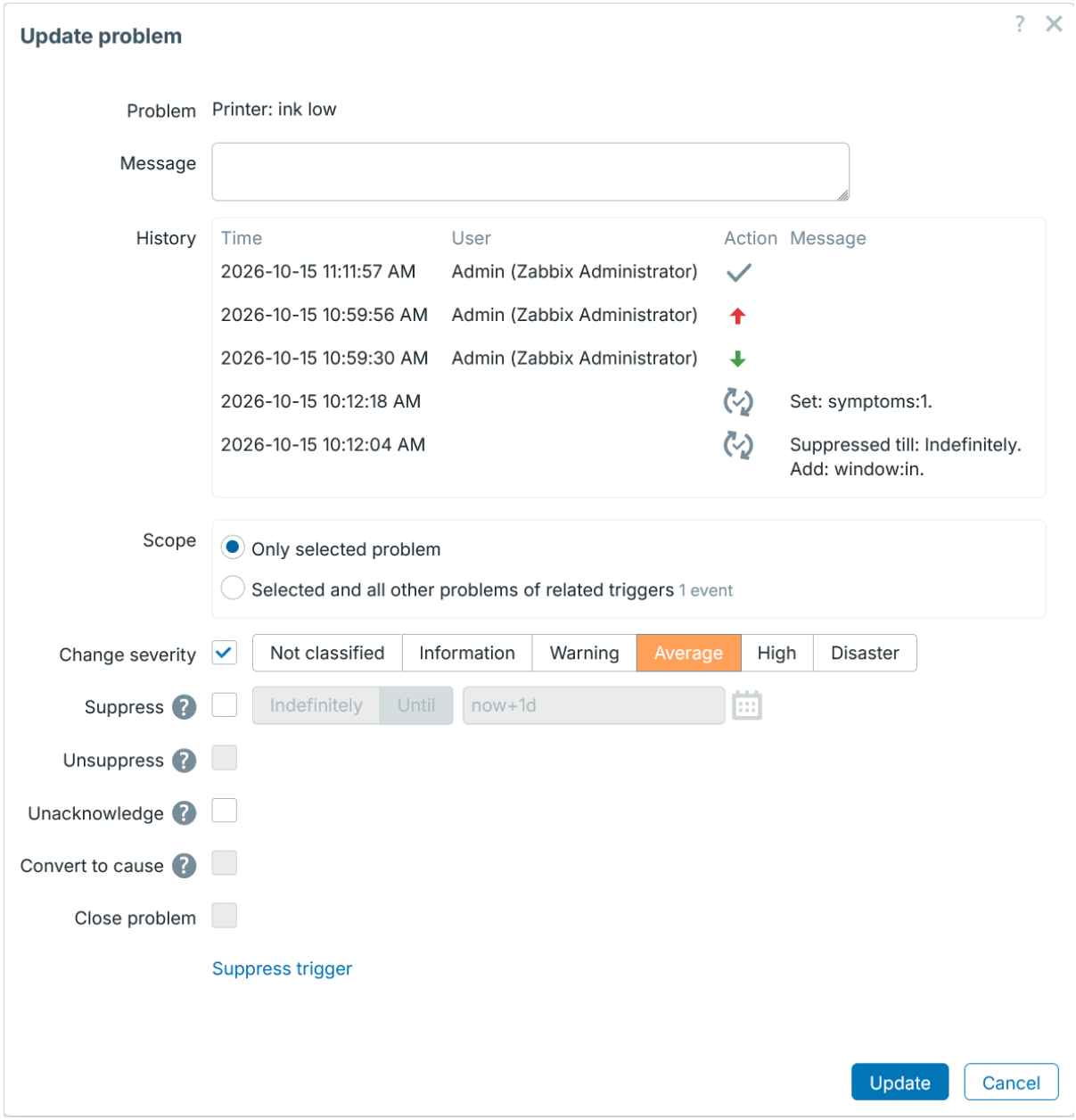
Vedi anche: Ricevere notifiche sugli item unsupported

Riconoscimento e soppressione dei problemi

Gli eventi di problema in Zabbix possono essere riconosciuti o soppressi dagli utenti.

Aggiornamento dei problemi Il popup di aggiornamento del problema consente di:

- Commentare il problema.
- Visualizzare i commenti e le azioni eseguite finora.
- Modificare la gravità del problema.
- Sopprimere/rimuovere la soppressione del problema.
- Confermare/annullare la conferma del problema.
- Convertire il problema sintomo in un problema causa.
- Chiudere manualmente il problema.
- Creare una manutenzione ad hoc per il trigger.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Problema	Se è selezionato un solo problema, viene visualizzato il nome del problema. Se sono selezionati più problemi, viene visualizzato *N problemi selezionati*.
Messaggio	Inserire il testo del commento sul problema (massimo 2048 caratteri).
Cronologia	Vengono elencate le attività e i commenti precedenti relativi al problema, insieme all'ora e ai dettagli dell'utente. Per il significato delle icone utilizzate per indicare le azioni degli utenti, vedere la pagina dettagli dell'evento. La cronologia viene visualizzata solo se per l'aggiornamento è selezionato un singolo problema.
Ambito	Definire l'ambito di azioni quali la modifica della gravità, la conferma o la chiusura manuale dei problemi: **Solo problema selezionato** - avrà effetto solo su questo evento **Problema selezionato e tutti gli altri problemi dei trigger correlati** - in caso di conferma/chiusura del problema, avrà effetto su questo evento e su tutti gli altri problemi che non sono ancora stati confermati/chiusi. Se l'ambito contiene problemi già confermati o chiusi, questi problemi non verranno confermati/chiusi nuovamente. D'altra parte, il numero di operazioni di modifica del messaggio e della gravità non è limitato.

Parametro	Descrizione
Modifica gravità	Selezionare la casella di controllo e fare clic sul pulsante della gravità per aggiornare la gravità del problema. La casella di controllo per la modifica della gravità è disponibile se esistono autorizzazioni di lettura-scrittura per almeno uno dei problemi selezionati. Facendo clic su *Aggiorna*, verranno aggiornati solo i problemi per i quali sono disponibili autorizzazioni di lettura-scrittura. Se non esistono autorizzazioni di lettura-scrittura per nessuno dei trigger selezionati, la casella di controllo è disabilitata.
Sopprimi	Selezionare la casella di controllo per sopprimere il problema: **Indefinitamente** - sopprime il problema indefinitamente **Fino a** - sopprime il problema fino a un'ora specificata. Sono supportati entrambi i formati temporali assoluto e relativo, ad esempio: `now+1d` - per un giorno a partire da ora (predefinito) `now/w` - fino alla fine della settimana corrente `2022-05-28 12:00:00` - fino alla data/ora assoluta Un periodo semplice (ad esempio `1d`, `1w`) non è supportato. La disponibilità di questa opzione dipende dalle impostazioni del ruolo utente "Sopprimi problemi". Vedere anche: Soppressione dei problemi
Rimuovi soppressione	Selezionare la casella di controllo per rimuovere la soppressione del problema. Questa casella di controllo è attiva solo se almeno uno dei problemi selezionati è soppresso. La disponibilità di questa opzione dipende dalle impostazioni del ruolo utente "Sopprimi problemi".
Conferma	Selezionare la casella di controllo per confermare il problema. Questa casella di controllo è disponibile se tra quelli selezionati è presente almeno un problema non confermato. Non è possibile aggiungere un'altra conferma a un problema già confermato, anche se è possibile aggiungere un altro commento.
Annulla conferma	Selezionare la casella di controllo per annullare la conferma del problema. Questa casella di controllo è disponibile se tra quelli selezionati è presente almeno un problema confermato.
Converti in causa	Selezionare la casella di controllo per convertire i problemi sintomo in problemi causa.
Chiudi problema	Selezionare la casella di controllo per chiudere manualmente i problemi selezionati. La casella di controllo per la chiusura di un problema è disponibile se l'opzione *Consenti chiusura manuale* è selezionata nella configurazione del trigger per almeno uno dei problemi selezionati. Facendo clic su *Aggiorna*, verranno chiusi solo i problemi che possono essere chiusi. Se nessun problema può essere chiuso manualmente, la casella di controllo è disabilitata. I problemi già chiusi non verranno chiusi nuovamente.
Sopprimi trigger	Fare clic per avviare una manutenzione ad hoc per il trigger. Verrà aperto un modulo di configurazione del periodo di manutenzione. In esso sarà preselezionato un periodo di manutenzione una tantum di 1h (configurabile nelle impostazioni e nel profilo dell'utente) a partire da ora. Anche il trigger corrispondente sarà preselezionato. Il nome della manutenzione sarà precompilato nel formato `Ad-hoc: <event name>`.

Conferma del problema Se un utente riceve una notifica di un evento di problema, può andare nel frontend di Zabbix, aprire la finestra popup di aggiornamento del problema di quel problema usando uno dei modi elencati di seguito e confermare il problema. Durante la conferma, può inserire un commento, ad esempio per indicare che sta lavorando al problema o qualsiasi altra osservazione desideri aggiungere.

In questo modo, se un altro utente del sistema individua lo stesso problema, può vedere immediatamente se è già stato confermato e quali commenti sono stati inseriti finora.

In questo modo, il flusso di lavoro per la risoluzione dei problemi con più di un utente del sistema può svolgersi in modo coordinato.

Lo stato di conferma viene utilizzato anche quando si definiscono le operazioni dell'azione. Ad esempio, è possibile definire che una notifica venga inviata a un manager di livello superiore solo se un evento non è stato confermato per un certo periodo di tempo.

Per confermare gli eventi e aggiungere commenti, un utente deve avere almeno i permessi di sola lettura sui trigger corrispondenti. Per modificare la gravità del problema o chiudere il problema, un utente deve avere i permessi di lettura e scrittura sui trigger corrispondenti.

Esistono **diversi** modi per accedere alla finestra popup di aggiornamento del problema, che consente di confermare un problema.

- È possibile selezionare i problemi in *Monitoring* > *Problems* e quindi fare clic su *Mass update* sotto l'elenco

- È possibile fare clic su *Update* nella colonna *Update* di un problema in:
  - *Dashboards* (widget *Problems* e *Problems by severity*)
  - *Monitoring > Problems*
  - *Monitoring > Problems > Event details*
- È possibile fare clic su una cella di un problema non risolto in:
  - *Dashboards* (widget *Trigger overview*)

Il menu popup contiene un'opzione *Update* che porta alla finestra di aggiornamento del problema.

In base alle informazioni di riconoscimento, è possibile configurare la modalità di visualizzazione del numero di problemi nella dashboard o nelle mappe. A tale scopo, è necessario effettuare le selezioni nell'opzione *Visualizzazione dei problemi*, disponibile sia nella configurazione della mappa sia nel widget della dashboard *Problemi per gravità*.

È possibile visualizzare il numero totale di problemi, il numero di problemi non riconosciuti separatamente dal totale oppure solo il numero di problemi non riconosciuti.

In base alle informazioni sugli aggiornamenti dei problemi (riconoscimento e così via), è possibile configurare le operazioni di aggiornamento: inviare un messaggio o eseguire comandi remoti.

Soppressione dei problemi La soppressione dei problemi consente di nascondere temporaneamente un problema per gestirlo in un secondo momento. È utile per ripulire l'elenco dei problemi e dare la massima priorità alle questioni più urgenti. Ad esempio, a volte durante il fine settimana può verificarsi un problema che non è abbastanza urgente da essere gestito immediatamente; in questo caso può essere "posticipato" fino a lunedì mattina.

La soppressione dei problemi consente di nascondere un *singolo* problema, a differenza della soppressione tramite la manutenzione dell'host, durante la quale vengono nascosti tutti i problemi dell'host in manutenzione.

Le operazioni per le azioni trigger verranno sospese per i problemi soppressi nello stesso modo in cui avviene con la manutenzione dell'host.

Un problema può essere soppresso tramite la finestra **aggiornamento del problema**, in cui la soppressione è una delle opzioni di aggiornamento del problema insieme a commentare, modificare la gravità, confermare, ecc.

Un problema può anche essere desoppresso tramite la stessa finestra di aggiornamento del problema.

Visualizzazione

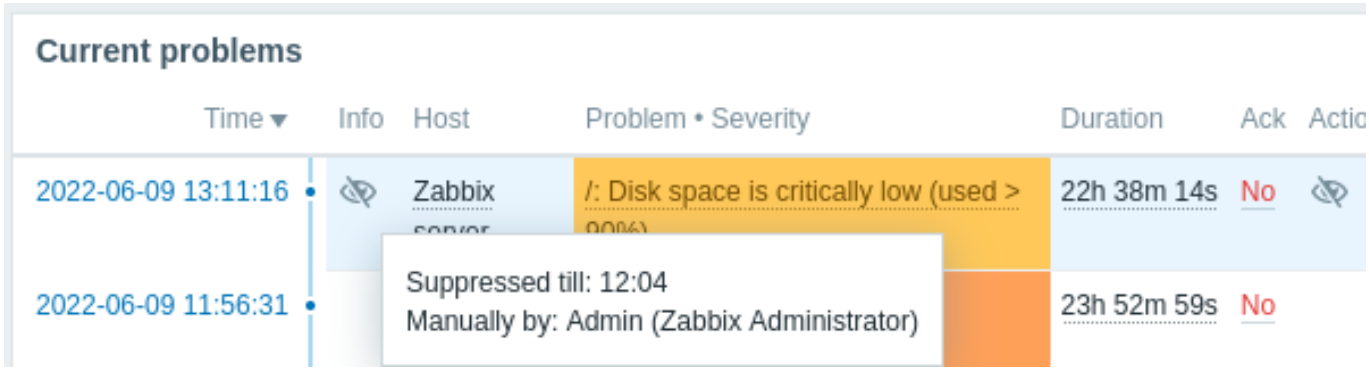
Una volta soppresso, il problema viene contrassegnato da un'icona di soppressione lampeggiante  nella colonna *Info*, prima di essere nascosto.

Se la soppressione è dovuta alla manutenzione dell'host, l'icona di soppressione nella colonna *Info* viene mostrata in arancione. Se si applicano sia la manutenzione sia la soppressione manuale, la soppressione manuale ha la priorità nel colore dell'icona.

L'icona di soppressione lampeggia mentre l'attività di soppressione è nella lista di attesa. Una volta che il task manager ha soppresso il problema, l'icona smette di lampeggiare. Se l'icona di soppressione continua a lampeggiare per molto tempo, ciò può indicare un problema del server, ad esempio se il server è inattivo e il task manager non riesce a completare l'attività. La stessa logica si applica alla desoppressione. Nel breve periodo dopo l'invio dell'attività e prima che il server l'abbia completata, l'icona di desoppressione lampeggia.

Un problema soppresso può essere sia nascosto sia mostrato, a seconda delle impostazioni del filtro/widget dei problemi.

Quando viene mostrato nell'elenco dei problemi, un problema soppresso è contrassegnato dall'icona di soppressione e i dettagli della soppressione vengono mostrati al passaggio del mouse:



I dettagli della soppressione vengono inoltre visualizzati in un popup quando si posiziona il mouse sull'icona di soppressione nella colonna *Azioni*.

Chiusura manuale dei problemi

Panoramica Sebbene in generale gli eventi di problema vengano risolti automaticamente quando lo stato del trigger passa da `Problem` a `OK`, possono esserci casi in cui è difficile determinare se un problema sia stato risolto tramite un'espressione del trigger. In tali casi, il problema deve essere risolto manualmente.

Ad esempio, *syslog* può segnalare che alcuni parametri del kernel devono essere regolati per ottenere prestazioni ottimali. In questo caso il problema viene segnalato agli amministratori Linux, che lo risolvono e poi chiudono manualmente il problema.

I problemi possono essere chiusi manualmente solo per i trigger con l'opzione *Allow manual close* abilitata.

Quando un problema viene "chiuso manualmente", Zabbix genera una nuova attività interna per Zabbix server. Quindi il processo *task manager* esegue questa attività e genera un evento OK, chiudendo quindi l'evento di problema.

L'evento *OK* generato include l'insieme completo dei tag dell'evento risolti per quell'evento (inclusi i tag ereditati da template, host e trigger). Questi tag sono disponibili nelle notifiche e in macro come `{EVENT.RECOVERY.TAGS}` e `{EVENT.RECOVERY.TAGSJSON}`.

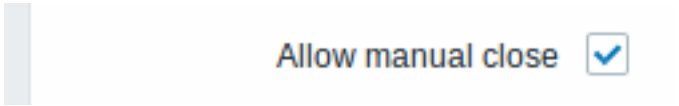
Un problema chiuso manualmente non significa che il trigger sottostante non entrerà mai più nello stato *Problem*. L'espressione del trigger viene rivalutata e può risultare in un problema:

- Quando arrivano nuovi dati per qualsiasi item incluso nell'espressione del trigger (si noti che i valori scartati da una fase di preprocessing di throttling non sono considerati come ricevuti e non causeranno la rivalutazione dell'espressione del trigger);
- Quando nell'espressione vengono utilizzate funzioni date and time e/o nodata().

Configurazione Sono necessari due passaggi per chiudere manualmente un problema.

Configurazione del trigger

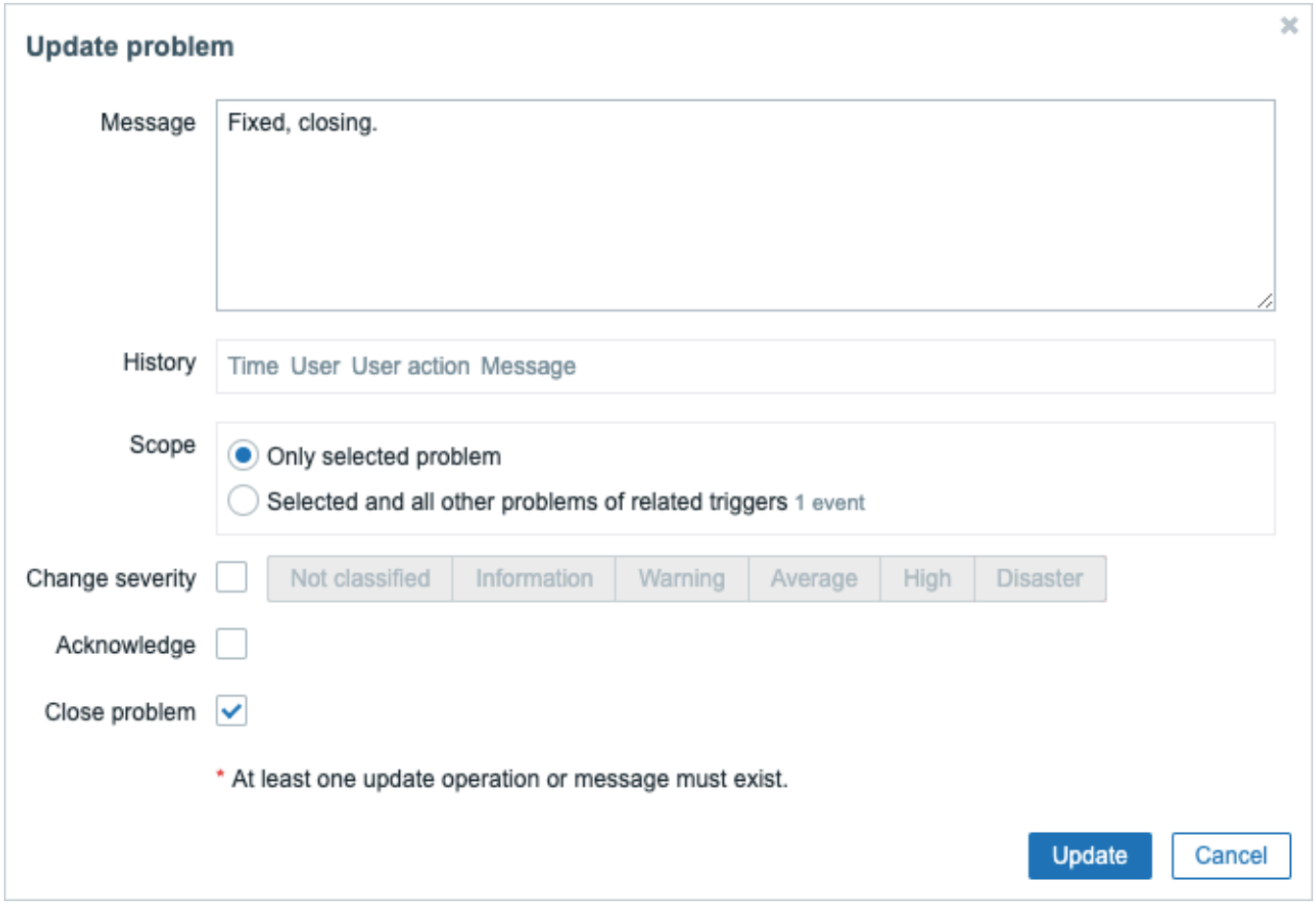
Nella configurazione del trigger, abilita l'opzione *Consenti chiusura manuale*.



Finestra di aggiornamento del problema

Se si verifica un problema per un trigger con il flag *Manual close*, è possibile aprire la finestra popup di aggiornamento del problema di quel problema e chiuderlo manualmente.

Per chiudere il problema, selezionare l'opzione *Close problem* nel modulo e fare clic su *Update*.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

La richiesta viene elaborata da Zabbix server. Normalmente occorrono alcuni secondi per chiudere il problema. Durante tale processo, *CLOSING* viene visualizzato in *Monitoring* > *Problems* come stato del problema.

Verifica È possibile verificare che un problema sia stato chiuso manualmente:

- nei dettagli dell'evento, disponibili tramite *Monitoring* > *Problems*;
- utilizzando la macro `{EVENT.UPDATE.HISTORY}` nei messaggi di notifica, che fornirà queste informazioni.

Periodi di manutenzione

Panoramica

La manutenzione viene utilizzata per sopprimere i problemi durante periodi di tempo predefiniti.

È possibile definire periodi di manutenzione per le seguenti entità:

- Gruppi di host
- Host
- Trigger
- Nomi degli eventi
- Tag

Se la manutenzione viene definita per tag o nomi degli eventi specifici, verrà attivata solo per i trigger corrispondenti; tutti gli altri trigger dell'host o del gruppo di host non saranno in manutenzione.

Esistono due tipi di manutenzione: *Con raccolta dati* e *Senza raccolta dati*.

Durante la manutenzione *Con raccolta dati*, i trigger vengono elaborati normalmente e gli eventi vengono creati quando necessario. Tuttavia, se l'opzione *Sospendi le operazioni per i problemi soppressi* è selezionata nella configurazione dell'azione, le escalation dei problemi vengono sospese per le entità in manutenzione. In questo caso, i passaggi dell'escalation che possono includere l'invio di notifiche o comandi remoti verranno ignorati per tutta la durata del periodo di manutenzione. Si noti che le operazioni di ripristino e aggiornamento dei problemi non vengono soppresse durante la manutenzione, ma solo le escalation. Le notifiche di ripristino non vengono inviate se il problema è iniziato durante la manutenzione.

Ad esempio, se i passaggi dell'escalation sono pianificati a 0, 30 e 60 minuti dall'inizio di un problema e si verifica una manutenzione della durata di mezz'ora, compresa tra 10 e 40 minuti dopo il verificarsi di un problema reale, i passaggi due e tre verranno eseguiti mezz'ora più tardi, ovvero a 60 e 90 minuti (a condizione che il problema persista). Analogamente, se un problema si verifica durante la manutenzione, l'escalation inizierà dopo la manutenzione.

Per ricevere normalmente le notifiche dei problemi durante la manutenzione (senza ritardi), è necessario deselezionare l'opzione *Sospendi le operazioni per i problemi soppressi* nella configurazione dell'azione.

Note:
Se almeno un host (utilizzato nell'espressione del trigger) non è in modalità di manutenzione, Zabbix invierà una notifica del problema.

Il server Zabbix deve essere in esecuzione durante la manutenzione. Le manutenzioni vengono ricalcolate ogni minuto o non appena la cache di configurazione viene ricaricata, se sono presenti modifiche al periodo di manutenzione.

I processi timer verificano se lo stato dell'host deve essere modificato in modalità di manutenzione o rimosso da tale modalità al secondo 0 di ogni minuto. Inoltre, ogni secondo il processo timer verifica se è necessario avviare o interrompere una manutenzione in base alla presenza di modifiche ai periodi di manutenzione dopo l'aggiornamento della configurazione. Pertanto, la velocità di avvio o interruzione dei periodi di manutenzione dipende dall'intervallo di aggiornamento della configurazione (10 secondi per impostazione predefinita). Si noti che le modifiche al periodo di manutenzione non includono le impostazioni *Attivo da/Attivo fino a*. Inoltre, se un'entità (gruppo di host/host/trigger/ecc.) viene aggiunta a un periodo di manutenzione attivo esistente, le modifiche verranno attivate dal processo timer solo all'inizio del minuto successivo.

Si noti che quando un host entra in manutenzione, i processi timer del server Zabbix leggeranno tutti i problemi aperti per verificare se è necessario sopprimerli. Ciò può influire sulle prestazioni se sono presenti molti problemi aperti. Il server Zabbix leggerà inoltre tutti i problemi aperti all'avvio, anche se in quel momento non è configurata alcuna manutenzione.

Si noti che il server Zabbix (o il proxy) raccoglie sempre i dati indipendentemente dal tipo di manutenzione (inclusa la manutenzione *Senza raccolta dati*). I dati vengono successivamente ignorati dal server se è impostata l'opzione *Senza raccolta dati*.

Al termine della manutenzione *Senza raccolta dati*, i trigger che utilizzano la funzione `nodata()` non si attiveranno prima del controllo successivo durante il periodo che stanno verificando.

Se un item di tipo log viene aggiunto mentre un host è in manutenzione e la manutenzione termina, verranno raccolte solo le nuove voci del file di log successive alla fine della manutenzione.

Se viene inviato un valore con timestamp per un host che si trova in manutenzione *Senza raccolta dati* (ad esempio utilizzando Zabbix sender), tale valore verrà eliminato; tuttavia, è possibile inviare un valore con timestamp relativo a un periodo di manutenzione scaduto, che verrà accettato.

Se il periodo di manutenzione o l'opzione di manutenzione (gruppo di host, host, trigger, ecc.) vengono modificati dall'utente, le modifiche avranno effetto solo dopo la sincronizzazione della cache di configurazione.

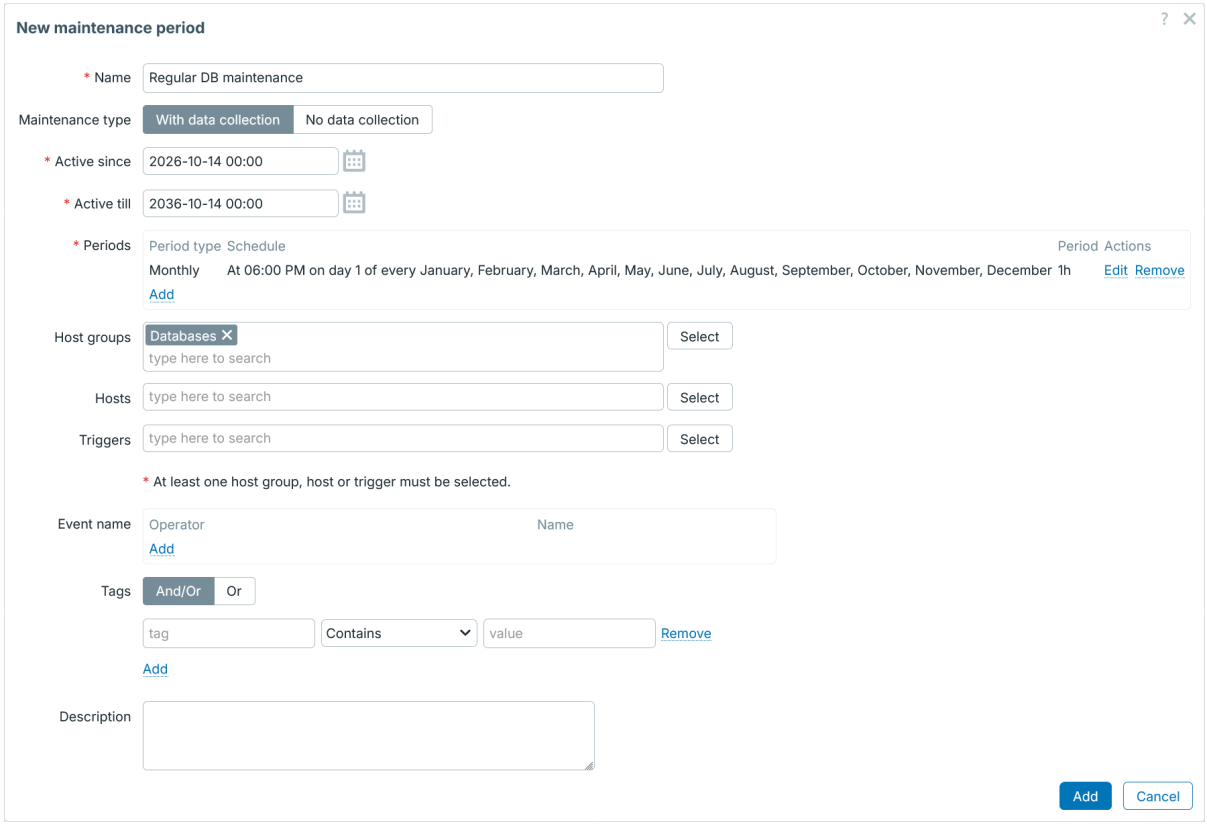
Configurazione

Per avviare immediatamente un periodo di manutenzione:

1. Vai a *Monitoring* > *Problems*.
2. Fai clic sul nome del problema.
3. Seleziona l'opzione di manutenzione dal menu dell'evento (host, trigger, nome dell'evento, tag).
4. Modifica i parametri di manutenzione nel modulo.

Per configurare un periodo di manutenzione pianificato:

1. Vai a *Data collection* > *Maintenance*.
2. Fai clic su *Create maintenance period* (oppure sul nome di un periodo di manutenzione esistente).
3. Inserisci i parametri di manutenzione nel modulo.



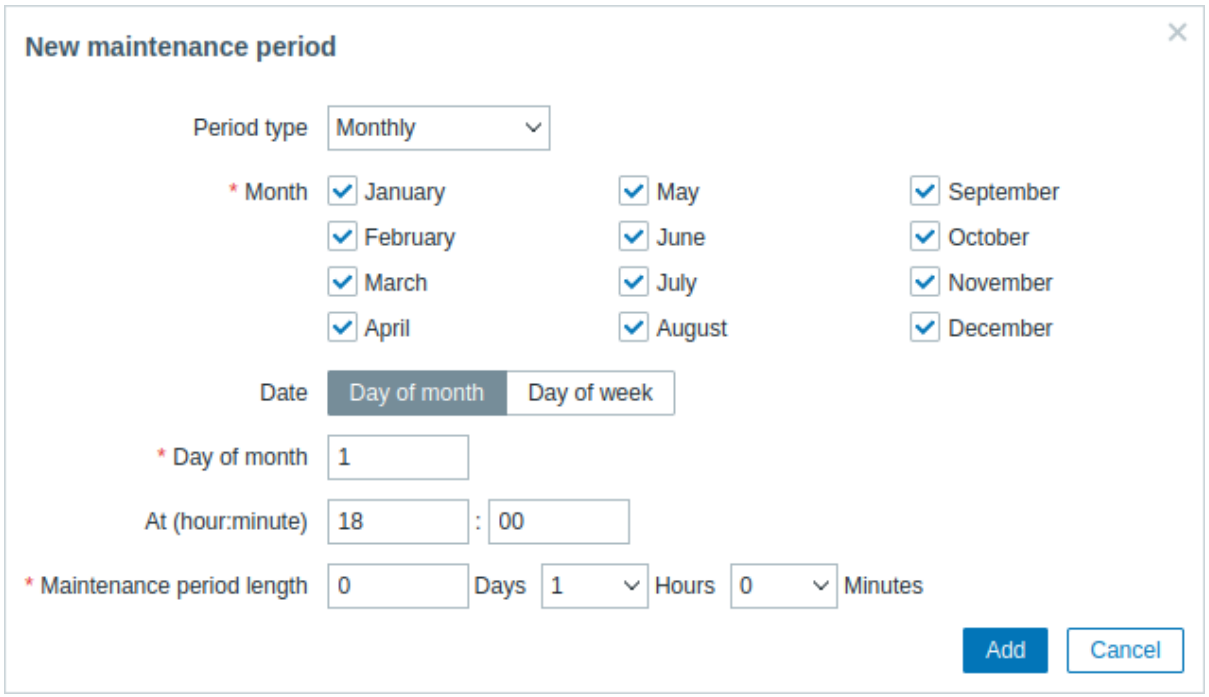
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Name	Nome del periodo di manutenzione.
Maintenance type	È possibile impostare due tipi di manutenzione: **With data collection** - i dati vengono raccolti dal server durante la manutenzione e i trigger vengono elaborati; **No data collection** - i dati possono essere comunque raccolti, ma non vengono salvati nel database durante la manutenzione e i trigger (inclusa la funzione `nodata()`) non si attivano. Per informazioni su come ciascun tipo influisce sul report di disponibilità, vedere Effect of maintenance periods.
Active since	La data e l'ora a partire dalle quali l'esecuzione dei periodi di manutenzione diventa attiva. *Nota:* l'impostazione della sola ora non attiva un periodo di manutenzione; i periodi di manutenzione devono essere configurati in *Periods* (vedere di seguito).
Active till	La data e l'ora fino alle quali l'esecuzione dei periodi di manutenzione rimane attiva.
Periods	Questo blocco consente di definire i giorni e gli orari esatti in cui si svolge la manutenzione. Facendo clic su Add si apre una finestra popup con un modulo flessibile *Maintenance period*, in cui è possibile definire la pianificazione della manutenzione. Per una descrizione dettagliata, vedere Maintenance periods.
Host groups	Seleziona i gruppi di host per i quali verrà attivata la manutenzione. La manutenzione verrà attivata per tutti gli host appartenenti ai gruppi di host specificati. Questo campo supporta il completamento automatico: iniziando a digitare, verrà visualizzato un elenco a discesa di tutti i gruppi di host disponibili. La specifica di un gruppo di host padre seleziona implicitamente tutti i gruppi di host annidati. Pertanto, la manutenzione verrà attivata anche sugli host appartenenti ai gruppi annidati.
Hosts	Seleziona gli host per i quali verrà attivata la manutenzione. Questo campo supporta il completamento automatico: iniziando a digitare, verrà visualizzato un elenco a discesa di tutti gli host disponibili.
Triggers	Seleziona i trigger per i quali verrà attivata la manutenzione. Questo campo supporta il completamento automatico: iniziando a digitare, verrà visualizzato un elenco a discesa di tutti i trigger disponibili.

Parametro	Descrizione
Event name	Specifica i nomi degli eventi per i quali verrà attivata la manutenzione. È possibile impostare diverse condizioni. Per ogni condizione sono disponibili due operatori: **Contains** - include i nomi dei tag specificati quando i valori dei tag contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole) **Does not contain** - esclude i nomi dei tag specificati quando i valori dei tag contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole)
Tags	Specifica i tag per i quali verrà attivata la manutenzione. È possibile impostare diverse condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Contains** - include i nomi dei tag specificati quando i valori dei tag contengono la stringa inserita (corrispondenza di sottostringa, con distinzione tra maiuscole e minuscole); **Equals** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole). **Does not equal** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole) **Does not contain** - esclude i nomi dei tag specificati quando i valori dei tag contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole) Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - devono essere soddisfatte tutte le condizioni; le condizioni con lo stesso nome di tag vengono raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione. I tag possono essere specificati solo se è selezionato il tipo di manutenzione *With data collection*.
Description	Descrizione del periodo di manutenzione.

Periodi di manutenzione

La finestra del periodo di manutenzione serve per pianificare il tempo di una manutenzione ricorrente o una tantum. Il modulo è dinamico e i campi disponibili cambiano in base al *Tipo di periodo* selezionato.



Tipo di periodo	Descrizione
Solo una volta	Configura un periodo di manutenzione una tantum: *Data* - data e ora del periodo di manutenzione; *Durata del periodo di manutenzione* - per quanto tempo la manutenzione sarà attiva.

Tipo di periodo	Descrizione
Giornaliero	Configura un periodo di manutenzione giornaliero: *Ogni giorno(i)* - frequenza della manutenzione (1 - *(predefinito)* ogni giorno, 2 - ogni due giorni, ecc.); *Alle (ora:minuto)* - ora del giorno in cui inizia la manutenzione; *Durata del periodo di manutenzione* - per quanto tempo la manutenzione sarà attiva. Quando il parametro *Ogni giorno(i)* è maggiore di "1", il giorno di inizio è quello in cui cade l'orario *Attivo da*. Esempi: - se *Attivo da* è impostato su "2021-01-01 12:00", *Ogni giorno(i)* è impostato su "2" e *Alle (ora:minuto)* è impostato su "23:00", allora il primo periodo di manutenzione inizierà il 1 gennaio alle 23:00, mentre il secondo periodo di manutenzione inizierà il 3 gennaio alle 23:00; - se *Attivo da* è impostato su "2021-01-01 12:00", *Ogni giorno(i)* è impostato su "2" e *Alle (ora:minuto)* è impostato su "01:00", allora il primo periodo di manutenzione inizierà il 3 gennaio alle 01:00, mentre il secondo periodo di manutenzione inizierà il 5 gennaio alle 01:00.
Settimanale	Configura un periodo di manutenzione settimanale: *Ogni settimana(e)* - frequenza della manutenzione (1 - *(predefinito)* ogni settimana, 2 - ogni due settimane, ecc.); *Giorno della settimana* - in quale giorno deve svolgersi la manutenzione; *Alle (ora:minuto)* - ora del giorno in cui inizia la manutenzione; *Durata del periodo di manutenzione* - per quanto tempo la manutenzione sarà attiva. Quando il parametro *Ogni settimana(e)* è maggiore di "1", la settimana di inizio è quella in cui cade l'orario *Attivo da*. Per esempi, vedere la descrizione del parametro *Giornaliero* sopra.
Mensile	Configura un periodo di manutenzione mensile: *Mese* - seleziona tutti i mesi durante i quali viene eseguita la manutenzione regolare; *Data:* ***Giorno del mese*** - seleziona questa opzione se la manutenzione deve svolgersi nello stesso giorno di ogni mese (ad esempio, ogni 1° giorno del mese), quindi seleziona il giorno richiesto nel campo *Giorno del mese* che appare; *Data:* ***Giorno della settimana*** - seleziona questa opzione se la manutenzione deve svolgersi solo in determinati giorni (ad esempio, ogni primo lunedì del mese), quindi seleziona (nel menu a discesa) la settimana del mese richiesta (prima, seconda, terza, quarta o ultima) e poi seleziona le caselle di controllo per il/i giorno/i di manutenzione; *Alle (ora:minuto)* - ora del giorno in cui inizia la manutenzione; *Durata del periodo di manutenzione* - per quanto tempo la manutenzione sarà attiva.

Attention:
Quando si crea un periodo di manutenzione, viene utilizzato il fuso orario dell'utente che lo crea. Tuttavia, quando vengono pianificati periodi di manutenzione ricorrenti (*Daily*, *Weekly*, *Monthly*), viene utilizzato il fuso orario del server Zabbix. Per garantire un comportamento prevedibile dei periodi di manutenzione ricorrenti, è necessario utilizzare un fuso orario comune per tutte le parti di Zabbix.

Quando hai terminato, fai clic su *Add* per aggiungere il periodo di manutenzione alla sezione *Periods*.

Nota che le modifiche dell'ora legale (DST) non influiscono sulla durata della manutenzione. Ad esempio, supponiamo di avere una manutenzione configurata di due ore che in genere inizia alle 01:00 e termina alle 03:00:

- se dopo un'ora di manutenzione (alle 02:00) si verifica un cambio DST e l'ora corrente passa da 02:00 a 03:00, la manutenzione continuerà per un'altra ora (fino alle 04:00);
- se dopo due ore di manutenzione (alle 03:00) si verifica un cambio DST e l'ora corrente passa da 03:00 a 02:00, la manutenzione terminerà, perché sono trascorse due ore;
- se un periodo di manutenzione inizia durante l'ora che viene saltata da un cambio DST, la manutenzione non inizierà.

Se un periodo di manutenzione è impostato su "1 day" (il periodo effettivo della manutenzione è di 24 ore, poiché Zabbix calcola i giorni in ore), inizia alle 00:00 e termina alle 00:00 del giorno successivo:

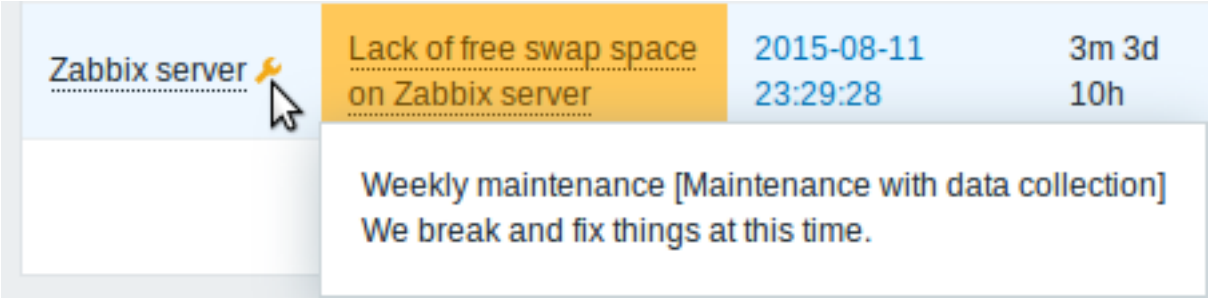
- la manutenzione terminerà alle 01:00 del giorno successivo se l'ora corrente avanza di un'ora;
- la manutenzione terminerà alle 23:00 di quel giorno se l'ora corrente torna indietro di un'ora.

Visualizzazione

Visualizzazione degli host in manutenzione

Un'icona a forma di chiave inglese arancione accanto al nome dell'host indica che questo host è in manutenzione in:

- *Dashboards*
- *Monitoring* > *Problems*
- *Inventory* > *Hosts* > *Host inventory details*
- *Data collection* > *Hosts* (vedere la colonna *Status*)



I dettagli della manutenzione vengono visualizzati quando il puntatore del mouse viene posizionato sull'icona.

Inoltre, gli host in manutenzione hanno uno sfondo arancione in *Monitoring* > *Maps*.

Visualizzazione dei problemi soppressi

Normalmente, i problemi per gli host in manutenzione vengono soppressi, cioè non vengono visualizzati nel frontend.

I problemi soppressi possono anche essere nascosti perché sono stati soppressi manualmente.

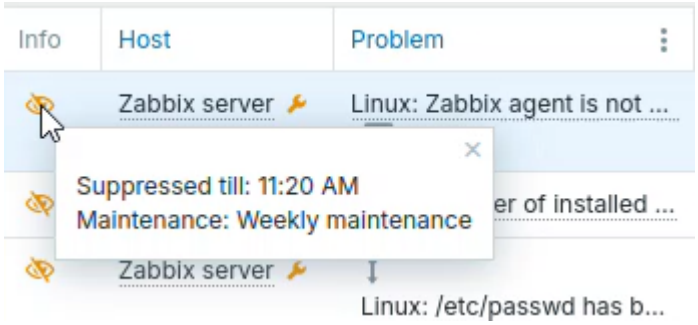
È possibile visualizzare i problemi soppressi selezionando l'opzione *Mostra problemi soppressi* in queste posizioni:

- *Dashboard* (nella configurazione dei widget *Host con problemi*, *Problemi*, *Problemi per gravità*, *Panoramica trigger*)
- *Monitoraggio* > *Problemi* (nel filtro)
- *Monitoraggio* > *Mappe* (nella configurazione della mappa)
- Notifiche globali (nella configurazione del profilo utente)

Quando i problemi soppressi vengono visualizzati, viene mostrata la seguente icona: . Se la soppressione è dovuta alla manutenzione dell'host, vengono mostrate le seguenti icone:

-  - il problema è stato soppresso a causa della manutenzione dell'host.
-  - il problema è stato desoppresso dopo la manutenzione dell'host.

Passando il puntatore sull'icona vengono visualizzati ulteriori dettagli. Nei dettagli dell'evento, è possibile verificare se l'evento è stato soppresso a causa della manutenzione o soppresso manualmente. Se si applicano sia la manutenzione sia la soppressione manuale, la soppressione manuale ha la priorità nel colore dell'icona.



Calcolo delle code durante la manutenzione

Le code visualizzate nel frontend di Zabbix (Administration > Queue) sono calcolate dal server di Zabbix. Non includono gli item in manutenzione *No data collection*, e la lunghezza della coda è sempre zero per questi item, anche quando i loro valori sono in ritardo. Gli item in manutenzione *With data collection* vengono comunque conteggiati nella coda.

Il proxy di Zabbix non è a conoscenza dei periodi di manutenzione perché non esiste una sincronizzazione della configurazione di manutenzione tra il server di Zabbix e il proxy. I controlli interni calcolati sui proxy di Zabbix (ad esempio, `zabbix[queue,,]` e `zabbix[stats,,,queue,,]`) segnalano gli item in ritardo indipendentemente dallo stato di manutenzione sul server di Zabbix.

Di conseguenza, per gli stessi item in manutenzione *No data collection* possono essere riportate lunghezze della coda diverse dal frontend di Zabbix e dai controlli interni sui proxy di Zabbix.

Etichettatura

I tag hanno molteplici scopi, in particolare per contrassegnare gli eventi. Quando le entità vengono taggate, qualsiasi nuovo evento relativo a un'entità taggata erediterà i suoi tag. Ad esempio:

- con template taggati - qualsiasi problema dell'host (creato dai trigger del template) erediterà i tag del template.
- con host taggati - qualsiasi problema dell'host erediterà i tag dell'host.
- con item/scenari web taggati - qualsiasi problema di item/scenario web erediterà i tag dell'item/scenario web.
- con trigger taggati - qualsiasi problema creato dal trigger erediterà i tag del trigger.

Un evento di problema eredita tutti i tag dall'intera catena di entità - template, host, item/scenari web, trigger. Le combinazioni identiche `tag:value` (dopo la risoluzione delle macro) vengono unite in una sola, evitando così duplicazioni.

Gli eventi di ripristino generati da una chiusura manuale includono anche i tag dell'evento risolti ed ereditati da template, host, item/scenari web e trigger. Questi tag sono disponibili nelle notifiche e tramite macro come `{EVENT.RECOVERY.TAGS}` e `{EVENT.RECOVERY.TAGSJSON}`.

I tag evento personalizzati offrono maggiore flessibilità. Ad esempio:

- correlazione degli eventi può essere configurata in base ai tag evento.
- condizioni delle azioni possono essere configurate in base ai tag evento.
- i problemi degli item possono essere raggruppati in base ai tag evento.
- i tag dei problemi possono essere usati per mappare i problemi ai servizi.

Le entità possono essere taggate con lo stesso nome di tag ma con valori di tag diversi (ad esempio, `component:memory` e `component:storage`). Allo stesso modo, un'entità può avere un tag senza valore e lo stesso tag con un valore (ad esempio, `database` e `database:postgresql`). Tali tag non sono considerati duplicati.

I tag sono composti da un nome di tag e da un valore di tag. Quando si taggano le entità, è possibile usare solo il nome oppure associarlo a un valore (ad esempio, `mysql`, `jira`, `target:mysql`, `service:jira`, ecc.).

I tag possono essere definiti per varie entità:

- Template
- Host
- Item
- Scenari web
- Trigger
- Servizi
- Item e trigger dei template
- Prototipi di host, item e trigger

Nelle liste *Template*, *Host*, *Item*, *Trigger*, *Scenari web* e nei rispettivi prototipi, la colonna *Tag* mostra sia i tag normali sia quelli ereditati. Passando il mouse sopra o facendo clic su un tag *ereditato* appare un tooltip JavaScript con il testo "Inherited tag". Se un tag ereditato esiste anche come tag normale, viene mostrato un testo tooltip diverso a seconda della lista in cui appare (ad esempio, "Inherited and template tag" nella lista template oppure "Inherited and host tag" nella lista host). In altre parti del frontend, i tag normali e quelli ereditati vengono visualizzati insieme (i duplicati vengono rimossi), senza icone e senza testo tooltip aggiuntivo.

> **Note:**
> Fare riferimento alle linee guida ufficiali di Zabbix per raccomandazioni generali sulla definizione dei tag, insieme alle indicazioni specifiche per template, item, trigger e regole di discovery a basso livello.

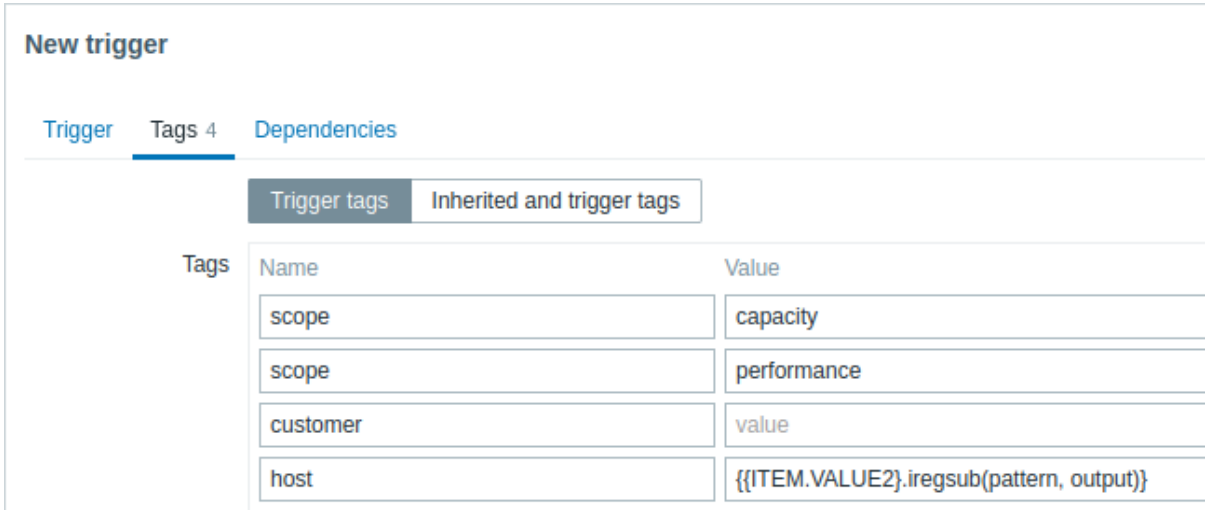
Casi d'uso Alcuni casi d'uso comuni per il tagging sono i seguenti:

1. Contrassegnare gli eventi di trigger:
   - Definire un tag del trigger (ad esempio, `scope:performance`).
   - I problemi creati da questo trigger avranno il tag del trigger.
2. Contrassegnare i problemi ereditati dal template:
   - Definire un tag del template (ad esempio, `target:mysql`).
   - I problemi creati dai trigger di questo template avranno il tag del template.
3. Contrassegnare i problemi dell'host:
   - Definire un tag dell'host (ad esempio, `service:jira`).
   - I problemi creati dai trigger di questo host avranno il tag dell'host.
4. Filtrare gli item correlati:

- Definire un tag dell'item (ad esempio, `component:cpu`).
- In *Monitoring* > *Latest data*, gli item possono essere filtrati in base al tag `component:cpu`.

5. Usare le informazioni estratte dal valore dell'item come valore del tag:
   - Definire un tag con una macro come valore del tag (ad esempio, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - In *Monitoring* > *Problems*, i problemi avranno il valore del tag risolto nei dati estratti dal valore dell'item.
6. Identificare i problemi in un file di log e chiuderli separatamente:
   - Definire un tag del trigger per il trigger dell'item di monitoraggio dei log che estrarrà valori dal valore dell'item usando una macro (ad esempio, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Nella configurazione del trigger, impostare event correlation:
     - impostare *PROBLEM event generation mode* su "Multiple";
     - impostare *OK event closes* su "All problems if tag values match";
     - impostare il tag per la corrispondenza.
   - I problemi creati dal trigger dell'item di log avranno il tag del trigger e verranno chiusi singolarmente.
7. Filtrare le notifiche:
   - Definire tag del trigger (ad esempio, `scope:security` per trigger1 e `scope:availability` per trigger2).
   - Usare il filtro dei tag nelle action conditions per ricevere notifiche solo sugli eventi che corrispondono ai dati del tag.
8. Identificare i problemi nelle notifiche:
   - Definire tag del trigger.
   - Usare la macro {EVENT.TAGS} nella notifica del problema.
   - La notifica del problema conterrà i tag del trigger, rendendo più facile identificare a quale applicazione/servizio appartiene la notifica.
9. Semplificare le attività di configurazione usando i tag del template:
   - Definire un tag del trigger del template.
   - I trigger creati da questo trigger del template avranno il suo tag.
10. Creare trigger con tag da low-level discovery (LLD):
    - Definire un tag del prototype del trigger con una macro LLD nel nome o nel valore del tag (ad esempio, `scope:{#FSNAME}`).
    - I trigger creati dal prototype del trigger avranno il suo tag.
11. Corrispondere i servizi usando i tag del servizio:
    - Definire service tags.
    - Configurare service actions per i servizi con tag corrispondenti.
    - Inoltre, usare i tag del servizio per collegare un servizio a un SLA per i calcoli SLA.
12. Collegare i servizi ai problemi usando i tag di problema del servizio:
    - Definire un problem tag nella configurazione del servizio (ad esempio, `target:mysql`).
    - I problemi con un tag corrispondente verranno automaticamente correlati al servizio e lo stato del servizio cambierà in base alle regole di calcolo dello stato del servizio configurate.
13. Sopprimere i problemi quando un host è in modalità di manutenzione:
    - Definire i tag nella configurazione del periodo di manutenzione.
    - I problemi con i tag definiti verranno soppressi.
14. Concedere l'accesso ai gruppi di utenti:
    - Definire i tag nella configurazione del gruppo di utenti.
    - Gli utenti del gruppo di utenti potranno visualizzare solo i problemi con i tag definiti.

Configurazione I tag possono essere definiti in una scheda dedicata, ad esempio, nella configurazione del trigger:



Supporto delle macro Le macro integrate e le macro utente nei tag vengono risolte al momento dell'evento. Fino a quando l'evento non si è verificato, queste macro verranno mostrate nel frontend di Zabbix come non risolte.

Le macro di discovery a basso livello vengono risolte durante il processo di discovery.

Le seguenti macro possono essere utilizzate nei nomi e nei valori dei tag dei trigger:

- macro integrate {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} e {HOST.ID}
- macro integrate {INVENTORY.*} (per fare riferimento ai valori dell'inventario dell'host da uno o più host in un'espressione del trigger)
- Macro utente e macro utente con contesto (il contesto può includere macro di discovery a basso livello)
- Macro di discovery a basso livello (solo nei tag del prototipo di trigger)

Le seguenti macro possono essere utilizzate nei nomi e nei valori dei tag di template, host e item/scenario web:

- macro integrate {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID}
- macro integrate {INVENTORY.*}
- Macro utente
- Macro di discovery a basso livello (solo nei tag del prototipo di host e item)

Le seguenti macro possono essere utilizzate nelle notifiche basate su trigger:

- macro integrate {EVENT.TAGS} e {EVENT.RECOVERY.TAGS} (queste macro verranno risolte in un elenco separato da virgole di tag dell'evento o di tag dell'evento di recupero)
- macro integrate {EVENT.TAGSJSON} e {EVENT.RECOVERY.TAGSJSON} (queste macro verranno risolte in un array JSON contenente oggetti tag dell'evento objects o oggetti tag dell'evento di recupero)

Estrazione di sottostringa nei tag dei trigger

L'estrazione di sottostringa è supportata per compilare il nome del tag o il valore del tag, usando una funzione macro. La funzione applica un'espressione regolare al valore ottenuto dalla macro supportata. Ad esempio:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

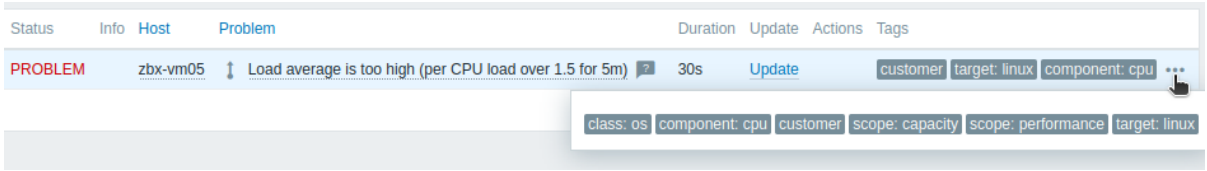
Se il nome o il valore del tag supera i 255 caratteri dopo la risoluzione della macro, verrà troncato a 255 caratteri.

Vedi anche: Uso delle funzioni macro nelle macro di discovery a basso livello per l'etichettatura degli eventi.

Visualizzazione dei tag degli eventi I tag, se definiti, possono essere visualizzati con i nuovi eventi in:

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Event details*
- *Dashboards* > *Problems* widget

L'ordine e il numero dei tag visualizzati sono determinati dalle opzioni di filtro *Tag display priority* e *Show tags* in *Monitoring* > *Problems* o nel widget *Problems* della dashboard. Si noti che è possibile visualizzare al massimo tre tag; se i tag sono più di tre, passando il mouse sui tre puntini vengono mostrati tutti i tag in una finestra pop-up.



Complex event processing

Overview Complex event processing (CEP) performs rule-based processing for Zabbix events.

For example, you might want to:

- group related events and mark first event in a series as cause (the following as symptoms)
- specify a time-to-live period for events

- specify a maximum number for certain events
- be notified about similar events by location
- be notified if some event happens too often (or not at all) within a time period

In addition to single event processing, you may define **time windows** for event analysis. Within a time window, all matching events form a single context and can be analyzed as a whole.

A sliding time window moves forward with the time. Older events leave the window (are ”evicted”), while new events may enter.

A tumbling time window does not move forward. When the window expires, a new window with the same duration is started. *Cause and symptom grouping* windows are a tumbling time window.

The time window concept presents logical points for executing operations. For example, an event can be transformed or ”enriched” when it enters the window - severity can be raised or lowered, tags can be added or adjusted, suppression can be activated, etc.

Similar operations can be executed when the event leaves the window or when a pattern is found.

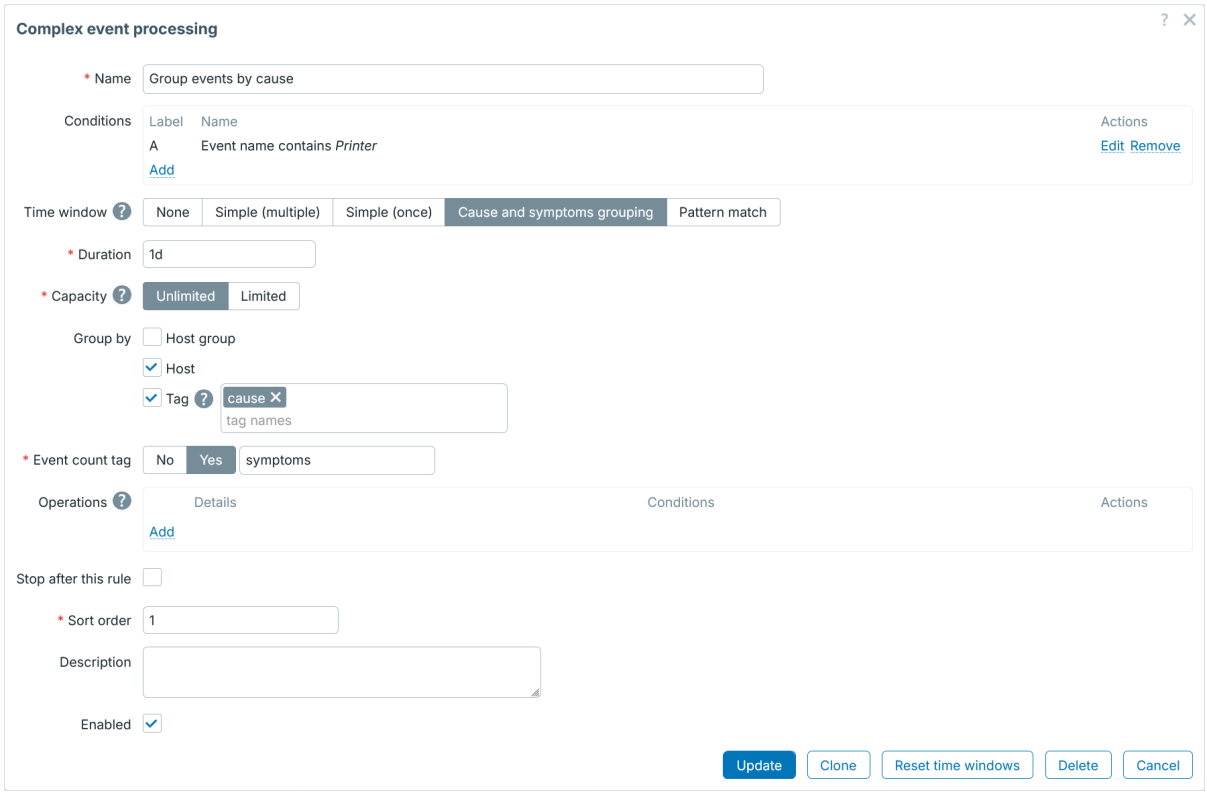
Event stream analysis is defined by CEP rules. A CEP rule consists of a time window (optional), conditions and operations. Operations are tied to an execution point.

Use cases

- Group related events, with the first such event automatically marked as cause (all following as symptoms).
- Enrich event with higher severity/specific tag if too many 401 events happen within a minute (possible bot activity).

Configuration To create a CEP rule in Zabbix frontend, do the following:

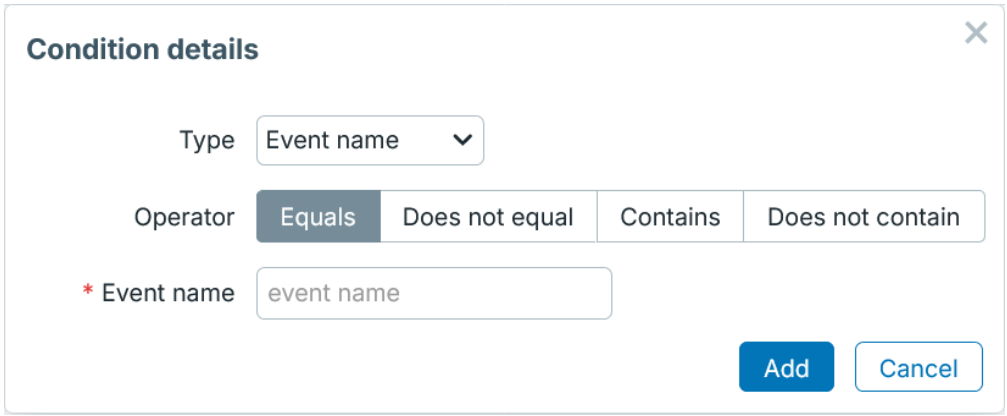
1. Go to *Data collection* > *Event processing*.
2. Click *Create complex event processing* in the upper-right corner of the screen.
3. Enter parameters of the rule in the form.



Parameter	Description
Name	Specify a unique CEP rule name.
Conditions	Specify conditions for associating events with this CEP rule.

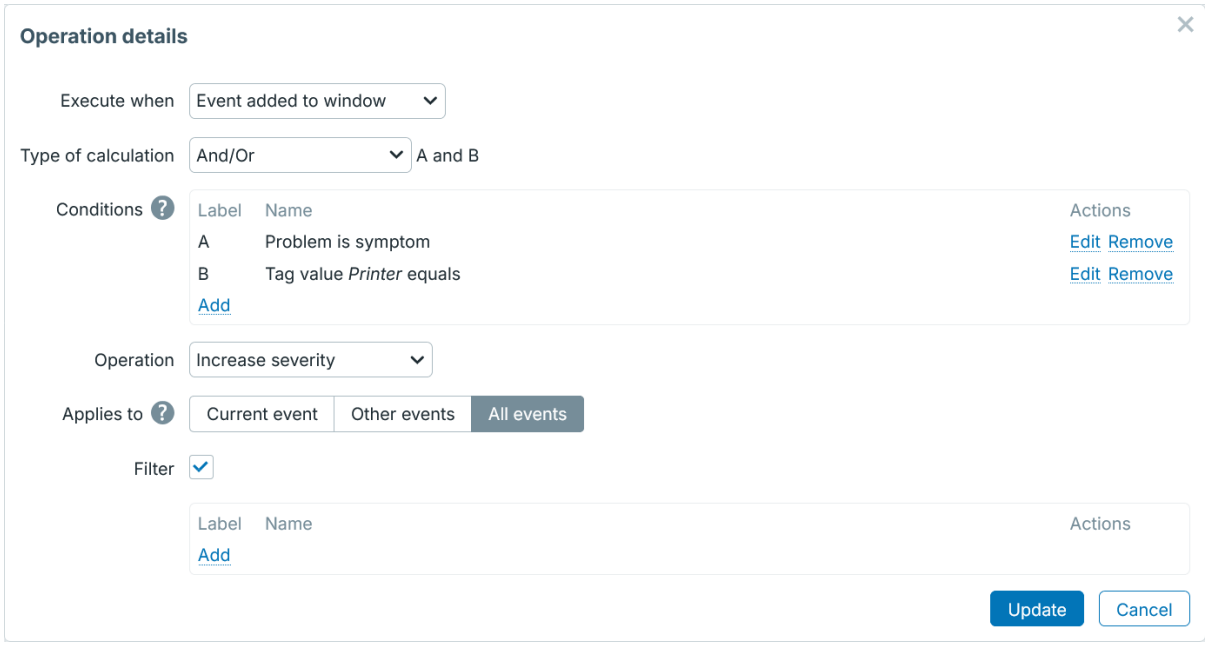
Parameter	Description
Time window	Select the time window type: **None** - process single events with no time window (default) **Simple (multiple)** - process events in a sliding time window (event can be added to multiple windows of this type) **Simple (once)** - process events in a sliding time window (event can be added only to one window of this type) **Cause and symptom grouping** - process first event in time window as cause, consecutive events as symptoms **Pattern match** - analyze sliding time window events based on a JavaScript script. The window is examined every second; events older than the duration are evicted, and the script is then run over the events that remain in it. When the script reports a match, the pattern match operations are executed for every event the window holds; the events stay in the window and are examined again a second later, until they age out. A window can be forcibly closed by the "Close window" operation.
Duration	Limit the time window duration (in seconds, 365 days max). The window duration is started by the first matching event. In *Simple* and *Pattern match* windows, the window is moving so older events will leave the window (are evicted) after the duration passes. In *Cause and symptom grouping* windows, window duration is a fixed timeframe for setting cause and symptom relationship. The window is started when the first event matches. When the duration expires, the window is closed and a new one is started. No event is evicted when the duration expires; periods run back to back. Time suffixes are supported (`10m`, `1h`, etc). User macros are supported.
Capacity	Specify the event capacity within the time window: **Unlimited** - event count is not limited **Limited** - event count is limited by the specified number Once the limit is surpassed, the latest event is evicted. User macros are supported.
Group by	Group events by host group, host, and/or event tag. This option is available for CEP rules with a time window. If specified, the rule will create multiple windows based on specified criteria; otherwise events will be grouped into a single window group.
Event count tag	Specify name for a tag displaying symptom event count. Symptom event count will be displayed as `<tag name>:<event count>` (for example: `symptoms:2`). This field is only available if *Cause and symptom grouping* is selected as time window type.
Script	Specify the script for identifying event patterns. This field is only available if *Pattern match* is selected as time window type.
Operations	Click on *Add* to specify operations.
Stop after this rule	Mark the checkbox if no further processing is required after this rule. Otherwise, continue with the next CEP rule according to the sort order.
Sort order	Specify position of this CEP rule in the processing order. If the order priority is the same, the rules will be executed alphabetically.
Description	Add the CEP rule description.

CEP rule condition



Parameter	Description
Type	Select condition type. Event filtering can be based on event name, tag name, tag value, severity, host group, host (visible name), or time period.

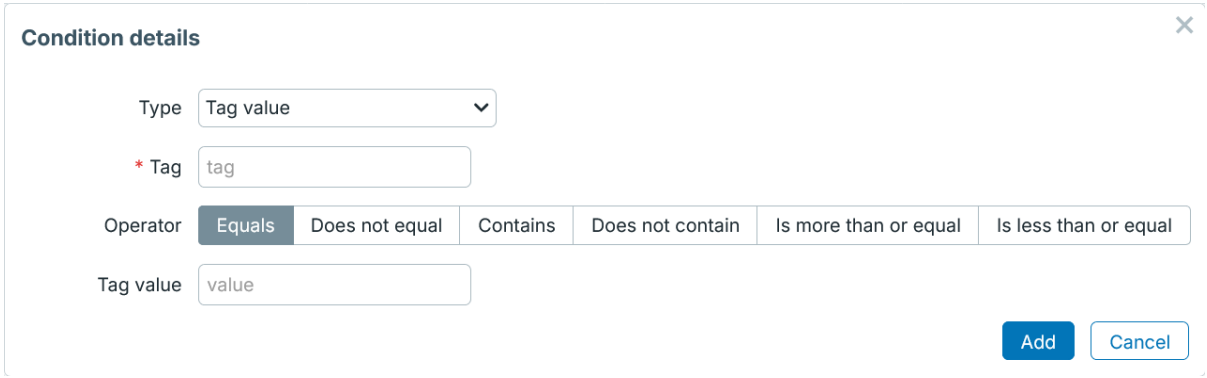
CEP rule operation



Parameter	Description
Execute when	Select execution point: **Event occurred** - when event occurs. This is the only stage where *Discard* operation is supported, since event is not actually created yet and can be discarded. Window-related operations are not available at this point. Changes made to the event in *Event occurred* phase (except closing) are visible to the next CEP rules trying to match this event. For example, if rule A added tag X and rule B has condition "Tag X exists", then it will match rule A and B. Of course, rule A should not be configured to stop after matching. **Event added to window** - when event is added to time window. Operations are executed after event is added to matched CEP-rule windows. If window is at its capacity, then *Event evicted* operations are executed instead. **Event evicted** - when event leaves the window. Operations are executed before event is evicted from a window because of duration or capacity. **Event closed** - when event is closed. **Window closed** - when time window is closed. Operations are executed on all window events when window is closed because of *Close window* operation or as expired cause-symptom window. **Pattern matched** - when pattern is matched. Operations are executed on all window events when the pattern match JavaScript returns `true`. Note that the *Event occurred*, *Event added to window* and *Event evicted* have fixed execution order, while *Window closed* and *Pattern matched* operate independently.
Conditions	Select additional conditions for the operation.

Parameter	Description
Operation	Select operation. **Events**: *Set name* - set event name *Close* - close event *Discard* - discard event. Only available at *Event occurred* stage. *Set severity* - set event severity *Increase severity* - increase event severity *Decrease severity* - decrease event severity *Suppress* - suppress event *Unsuppress* - unsuppress event *Clone first* - clone first event. Useful for producing a summary event based on the first event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. *Clone last* - clone last event. Useful for producing a summary event based on the last event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. For event cloning to work either the source (first/last) event must be resolved or its trigger must have multi-event generation enabled. **Tags**: *Add* - add tag *Set* - set tag and value *Set value* - set tag value *Increase value* - increase tag value *Decrease value* - decrease tag value *Rename* - rename tag *Remove* - remove tag **Window**: *Close (window)* - close event window
Applies to	Select operation scope: **Current event** - apply operation to the current event **Other events** - apply operation to all events except the current one **All events** - apply operation to all events
Filter	Select additional filter for the operation scope. This field is available if *Applies to* is set ”Other events” or ”All events”.

CEP rule operation condition



Parameter	Description
Type	Select condition type. Event filtering can be based on tag name, tag value, problem is opened/closed, problem is symptom/not symptom, first event in time window, last event in time window, problem is suppressed/not suppressed, event is cloned/not cloned.

Correlazione degli eventi

Panoramica

La correlazione degli eventi consente di correlare gli eventi di problema con la relativa risoluzione in modo molto preciso e flessibile.

La correlazione degli eventi può essere definita:

- A livello di trigger - un trigger può essere utilizzato per associare problemi distinti alla relativa soluzione;
- Globalmente - i problemi possono essere correlati alla relativa soluzione proveniente da un trigger/metodo di polling diverso utilizzando regole di correlazione globali.

Correlazione degli eventi basata su trigger

Panoramica

La correlazione degli eventi basata sui trigger consente di correlare separatamente i problemi segnalati da un trigger.

Sebbene generalmente un evento OK possa chiudere tutti gli eventi problema creati da un trigger, in alcuni casi è necessario un approccio più dettagliato. Ad esempio, durante il monitoraggio dei file di log, potrebbe essere necessario individuare determinati problemi in un file di log e chiuderli singolarmente anziché tutti insieme.

Questo si verifica con i trigger che hanno il parametro *PROBLEM event generation mode* impostato su *Multiple*. Questi trigger vengono normalmente utilizzati per il monitoraggio dei log, l'elaborazione dei trap e così via.

In Zabbix è possibile correlare gli eventi problema in base ai tag. I tag vengono utilizzati per estrarre i valori e creare un'identificazione per gli eventi problema. Sfruttando questa funzionalità, è inoltre possibile chiudere i problemi singolarmente in base alla corrispondenza dei tag.

In altre parole, lo stesso trigger può creare eventi separati identificati dal tag dell'evento. Pertanto, gli eventi problema possono essere identificati uno per uno e chiusi separatamente in base all'identificazione tramite il tag dell'evento.

Come funziona

Nel monitoraggio dei log potresti incontrare righe simili a queste:

```
Line1: Service 1 stopped
Line2: Service 2 stopped
Line3: Service 1 was restarted
Line4: Service 2 was restarted
```

L'idea della correlazione degli eventi è poter associare l'evento di problema dalla Line1 alla risoluzione della Line3 e l'evento di problema dalla Line2 alla risoluzione della Line4, chiudendo questi problemi uno per uno:

```
Line1: Service 1 stopped
Line3: Service 1 was restarted #problem from Line 1 closed

Line2: Service 2 stopped
Line4: Service 2 was restarted #problem from Line 2 closed
```

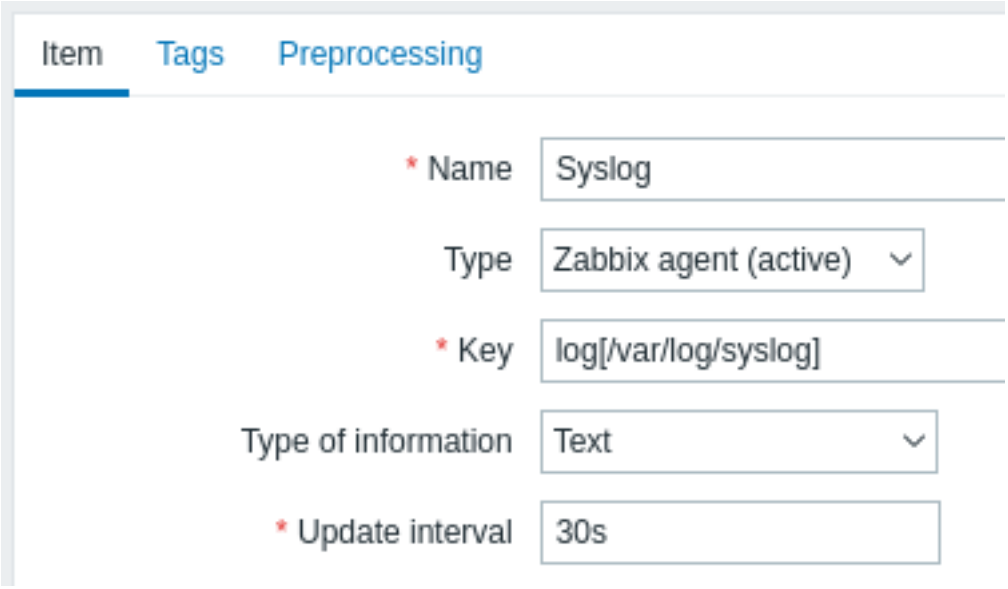
Per fare ciò, devi assegnare un tag a questi eventi correlati, ad esempio "Service 1" e "Service 2". Questo può essere fatto applicando una espressione regolare alla riga di log per estrarre il valore del tag. Poi, quando gli eventi vengono creati, vengono taggati rispettivamente come "Service 1" e "Service 2" e il problema può essere associato alla risoluzione.

Configurazione

Item

Per iniziare, potresti voler configurare un item che monitori un file di log, ad esempio:

```
log[/var/log/syslog]
```



Una volta configurato l'item, attendi un minuto affinché le modifiche di configurazione vengano acquisite, quindi vai a Latest data per verificare che l'item abbia iniziato a raccogliere dati.

Trigger

Con l'item in funzione, è necessario configurare il trigger. È importante decidere quali voci nel file di log meritano attenzione. Ad esempio, la seguente espressione del trigger cercherà una stringa come 'Stopping' per segnalare potenziali problemi:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Per assicurarti che ogni riga contenente la stringa "Stopping" sia considerata un problema, imposta anche la modalità *Problem event generation mode* nella configurazione del trigger su 'Multiple'.

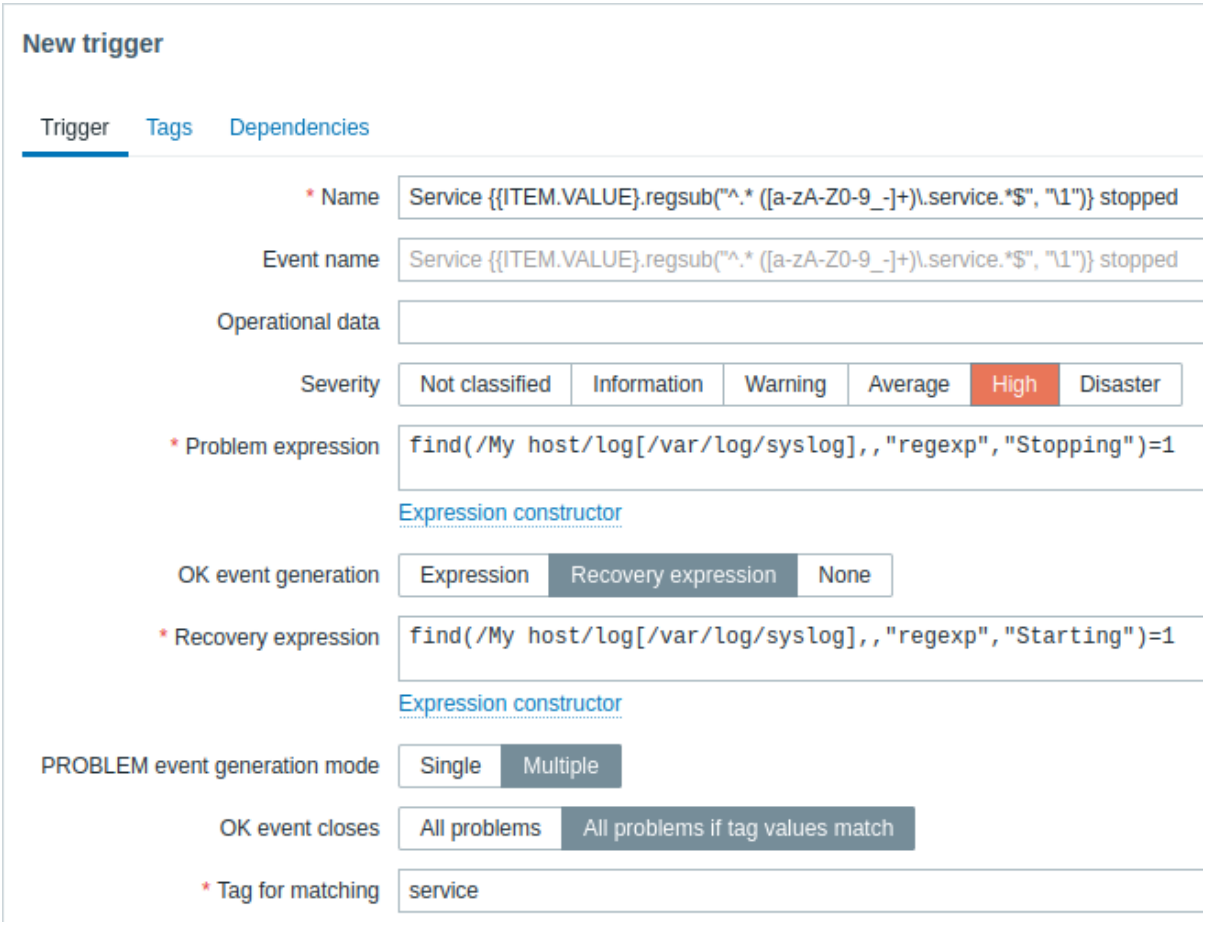
Definisci quindi un'espressione di ripristino. La seguente espressione di ripristino risolverà tutti i problemi se viene trovata una riga di log contenente la stringa "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

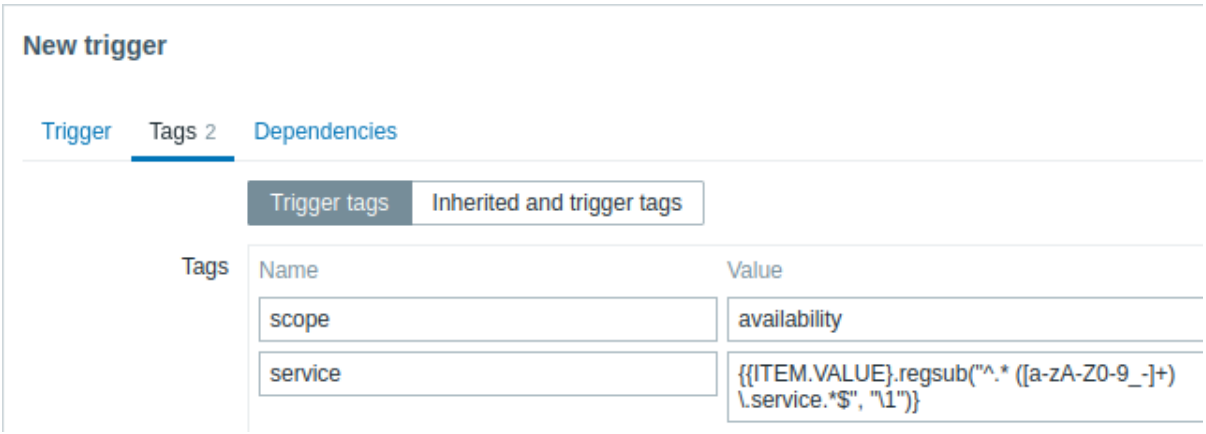
Poiché non vogliamo questo, è importante assicurarsi in qualche modo che i corrispondenti problemi root vengano chiusi, non solo tutti i problemi. È qui che il tagging può aiutare.

Problemi e risoluzioni possono essere associati specificando un tag nella configurazione del trigger. È necessario impostare le seguenti opzioni:

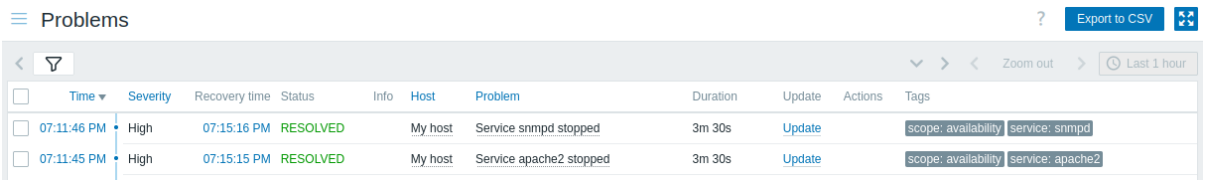
- *Problem event generation mode*: Multiple
- *OK event closes*: All problems if tag values match
- Inserisci il nome del tag per la corrispondenza degli eventi



- configura i tag per estrarre i valori dei tag dalle righe di log



Se configurato correttamente, potrai vedere gli eventi di problema taggati per applicazione e associati alla loro risoluzione in *Monitoring > Problems*.



Warning:
Poiché è possibile una configurazione errata, quando possono essere creati tag di evento simili per problemi **non correlati**, esamina i casi descritti di seguito!

- Le macro indicizzate fanno sempre riferimento al campo *Expression* della configurazione del trigger, non alla *Recovery expression*. Per esempio, in un evento di ripristino, **{ITEM.VALUE1}** risolverà al valore più recente del primo item **nell'espressione del problema** al momento del ripristino. Se l'espressione di ripristino si basa su un item diverso e il valore dell'item dell'espressione del problema cambia al momento del ripristino, gli eventi avranno tag diversi e non

verranno correlati.

- Con due applicazioni che scrivono messaggi di errore e di ripristino nello stesso file di log, un utente può decidere di usare due tag *service* nello stesso trigger con valori di tag diversi, utilizzando espressioni regolari separate nei valori dei tag per estrarre i nomi di, ad esempio, service A e service B dalla macro {ITEM.VALUE} (ad esempio, quando i formati dei messaggi differiscono). Tuttavia, questo potrebbe non funzionare come previsto se non c'è corrispondenza con le espressioni regolari. Le regexp senza corrispondenza produrranno valori di tag vuoti e un singolo valore di tag vuoto sia negli eventi di problema sia in quelli OK è sufficiente per correlare gli eventi. Quindi un messaggio di ripristino di service A potrebbe chiudere accidentalmente un messaggio di errore di service B.
- I tag effettivi e i valori dei tag diventano visibili solo quando un trigger si attiva. Se l'espressione regolare utilizzata non è valida, viene sostituita silenziosamente con una stringa *UNKNOWN*. Se l'evento di problema iniziale con un valore di tag *UNKNOWN* viene perso, potrebbero comparire successivi eventi OK con lo stesso valore di tag *UNKNOWN* che potrebbero chiudere eventi di problema che non avrebbero dovuto chiudere.
- Se un utente usa la macro {ITEM.VALUE} senza funzioni macro come valore del tag, si applica il limite di 255 caratteri. Quando i messaggi di log sono lunghi e i primi 255 caratteri non sono specifici, questo può anche generare tag di evento simili per problemi non correlati.

Correlazione globale degli eventi

Panoramica

La correlazione globale degli eventi consente di analizzare tutte le metriche monitorate da Zabbix e creare correlazioni.

È possibile correlare eventi creati da trigger completamente diversi e applicare le stesse operazioni a tutti. Creando regole di correlazione intelligenti, è possibile evitare migliaia di notifiche ripetitive e concentrarsi sulle cause principali di un problema.

La correlazione globale degli eventi è un meccanismo potente che consente di svincolarsi dalla logica di rilevamento e risoluzione dei problemi basata su un singolo trigger. Finora, un singolo evento di problema veniva creato da un trigger e la risoluzione del problema dipendeva dallo stesso trigger. Non era possibile risolvere con un trigger diverso un problema creato da un altro trigger. Con la correlazione degli eventi basata sui tag degli eventi, invece, è possibile.

Ad esempio, un trigger di log può segnalare problemi dell'applicazione, mentre un trigger di polling può segnalare che l'applicazione è attiva e funzionante. Utilizzando i tag degli eventi, è possibile assegnare al trigger di log il tag *status:down* e al trigger di polling il tag *status:up*. Quindi, in una regola di correlazione globale, è possibile mettere in relazione questi trigger e assegnare a questa correlazione un'operazione appropriata, come la chiusura dei vecchi eventi.

In un altro caso d'uso, la correlazione globale può identificare trigger simili e applicare la stessa operazione a tutti. E se fosse possibile ricevere un solo report di problema per ogni problema relativo a una porta di rete? Non sarebbe necessario segnalarli tutti. Anche questo è possibile con la correlazione globale degli eventi.

La correlazione globale degli eventi viene configurata nelle **regole di correlazione**. Una regola di correlazione definisce come i nuovi eventi di problema vengono associati agli eventi di problema esistenti e cosa fare in caso di corrispondenza (chiudere il nuovo evento, chiudere i vecchi eventi corrispondenti generando i relativi eventi OK). Se un problema viene chiuso dalla correlazione globale, ciò viene indicato nella colonna *Info* di *Monitoring* > *Problems*.

La configurazione delle regole di correlazione globale è disponibile solo per gli utenti con livello Super Admin.

> **Attention:**
> La correlazione degli eventi deve essere configurata con molta attenzione, poiché può influire negativamente sulle prestazioni dell'elaborazione degli eventi o, se configurata in modo errato, chiudere più eventi del previsto (nel peggiore dei casi, potrebbero essere chiusi tutti gli eventi di problema).

Per configurare le regole di correlazione globale in modo **sicuro**, osservare i seguenti suggerimenti importanti:

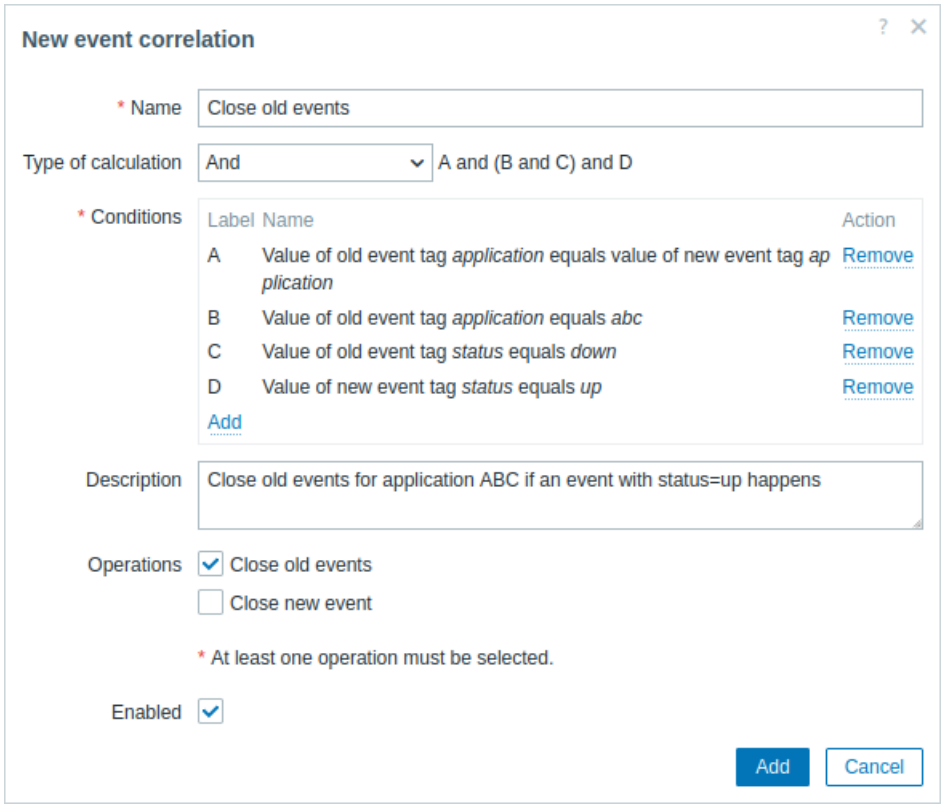
- Ridurre l'ambito della correlazione. Impostare sempre un tag univoco per il nuovo evento associato ai vecchi eventi e utilizzare la condizione di correlazione *New event tag name*.
- Aggiungere una condizione esplicita per i vecchi eventi quando si utilizza **Close old events**. Aggiungere sempre almeno una condizione *Old event* (ad esempio, *Old event tag name*, *Old event tag value* o *Event tag pair*) quando si seleziona **Close old events**; in caso contrario, la regola potrebbe trovare corrispondenze e chiudere eventi di problema esistenti non correlati (nel peggiore dei casi, tutti i problemi). Preferire *Event tag pair* per la corrispondenza dei valori dinamici (host:port, ID di sessione e così via) e restringere ulteriormente la corrispondenza in base all'host o al gruppo di host, quando possibile.
- Evitare di utilizzare nomi di tag comuni che potrebbero finire per essere utilizzati da configurazioni di correlazione diverse.
- Limitare il numero di regole di correlazione a quelle realmente necessarie.

Vedere anche: known issues.

Configurazione

Per configurare globalmente le regole di correlazione degli eventi:

1. Vai su *Data collection* > *Event correlation*.
2. Fai clic su *Create event correlation* a destra (oppure sul nome della correlazione per modificare una regola esistente).
3. Inserisci i parametri della regola di correlazione nel modulo.

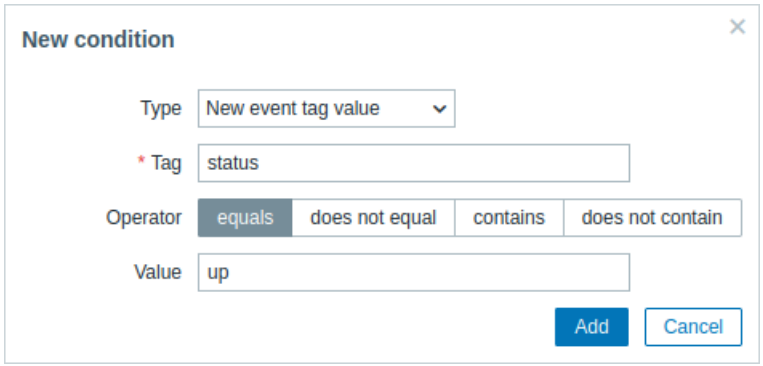


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parameter	Description
Name	Nome univoco della regola di correlazione.
Type of calculation	Sono disponibili le seguenti opzioni di calcolo delle condizioni: **And** - tutte le condizioni devono essere soddisfatte **Or** - è sufficiente che una condizione sia soddisfatta **And/Or** - AND con tipi di condizione diversi e OR con lo stesso tipo di condizione **Custom expression** - una formula di calcolo definita dall'utente per valutare le condizioni dell'azione. Deve includere tutte le condizioni (rappresentate da lettere maiuscole A, B, C, ...) e può includere spazi, tabulazioni, parentesi ( ), **and** (case sensitive), **or** (case sensitive), **not** (case sensitive).
Conditions	Elenco delle condizioni. Vedi sotto per i dettagli sulla configurazione di una condizione.
Description	Descrizione della regola di correlazione.
Operations	Seleziona la casella di controllo dell'operazione da eseguire quando l'evento viene correlato. Sono disponibili le seguenti operazioni: **Close old events** - chiude i vecchi eventi quando si verifica un nuovo evento. Aggiungi sempre una condizione basata sul vecchio evento quando usi l'operazione *Close old events*, altrimenti tutti i problemi esistenti potrebbero essere chiusi. **Close new event** - chiude il nuovo evento quando si verifica. **Warning!** Non lasciare vuote le condizioni del vecchio/nuovo evento quando usi *Close old events*/*Close new event*. Se selezioni l'operazione *Close old events* senza aggiungere una condizione che corrisponda al vecchio evento, Zabbix potrebbe abbinare tutti i vecchi eventi esistenti e chiuderli. Aggiungi sempre una condizione esplicita del vecchio evento (ad esempio, *Old event tag name* o *Event tag pair*) quando usi *Close old events*. Ad esempio, una regola che usa solo una *New event condition* e l'operazione *Close old events* corrisponderà a tutti i vecchi eventi che soddisfano i criteri del vecchio evento (mancanti), chiudendo di fatto i vecchi problemi.

Parameter	Description
Enabled	Se selezioni questa casella di controllo, la regola di correlazione sarà abilitata.

Per configurare i dettagli di una nuova condizione, fai clic su Add nel blocco Conditions. Si aprirà una finestra popup in cui potrai modificare i dettagli della condizione.



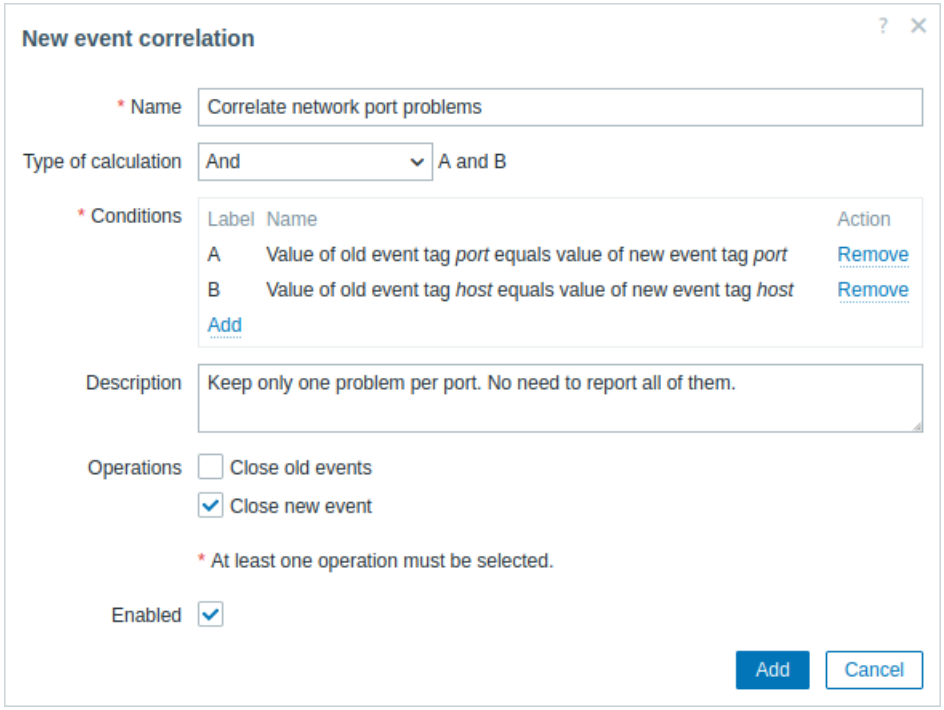
Parameter	Description
New condition	Seleziona una condizione per correlare gli eventi. *Nota* che se non viene specificata alcuna condizione del vecchio evento, tutti i vecchi eventi potrebbero essere abbinati e chiusi. Allo stesso modo, se non viene specificata alcuna condizione del nuovo evento, tutti i nuovi eventi potrebbero essere abbinati e chiusi. Sono disponibili le seguenti condizioni: **Old event tag name** - specifica il nome del tag del vecchio evento per la corrispondenza. **New event tag name** - specifica il nome del tag del nuovo evento per la corrispondenza. **New event host group** - specifica il gruppo host del nuovo evento per la corrispondenza. **Event tag pair** - specifica il nome del tag del nuovo evento e il nome del tag del vecchio evento per la corrispondenza. In questo caso ci sarà corrispondenza se i **valori** dei tag in entrambi gli eventi coincidono. I *nomi* dei tag non devono coincidere. Questa opzione è utile per confrontare valori runtime, che potrebbero non essere noti al momento della configurazione (vedi anche Example). **Old event tag value** - specifica il nome e il valore del tag del vecchio evento per la corrispondenza, usando i seguenti operatori: *equals* - ha il valore del tag del vecchio evento *does not equal* - non ha il valore del tag del vecchio evento *contains* - contiene la stringa nel valore del tag del vecchio evento *does not contain* - non contiene la stringa nel valore del tag del vecchio evento **New event tag value** - specifica il nome e il valore del tag del nuovo evento per la corrispondenza, usando i seguenti operatori: *equals* - ha il valore del tag del nuovo evento *does not equal* - non ha il valore del tag del nuovo evento *contains* - contiene la stringa nel valore del tag del nuovo evento *does not contain* - non contiene la stringa nel valore del tag del nuovo evento

Warning:
Poiché è possibile una configurazione errata, quando per problemi **non correlati** possono essere creati tag di evento simili, esamina attentamente i casi descritti di seguito!

- I tag effettivi e i valori dei tag diventano visibili solo quando si attiva un trigger. Se l'espressione regolare utilizzata non è valida, viene sostituita silenziosamente con una stringa *UNKNOWN*. Se l'evento di problema iniziale con un valore di tag *UNKNOWN* viene perso, potrebbero comparire successivi eventi OK con lo stesso valore di tag *UNKNOWN* che potrebbero chiudere eventi di problema che non avrebbero dovuto essere chiusi.
- Se un utente usa la macro {ITEM.VALUE} senza funzioni macro come valore del tag, si applica il limite di 255 caratteri. Quando i messaggi di log sono lunghi e i primi 255 caratteri non sono specifici, ciò può anche generare tag di evento simili per problemi non correlati.

Esempio

Impedisci eventi di problema ripetitivi provenienti dalla stessa porta di rete.



Questa regola di correlazione globale correlerà i problemi se i valori dei tag *host* e *port* esistono nel trigger e sono uguali nell'evento originale e in quello nuovo.

L'operazione chiuderà i nuovi eventi di problema sulla stessa porta di rete, mantenendo aperto solo il problema originale.

Scoperta

Panoramica La discovery fornisce metodi automatici per aggiungere host e configurare il monitoraggio senza configurazione manuale. Zabbix supporta più metodi di discovery per adattarsi a scenari diversi.

Metodi di discovery

- Network discovery - rileva automaticamente host e servizi disponibili nella rete.
- Active agent autoregistration - consente agli agent attivi di registrarsi automaticamente con il server.
- Low-level discovery - crea automaticamente item, trigger, grafici e host per entità rilevate come file system o interfacce di rete utilizzando prototipi e regole di discovery annidate.

Rilevamento della rete

Panoramica

Zabbix offre una funzionalità di discovery automatico della rete che è efficace e molto flessibile.

Con il network discovery configurato correttamente, puoi:

- Velocizzare il deployment di Zabbix.
- Semplificare l'amministrazione.
- Usare Zabbix in ambienti in rapida evoluzione senza un'eccessiva amministrazione.

Il network discovery di Zabbix si basa sulle seguenti informazioni:

- Intervalli IP
- Disponibilità di servizi esterni (FTP, SSH, WEB, POP3, IMAP, TCP, ecc.)
- Informazioni ricevute da Zabbix agent (è supportata solo la modalità non crittografata)
- Informazioni ricevute da SNMP agent

NON fornisce il discovery della topologia di rete.

Il network discovery è sostanzialmente composto da due fasi: discovery e azioni.

Discovery

Zabbix esegue periodicamente la scansione degli intervalli IP definiti nelle regole di discovery di rete. La frequenza del controllo è configurabile separatamente per ciascuna regola.

Ogni regola ha un insieme di controlli dei servizi definiti da eseguire per l'intervallo IP.

Le regole di discovery vengono elaborate dal discovery manager. Il discovery manager crea un job per ogni regola con un elenco di task (controlli di rete). I controlli di rete vengono eseguiti in parallelo dai discovery worker disponibili (il numero è configurabile nel frontend per ciascuna regola). L'eccezione sono i controlli SNMPv3, che vengono elaborati da un singolo worker. Solo i controlli con lo stesso IP e la stessa porta vengono pianificati in sequenza, perché alcuni dispositivi non accettano connessioni parallele sulla stessa porta.

La dimensione della coda dei controlli di rete è limitata a 2000000 o a circa 4 GB di memoria. Se la coda diventa piena, la regola di discovery verrà saltata e nel log verrà stampato un messaggio di avviso. È possibile usare l'`zabbix[discovery_queue]` item interno per monitorare il numero di controlli di discovery nella coda.

I controlli di discovery vengono elaborati indipendentemente dagli altri controlli. Se alcuni controlli non trovano un servizio (o falliscono), gli altri controlli verranno comunque elaborati.

Note:
Se una regola di discovery viene modificata durante l'esecuzione, l'esecuzione corrente di discovery verrà interrotta.

Ogni controllo di un servizio e di un host (IP) eseguito dal modulo di discovery di rete genera un evento di discovery.

Event	Check of service result
Service Discovered	Il servizio è 'up' dopo essere stato 'down' oppure quando viene scoperto per la prima volta.
Service Up	Il servizio è 'up', dopo essere già stato 'up'.
Service Lost	Il servizio è 'down' dopo essere stato 'up'.
Service Down	Il servizio è 'down', dopo essere già stato 'down'.
Host Discovered	Almeno un servizio di un host è 'up' dopo che tutti i servizi di quell'host erano 'down', oppure viene scoperto un servizio che appartiene a un host non registrato.
Host Up	Almeno un servizio di un host è 'up', dopo che almeno un servizio era già 'up'.
Host Lost	Tutti i servizi di un host sono 'down' dopo che almeno uno era 'up'.
Host Down	Tutti i servizi di un host sono 'down', dopo che erano già 'down'.

Azioni

Gli eventi di discovery possono costituire la base per azioni pertinenti, quali:

- Invio di notifiche
- Aggiunta/rimozione di host
- Abilitazione/disabilitazione di host
- Aggiunta di host a un gruppo
- Rimozione di host da un gruppo
- Aggiunta di tag a un host
- Rimozione di tag da un host
- Collegamento di un template agli host/scollegamento di un template dagli host
- Esecuzione di script remoti

Queste azioni possono essere configurate in base al tipo di dispositivo, IP, stato, uptime/downtime, ecc. Per i dettagli completi sulla configurazione delle azioni per eventi basati sulla network discovery, vedere le pagine operation e conditions dell'azione.

Poiché le azioni di network discovery sono basate su eventi, verranno attivate sia quando un host rilevato è online sia quando è offline. Si consiglia vivamente di aggiungere la condition dell'azione *Discovery status: up* per evitare che azioni come *Add host* vengano attivate in caso di eventi *Service Lost*/*Service Down*. In caso contrario, se un host rilevato viene rimosso manualmente, continuerà comunque a generare eventi *Service Lost*/*Service Down* e verrà ricreato durante il successivo ciclo di discovery.

Note:
Il collegamento di template a un host rilevato non riuscirà nel suo complesso se uno qualsiasi dei template collegabili contiene un'entità univoca (ad esempio una chiave item) uguale a un'entità univoca (ad esempio una chiave item) già esistente sull'host o su un altro dei template collegabili.

Creazione dell'host

Un host viene aggiunto se viene selezionata l'operazione *Add host*. Un host viene aggiunto anche, se l'operazione *Add host* è assente, quando si selezionano operazioni che producono azioni su un host. Tali operazioni sono:

- Enable host.
- Disable host.
- Add host to a host group.
- Link template to a host.

Gli host creati vengono aggiunti al gruppo *Discovered hosts* (per impostazione predefinita, configurabile in *Administration* > *General* > *Other*). Se si desidera che gli host vengano aggiunti a un altro gruppo, aggiungere un'operazione *Remove from host groups* (specificando "Discovered hosts") e aggiungere anche un'operazione *Add to host groups* (specificando un altro host group), perché un host deve appartenere a un host group.

L'indirizzo IP del dispositivo rilevato, insieme alla sorgente di discovery (Zabbix server, Zabbix proxy o proxy group) e al tipo di interfaccia, viene usato come criterio per trovare un host nel sistema. Se esiste già un host con lo stesso indirizzo IP, lo stesso tipo di interfaccia e la stessa sorgente di discovery, tale host sarà il target per l'esecuzione delle operazioni. Se la sorgente di discovery differisce, l'entità rilevata viene considerata un host diverso e può essere creato un nuovo host.

Se l'indirizzo IP dell'host rilevato viene modificato o l'interfaccia viene eliminata, al successivo discovery verrà creato un nuovo host.

Denominazione degli host

Quando si aggiungono host, il nome host è il risultato di una ricerca DNS inversa oppure l'indirizzo IP se la ricerca inversa non riesce. La ricerca viene eseguita dal server Zabbix o dal proxy Zabbix, a seconda di quale dei due stia effettuando il discovery. Se la ricerca non riesce sul proxy, non viene ritentata sul server. Se esiste già un host con tale nome, al nome dell'host successivo verrà aggiunto **_2**, poi **_3** e così via.

È anche possibile ignorare la ricerca DNS/IP e utilizzare invece il valore di un item come nome host, ad esempio:

- È possibile individuare più server con Zabbix agent in esecuzione utilizzando un item Zabbix agent per il discovery e assegnare automaticamente nomi corretti in base al valore stringa restituito da questo item
- È possibile individuare più dispositivi di rete SNMP utilizzando un item SNMP agent per il discovery e assegnare automaticamente nomi corretti in base al valore stringa restituito da questo item

Se il nome host è stato impostato utilizzando il valore di un item, non viene aggiornato durante i successivi controlli di discovery. Se non è possibile impostare il nome host utilizzando il valore di un item, viene usato il valore predefinito (nome DNS).

Se esiste già un host con l'indirizzo IP individuato e l'origine del discovery (server Zabbix, proxy o gruppo di proxy) non è cambiata, non viene creato un nuovo host. Se l'origine del discovery è diversa, l'entità individuata viene trattata come distinta e può essere creato un nuovo host. Tuttavia, se l'azione di discovery contiene operazioni (collegamento di template, aggiunta a un gruppo di host, ecc.), queste vengono eseguite sull'host esistente che corrisponde per indirizzo IP, tipo di interfaccia e origine del discovery.

Rimozione degli host

Gli host rilevati da una regola di network discovery vengono rimossi automaticamente da *Monitoring* > *Discovery* se un'entità rilevata non è più nell'intervallo IP della regola. Gli host vengono rimossi immediatamente.

Creazione delle interfacce quando si aggiungono host

Quando gli host vengono aggiunti in seguito al rilevamento di rete, vengono create le interfacce in base a queste regole:

- I servizi rilevati - ad esempio, se un controllo SNMP ha avuto esito positivo, verrà creata un'interfaccia SNMP.
- Se un host risponde sia alle richieste dell'agent Zabbix sia a quelle SNMP, verranno create entrambe le tipologie di interfaccia.
- Se i criteri di unicità sono i dati restituiti dall'agent Zabbix o da SNMP, la prima interfaccia trovata per un host verrà creata come predefinita. Gli altri indirizzi IP verranno ignorati, a meno che l'interfaccia predefinita non vada giù. Nel caso in cui l'interfaccia predefinita sia giù, il più piccolo degli altri indirizzi IP verrà usato come sostituto dell'interfaccia predefinita. Le condizioni dell'azione (come Host IP) non influiscono sull'aggiunta delle interfacce. *Nota* che questo funzionerà se tutte le interfacce vengono rilevate dalla stessa regola di rilevamento. Se una regola di rilevamento diversa rileva un'interfaccia diversa dello stesso host, verrà aggiunto un host aggiuntivo.
- Se un host ha risposto solo ai controlli dell'agent, verrà creato con una sola interfaccia agent. Se in seguito iniziasse a rispondere anche a SNMP, verrebbero aggiunte ulteriori interfacce SNMP.
- Se inizialmente sono stati creati 3 host separati, rilevati in base ai criteri di unicità "IP", e poi la regola di rilevamento viene modificata in modo che gli host A, B e C abbiano un risultato identico dei criteri di unicità, B e C vengono creati come interfacce aggiuntive per A, il primo host. Gli host singoli B e C rimangono. In *Monitoring* > *Discovery* le interfacce aggiunte verranno visualizzate nella colonna "Discovered device", con carattere nero e rientrate, mentre la colonna "Monitored host" mostrerà solo A, il primo host creato. "Uptime/Downtime" non viene misurato per gli IP considerati interfacce aggiuntive.

Modifica dell'impostazione del proxy

Gli host rilevati da proxy diversi non vengono sempre trattati come host diversi. I controlli di discovery e di unicità dipendono dalla struttura del gruppo di proxy: quando un proxy esegue una regola di discovery e crea un host, quell'host viene aggiunto al gruppo di proxy padre del proxy, non assegnato al proxy stesso. Quando Zabbix valuta l'unicità dell'IP durante la discovery, controlla gli host monitorati dal gruppo di proxy padre. Gli host monitorati dai singoli proxy all'interno di quel gruppo, incluso il proxy che ha eseguito la discovery, vengono ignorati per il controllo di unicità, il che può causare host duplicati se più proxy monitorano subnet sovrapposte.

Sebbene questo comportamento consenta alla discovery di operare su intervalli IP sovrapposti utilizzati da subnet diverse, modificare il proxy assegnato a una subnet già monitorata è più complesso, perché le modifiche del proxy devono essere applicate in modo coerente agli host rilevati e all'appartenenza al gruppo di proxy padre per evitare duplicati.

Ad esempio, i passaggi per sostituire il proxy in una regola di discovery:

1. Disabilitare la regola di discovery.
2. Sincronizzare la configurazione del proxy.
3. Sostituire il proxy nella regola di discovery.
4. Sostituire il proxy per tutti gli host rilevati da questa regola (assicurarsi che gli host nel gruppo di proxy padre e tutti gli host monitorati dai singoli proxy in quel gruppo vengano aggiornati per evitare duplicati).
5. Abilitare la regola di discovery.

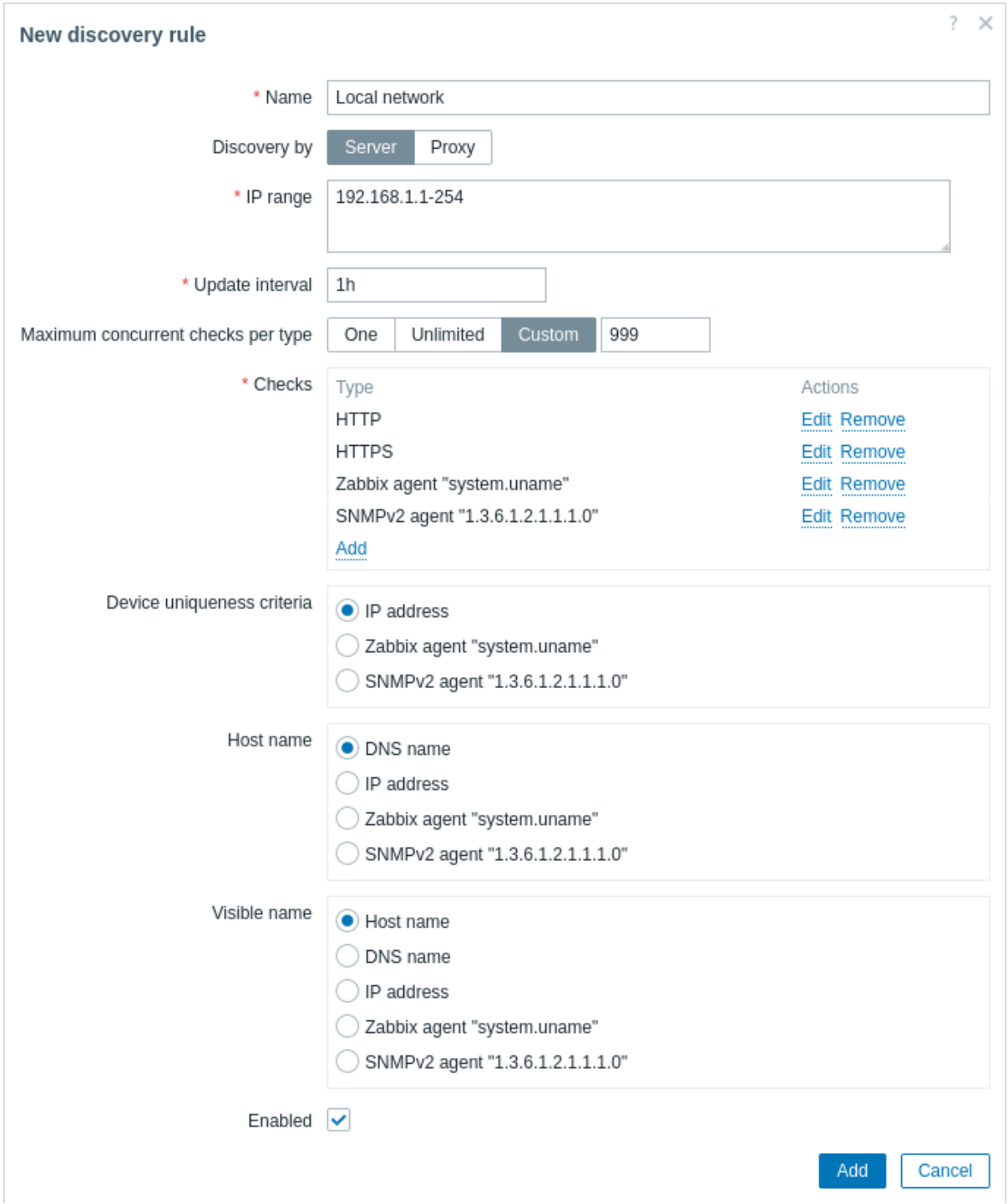
Configurazione di una regola di discovery di rete

Panoramica

Per configurare una regola di discovery di rete usata da Zabbix per individuare host e servizi:

1. Vai su *Data collection* > *Discovery*.
2. Fai clic su *Create discovery rule* (oppure sul nome della regola per modificare una regola esistente).
3. Modifica gli attributi della regola di discovery.

Attributi della regola



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parameter	Description
Name	Nome univoco della regola. Ad esempio, ”Local network”.
Discovery by	La discovery viene eseguita da: **Server** - da Zabbix server **Proxy** - da Zabbix proxy (selezionato nel campo del nome del proxy)
IP range	L’intervallo di indirizzi IP per la discovery. Può avere i seguenti formati: IP singolo: 192.168.1.33 Intervallo di indirizzi IP: 192.168.1-10.1-255. L’intervallo è limitato dal numero totale di indirizzi coperti (meno di 64K). Maschera IP: 192.168.4.0/24 maschere IP supportate: /16 - /30 per indirizzi IPv4 /112 - /128 per indirizzi IPv6 Elenco: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Questo campo supporta spazi, tabulazioni e più righe.

Parameter	Description
Update interval	Questo parametro definisce con quale frequenza Zabbix eseguirà la regola. L'intervallo viene misurato dopo la fine dell'esecuzione della precedente istanza di discovery, quindi non c'è sovrapposizione. Sono supportati i suffissi di tempo, ad esempio 30s, 1m, 2h, 1d. Sono supportate le macro utente. *Nota* che se viene usata una macro utente e il suo valore viene modificato (ad esempio, 1w → 1h), il controllo successivo verrà eseguito in base al valore precedente (molto più avanti nel futuro con i valori dell'esempio).
Maximum concurrent checks per type	Imposta il numero massimo di thread di discovery (worker) per controllo di servizio per l'elaborazione parallela dei controlli di discovery: **One** - un thread **Unlimited** - numero illimitato di thread (ma non superiore a quello del parametro StartDiscoverers) **Custom** - imposta un numero personalizzato di thread (0-999). I controlli di servizio asincroni SNMPv1 e SNMPv2 vengono elaborati in parallelo dai worker disponibili. Tutte le regole di discovery con controlli di servizio asincroni SNMPv3 vengono elaborate da un solo worker a causa delle peculiarità dell'implementazione di libsnmp; pertanto, aumentare il numero di worker non aumenterà la velocità di discovery.
Checks	Zabbix utilizzerà questo elenco di controlli per la discovery. Fai clic su Add per configurare un nuovo controllo in una finestra popup. Controlli supportati: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping. Una discovery basata su protocollo utilizza la funzionalità **net.tcp.service[]** per testare ciascun host, ad eccezione di SNMP che interroga un OID SNMP. Zabbix agent viene testato interrogando un item in modalità non crittografata. Per maggiori dettagli, consulta gli item dell'agent. Il parametro 'Ports' può essere uno dei seguenti: Porta singola: 22 Intervallo di porte: 22-45 Elenco: 22-45,55,60-70 A partire da Zabbix 7.0, tutti i controlli di servizio vengono eseguiti in modo asincrono, ad eccezione dei controlli LDAP. A partire da Zabbix 7.0, il controllo HTTP/HTTPS viene eseguito tramite libcurl. Se Zabbix server/proxy è compilato senza libcurl, i controlli HTTP funzioneranno come nelle versioni precedenti (cioè come controlli TCP), ma i controlli HTTPS non funzioneranno.
Device uniqueness criteria	I criteri di unicità possono essere: **IP address** - non elaborare dispositivi con IP singolo multipli. Se esiste già un dispositivo con lo stesso IP, verrà considerato già scoperto e non verrà aggiunto un nuovo host. **<discovery check>** - controllo di Zabbix agent o SNMP agent. Nota che i criteri di unicità usati durante la discovery non sono gli stessi dell'identificazione dell'host nel sistema durante l'esecuzione delle azioni. I criteri di unicità durante la discovery definiscono se due o più dispositivi scoperti sono uguali (o diversi), mentre in Zabbix solo l'indirizzo IP è il criterio per l'identificazione dell'host (vedi Host creation).
Host name	Imposta il nome tecnico dell'host creato usando: **DNS name** - nome DNS (predefinito) **IP address** - indirizzo IP **<discovery check>** - valore stringa ricevuto dal controllo di discovery (ad esempio, controllo Zabbix agent, SNMP agent) Vedi anche: Host naming.
Visible name	Imposta il nome visibile dell'host creato usando: **Host name** - nome tecnico dell'host (predefinito) **DNS name** - nome DNS **IP address** - indirizzo IP **<discovery check>** - valore stringa ricevuto dal controllo di discovery (ad esempio, controllo Zabbix agent, SNMP agent) Vedi anche: Host naming.
Enabled	Con la casella di controllo selezionata, la regola è attiva e verrà eseguita da Zabbix server. Se non selezionata, la regola non è attiva. Non verrà eseguita.

Superamento del limite dei descrittori di file

In caso di un numero elevato di controlli concorrenti, è possibile esaurire il limite dei descrittori di file per il discovery manager.

Il numero di descrittori di file richiesti per il rilevamento equivale al numero di discovery worker * 1000. Per impostazione predefinita, ci sono 5 discovery worker, mentre il limite soft del sistema è di circa 1024.

Se questo limite viene raggiunto, Zabbix ridurrà il numero predefinito di controlli concorrenti per tipo per ciascun worker e scriverà un avviso nel file di log. Tuttavia, se l'utente ha impostato un valore più alto per *Maximum concurrent checks per type* rispetto al valore calcolato da Zabbix, Zabbix utilizzerà il valore definito dall'utente per un worker.

Uno scenario reale

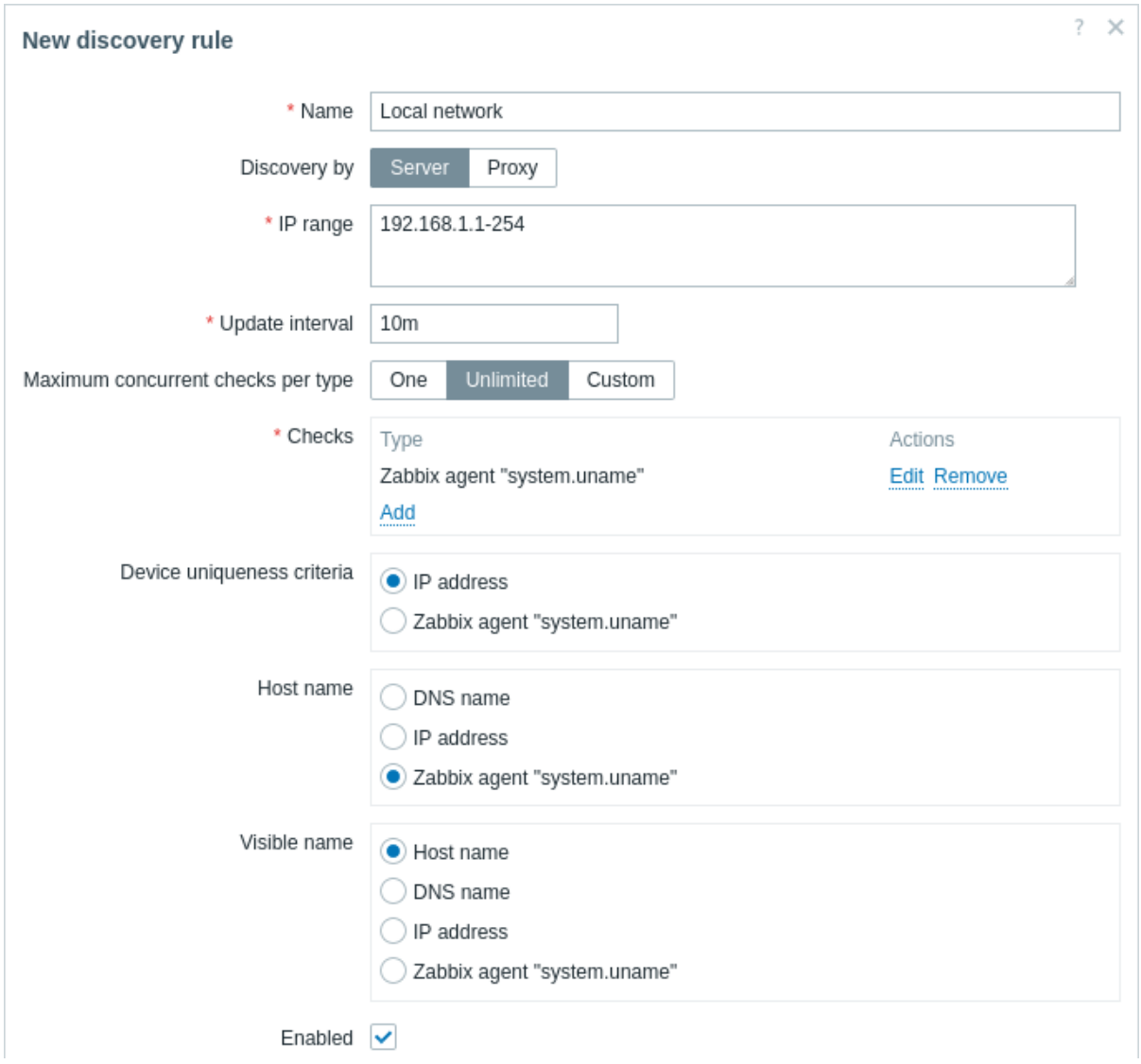
In questo esempio, desideriamo configurare il rilevamento della rete per la rete locale con un intervallo IP di 192.168.1.1-192.168.1.254.

Nel nostro scenario vogliamo:

- rilevare gli host che hanno Zabbix agent in esecuzione
- eseguire il rilevamento ogni 10 minuti
- aggiungere un host al monitoraggio se il tempo di attività dell'host è superiore a 1 ora
- rimuovere gli host se il tempo di inattività dell'host è superiore a 24 ore
- aggiungere gli host Linux al gruppo "Linux servers"
- aggiungere gli host Windows al gruppo "Windows servers"
- utilizzare il template *Linux* per gli host Linux
- utilizzare il template *Windows* per gli host Windows

Passo 1

Definizione di una regola di network discovery per il nostro intervallo IP.



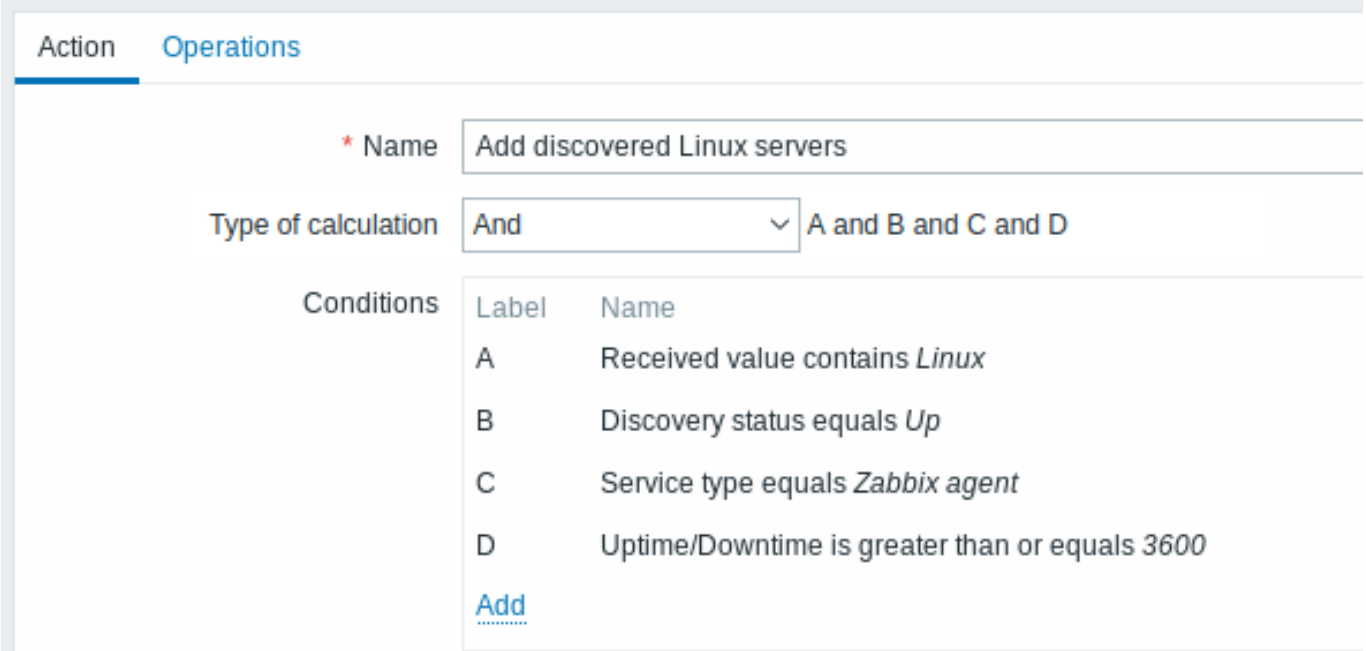
Zabbix proverà a individuare gli host nell'intervallo IP 192.168.1.1-192.168.1.254 collegandosi agli agent Zabbix e ottenendo il valore dalla chiave **system.uname**. Il valore ricevuto dall'agent può essere utilizzato per assegnare un nome agli host e anche per applicare azioni diverse per sistemi operativi differenti. Ad esempio, collegare i server Windows al template *Windows* e i server Linux al template *Linux*.

La regola verrà eseguita ogni 10 minuti.

Quando questa regola viene aggiunta, Zabbix avvierà automaticamente il discovery e la generazione degli eventi basati sul discovery per l'ulteriore elaborazione.

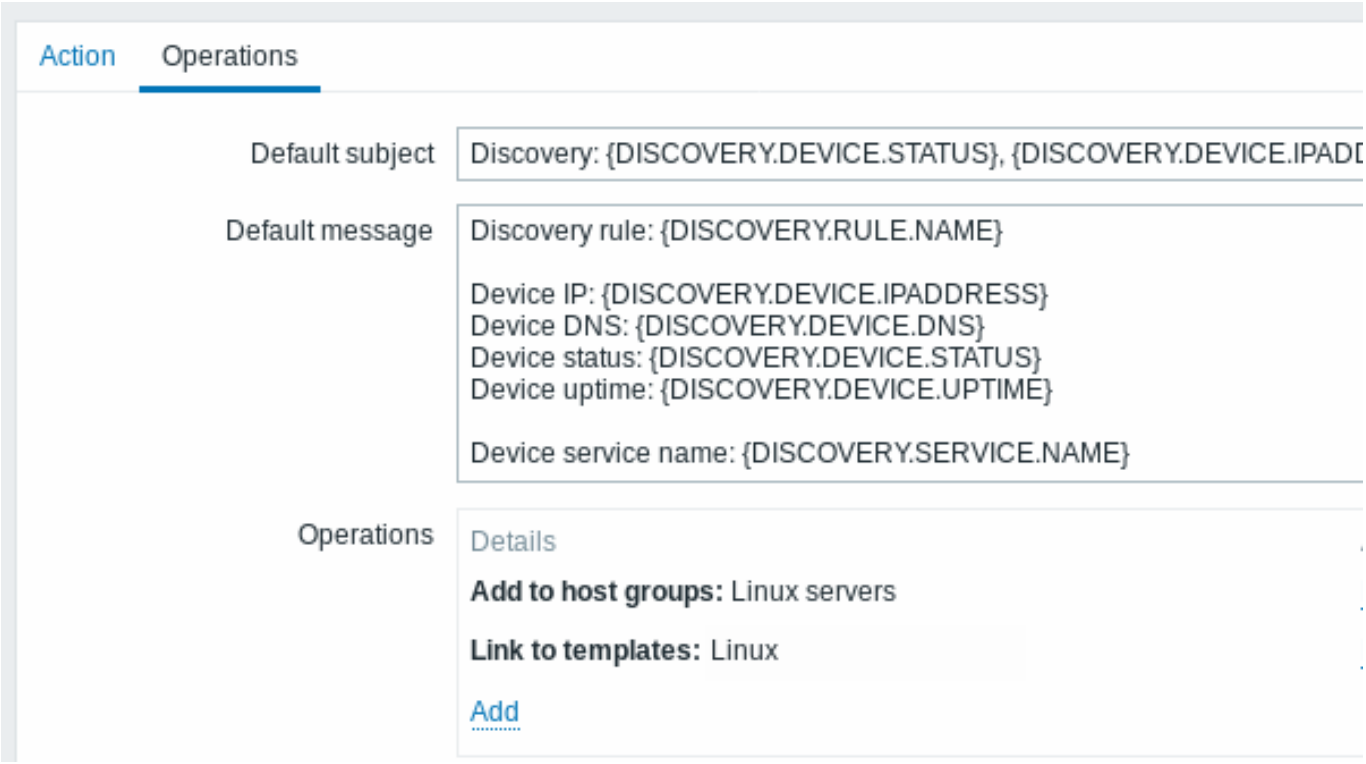
Passaggio 2

Definizione di un'azione di discovery per aggiungere i server Linux rilevati al rispettivo gruppo/template.



L'azione verrà attivata se:

- il servizio "Zabbix agent" è "up"
- il valore di system.uname (la chiave dello Zabbix agent che abbiamo usato nella definizione della regola) contiene "Linux"
- l'uptime è di 1 ora (3600 secondi) o superiore

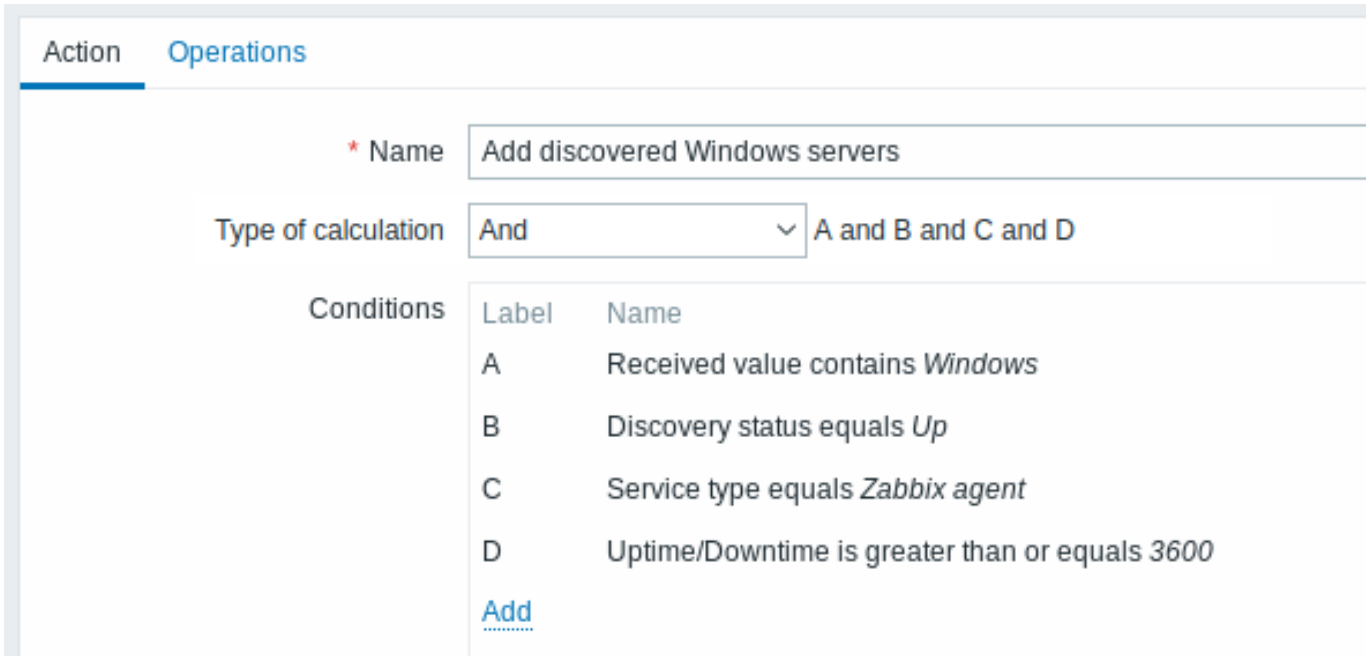


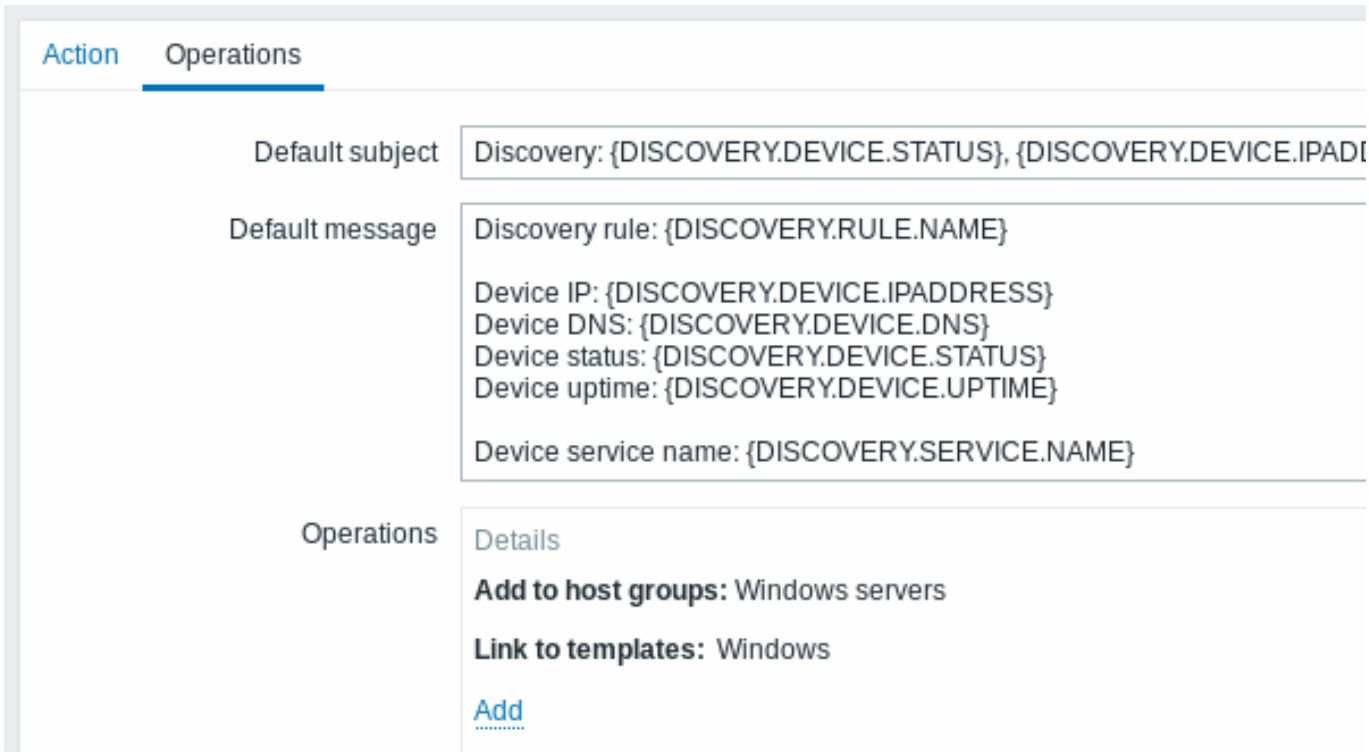
L'azione eseguirà le seguenti operazioni:

- aggiungere il host rilevato al gruppo "Linux servers" (e aggiungere anche il host se non era stato aggiunto in precedenza)
- collegare il host al template *Linux*. Zabbix inizierà automaticamente a monitorare il host utilizzando item e trigger del template "Linux".

Passaggio 3

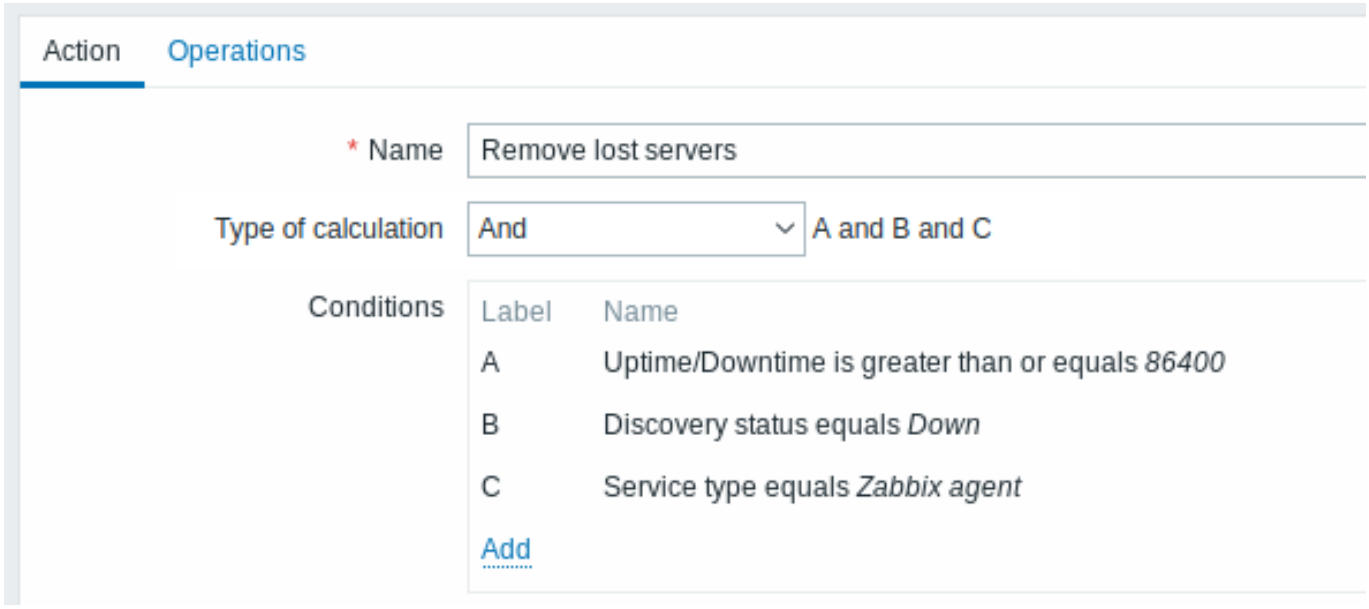
Definizione di un’azione di discovery per aggiungere i server Windows rilevati al rispettivo gruppo/template.

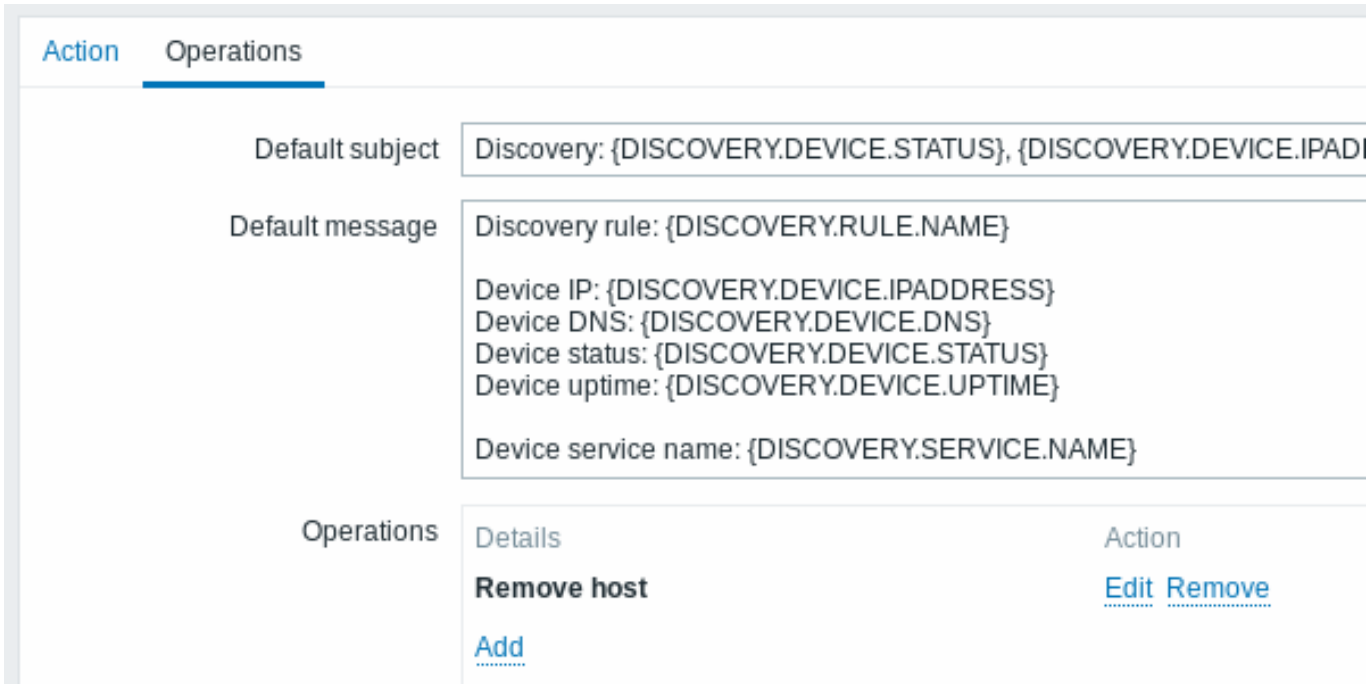




Passaggio 4

Definizione di un’azione di discovery per rimuovere i server persi.





Un server verrà rimosso se il servizio ”Zabbix agent” è ’down’ per più di 24 ore (86400 secondi).

Autoregistrazione dell’agent attivo

Panoramica

È possibile consentire l’autoregistrazione dell’agent attivo di Zabbix, dopo la quale il server può iniziare a monitorarlo. In questo modo, nuovi host possono essere aggiunti al monitoraggio senza configurarli manualmente sul server.

L’autoregistrazione può avvenire quando un agent attivo precedentemente sconosciuto richiede i controlli.

La funzionalità può essere molto utile per il monitoraggio automatico di nuovi nodi cloud. Non appena si dispone di un nuovo nodo nel Zabbix Cloud, verrà avviata automaticamente la raccolta dei dati di prestazioni e disponibilità dell’host.

L’autoregistrazione dell’agent attivo supporta anche il monitoraggio degli host aggiunti con controlli passivi. Quando l’agent attivo richiede i controlli, se nel file di configurazione sono definiti i parametri di configurazione `ListenIP` o `ListenPort`, questi vengono inviati al server. Se sono specificati più indirizzi IP, viene inviato al server il primo.

Il server, quando aggiunge il nuovo host autoregistrato, utilizza l’indirizzo IP e la porta ricevuti per configurare l’agent. Se non viene ricevuto alcun valore per l’indirizzo IP, viene utilizzato quello impiegato per la connessione in ingresso. Se non viene ricevuto alcun valore per la porta, viene utilizzato 10050.

È possibile specificare che l'host debba essere autoregistrato con un nome DNS come interfaccia predefinita dell'agent.

L'autoregistrazione viene rieseguita:

- se le informazioni dei metadati host cambiano:
  - a causa della modifica di HostMetadata e del riavvio dell'agent
  - a causa della modifica del valore restituito da HostMetadataItem
- per gli host creati manualmente con metadati mancanti
- se un host viene modificato manualmente per essere monitorato da un altro Zabbix proxy
- se l'autoregistrazione per lo stesso host proviene da un nuovo Zabbix proxy

L'intervallo heartbeat dell'autoregistrazione dell'agent attivo per Zabbix server e Zabbix proxy è di 120 secondi. Quindi, nel caso in cui un host rilevato venga eliminato, l'autoregistrazione verrà rieseguita dopo 120 secondi.

Configurazione

Specificare il server

Assicurarsi che il server Zabbix sia identificato nel file di configurazione dell'agent (`zabbix_agentd.conf`):

```
ServerActive=10.0.0.1
```

A meno che non si definisca esplicitamente un *Hostname* in `zabbix_agentd.conf`, il server utilizzerà il nome host di sistema della posizione dell'agent per denominare l'host. In Linux, il nome host di sistema può essere ottenuto eseguendo il comando `hostname`.

Se *Hostname* è definito nella configurazione dell'agent Zabbix come un elenco di host separati da virgole, verranno creati host per tutti i nomi host elencati.

Riavviare l'agent dopo aver apportato modifiche al file di configurazione.

Azione per l'autoregistrazione dell'agent attivo

Quando il server riceve una richiesta di autoregistrazione da un agent, richiama un'azione. Per l'autoregistrazione dell'agent deve essere configurata un'azione con origine dell'evento "Autoregistrazione".

> **Note:**
> Non è necessario configurare il network discovery per consentire agli agent attivi di autoregistrarsi.

Nel frontend di Zabbix:

1. Vai a *Avvisi* > *Azioni*.
2. Seleziona *Azioni di autoregistrazione*.
3. Fai clic su *Crea azione*.
4. Nella scheda *Azione*, assegna un nome all'azione.
5. Facoltativamente, specifica le condizioni. Nelle condizioni per il nome host/metadati host puoi utilizzare una corrispondenza per sottostringa o per espressione regolare. Se intendi utilizzare il tipo di condizione *Metadati host*, consulta Utilizzo dei metadati host.
6. Nella scheda *Operazioni*, aggiungi le operazioni pertinenti, come *Aggiungi host*, *Aggiungi al gruppo di host* (ad esempio, *Host rilevati*), *Collega template* e così via.

> **Note:**
> Se è probabile che gli host che si autoregistreranno siano supportati solo per il monitoraggio attivo (ad esempio, host isolati dal firewall del server Zabbix), potresti voler creare un template specifico, come *Template_Linux-active*, a cui collegarli.

Gli host creati vengono aggiunti al gruppo *Host rilevati* (per impostazione predefinita, configurabile in *Amministrazione* > *Generale* > *Altri*). Se desideri che gli host vengano aggiunti a un altro gruppo, aggiungi un'operazione *Rimuovi dal gruppo di host* (specificando "Host rilevati") e anche un'operazione *Aggiungi al gruppo di host* (specificando un altro gruppo di host), perché un host deve appartenere a un gruppo di host.

Autoregistrazione sicura

È possibile configurare una modalità sicura di autoregistrazione utilizzando l'autenticazione basata su PSK con connessioni crittografate.

Il livello di crittografia viene configurato globalmente in *Administration* > *General* > *Autoregistration*. È possibile selezionare nessuna crittografia, la crittografia TLS con autenticazione PSK oppure entrambe (in modo che alcuni host possano registrarsi senza crittografia, mentre altri tramite una connessione crittografata).

L'autenticazione tramite PSK viene verificata dal server Zabbix prima di aggiungere un host. Se la verifica ha esito positivo, l'host viene aggiunto e *Connections from/to host* viene impostato su "PSK", utilizzando un'identità e una chiave precondivisa uguali a quelle configurate nelle impostazioni globali di autoregistrazione.

Attention:
Per garantire la sicurezza dell'autoregistrazione nelle installazioni che utilizzano proxy, è necessario abilitare la crittografia tra il server Zabbix e il proxy.

Utilizzo del DNS come interfaccia predefinita

HostInterface e HostInterfaceItem parametri di configurazione consentono di specificare un valore personalizzato per l'interfaccia del host durante l'autoregistrazione.

Più precisamente, sono utili se il host deve essere autoregistrato con un nome DNS come interfaccia agent predefinita anziché con il relativo indirizzo IP. In questo caso, il nome DNS deve essere specificato o restituito come valore di uno dei parametri HostInterface o HostInterfaceItem. Se il valore di uno di questi parametri cambia, ad esempio da un indirizzo IP a un nome DNS o viceversa, l'interfaccia predefinita del host autoregistrato verrà aggiornata di conseguenza. Questo aggiornamento viene applicato al host esistente, senza crearne uno nuovo. Per inviare il nuovo valore, è necessario riavviare l'agent affinché avvii nuovamente il processo di autoregistrazione.

Note:
Se i parametri HostInterface o HostInterfaceItem non sono configurati, viene utilizzato il parametro listen_dns. Questo valore viene determinato eseguendo una ricerca DNS inversa dell'indirizzo IP dell'agent. Se il DNS inverso non è configurato correttamente o restituisce un nome non valido, l'autoregistrazione potrebbe non riuscire o produrre risultati errati a causa di un valore dell'interfaccia non valido.

Utilizzo dei metadati dell'host

Quando l'agent invia una richiesta di autoregistrazione al server, invia il proprio nome host. In alcuni casi (ad esempio, nei nodi cloud Amazon), un nome host non è sufficiente affinché il server Zabbix possa distinguere gli host rilevati. I metadati dell'host possono essere utilizzati facoltativamente per inviare altre informazioni da un agent al server.

I metadati dell'host vengono configurati nel file di configurazione dell'agent (`zabbix_agentd.conf`). Esistono 2 modi per specificare i metadati dell'host nel file di configurazione:

```
HostMetadata
HostMetadataItem
```

Vedere la descrizione delle opzioni nel link precedente.

Il parametro HostMetadataItem può restituire fino a 65535 punti di codice UTF-8. Un valore più lungo verrà troncato.

Si noti che in MySQL la lunghezza massima effettiva in caratteri sarà inferiore se il valore restituito contiene caratteri multibyte. Ad esempio, un valore contenente solo caratteri di 3 byte sarà limitato a un totale di 21844 caratteri, mentre un valore contenente solo caratteri di 4 byte sarà limitato a 16383 simboli.

Attention:
Un tentativo di autoregistrazione avviene ogni volta che un agent attivo invia una richiesta al server per aggiornare i controlli attivi. L'intervallo tra le richieste è specificato nel parametro RefreshActiveChecks dell'agent. La prima richiesta viene inviata immediatamente dopo il riavvio dell'agent.

Esempi

Autoregistrazione per OS utilizzando HostMetadata

Supponiamo di voler fare in modo che gli host vengano autoregistrati dal server Zabbix. Nella rete sono presenti agent Zabbix attivi (vedere la sezione "Configuration" sopra). Nella rete sono presenti host Windows e host Linux e nel frontend Zabbix sono disponibili i template "Linux by Zabbix agent" e "Windows by Zabbix agent". Quindi, al momento della registrazione dell'host, si desidera che al host in fase di registrazione venga applicato il template Linux/Windows appropriato. Per impostazione predefinita, durante l'autoregistrazione viene inviato al server solo il nome host, il che potrebbe non essere sufficiente. Per assicurarsi che al host venga applicato il template corretto, è necessario utilizzare i metadati dell'host.

Configurazione del frontend

La prima cosa da fare è configurare il frontend. Creare 2 azioni. La prima azione:

- Nome: Registrazione automatica del host Linux
- Condizioni: I metadati del host contengono *Linux*
- Operazioni: Collegare i template: Linux by Zabbix agent

Note:
In questo caso è possibile saltare un'operazione "Add host". Il collegamento di un template a un host richiede prima l'aggiunta del host, quindi il server lo farà automaticamente.

La seconda azione:

- Nome: Registrazione automatica del host Windows
- Condizioni: I metadati del host contengono *Windows*
- Operazioni: Collegare i template: Windows by Zabbix agent

Configurazione dell'agent

Ora è necessario configurare gli agent. Aggiungere la riga seguente ai file di configurazione dell'agent:

```
HostMetadataItem=system.uname
```

In questo modo ci si assicura che i metadati del host contengano "Linux" o "Windows" a seconda del host su cui è in esecuzione un agent. Un esempio di metadati del host in questo caso:

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-OPXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

Non dimenticare di riavviare l'agent dopo aver apportato qualsiasi modifica al file di configurazione.

Utilizzo di HostMetadata per controllare l'autoregistrazione e i template

Passaggio 1 - Proteggere la registrazione con HostMetadata

Utilizzare i metadati dell'host per consentire una protezione di base contro la registrazione di host indesiderati.

Configurazione del frontend

Creare un'azione nel frontend, utilizzando un codice segreto difficile da indovinare per impedire host indesiderati:

- Nome: Azione di autoregistrazione Linux
- Condizioni:
  - Tipo di calcolo: AND
  - Condizione (A): I metadati dell'host contengono //Linux//
  - Condizione (B): I metadati dell'host contengono //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//
- Operazioni:
  - Invia messaggio agli utenti: Admin tramite tutti i supporti
  - Aggiungi ai gruppi host: Server Linux
  - Collega template: Linux by Zabbix agent

Si noti che questo metodo da solo non fornisce una protezione efficace, poiché i dati vengono trasmessi in testo semplice. Per rendere immediatamente effettive le modifiche è necessario ricaricare la cache di configurazione.

Configurazione dell'agent

Aggiungere la riga seguente al file di configurazione dell'agent:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

dove "Linux" è una piattaforma e il resto della stringa è il testo segreto difficile da indovinare.

Non dimenticare di riavviare l'agent dopo aver apportato qualsiasi modifica al file di configurazione.

Passaggio 2 - Aggiungere il template al host registrato

È possibile aggiungere template aggiuntivi per un host già registrato. In questo caso, il template MySQL by Zabbix agent verrà collegato solo ai host il cui HostMetadata contiene il token `MySQL`.

Configurazione del frontend

Aggiornare l'azione nel frontend:

- Nome: Azione di autoregistrazione Linux
- Condizioni:
  - Tipo di calcolo: AND
  - Condizione (A): I metadati dell'host contengono Linux
  - Condizione (B): I metadati dell'host contengono 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
  - Condizione (C): I metadati dell'host contengono MySQL
- Operazioni:
  - Invia messaggio agli utenti: Admin tramite tutti i supporti

- Aggiungi ai gruppi host: Server Linux
- Collega i template: Linux by Zabbix agent
- Collega i template: MySQL by Zabbix Agent

Configurazione dell'agent

Aggiornare la riga seguente nel file di configurazione dell'agent:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Non dimenticare di riavviare l'agent dopo aver apportato qualsiasi modifica al file di configurazione.

Scoperta a basso livello

Panoramica La low-level discovery (LLD) rileva automaticamente le entità (ad esempio, file system, interfacce di rete) su un host e crea gli item, trigger e grafici corrispondenti senza configurazione manuale per ciascuna.

Per usare LLD, si crea una **regola di discovery** che raccoglie dati JSON che descrivono le entità e dei **prototype** (item, trigger e grafico) per ogni entità.

Il JSON restituito dalla regola deve essere un array di oggetti, in cui ogni oggetto rappresenta una singola entità scoperta tramite coppie chiave-valore. Ad esempio, una regola di discovery con `net.if.discovery` potrebbe restituire:

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

Un insieme corrispondente di item, trigger e grafici viene quindi creato dai prototype: uno per ogni interfaccia (`lo` e `eth0`).

Se gli oggetti contengono chiavi nel formato `{#MACRO}`, tali macro vengono usate direttamente nei prototype. In caso contrario, oppure se sono necessarie macro aggiuntive/personalizzate, le macro possono essere definite manualmente e mappate ai campi JSON tramite JSONPath.

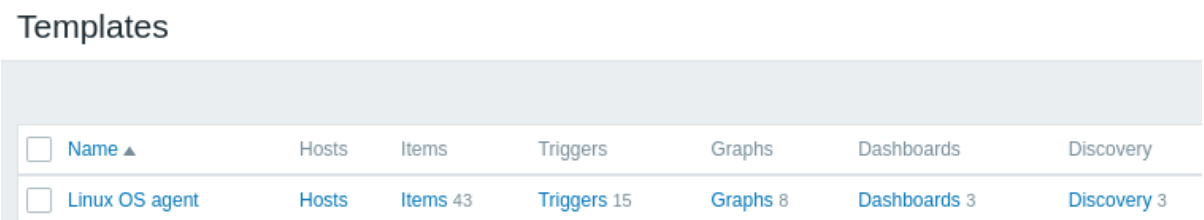
LLD può anche creare host (ad esempio, per macchine virtuali rilevate su un hypervisor) e supporta regole di discovery nidificate per il rilevamento multilivello.

Le entità che non vengono più rilevate possono essere automaticamente disabilitate o eliminate.

Configurazione della discovery a basso livello Illustreremo la discovery a basso livello con un esempio di discovery del file system.

Per configurare la discovery, procedere come segue:

1. Andare su *Data collection* > *Templates* o *Hosts*.
2. Fare clic su *Discovery* nella riga del template/host appropriato.



3. Fare clic su *Create discovery rule* nell'angolo in alto a destra della schermata.
4. Compilare il modulo della regola di discovery con i dettagli richiesti.

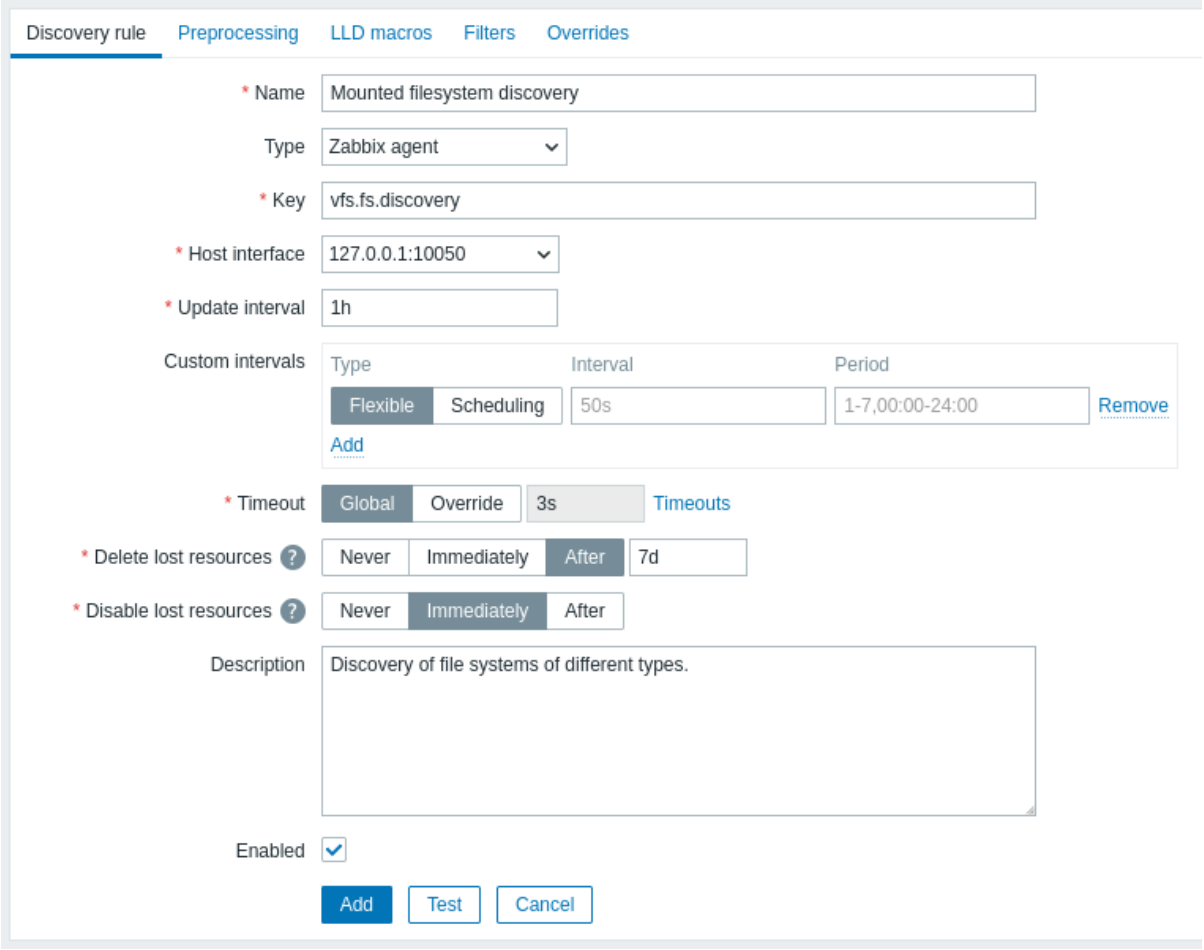
Regola di discovery

Il modulo della regola di discovery contiene cinque schede che rappresentano, da sinistra a destra, il flusso dei dati durante la discovery:

- *Regola di discovery* - specificare, soprattutto, l'item integrato o lo script personalizzato per recuperare i dati di discovery.
- *Preprocessing* - applicare il preprocessing ai dati rilevati.
- *Macro LLD* - consentire l'estrazione dei valori delle macro da utilizzare negli item rilevati, nei trigger e così via.
- *Filtri* - consentire il filtraggio dei valori rilevati.

- *Override* - consentire la modifica di item, trigger, grafici o prototipi di host quando vengono applicati a oggetti rilevati specifici.

La scheda **Regola di discovery** contiene la chiave dell'item da utilizzare per la discovery (oltre ad alcuni attributi generali della regola di discovery):



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Nome	Nome della regola di discovery.
Tipo	Tipo di controllo da eseguire per la discovery. In questo esempio utilizziamo un tipo di item *Zabbix agent*. La regola di discovery può anche essere un item dipendente, che dipende da un item normale. Non può dipendere da un'altra regola di discovery. Per un item dipendente, selezionare il tipo corrispondente (*Item dipendente*) e specificare l'item master nel campo 'Item master'. L'item master deve esistere.
Chiave	Inserire la chiave dell'item di discovery (fino a 2048 caratteri). Ad esempio, è possibile utilizzare la chiave dell'item integrato "vfs.fs.discovery" per restituire una stringa JSON con l'elenco dei file system presenti nel computer, i relativi tipi e le opzioni di mount. Si noti che un'altra opzione per la discovery dei file system consiste nell'utilizzare i risultati della discovery tramite la chiave agent "vfs.fs.get" (vedere l'esempio).

Parametro	Descrizione
Intervallo di aggiornamento	Questo campo specifica la frequenza con cui Zabbix esegue la discovery. All'inizio, quando si sta configurando la discovery dei file system, potrebbe essere utile impostare un intervallo breve; tuttavia, una volta verificato il corretto funzionamento, è possibile impostarlo su 30 minuti o più, poiché i file system generalmente non cambiano molto spesso. Sono supportati i suffissi temporali, ad esempio 30s, 1m, 2h, 1d. Sono supportate le macro utente. *Nota*: l'intervallo di aggiornamento può essere impostato su '0' solo se esistono intervalli personalizzati con un valore diverso da zero. Se impostato su '0' e se esiste un intervallo personalizzato (flessibile o pianificato) con un valore diverso da zero, l'item verrà interrogato per la durata dell'intervallo personalizzato. Le nuove regole di discovery verranno controllate entro 60 secondi dalla loro creazione, a meno che non abbiano un intervallo di aggiornamento Pianificazione o Flessibile e *Intervallo di aggiornamento* sia impostato su 0. *Nota*: per una regola di discovery esistente, la discovery può essere eseguita immediatamente premendo il pulsante *Esegui ora*.
Intervalli personalizzati	È possibile creare regole personalizzate per il controllo dell'item: **Flessibile** - creare un'eccezione all'*Intervallo di aggiornamento* (intervallo con frequenza diversa) **Pianificazione** - creare una pianificazione personalizzata per le interrogazioni. Per informazioni dettagliate, vedere Intervalli personalizzati.
Timeout	Impostare il timeout del controllo di discovery. Selezionare l'opzione di timeout: **Globale** - viene utilizzato il timeout del proxy/globale (visualizzato nel campo *Timeout* disabilitato); **Override** - viene utilizzato un timeout personalizzato (impostato nel campo *Timeout*; intervallo consentito: 1 - 600s). Sono supportati i suffissi temporali, ad esempio 30s e 1m, e le macro utente. Facendo clic sul link *Timeout* è possibile configurare i timeout del proxy o i timeout globali (se non viene utilizzato un proxy). Il link *Timeout* è visibile solo agli utenti di tipo *Super admin* con autorizzazioni per le sezioni frontend *Amministrazione* > *Generale* o *Amministrazione* > *Proxy*.
Elimina risorse non più rilevate	Specificare dopo quanto tempo l'entità rilevata verrà eliminata quando il suo stato di discovery diventa "Non più rilevata": *Mai* - non verrà eliminata; *Immediatamente* - verrà eliminata immediatamente; *Dopo* - verrà eliminata dopo il periodo di tempo specificato. Il valore deve essere maggiore del valore di *Disabilita risorse non più rilevate*. Sono supportati i suffissi temporali, ad esempio 2h e 1d. Sono supportate le macro utente. *Nota*: non è consigliato utilizzare "Immediatamente", poiché una semplice modifica errata del filtro potrebbe causare l'eliminazione dell'entità insieme a tutti i dati storici. Si noti che le risorse disabilitate manualmente non verranno eliminate dalla discovery di basso livello.
Disabilita risorse non più rilevate	Specificare dopo quanto tempo l'entità rilevata verrà disabilitata quando il suo stato di discovery diventa "Non più rilevata": *Mai* - non verrà disabilitata; *Immediatamente* - verrà disabilitata immediatamente; *Dopo* - verrà disabilitata dopo il periodo di tempo specificato. Il valore deve essere maggiore dell'intervallo di aggiornamento della regola di discovery. Si noti che le risorse disabilitate automaticamente verranno nuovamente abilitate se vengono rilevate di nuovo dalla discovery di basso livello. Le risorse disabilitate manualmente non verranno nuovamente abilitate se vengono rilevate di nuovo. Questo campo non viene visualizzato se *Elimina risorse non più rilevate* è impostato su "Immediatamente". Sono supportati i suffissi temporali, ad esempio 2h e 1d. Sono supportate le macro utente.
Descrizione	Inserire una descrizione.
Abilitato	Se selezionato, la regola verrà elaborata.

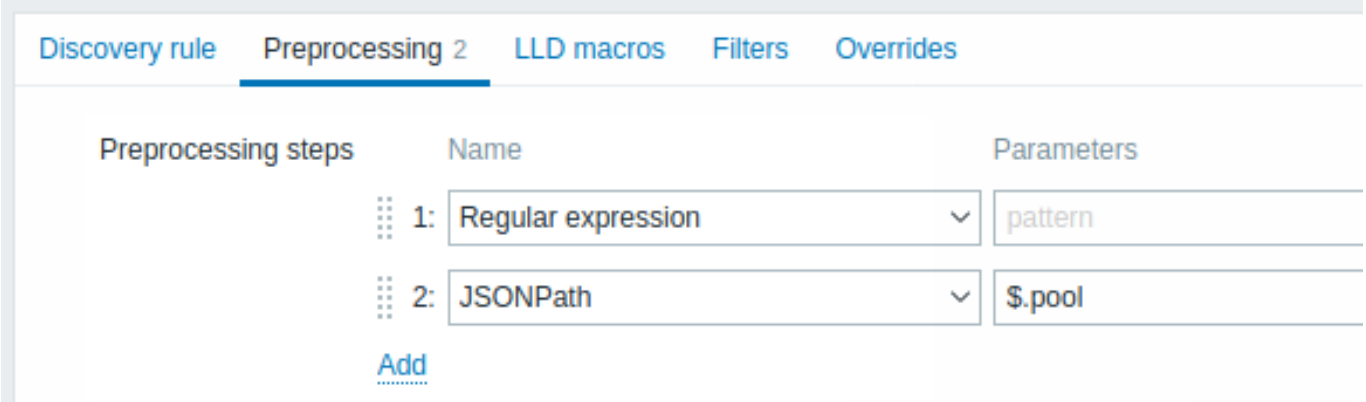
Note:
La cronologia della regola di discovery non viene conservata.

Preprocessing

La scheda **Preprocessing** consente di definire le regole di trasformazione da applicare al risultato della discovery. In questo passaggio è possibile utilizzare una o più trasformazioni. Le trasformazioni vengono eseguite nell'ordine in cui sono definite. Tutto il preprocessing viene eseguito dal server Zabbix.

Vedere anche:

- Dettagli del preprocessing
- Test del preprocessing



Tipo		
	Trasformazione	Descrizione
Testo		
	Espressione regolare	Confronta il valore ricevuto con l'espressione regolare <pattern> e sostituisce il valore con l'<output> estratto. L'espressione regolare supporta l'estrazione di un massimo di 10 gruppi acquisiti con la sequenza \N. Parametri: **pattern** - espressione regolare **output** - template di formattazione dell'output. Una sequenza di escape \N (dove N=1...9) viene sostituita con l'N-esimo gruppo corrispondente. Una sequenza di escape \0 viene sostituita con il testo corrispondente. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
	Sostituisci	Trova la stringa di ricerca e la sostituisce con un'altra stringa (oppure con niente). Tutte le occorrenze della stringa di ricerca verranno sostituite. Parametri: **search string** - stringa da trovare e sostituire, con distinzione tra maiuscole e minuscole (obbligatorio) **replacement** - stringa con cui sostituire la stringa di ricerca. La stringa di sostituzione può anche essere vuota, consentendo di fatto di eliminare la stringa di ricerca quando viene trovata. È possibile utilizzare sequenze di escape per cercare o sostituire interruzioni di riga, ritorni a capo, tabulazioni e spazi "\n \r \t \s"; la barra rovesciata può essere sottoposta a escape come "\\" e le sequenze di escape possono essere sottoposte a escape come "\\n". L'escape delle interruzioni di riga, dei ritorni a capo e delle tabulazioni viene eseguito automaticamente durante la low-level discovery.
Dati strutturati		
	JSONPath	Estrae un valore o un frammento dai dati JSON utilizzando la funzionalità JSONPath. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

Tipo	

XML XPath	Estrae un valore o un frammento dai dati XML utilizzando la funzionalità XPath.
	Per utilizzare questa opzione, il server Zabbix deve essere compilato con il supporto per libxml.
	Esempi:
	`number(/document/item/value)` estrarrà 10 da `<document><item><value>10</value></item></document>`
	`number(/document/item/@attribute)` estrarrà 10 da `<document><item attribute="10"></item></document>`
	`/document/item` estrarrà `<item><value>10</value></item>` da `<document><item><value>10</value></item></document>`
	Si noti che gli spazi dei nomi non sono supportati.
	Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

CSV in JSON	Converte i dati di un file CSV nel formato JSON. Per ulteriori informazioni, vedere: Preprocessing da CSV a JSON.
XML in JSON	Converte i dati in formato XML in JSON. Per ulteriori informazioni, vedere: Regole di serializzazione. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
SNMP	
Valore SNMP walk	Estrae il valore tramite il nome OID/MIB specificato e applica le opzioni di formattazione: **Invariato** - restituisce HEX-STRING come stringa esadecimale senza escape (*nota*: i suggerimenti di visualizzazione vengono comunque applicati); **UTF-8 da HEX-STRING** - converte HEX-STRING in una stringa UTF-8; **MAC da HEX-STRING** - converte HEX-STRING in una stringa di indirizzo MAC (in cui `' '` verrà sostituito da `':'`); **Intero da BITS** - converte i primi 8 byte di una stringa di bit espressa come sequenza di caratteri esadecimali (ad esempio, ”1A 2B 3C 4D”) in un intero senza segno a 64 bit. Nelle stringhe di bit più lunghe di 8 byte, i byte successivi verranno ignorati. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
SNMP walk in JSON	Converte i valori SNMP in JSON. Specificare un nome di campo nel JSON e il percorso OID SNMP corrispondente. I valori dei campi verranno popolati con i valori del percorso OID SNMP specificato. È possibile utilizzare questo passaggio di preprocessing per la discovery degli OID SNMP. Sono disponibili opzioni di formattazione dei valori simili a quelle del passaggio *Valore SNMP walk*. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Valore SNMP get	Applica le opzioni di formattazione al valore SNMP get: **UTF-8 da HEX-STRING** - converte HEX-STRING in una stringa UTF-8; **MAC da HEX-STRING** - converte HEX-STRING in una stringa di indirizzo MAC (in cui `' '` verrà sostituito da `':'`); **Intero da BITS** - converte i primi 8 byte di una stringa di bit espressa come sequenza di caratteri esadecimali (ad esempio, ”1A 2B 3C 4D”) in un intero senza segno a 64 bit. Nelle stringhe di bit più lunghe di 8 byte, i byte successivi verranno ignorati. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Script personalizzati	

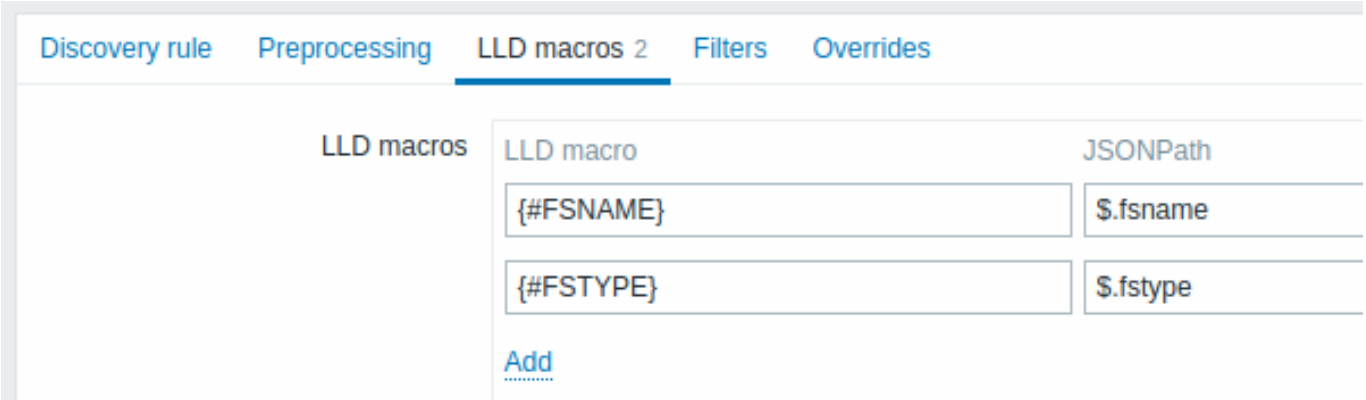
Tipo	
JavaScript	Inserire il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona a forma di matita accanto ad esso. La lunghezza JavaScript disponibile dipende dal database utilizzato. Per ulteriori informazioni, vedere: Preprocessing JavaScript
Convalida	
Non corrisponde all'espressione regolare	Specificare un'espressione regolare a cui il valore non deve corrispondere. Ad esempio, `Error:(.*?)\.` Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Verifica della presenza di errori in JSON	Verifica la presenza di un messaggio di errore a livello applicativo situato in JSONPath. Interrompe l'elaborazione se la verifica ha esito positivo e il messaggio non è vuoto; in caso contrario, continua l'elaborazione con il valore precedente a questo passaggio di preprocessing. Si noti che questi errori del servizio esterno vengono segnalati all'utente così come sono, senza aggiungere informazioni sul passaggio di preprocessing. Ad esempio, `$.errors`. Se viene ricevuto un JSON come `{"errors":"e1"}`, il passaggio di preprocessing successivo non verrà eseguito. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Verifica della presenza di errori in XML	Verifica la presenza di un messaggio di errore a livello applicativo situato in Xpath. Interrompe l'elaborazione se la verifica ha esito positivo e il messaggio non è vuoto; in caso contrario, continua l'elaborazione con il valore precedente a questo passaggio di preprocessing. Si noti che questi errori del servizio esterno vengono segnalati all'utente così come sono, senza aggiungere informazioni sul passaggio di preprocessing. Non verrà segnalato alcun errore se non è possibile analizzare XML non valido. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Corrisponde all'espressione regolare	Specificare un'espressione regolare a cui il valore deve corrispondere. Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: eliminare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Limitazione	
Elimina i valori invariati con heartbeat	Elimina un valore se non è cambiato entro il periodo di tempo definito (in secondi). Sono supportati valori interi positivi per specificare i secondi (minimo: 1 secondo). In questo campo è possibile utilizzare i suffissi temporali (ad esempio, `30s`, `1m`, `2h`, `1d`). In questo campo è possibile utilizzare macro utente e macro di low-level discovery. Per un discovery item è possibile specificare una sola opzione di limitazione. Ad esempio, `1m`. Se un testo identico viene passato due volte a questa regola entro 60 secondi, verrà eliminato. *Nota*: la modifica dei prototipi degli item non reimposta la limitazione. La limitazione viene reimpostata solo quando vengono modificati i passaggi di preprocessing.
Prometheus	
Prometheus in JSON	Converte le metriche Prometheus richieste in JSON. Per ulteriori informazioni, vedere Controlli Prometheus.

Si noti che, se la discovery rule è stata applicata al host tramite un template, il contenuto di questa scheda è di sola lettura.

Macro personalizzate

La scheda **LLD macros** consente di specificare LLD macros personalizzate quando il JSON restituito non contiene già chiavi nel formato `{#MACRO}`.

Ad esempio, la chiave dell'item agent `vfs.fs.get` restituisce dati JSON filesystem senza alcuna LLD macro predefinita.
Qui, si definiscono manualmente le macro e le si mappano ai valori nel JSON usando espressioni JSONPath:



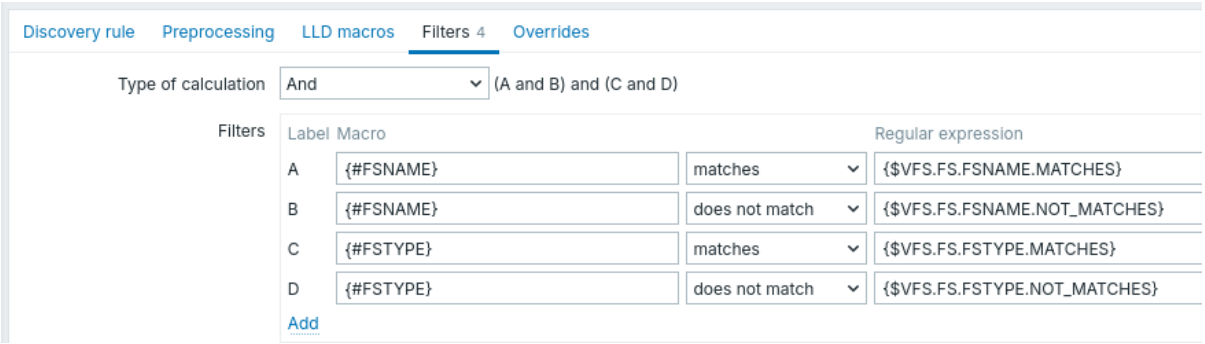
Parameter	Description
LLD macro	Nome della LLD macro nel formato `{#MACRO}`.
JSONPath	Espressione che definisce quale valore estrarre da ciascuna entità rilevata nell'array JSON. Ad esempio, `$.foo` estrae `"bar"` e `"baz"` da `[{"foo":"bar"}, {"foo":"baz"}]`. JSONPath supporta sia la notazione con punto sia quella con parentesi quadre. La notazione con parentesi quadre deve essere usata per chiavi contenenti caratteri speciali o Unicode, ad esempio, `$['unicode + special chars #1']['unicode + special chars #2']`.

La regola di discovery deve restituire un array JSON alla radice (`$.`).
Per compatibilità con le versioni precedenti, è accettato anche un oggetto JSON contenente un singolo array `"data"`. In questo caso, l'array `"data"` viene automaticamente estratto:

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

Filtro

Un filtro può essere utilizzato per generare item, trigger e grafici reali solo per le entità che corrispondono ai criteri. La scheda **Filtri** contiene le definizioni dei filtri delle regole di discovery che consentono di filtrare i valori rilevati:



Parametro	Descrizione
Tipo di calcolo	Sono disponibili le seguenti opzioni per il calcolo dei filtri: **And** - tutti i filtri devono essere superati; **Or** - è sufficiente superare un filtro; **And/Or** - utilizza *And* con nomi di macro diversi e *Or* con lo stesso nome di macro; **Espressione personalizzata** - consente di definire un calcolo personalizzato dei filtri. La formula deve includere tutti i filtri presenti nell'elenco. Limitata a 255 simboli.

Parametro	Descrizione
Filtri	Sono disponibili i seguenti operatori per le condizioni dei filtri: *corrisponde*, *non corrisponde*, *esiste*, *non esiste*. Gli operatori *corrisponde* e *non corrisponde* richiedono una Perl Compatible Regular Expression (PCRE). È possibile inserire un'espressione regolare o fare riferimento a un'espressione regolare globale nel campo *Espressione regolare*. Gli operatori *esiste* e *non esiste* consentono di filtrare le entità in base alla presenza o all'assenza della macro LLD specificata nella risposta. Se nella risposta manca una macro del filtro, l'entità rilevata verrà ignorata, a meno che per questa macro non sia specificata una condizione "non esiste". Verrà visualizzato un avviso se l'assenza di una macro influisce sul risultato dell'espressione. Ad esempio, se {#B} manca in: {#A} corrisponde a 1 e {#B} corrisponde a 2 - verrà visualizzato un avviso {#A} corrisponde a 1 oppure {#B} corrisponde a 2 - nessun avviso.

Ad esempio, se si è interessati solo ai file system C:, D: ed E:, è possibile inserire {#FSNAME} nel campo *Macro* e ^C|^D|^E nel campo *Espressione regolare*.
 È inoltre possibile filtrare in base ai tipi di file system utilizzando la macro {#FSTYPE} (ad esempio, ^ext|^reiserfs) e in base ai tipi di unità (supportati solo dall'agent Windows) utilizzando la macro {#FSDRIVETYPE} (ad esempio, fixed).

Per testare un'espressione regolare è possibile utilizzare grep -E, ad esempio:

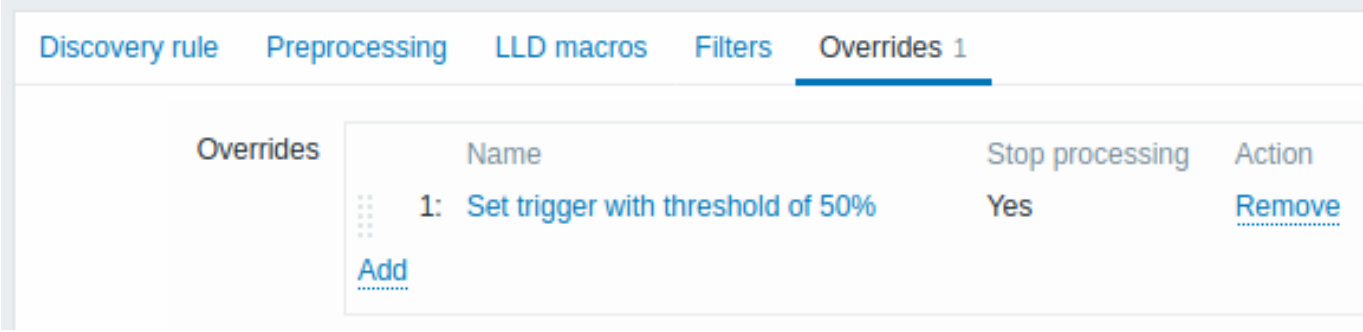
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
Un errore o un refuso nell'espressione regolare utilizzata nella regola LLD (ad esempio, un'espressione regolare errata per "File system da rilevare") può causare l'eliminazione di migliaia di elementi di configurazione, valori storici ed eventi per numerosi host.

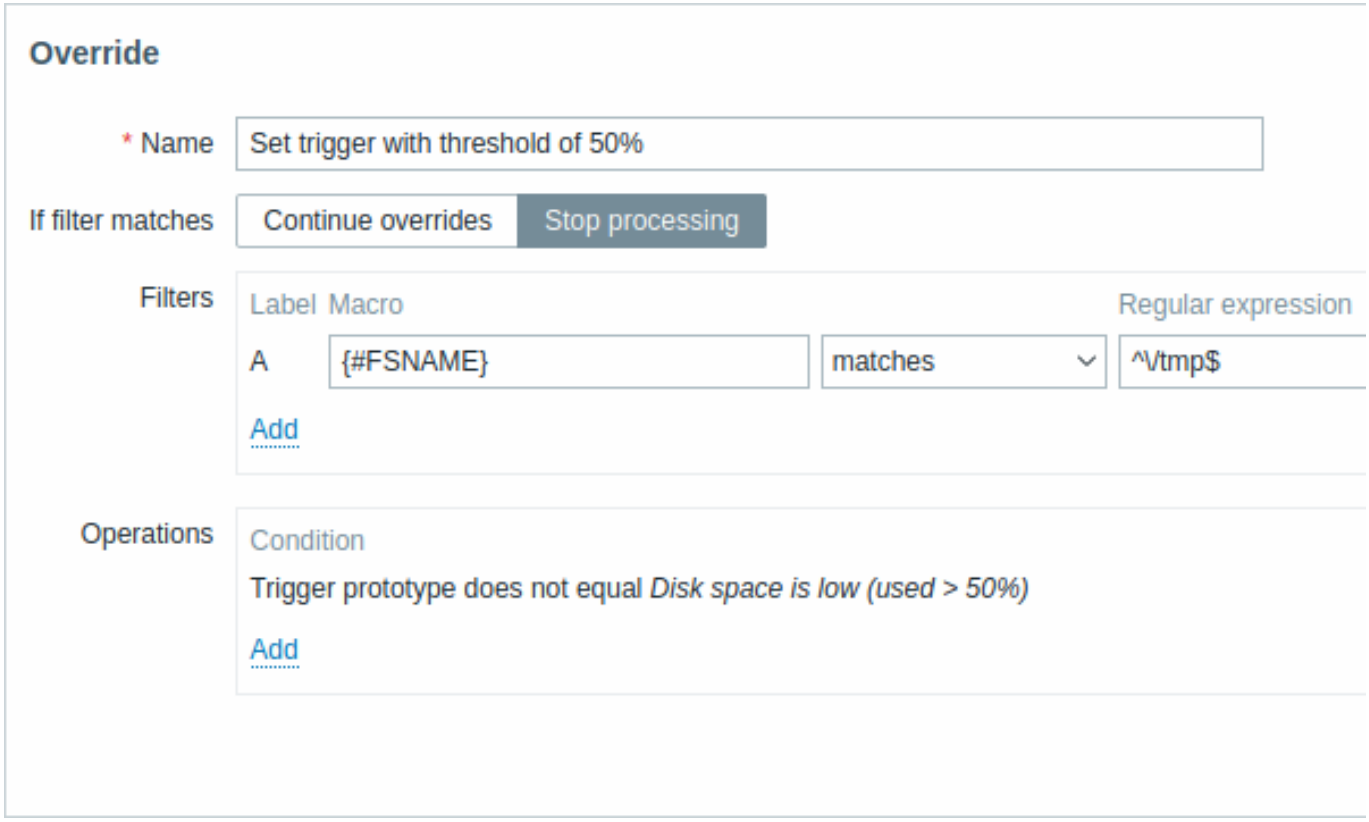
Attention:
Il database Zabbix in MySQL deve essere creato con distinzione tra maiuscole e minuscole affinché i nomi dei file system che differiscono solo per la combinazione di maiuscole e minuscole vengano rilevati correttamente.

Sovrascritture

La scheda **Sovrascritture** consente di impostare regole per modificare l'elenco dei prototipi di item, trigger, grafici, host e discovery, oppure i relativi attributi, per gli oggetti rilevati che soddisfano determinati criteri.



Le sovrascritture, se presenti, vengono visualizzate in un elenco riordinabile tramite trascinamento e vengono eseguite nell'ordine in cui sono definite. Per configurare i dettagli di una nuova sovrascrittura, fare clic su Add nel blocco *Sovrascritture*. Per modificare una sovrascrittura esistente, fare clic sul nome della sovrascrittura. Si aprirà una finestra popup che consente di modificare i dettagli della regola di sovrascrittura.

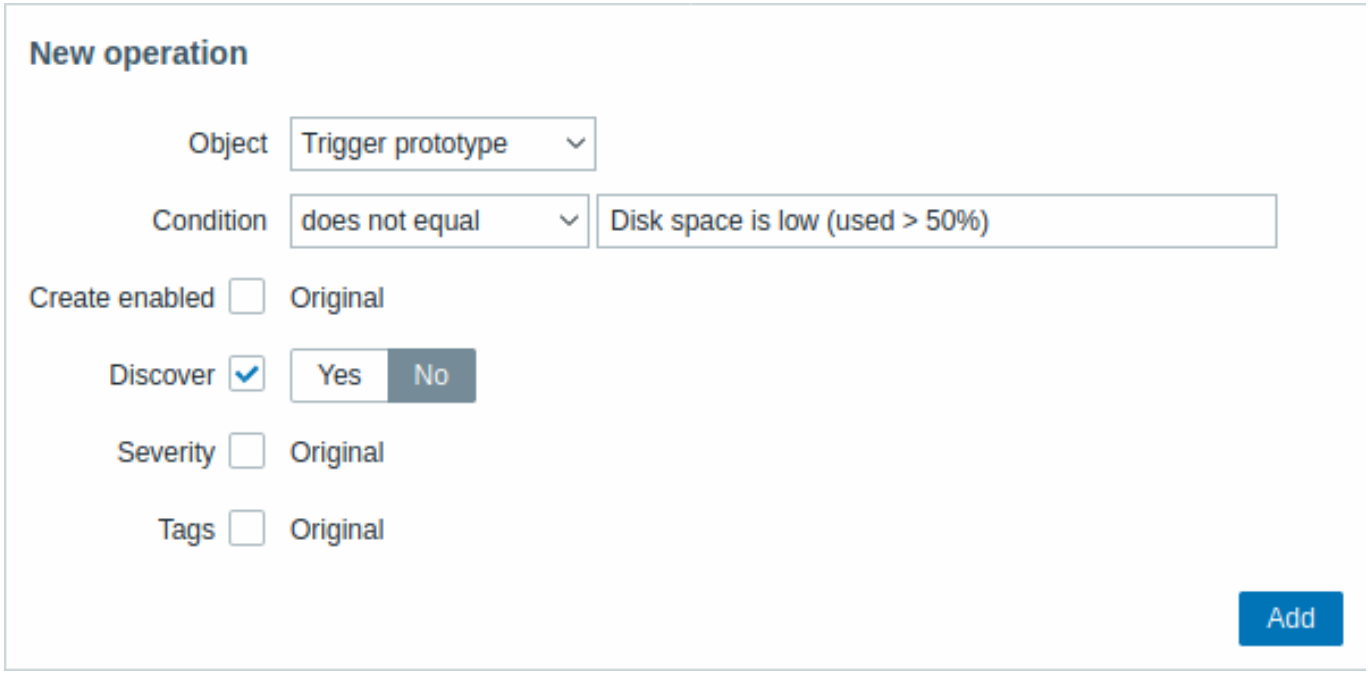


Tutti i parametri obbligatori sono contrassegnati da asterischi rossi.

Parametro	Descrizione
Nome	Un nome univoco della sovrascrittura (per regola LLD).
Se il filtro corrisponde	Definisce se le sovrascritture successive devono essere elaborate quando le condizioni del filtro sono soddisfatte: **Continua con le sovrascritture** - le sovrascritture successive verranno elaborate. **Interrompi elaborazione** - verranno eseguite le operazioni precedenti, se presenti, e quelle di questa sovrascrittura; le sovrascritture successive verranno ignorate per le righe LLD corrispondenti.
Filtri	Determina a quali entità rilevate deve essere applicata la sovrascrittura. I filtri delle sovrascritture vengono elaborati dopo i filtri della regola di discovery e hanno la stessa funzionalità.
Operazioni	Le operazioni di sovrascrittura vengono visualizzate con i seguenti dettagli: **Condizione** - un tipo di oggetto e una condizione da soddisfare per il nome dell'oggetto; ad esempio: il prototipo di trigger non è uguale a *Spazio su disco insufficiente (utilizzato > 50%)*. **Azioni** - vengono visualizzati i collegamenti per modificare e rimuovere un'operazione.

Configurazione di un'operazione

Per configurare i dettagli di una nuova operazione, fare clic su Add nel blocco Operazioni. Per modificare un'operazione esistente, fare clic su Edit accanto all'operazione. Si aprirà una finestra popup in cui è possibile modificare i dettagli dell'operazione.



Parametro	Descrizione
Oggetto	Sono disponibili cinque tipi di oggetti: Prototipo di item Prototipo di trigger Prototipo di grafico Prototipo di host Prototipo di discovery
Condizione	Consente di filtrare le entità a cui deve essere applicata l'operazione.
Operatore	Operatori supportati: **è uguale a** - applica a questo prototipo **non è uguale a** - applica a tutti i prototipi tranne questo **contiene** - applica se il nome del prototipo contiene questa stringa **non contiene** - applica se il nome del prototipo non contiene questa stringa **corrisponde** - applica se il nome del prototipo corrisponde a un'espressione regolare **non corrisponde** - applica se il nome del prototipo non corrisponde a un'espressione regolare
Pattern	Un'espressione regolare o una stringa da cercare.
Oggetto: *Prototipo di item*	
Crea abilitato	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di item: *Sì* - l'item verrà aggiunto in stato abilitato. *No* - l'item verrà aggiunto a un'entità rilevata, ma in stato disabilitato.
Rileva	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di item: *Sì* - l'item verrà aggiunto. *No* - l'item non verrà aggiunto.
Intervallo di aggiornamento	Quando la casella è selezionata, vengono visualizzate due opzioni che consentono di impostare un intervallo diverso per l'item: *Ritardo* - intervallo di aggiornamento dell'item. Sono supportate le macro utente, le macro LLD e i suffissi temporali (ad es. 30s, 1m, 2h, 1d). Deve essere impostato su 0 se viene utilizzato *Intervallo personalizzato*. *Intervallo personalizzato* - fare clic su Add per specificare intervalli flessibili o pianificati. Per informazioni dettagliate, vedere Intervalli personalizzati.

Parametro	Descrizione
Cronologia	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di impostare periodi diversi di conservazione della cronologia per l'item: *Non memorizzare* - se selezionato, la cronologia non verrà memorizzata. *Memorizza fino a* - se selezionato, a destra verrà visualizzato un campo di input per specificare il periodo di conservazione. Sono supportate le macro utente e le macro LLD.
Trend	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di impostare periodi diversi di conservazione dei trend per l'item: *Non memorizzare* - se selezionato, i trend non verranno memorizzati. *Memorizza fino a* - se selezionato, a destra verrà visualizzato un campo di input per specificare il periodo di conservazione. Sono supportate le macro utente e le macro LLD.
Tag	Quando la casella è selezionata, viene visualizzato un nuovo blocco che consente di specificare coppie tag-valore. Sono supportate le macro utente e le macro LLD. Questi tag verranno aggiunti ai tag specificati nel prototipo di item, anche se i nomi dei tag coincidono.
Oggetto: *Prototipo di trigger*	
Crea abilitato	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di trigger: *Sì* - il trigger verrà aggiunto in stato abilitato. *No* - il trigger verrà aggiunto a un'entità rilevata, ma in stato disabilitato.
Rileva	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di trigger: *Sì* - il trigger verrà aggiunto. *No* - il trigger non verrà aggiunto.
Gravità	Quando la casella è selezionata, vengono visualizzati i pulsanti della gravità del trigger, che consentono di modificarne la gravità.
Tag	Quando la casella è selezionata, viene visualizzato un nuovo blocco che consente di specificare coppie tag-valore. Sono supportate le macro utente e le macro LLD. Questi tag verranno aggiunti ai tag specificati nel prototipo di trigger, anche se i nomi dei tag coincidono.
Oggetto: *Prototipo di grafico*	
Rileva	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di grafico: *Sì* - il grafico verrà aggiunto. *No* - il grafico non verrà aggiunto.
Oggetto: *Prototipo di host*	
Crea abilitato	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di host: *Sì* - l'host verrà creato in stato abilitato. *No* - l'host verrà creato in stato disabilitato.
Rileva	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di host: *Sì* - l'host verrà rilevato. *No* - l'host non verrà rilevato.
Collega template	Quando la casella è selezionata, viene visualizzato un campo di input per specificare i template. Iniziare a digitare il nome del template oppure fare clic su *Seleziona* accanto al campo e selezionare i template dall'elenco in una finestra popup. I template di questa sovrascrittura vengono aggiunti a tutti i template già collegati al prototipo di host.

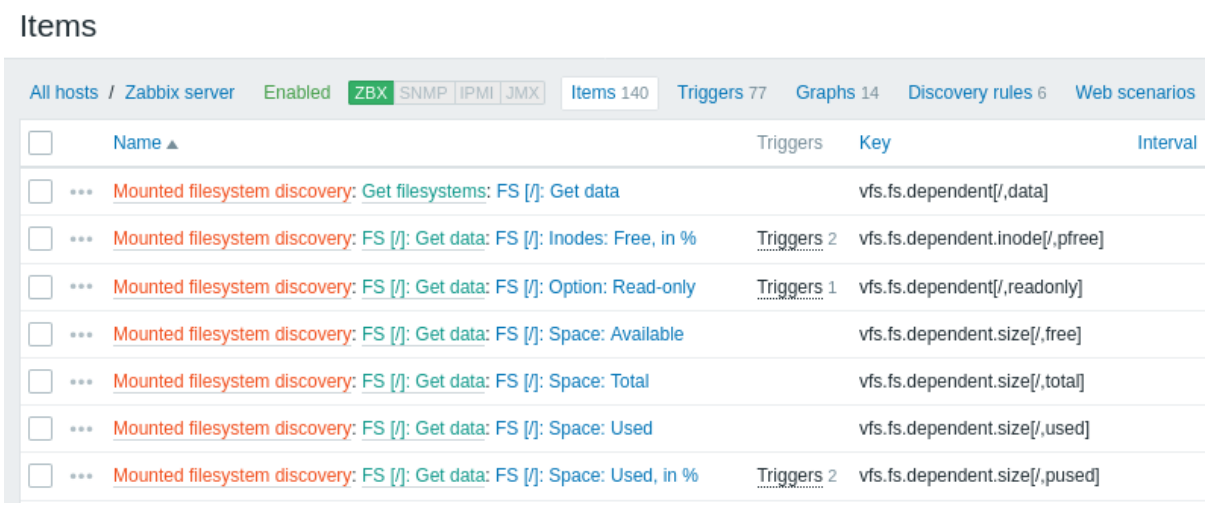
Parametro		Descrizione
	Tag	Quando la casella è selezionata, viene visualizzato un nuovo blocco che consente di specificare coppie tag-valore. Sono supportate le macro utente e le macro LLD. Questi tag verranno aggiunti ai tag specificati nel prototipo di host, anche se i nomi dei tag coincidono.
	Inventario dell'host	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di selezionare una diversa modalità di inventario per il prototipo di host: *Disabilitato* - non compilare l'inventario dell'host *Manuale* - fornire manualmente i dettagli *Automatico* - compilare automaticamente i dati dell'inventario dell'host in base alle metriche raccolte.
Oggetto: *Prototipo di discovery*		
	Crea abilitato	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di discovery: *Sì* - la regola di discovery verrà aggiunta in stato abilitato. *No* - la regola di discovery verrà aggiunta in stato disabilitato.
	Rileva	Quando la casella è selezionata, vengono visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di discovery: *Sì* - la regola di discovery verrà aggiunta. *No* - la regola di discovery non verrà aggiunta.
	Intervallo di aggiornamento	Quando la casella è selezionata, vengono visualizzate due opzioni che consentono di impostare un intervallo diverso per la regola di discovery: *Ritardo* - intervallo di aggiornamento della regola. Sono supportate le macro utente, le macro LLD e i suffissi temporali (ad es. 30s, 1m, 2h, 1d). Deve essere impostato su 0 se viene utilizzato *Intervallo personalizzato*. *Intervallo personalizzato* - fare clic su Add per specificare intervalli flessibili o pianificati. Per informazioni dettagliate, vedere Intervalli personalizzati.

Pulsanti del modulo

I pulsanti nella parte inferiore del modulo consentono di eseguire diverse operazioni.

Add	Aggiunge una regola di discovery. Questo pulsante è disponibile solo per le nuove regole di discovery.
Update	Aggiorna le proprietà di una regola di discovery. Questo pulsante è disponibile solo per le regole di discovery esistenti.
Clone	Crea un'altra regola di discovery basata sulle proprietà della regola di discovery corrente.
Execute now	Esegue immediatamente la discovery in base alla regola di discovery. La regola di discovery deve già esistere. Vedere ulteriori dettagli. *Nota* che, quando si esegue immediatamente la discovery, la cache di configurazione non viene aggiornata; pertanto, il risultato non rifletterà le modifiche più recenti alla configurazione della regola di discovery.
Test	Verifica la configurazione della regola di discovery. Utilizzare questo pulsante per verificare le impostazioni di configurazione, come la connettività e la correttezza dei parametri, senza applicare definitivamente alcuna modifica.
Delete	Elimina la regola di discovery.
Cancel	Annulla la modifica delle proprietà della regola di discovery.

Entità rilevate Le schermate seguenti illustrano come appaiono item, trigger e grafici rilevati nella configurazione dell'host. Le entità rilevate sono precedute da un collegamento arancione alla regola di discovery da cui provengono.



Si noti che le entità rilevate non verranno create nel caso in cui esistano già entità con gli stessi criteri di unicità, ad esempio un item con la stessa chiave o un grafico con lo stesso nome. In questo caso, nel frontend viene visualizzato un messaggio di errore che indica che la regola di low-level discovery non ha potuto creare determinate entità. La regola di discovery stessa, tuttavia, non diventerà unsupported perché alcune entità non hanno potuto essere create e hanno dovuto essere saltate. La regola di discovery continuerà a creare/aggiornare altre entità.

Se un'entità rilevata (host, file system, interfaccia, ecc.) smette di essere rilevata (o non supera più il filtro), le entità create sulla sua base possono essere automaticamente disabilitate ed eventualmente eliminate.

Le risorse perse possono essere automaticamente disabilitate in base al valore del parametro *Disable lost resources*. Questo riguarda host, item e trigger persi.

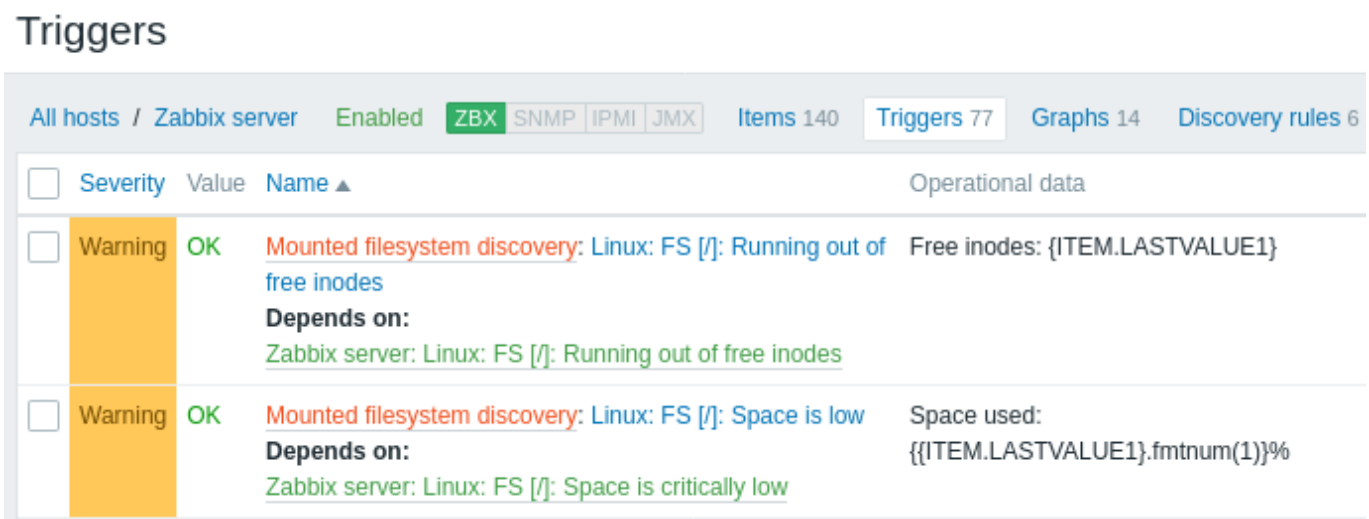
Le risorse perse possono essere automaticamente eliminate in base al valore del parametro *Delete lost resources*. Questo riguarda host, gruppi di host, item, trigger e grafici persi.

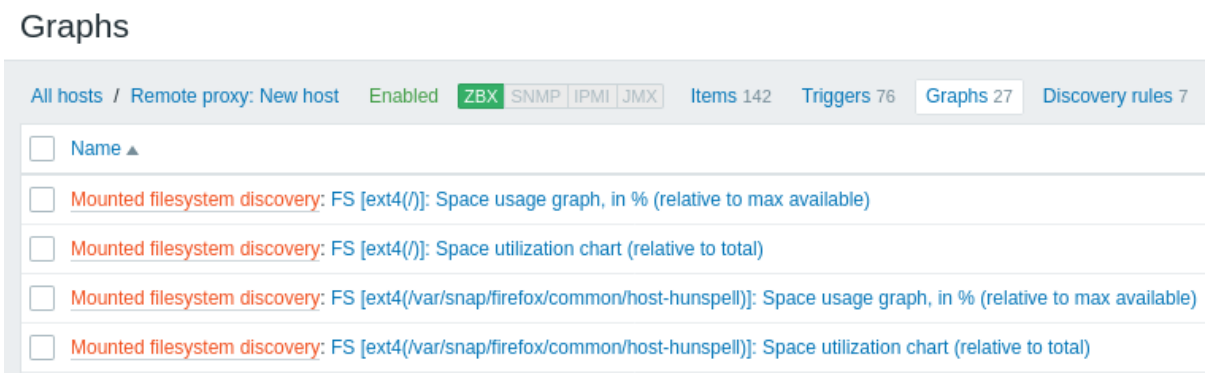
Quando le entità rilevate diventano 'Not discovered anymore', nell'elenco delle entità viene visualizzato un indicatore di durata. Spostando il puntatore del mouse su di esso, verrà visualizzato un messaggio che ne indica i dettagli di stato.



Se le entità sono state contrassegnate per l'eliminazione, ma non sono state eliminate nel momento previsto (regola di discovery o host dell'item disabilitati), verranno eliminate la volta successiva in cui la regola di discovery verrà elaborata.

Le entità che contengono altre entità contrassegnate per l'eliminazione non verranno aggiornate se modificate a livello di regola di discovery. Ad esempio, i trigger basati su LLD non verranno aggiornati se contengono item contrassegnati per l'eliminazione.





Altri tipi di discovery Maggiori dettagli e istruzioni su altri tipi di discovery pronti all'uso sono disponibili nelle sezioni seguenti:

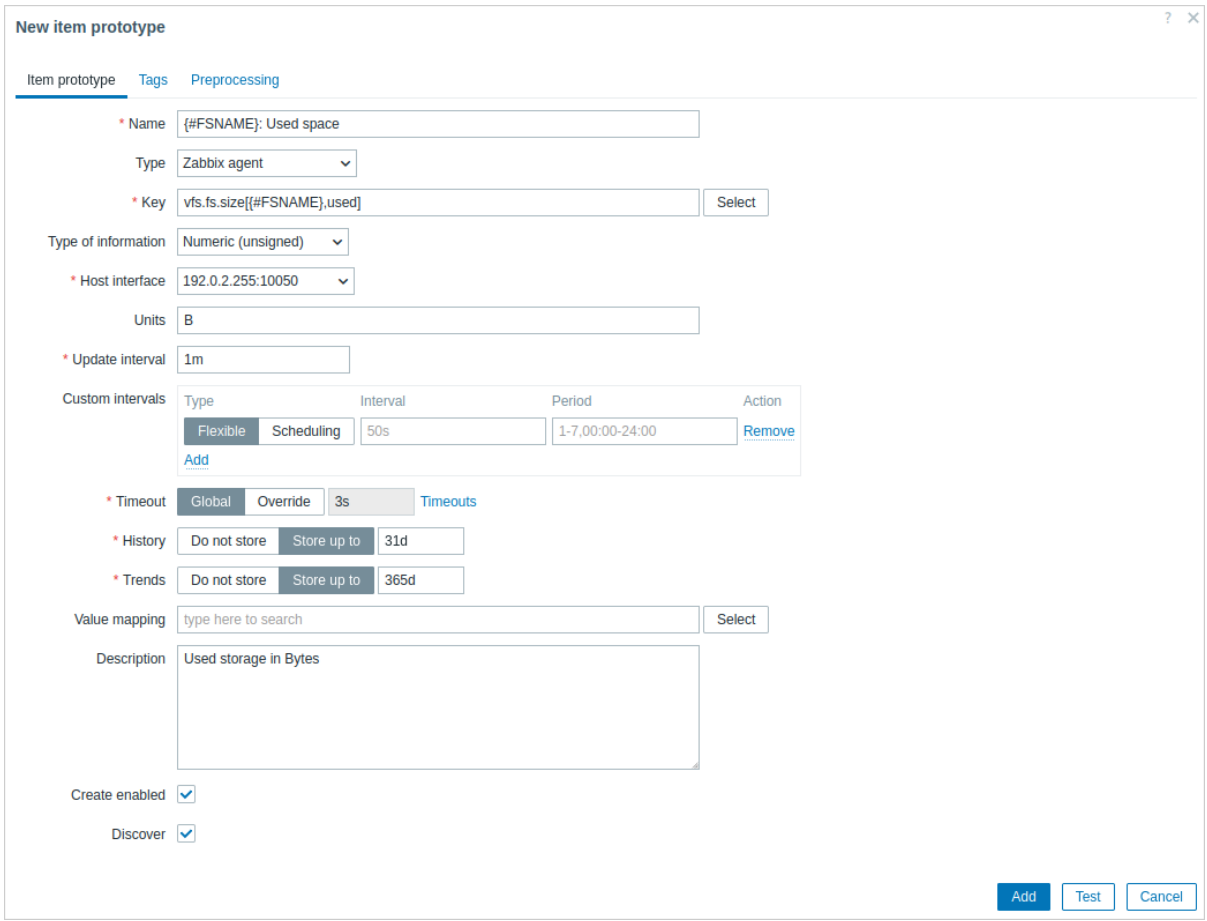
- discovery di interfacce di rete
- discovery di CPU e core della CPU
- discovery di OID SNMP
- discovery di oggetti JMX;
- discovery tramite query SQL ODBC
- discovery di servizi Windows
- discovery di interfacce host in Zabbix

Per maggiori dettagli sul formato JSON per gli item di discovery e un esempio su come implementare il proprio rilevatore di file system come script Perl, vedere creazione di regole LLD personalizzate.

Prototipi di item

Dopo aver creato una regola, accedere agli item di tale regola e fare clic su *Crea prototipo di item* per creare un prototipo di item.

Notare come la macro `{#FSNAME}` venga utilizzata quando è richiesto il nome di un file system. L'uso di una macro di low-level discovery nella chiave dell'item è obbligatorio per assicurarsi che la discovery venga elaborata correttamente. Quando la regola di discovery viene elaborata, questa macro verrà sostituita con il file system rilevato.



Note:
Quando i parametri della chiave dell'item non vengono utilizzati, inserire la macro LLD tra le parentesi quadre della chiave dell'item `[...]` (ad esempio, `v_[{#MACRO}]`).

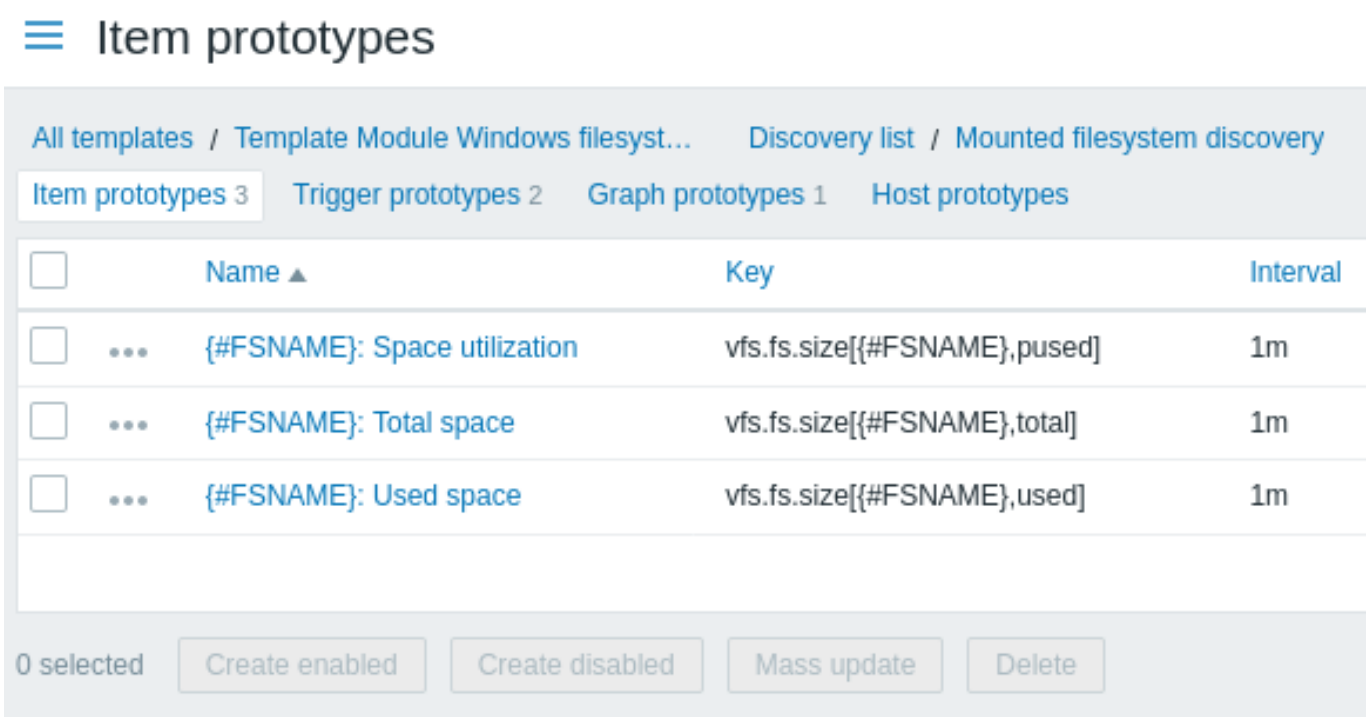
Le macro di low-level discovery e le macro utente sono supportate nella configurazione dei prototipi di item e nei parametri di pre-elaborazione dei valori degli item. Notare che, quando vengono utilizzate negli intervalli di aggiornamento, una singola macro deve riempire l'intero campo. Non sono supportate più macro nello stesso campo né macro combinate con testo.

Note:
L'escape contestuale delle macro di low-level discovery viene eseguito per un utilizzo sicuro nei parametri di pre-elaborazione delle espressioni regolari e di XPath.

Attributi specifici dei prototipi di item:

Parametro	Descrizione
Crea abilitato	Se selezionato, l'item verrà aggiunto in stato abilitato. Se non selezionato, l'item verrà aggiunto a un'entità rilevata, ma in stato disabilitato.
Rileva	Se selezionato (impostazione predefinita), l'item verrà aggiunto a un'entità rilevata. Se non selezionato, l'item non verrà aggiunto a un'entità rilevata, a meno che questa impostazione non venga sovrascritta nella regola di discovery.

È possibile creare diversi prototipi di item per ogni metrica del file system di interesse:



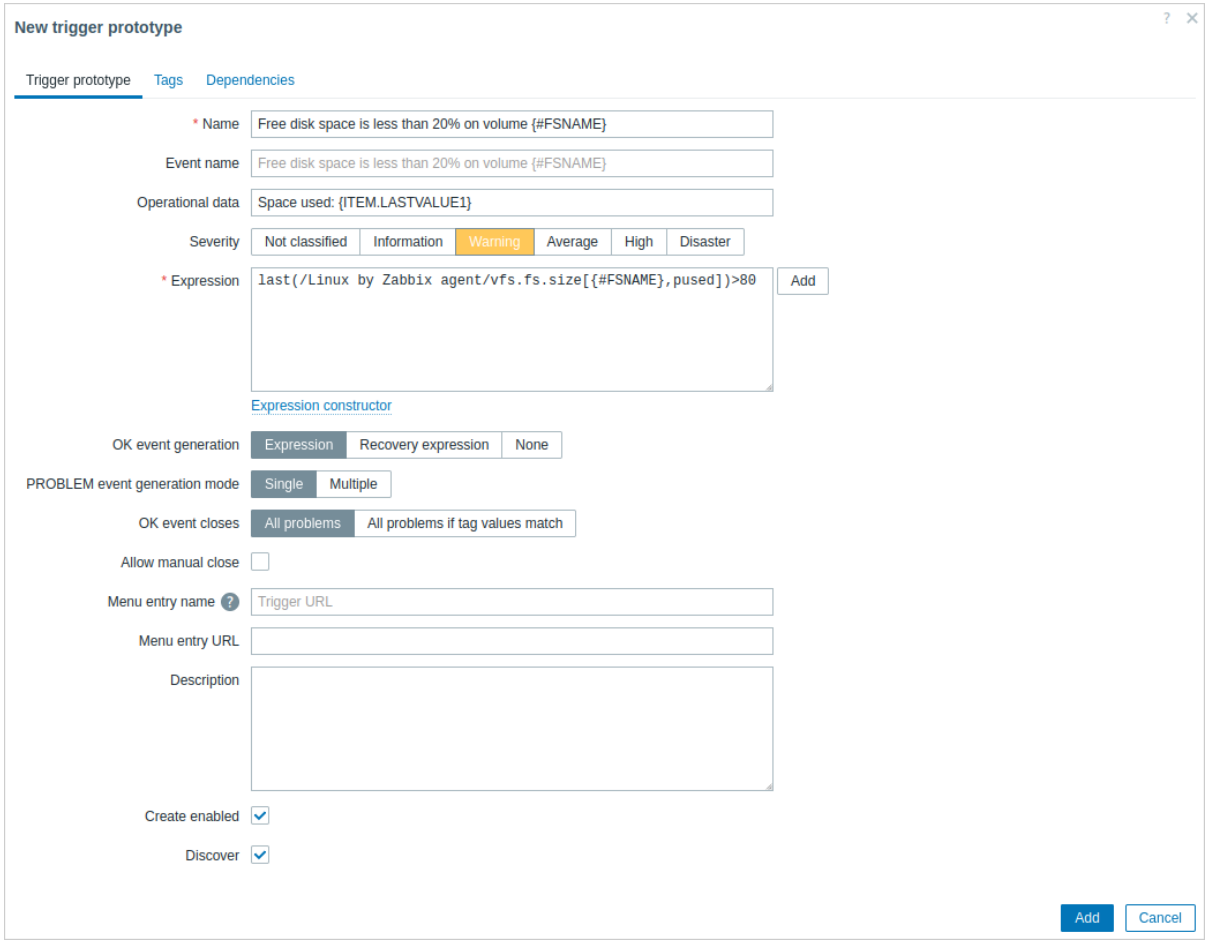
Fare clic sull'icona con tre punti per aprire il menu del prototipo di item specifico, con le seguenti opzioni:

- *Crea prototipo di trigger* - crea un prototipo di trigger basato su questo prototipo di item
- *Prototipi di trigger* - fare clic per visualizzare un elenco con i collegamenti ai prototipi di trigger già configurati per questo prototipo di item
- *Crea item dipendente* - crea un item dipendente per questo prototipo di item

L'opzione *Aggiornamento di massa* è disponibile se si desidera aggiornare contemporaneamente le proprietà di diversi prototipi di item.

Prototipi di trigger

Creiamo i prototipi di trigger in modo simile ai prototipi di item:

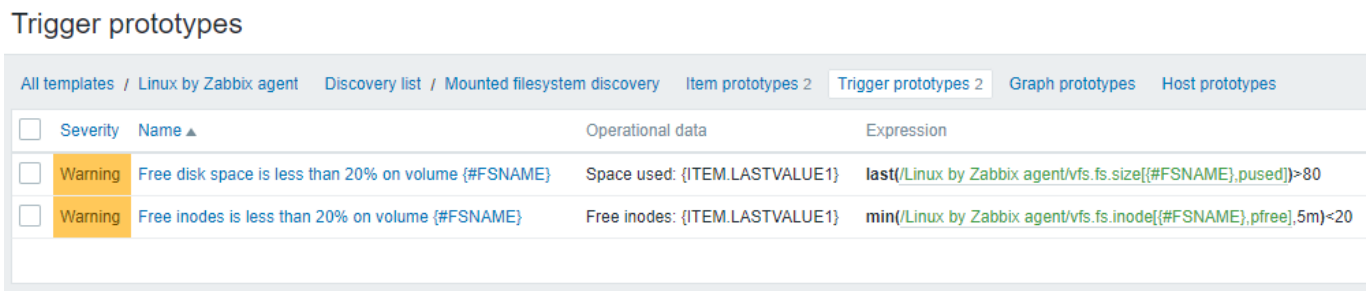


Attributi specifici per i prototipi di trigger:

Parameter	Description
Create enabled	Se selezionato, il trigger verrà aggiunto nello stato abilitato. Se non selezionato, il trigger verrà aggiunto a un'entità scoperta, ma nello stato disabilitato.
Discover	Se selezionato (impostazione predefinita), il trigger verrà aggiunto a un'entità scoperta. Se non selezionato, il trigger non verrà aggiunto a un'entità scoperta, a meno che questa impostazione non venga sovrascritta nella regola di discovery.

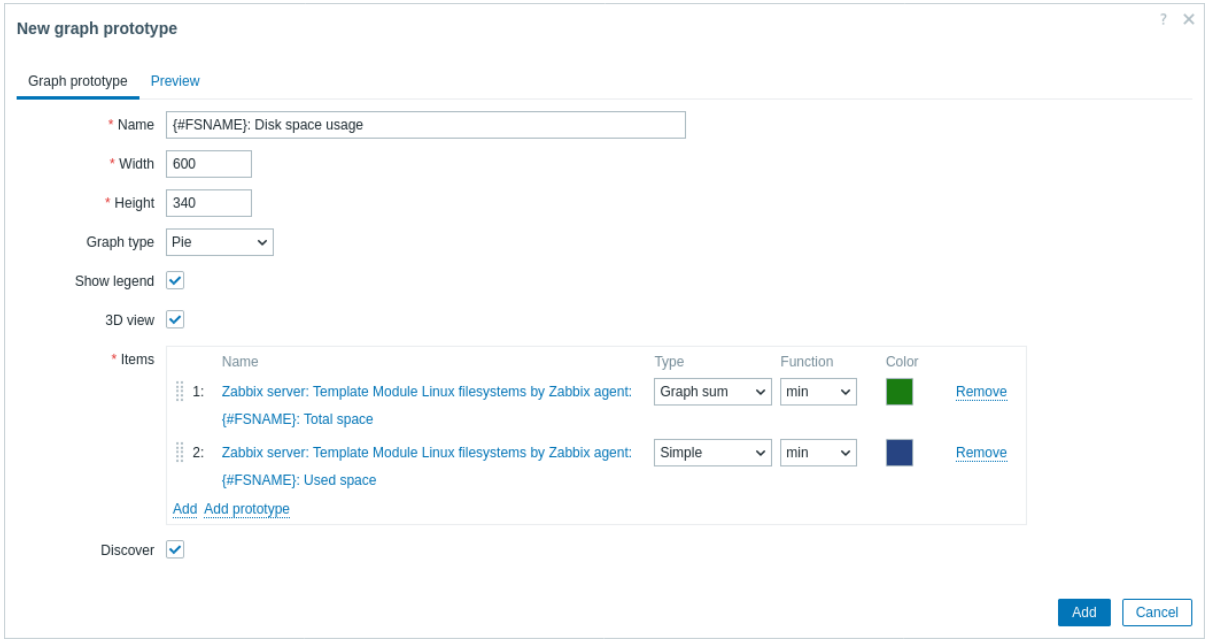
Quando i trigger reali vengono creati dai prototipi, può essere necessario avere flessibilità riguardo alla costante ('20' nel nostro esempio) usata per il confronto nell'espressione. Vedi come le macro utente con contesto possono essere utili per ottenere questa flessibilità.

Puoi definire dipendenze tra i prototipi di trigger. Per farlo, vai alla scheda *Dependencies*. Un prototipo di trigger può dipendere da un altro prototipo di trigger della stessa regola di low-level discovery (LLD) oppure da un trigger normale. Un prototipo di trigger non può dipendere da un trigger prototipo di una regola LLD diversa o da un trigger creato da un prototipo di trigger. Il prototipo di trigger dell'host non può dipendere da un trigger di un template.



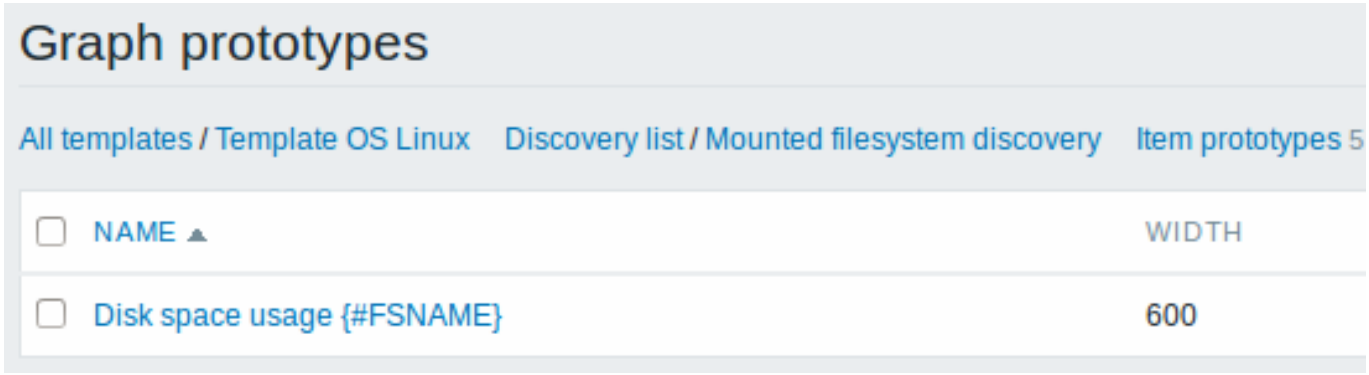
Prototipi di graph

Possiamo creare anche dei prototipi di graph:

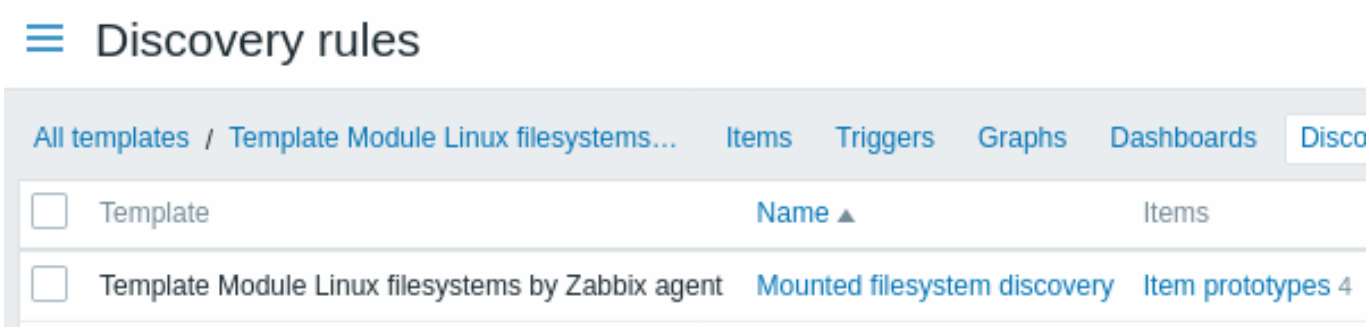


Attributi specifici dei prototipi di graph:

Parameter	Description
Discover	Se selezionato (impostazione predefinita), il graph verrà aggiunto a un'entità scoperta. Se non selezionato, il graph non verrà aggiunto a un'entità scoperta, a meno che questa impostazione non venga sovrascritta nella regola di discovery.



Infine, abbiamo creato una regola di discovery che appare come mostrato di seguito. Ha cinque prototipi di item, due prototipi di trigger e un prototipo di graph.



Prototipi host

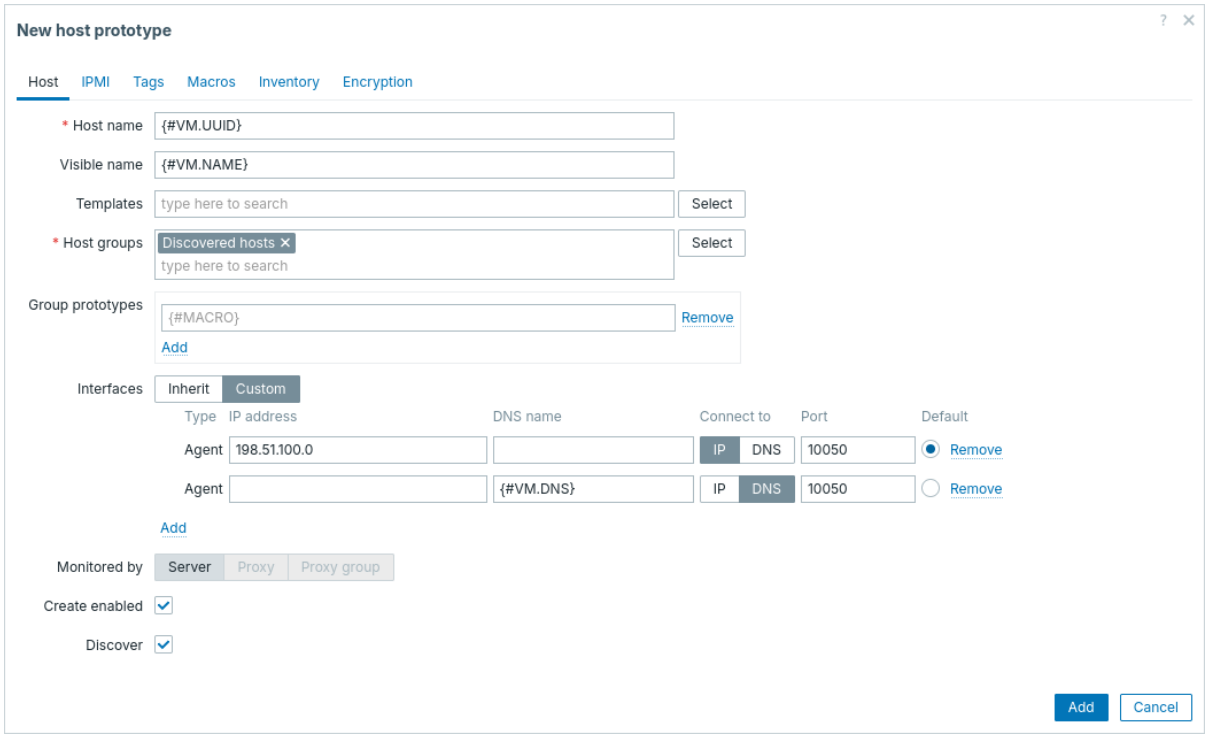
I prototipi di host sono modelli per la creazione di host tramite regole di low-level discovery. Prima di essere rilevati come host, questi prototipi non possono avere item e trigger, ad eccezione di quelli collegati dai template.

Configurazione

I host prototype vengono configurati nelle regole di discovery a basso livello.

Per creare un host prototype:

1. Vai a *Raccolta dati > Host*.
2. Fai clic su *Discovery* per l'host richiesto per passare all'elenco delle regole di discovery a basso livello configurate per quell'host.
3. Fai clic su *Host prototypes* per la regola di discovery richiesta.
4. Fai clic sul pulsante *Create host prototype* nell'angolo in alto a destra e completa il modulo di configurazione.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parameter	Description
Host name	Questo parametro deve contenere almeno una low-level discovery macro per garantire nomi host univoci per gli host creati.
Visible name	Sono supportate le low-level discovery macro.
Group prototypes	Consente di specificare i prototipi di gruppi host usando low-level discovery macro. Sulla base dei prototipi di gruppo specificati, verranno individuati, creati e collegati agli host creati i host groups; anche i gruppi individuati che sono già stati creati da altre regole di discovery a basso livello verranno collegati agli host creati. Tuttavia, i gruppi host individuati che corrispondono a gruppi host creati manualmente non verranno collegati agli host creati.
Interfaces	Imposta se gli host individuati ereditano l'IP dall'host a cui appartiene la regola di discovery (predefinito) oppure ottengono interfacce personalizzate. Sono supportate low-level discovery macro e user macro.
Monitored by	Questo campo viene sincronizzato automaticamente con il rispettivo campo nella configurazione dell'host.
Create enabled	Imposta lo stato degli host individuati; se non selezionato, gli host verranno creati come disabilitati.
Discover	Imposta se gli host verranno creati dal host prototype; se non selezionato, gli host non verranno creati dal host prototype (a meno che questa impostazione non venga sovrascritta nella regola di discovery a basso livello).

Note:
Sono supportate anche le low-level discovery macro per i valori dei tag e per i valori delle user macro del host prototype.
 Le *Value maps* non sono supportate per gli host prototype.

Per un esempio di configurazione di un host prototype, consulta *Virtual machine monitoring*.

Interfacce host

Per aggiungere interfacce personalizzate, imposta il selettore *Interfacce* da "Eredita" a "Personalizzato". Fai clic su Add e seleziona il tipo di interfaccia: Zabbix agent, SNMP, JMX, IPMI.

Note:
Se è selezionato *Personalizzato*, ma non è stata impostata alcuna interfaccia, gli host verranno creati senza interfacce.
Se è selezionato *Eredita* e il prototipo host appartiene a un template, tutti gli host rilevati erediteranno l'interfaccia host dall'host a cui il template è collegato.

Se vengono specificate più interfacce personalizzate, l'interfaccia primaria può essere impostata nella colonna *Predefinita*.

Per un esempio su come configurare un'interfaccia host personalizzata, vedi *Esempio di configurazione del monitoraggio VMware*.

Warning:
Un host verrà creato solo se un'interfaccia host contiene dati corretti.

Host scoperti

Nell'elenco degli host, gli host scoperti sono preceduti dal nome della regola di discovery che li ha creati.

Gli host scoperti ereditano la maggior parte dei parametri dai prototipi di host come *sola lettura*. Solo i seguenti parametri degli host scoperti possono essere configurati:

- *Template* - collegare nuovi template o scollegare quelli aggiunti manualmente. I template ereditati da un prototipo di host non possono essere scollegati.
- *Descrizione* - aggiungere o modificare la descrizione dell'host.
- *Stato* - abilitare o disabilitare l'host.
- *Tag* - aggiungere nuovi tag o rimuovere quelli aggiunti manualmente. I tag ereditati da un prototipo di host non possono essere rimossi. Si noti che i tag non possono avere duplicati (tag con lo stesso nome e valore). Se un nuovo tag viene aggiunto al prototipo di host e corrisponde a un tag aggiunto manualmente su un host scoperto, durante la discovery il tag aggiunto manualmente verrà sostituito con il tag ereditato.
- *Macro* - aggiungere nuove macro dell'host o rimuovere quelle aggiunte manualmente, così come quelle ereditate da un prototipo di host; modificare i valori delle macro e i tipi.

Note:
Gli host scoperti ereditano anche le macro utente dall'host in cui è configurata la regola di discovery. Queste macro ereditate possono essere rimosse e i loro valori e tipi possono essere modificati.

Gli host scoperti possono anche scoprire altri host. Ad esempio, se un prototipo di host (utilizzato per scoprire hypervisor) ha un template collegato che include una regola di low-level discovery con un proprio prototipo di host (utilizzato per scoprire macchine virtuali), Zabbix scoprirà sia gli hypervisor sia le loro macchine virtuali. Per scoprire anche i container su tali VM, è possibile creare regole di low-level discovery oppure collegare un nuovo template con prototipi di host alle VM scoperte, oppure preconfigurare il prototipo di host della VM collegando un template che includa a sua volta prototipi di host.

Gli host scoperti possono essere eliminati manualmente. Si noti, tuttavia, che verranno scoperti di nuovo se la discovery è abilitata per essi.

Gli host che non vengono più scoperti possono essere:

- disabilitati automaticamente (in base al valore di *Disable lost resources* della regola di discovery)
- eliminati automaticamente (in base al valore di *Delete lost resources* della regola di discovery).

Prototipi di discovery

Panoramica

I prototipi di discovery sono regole di **low-level discovery annidate** all'interno di una regola di discovery "parent", che consentono la discovery multilivello di oggetti con i propri item, trigger, ecc. Ad esempio, è possibile rilevare tutte le istanze di database su un server di database, quindi rilevare i tablespace per ogni istanza e infine rilevare le tabelle per ogni tablespace.

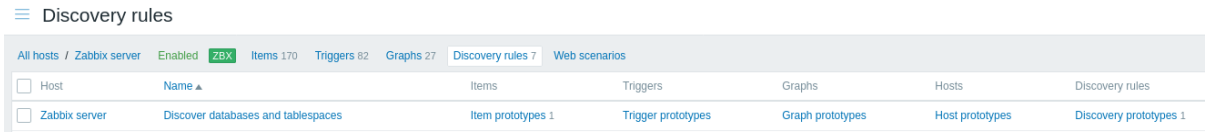
I prototipi di discovery hanno i propri prototipi di item, trigger, grafico, host e discovery. Un prototipo di discovery annidato utilizzerà lo stesso valore JSON della regola parent se si specifica un tipo *Nested*.

I livelli di annidamento per i prototipi di discovery sono illimitati.

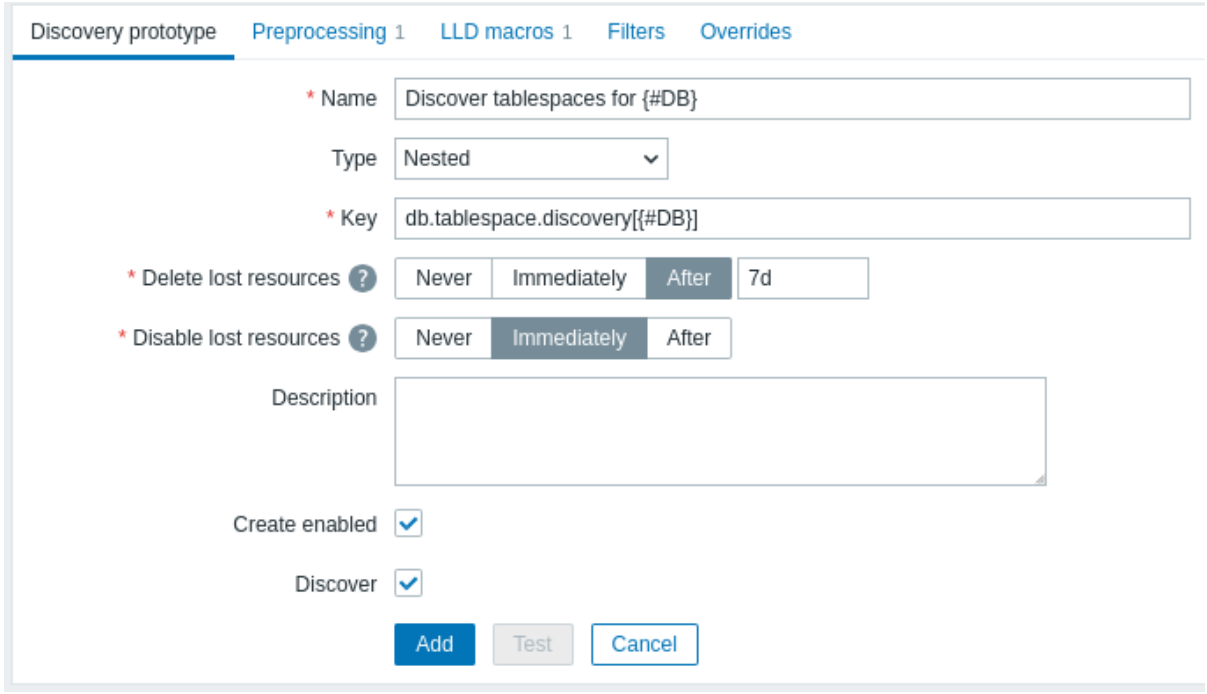
Configurazione

Per creare un prototipo di discovery:

- Fare clic su *Discovery prototypes* nella riga di una discovery rule esistente



- Fare clic su *Create discovery prototype*



I campi di configurazione di questo modulo sono condivisi con la normale low-level discovery.

Se si seleziona ”Nested” come *Type* nel modulo del discovery prototype aperto, allora le discovery rule (dal discovery prototype) vengono generate in base a un oggetto JSON ricavato dallo stesso valore JSON della discovery rule padre. Ad esempio, se il JSON originale è [<object A>, <object B>] e c’è un discovery rule prototype annidato, verranno generate due discovery rule basate rispettivamente sui dati di object A e object B.

In questo caso il discovery prototype viene attivato contemporaneamente alla regola padre. La regola annidata può quindi usare il preprocessing, per lavorare su un altro ”slice” degli stessi dati, già acquisiti dal padre.

Le macro LLD della discovery rule LLD padre sono disponibili per le discovery rule annidate.

Regole LLD annidate sugli host rilevati

Una regola di low-level discovery *Nested* può essere utilizzata in un template dell’host assegnato a un prototipo di host. Se una regola di discovery *Nested* esiste su un host rilevato, allora l’oggetto JSON utilizzato per rilevare l’host viene inviato anche a tutte le regole LLD di tipo annidato su questo host. Per maggiori dettagli, vedere l’esempio.

Le macro LLD della regola di discovery che ha creato l’host sono disponibili per le regole di discovery annidate.

Esempio

Illustriamo una possibile applicazione dei prototipi di individuazione, basata sulla ricezione del seguente esempio di JSON multilivello.

```
[
  {
    "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" }
    ]
  }
]
```

Caso 1

Rilevamento delle istanze di database su un server di database, quindi rilevamento dei tablespace per ciascuna istanza.

1. Si dispone di almeno un host relativo al rilevamento del server di database.
2. Creare una regola LLD per questo host denominata *Rileva database e tablespace*.
3. Passare alla scheda *Macro LLD* di questa regola e aggiungere la macro `{#DB}=$.database`.
4. Aggiungere un prototipo di item per questa regola denominato *Connessioni attive a {#DB}* (Tipo: Agent, Chiave: `db.connections[{#DB}]`).
5. Vengono rilevati gli item relativi a ciascun database:

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. Creare un prototipo di discovery per questa regola denominato *Rileva tablespace per {#DB}* (Tipo: Nested, Chiave: `db.tablespace.discovery[{#DB}]`).
7. Passare alla scheda *Preprocessing* di questo prototipo di discovery e aggiungere il passaggio `JSONPath=$.tablespaces`.
8. Passare alla scheda *Macro LLD* di questo prototipo di discovery e aggiungere la macro `{#TSNAME}=$.name`.
9. Creare un prototipo di item per questo prototipo di discovery denominato *Dimensione del tablespace {#TSNAME} per {#DB}* (Tipo: Agent, Chiave: `db.ts.size[{#DB}, {#TSNAME}]`).
10. Vengono rilevati gli item relativi a ciascun tablespace di ciascun database:

```
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.
```

con le chiavi `db.ts.size[db1,ts1]`, `db.ts.size[db1,ts2]`, ... `db.ts.size[db3,ts3]`.

Caso 2

Rilevamento delle istanze di database sul server di database rappresentandole come host rilevati, quindi rilevamento dei tablespace per ciascuna istanza.

1. È presente almeno un host (host radice) correlato al rilevamento del server di database.
2. Creare un template per rilevare i tablespace per ciascun database.
3. Creare un item in questo template denominato *Connessioni attive a {#DB}* (Tipo: Agent, Chiave: `db.connections[{#DB}]`).
4. Creare una regola LLD per questo template denominata *Rileva tablespace* (Tipo: Nested).
5. Passare alla scheda *Preprocessing* di questa regola e aggiungere il passaggio `JSONPath=$.tablespaces`.
6. Passare alla scheda *LLD Macros* di questa regola, aggiungere la macro `{#TSNAME}=$.name`.

7. Creare un prototipo di item per questa regola denominato *Dimensione del tablespace {#TSNAME} per {#DB}* (Tipo: Agent, Chiave: `db.ts.size[{#DB}, {#TSNAME}]`).
8. Tornare all'host radice e creare una regola LLD per questo host denominata *Rileva database e tablespace*.
9. Passare alla scheda *LLD Macros* di questa regola, aggiungere la macro `{#DB}=$.database`.
10. Aggiungere un prototipo di host per questa regola denominato *Host per il database {#DB}*.
11. Passare alla scheda *Macros* di questo prototipo di host, aggiungere la macro `{$DB}={#DB}` (per il nome e la chiave dell'item del passaggio 3).
12. Collegare il template del passaggio 2 a questo prototipo di host.
13. Gli host rilevati contengono gli item rilevati relativi a ciascun database e ai relativi tablespace:

Host	Item
Host per il database db1	Connessioni attive a db1
	Dimensione del tablespace ts1 per db1
	Dimensione del tablespace ts2 per db1
	Dimensione del tablespace ts3 per db1
Host per il database db2	Connessioni attive a db2
	Dimensione del tablespace ts1 per db2
	Dimensione del tablespace ts2 per db2
	Dimensione del tablespace ts3 per db2
Host per il database db3	Connessioni attive a db3
	Dimensione del tablespace ts1 per db3
	Dimensione del tablespace ts2 per db3
	Dimensione del tablespace ts3 per db3

Note sulla discovery a basso livello

Utilizzo delle macro LLD nei contesti delle macro utente

Le macro LLD possono essere utilizzate all'interno del contesto delle macro utente, ad esempio, nei prototipi di trigger.

Più regole LLD per lo stesso item

È possibile definire diverse regole di low-level discovery con lo stesso item di discovery.

A tale scopo, è necessario definire il parametro Alias dell'agent parametro, che consente di utilizzare chiavi dell'item di discovery modificate in diverse regole di discovery, ad esempio `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]`, ecc.

Limiti dei dati per i valori restituiti

Non esiste alcun limite per i dati JSON delle regole di low-level discovery se vengono ricevuti direttamente da Zabbix server. Questo perché i valori restituiti vengono elaborati senza essere memorizzati in un database.

Non esiste inoltre alcun limite per le regole di low-level discovery personalizzate. Tuttavia, se i dati di una regola di low-level discovery personalizzata vengono recuperati utilizzando un parametro utente, si applica il limite del valore restituito del parametro utente.

Se i dati devono passare attraverso Zabbix proxy, quest'ultimo deve memorizzare questi dati nel database. In tal caso, si applicano i limiti del database.

Esempi di regole di discovery

Questa sezione fornisce esempi pratici di configurazione delle regole di discovery per diversi tipi di entità, come filesystem, interfacce di rete, servizi, sensori, oggetti SNMP, applicazioni Java e database.

Ogni esempio illustra come può essere strutturata una regola di discovery, incluso il key dell'item o la query utilizzata, l'output JSON con le macro LLD e il modo in cui tali risultati vengono usati per creare prototipi di item, trigger e grafici.

- Discovery di dispositivi a blocchi
- Discovery di CPU e core CPU
- Discovery delle interfacce dell'host in Zabbix
- Discovery dei sensori IPMI
- Discovery di oggetti JMX

- Discovery dei filesystem montati
- Discovery delle interfacce di rete
- Discovery di OID SNMP
- Discovery di OID SNMP (legacy)
- Discovery dei servizi systemd
- Discovery delle istanze dei contatori di prestazioni di Windows
- Discovery dei servizi Windows
- Discovery tramite query WMI
- Discovery tramite query SQL ODBC
- Discovery tramite dati Prometheus

Regole LLD personalizzate

Panoramica

È anche possibile creare una regola LLD completamente personalizzata, in grado di individuare qualsiasi tipo di entità, ad esempio i database su un server di database.

Per farlo, è necessario creare un item personalizzato che restituisca una stringa JSON, specificando gli oggetti trovati e, facoltativamente, alcune loro proprietà. Il numero di macro per entità non è limitato: mentre le regole di discovery integrate restituiscono una o due macro (ad esempio, due per la discovery del filesystem), è possibile restituirne di più.

Esempio

Il formato richiesto della stringa JSON è illustrato al meglio con un esempio. Supponiamo che tu stia eseguendo un vecchio agent Zabbix 1.8 (uno che non supporta la chiave `vfs.fs.discovery`), ma che tu abbia comunque bisogno di individuare i file system. Ecco un semplice script Perl per Linux che rileva i file system montati e li restituisce in formato JSON, includendo sia il nome sia il tipo di file system. Un modo per utilizzarlo sarebbe come UserParameter con chiave `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:
I simboli consentiti per i nomi delle macro LLD sono **0-9** , **A-Z** , _ , **.**. Le lettere minuscole non sono supportate nei nomi.

Di seguito è mostrato un esempio del relativo output (riformattato per maggiore chiarezza). Il JSON per i controlli di discovery personalizzati deve seguire lo stesso formato.

```
[
    { "{#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"  }
]
```

Nell'esempio precedente è necessario che le chiavi corrispondano ai nomi delle macro LLD utilizzate nei prototipi; l'alternativa consiste nell'estrarre i valori delle macro LLD usando JSONPath {#FSNAME} → $.fsname e {#FSTYPE} → $.fstype, rendendo così possibile uno script di questo tipo:

```
####!/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";
```

Di seguito è mostrato un esempio del relativo output (riformattato per maggiore chiarezza).

```
[
    { "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"  }
]
```

Quindi, nel campo *Filter* della regola di discovery, puoi specificare {#FSTYPE} come macro e `rootfs|ext3` come espressione regolare.

> **Note:**
> Non è necessario usare nomi di macro come FSNAME/FSTYPE con regole LLD personalizzate; puoi usare liberamente i nomi che preferisci. Nel caso in cui venga usato JSONPath, allora la riga LLD sarà un elemento dell'array che può essere un oggetto, ma anche un altro array o un valore.

Tieni presente che, se si utilizza un parametro utente, il valore restituito è limitato a 16 MB. Per maggiori dettagli, vedi limiti dei dati per i valori restituiti da LLD.

Monitoraggio delle macchine virtuali

Panoramica Zabbix può utilizzare le regole di low-level discovery per individuare automaticamente hypervisor VMware e macchine virtuali, e creare host da monitorare in base a prototipi di host predefiniti.

Zabbix include anche template pronti all'uso per il monitoraggio di VMware vCenter o degli hypervisor ESXi.

La versione minima richiesta di VMware vCenter o vSphere è la 5.1.

Raccolta dati Il monitoraggio delle macchine virtuali consiste in due passaggi:

1. I processi Zabbix *vmware collector* elaborano i dati delle macchine virtuali: i processi ottengono le informazioni necessarie dai servizi web VMware tramite il protocollo SOAP, le pre-elaborano e le memorizzano nella memoria condivisa del server Zabbix.
2. I processi Zabbix *poller* recuperano i dati utilizzando il simple check di Zabbix VMware monitoring item keys.

Zabbix suddivide i dati raccolti in dati di configurazione VMware e dati dei contatori di prestazione VMware. Entrambi i tipi di dati vengono raccolti in modo indipendente dai processi *vmware collector*.

Le seguenti statistiche sono disponibili in base alle informazioni dei contatori di prestazione VMware:

- Datastore
- Dispositivo disco
- CPU
- Alimentazione
- Interfaccia di rete
- Item di contatori di prestazione personalizzati

Per l'elenco completo degli item che ottengono dati dai contatori di prestazione VMware, vedere *VMware monitoring item keys*.

Si noti che la frequenza di recupero degli eventi VMware dipende dall'intervallo di polling di vmware.eventlog, ma non può essere inferiore a 5 secondi.

Configurazione Se il server Zabbix viene compilato dai sorgenti, deve essere compilato con le opzioni di configurazione `--with-libcurl --with-libxml2` per abilitare il monitoraggio delle macchine virtuali. I pacchetti Zabbix sono già compilati con queste opzioni abilitate.

I seguenti parametri del file di configurazione del server Zabbix possono essere modificati per il monitoraggio delle macchine virtuali:

- `StartVMwareCollectors`

Note:
Si consiglia di abilitare un numero di collector superiore al numero di servizi VMware monitorati; altrimenti, il recupero delle statistiche dei contatori di prestazioni VMware potrebbe essere ritardato dal recupero dei dati di configurazione VMware (operazione che richiede tempo nelle installazioni di grandi dimensioni).

 In generale, il valore di `StartVMwareCollectors` non dovrebbe scendere sotto 2 e non dovrebbe superare il doppio del numero di servizi VMware monitorati: Numero di servizi < `StartVMwareCollectors` < (Numero di servizi * 2). Ad esempio, quando si monitora un servizio VMware, impostare `StartVMwareCollectors` su 2; quando si monitorano tre servizi, impostare `StartVMwareCollectors` su 5.

 Si noti che il numero richiesto di collector dipende anche dall'ampiezza dell'ambiente VMware e dai parametri di configurazione `VMwareFrequency` e `VMwarePerfFrequency`.

- `VMwareCacheSize`
- `VMwareFrequency`
- `VMwarePerfFrequency`
- `VMwareTimeout`

Attention:
Per supportare le metriche di capacità del datastore, assicurarsi che il valore della chiave VMware `vpxd.stats.maxQueryMetrics` sia impostato ad almeno 64. Per ulteriori informazioni, vedere l'articolo della *VMware Knowledge Base*.

Discovery

Zabbix può utilizzare regole di discovery a basso livello (ad esempio, `vmware.hv.discovery[{$VMWARE.URL}]`) per individuare automaticamente gli hypervisor VMware e le macchine virtuali. Inoltre, Zabbix può utilizzare i prototipi di host per generare automaticamente host reali per le entità individuate. Per ulteriori informazioni, vedere *Host prototypes*.

Esempi di configurazione

Per un esempio di base su come configurare Zabbix per il monitoraggio di VMware usando il template *VMware FQDN*, vedere *Monitor VMware with Zabbix*.

Per un esempio più dettagliato su come creare un host, una regola di discovery a basso livello e un prototipo di host per il monitoraggio di VMware, vedere *Setup example*.

Registrazione estesa I dati raccolti dai processi *vmware collector* possono essere registrati per un debug dettagliato utilizzando il livello di debug 5.
Il livello di debug può essere configurato nei file di configurazione del server e del proxy, oppure utilizzando l'opzione di controllo runtime `-R log_level_increase="vmware collector,N"`, dove `N` è il numero del processo.

Ad esempio, per aumentare il livello di debug da 4 a 5 per tutti i processi *vmware collector*, eseguire il comando seguente:

```
zabbix_server -R log_level_increase="vmware collector"
```

Per aumentare il livello di debug da 4 a 5 per il secondo processo *vmware collector*, eseguire il comando seguente:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Quando la registrazione estesa dei dati del VMware collector non è più necessaria, si consiglia di ridurre il livello di debug al valore predefinito (3) eseguendo il comando `-R log_level_decrease`.

Risoluzione dei problemi

- In caso di metriche non disponibili, assicurarsi che non siano state rese non disponibili o disattivate per impostazione predefinita nelle versioni recenti di VMware vSphere, oppure che non siano stati impostati limiti sulle query del database delle metriche di prestazioni. Per ulteriori informazioni, vedere ZBX-12094.
- Se `config.vpxd.stats.maxQueryMetrics` non è valido o supera il numero massimo di caratteri consentito, aggiungere un parametro `config.vpxd.stats.maxQueryMetrics` alle impostazioni di vCenter Server. Il valore di questo parametro deve essere uguale al valore di `maxQuerysize` nel file `web.xml` di VMware. Per ulteriori informazioni, vedere l'articolo *VMware Knowledge Base*.
- Se sospettate che la vostra installazione di Zabbix stia utilizzando troppa memoria, vedere Profiling excessive memory usage with tcmalloc.

Chiavi degli item per il monitoraggio VMware

Panoramica Questa pagina fornisce dettagli sui controlli semplici che possono essere utilizzati per monitorare gli ambienti VMware.
Le metriche sono raggruppate in base all'obiettivo di monitoraggio.

Chiavi item supportate Le chiavi item sono elencate senza parametri e informazioni aggiuntive. Fare clic sulla chiave item per visualizzare i dettagli completi.

Item key	Description	Item group
vmware.eventlog	Il registro eventi di VMware.	General service
vmware.fullname	Il nome completo del servizio VMware.	
vmware.version	La versione del servizio VMware.	
vmware.cl.perfcounter	Le metriche del contatore delle prestazioni del cluster VMware.	Cluster
vmware.cluster.alarms.get	I dati degli allarmi del cluster VMware.	
vmware.cluster.discovery	La discovery dei cluster VMware.	
vmware.cluster.property	La proprietà del cluster VMware.	
vmware.cluster.status	Lo stato del cluster VMware.	
vmware.cluster.tags.get	L'array dei tag del cluster VMware.	
vmware.datastore.alarms.get	I dati degli allarmi del datastore VMware.	Datastore
vmware.datastore.discovery	La discovery dei datastore VMware.	
vmware.datastore.hv.list	L'elenco degli hypervisor del datastore.	
vmware.datastore.perfcounter	Il valore del contatore delle prestazioni del datastore VMware.	
vmware.datastore.property	La proprietà del datastore VMware.	
vmware.datastore.read	La quantità di tempo per un'operazione di lettura dal datastore.	
vmware.datastore.size	Lo spazio del datastore VMware in byte o in percentuale sul totale.	
vmware.datastore.tags.get	L'array dei tag del datastore VMware.	
vmware.datastore.write	La quantità di tempo per un'operazione di scrittura sul datastore.	

Item key	Description	Item group
vmware.dc.alarms.get	I dati degli allarmi del datacenter VMware.	Datacenter
vmware.dc.discovery	La discovery dei datacenter VMware.	
vmware.dc.tags.get	L'array dei tag del datacenter VMware.	
vmware.dvswitch.discovery	La discovery degli vSphere Distributed Switch di VMware.	vSphere Distributed Switch
vmware.dvswitch.fetchports.get	I dati delle porte dello vSphere Distributed Switch VMware.	
vmware.hv.alarms.get	I dati degli allarmi dell'hypervisor VMware.	Hypervisor
vmware.hv.cluster.name	Il nome del cluster dell'hypervisor VMware.	
vmware.hv.connectionstate	Lo stato di connessione dell'hypervisor VMware.	
vmware.hv.cpu.usage	L'utilizzo del processore dell'hypervisor VMware (Hz).	
vmware.hv.cpu.usage.perf	L'utilizzo del processore dell'hypervisor VMware come percentuale durante l'intervallo.	
vmware.hv.cpu.utilization	L'utilizzo del processore dell'hypervisor VMware come percentuale durante l'intervallo, dipende dalla gestione dell'alimentazione o da HT.	
vmware.hv.datacenter.name	Il nome del datacenter dell'hypervisor VMware.	
vmware.hv.datastore.discovery	La discovery dei datastore dell'hypervisor VMware.	
vmware.hv.datastore.list	L'elenco dei datastore dell'hypervisor VMware.	
vmware.hv.datastore.multipath	Il numero di percorsi datastore disponibili.	
vmware.hv.datastore.read	La quantità media di tempo per un'operazione di lettura dal datastore.	
vmware.hv.datastore.size	Lo spazio del datastore VMware in byte o in percentuale sul totale.	
vmware.hv.datastore.write	La quantità media di tempo per un'operazione di scrittura sul datastore.	
vmware.hv.discovery	La discovery degli hypervisor VMware.	
vmware.hv.diskinfo.get	I dati del disco dell'hypervisor VMware.	
vmware.hv.fullname	Il nome completo del prodotto, incluse le informazioni sulla versione.	
vmware.hv.hw.cpu.freq	La frequenza del processore dell'hypervisor VMware.	
vmware.hv.hw.cpu.model	Il modello del processore dell'hypervisor VMware.	
vmware.hv.hw.cpu.num	Il numero di core del processore sull'hypervisor VMware.	
vmware.hv.hw.cpu.threads	Il numero di thread del processore sull'hypervisor VMware.	
vmware.hv.hw.memory	La dimensione totale della memoria dell'hypervisor VMware.	
vmware.hv.hw.model	Il modello dell'hypervisor VMware.	
vmware.hv.hw.sensors.get	Il valore dei sensori hardware dell'hypervisor VMware.	
vmware.hv.hw.serialnumber	Il numero di serie dell'hypervisor VMware.	
vmware.hv.hw.uuid	L'UUID BIOS dell'hypervisor VMware.	
vmware.hv.hw.vendor	Il nome del fornitore dell'hypervisor VMware.	
vmware.hv.maintenance	Lo stato di manutenzione dell'hypervisor VMware.	
vmware.hv.memory.size.ballooned	La dimensione della memoria ballooned dell'hypervisor VMware.	
vmware.hv.memory.used	La dimensione della memoria utilizzata dell'hypervisor VMware.	
vmware.hv.net.if.discovery	La discovery delle interfacce di rete dell'hypervisor VMware.	
vmware.hv.network.in	Le statistiche di input di rete dell'hypervisor VMware.	
vmware.hv.network.linkspeed	La velocità dell'interfaccia di rete dell'hypervisor VMware.	
vmware.hv.network.out	Le statistiche di output di rete dell'hypervisor VMware.	
vmware.hv.perfcounter	Il valore del contatore delle prestazioni dell'hypervisor VMware.	
vmware.hv.property	La proprietà dell'hypervisor VMware.	
vmware.hv.power	L'utilizzo di energia dell'hypervisor VMware.	
vmware.hv.sensor.health.state	Il sensore di riepilogo dello stato di salute dell'hypervisor VMware.	
vmware.hv.sensors.get	I sensori di stato del vendor HW dell'hypervisor VMware.	
vmware.hv.status	Lo stato dell'hypervisor VMware.	
vmware.hv.tags.get	L'array dei tag dell'hypervisor VMware.	
vmware.hv.uptime	L'uptime dell'hypervisor VMware.	
vmware.hv.version	La versione dell'hypervisor VMware.	
vmware.hv.vm.num	Il numero di macchine virtuali sull'hypervisor VMware.	
vmware.rp.cpu.usage	L'utilizzo della CPU in hertz durante l'intervallo nel Resource pool VMware.	Resource pool
vmware.rp.memory	Le metriche di memoria del resource pool VMware.	
vmware.alarms.get	I dati degli allarmi del virtual center VMware.	Virtual center
vmware.vm.alarms.get	I dati degli allarmi della macchina virtuale VMware.	Virtual machine
vmware.vm.attribute	Il valore dell'attributo personalizzato della macchina virtuale VMware.	
vmware.vm.cluster.name	Il nome della macchina virtuale VMware.	

Item key	Description	Item group
vmware.vm.consolidationneeded	Indica se la macchina virtuale VMware richiede consolidamento.	
vmware.vm.cpu.latency	La percentuale di tempo in cui la macchina virtuale non è in grado di eseguire perché è in contesa per l'accesso alla CPU fisica.	
vmware.vm.cpu.num	Il numero di processori sulla macchina virtuale VMware.	
vmware.vm.cpu.readiness	La percentuale di tempo in cui la macchina virtuale era pronta, ma non ha potuto essere pianificata per l'esecuzione sulla CPU fisica.	
vmware.vm.cpu.ready	Il tempo in cui la macchina virtuale era pronta, ma non ha potuto essere pianificata per l'esecuzione sulla CPU fisica.	
vmware.vm.cpu.swapwait	La percentuale di tempo CPU trascorsa in attesa dello swap-in.	
vmware.vm.cpu.usage	L'utilizzo del processore della macchina virtuale VMware (Hz).	
vmware.vm.cpu.usage.perf	L'utilizzo del processore della macchina virtuale VMware come percentuale durante l'intervallo.	
vmware.vm.datacenter.name	Il nome del datacenter della macchina virtuale VMware.	
vmware.vm.discovery	La discovery delle macchine virtuali VMware.	
vmware.vm.guest.memory.swapped	La quantità di memoria fisica guest che viene spostata nell'area di swap.	
vmware.vm.guest.osuptime	Il tempo totale trascorso dall'ultimo avvio del sistema operativo.	
vmware.vm.hv.maintenance	Lo stato di manutenzione dell'hypervisor della macchina virtuale VMware.	
vmware.vm.hv.name	Il nome dell'hypervisor della macchina virtuale VMware.	
vmware.vm.memory.size	La dimensione totale della memoria della macchina virtuale VMware.	
vmware.vm.memory.size.ballooned	La dimensione della memoria ballooned della macchina virtuale VMware.	
vmware.vm.memory.size.compressed	La dimensione della memoria compressa della macchina virtuale VMware.	
vmware.vm.memory.size.consumed	La quantità di memoria fisica host consumata per il backup delle pagine di memoria fisica guest.	
vmware.vm.memory.size.private	La dimensione della memoria privata della macchina virtuale VMware.	
vmware.vm.memory.size.shared	La dimensione della memoria condivisa della macchina virtuale VMware.	
vmware.vm.memory.size.swapped	La dimensione della memoria swap della macchina virtuale VMware.	
vmware.vm.memory.size.usage.guest	L'utilizzo della memoria guest della macchina virtuale VMware.	
vmware.vm.memory.size.usage.host	L'utilizzo della memoria host della macchina virtuale VMware.	
vmware.vm.memory.usage	La percentuale di memoria fisica host che è stata consumata.	
vmware.vm.net.if.discovery	La discovery delle interfacce di rete della macchina virtuale VMware.	
vmware.vm.net.if.in	Le statistiche di input dell'interfaccia di rete della macchina virtuale VMware.	
vmware.vm.net.if.out	Le statistiche di output dell'interfaccia di rete della macchina virtuale VMware.	
vmware.vm.net.if.usage	L'utilizzo della rete della macchina virtuale VMware durante l'intervallo.	
vmware.vm.perfcounter	Il valore del contatore delle prestazioni della macchina virtuale VMware.	
vmware.vm.powerstate	Lo stato di alimentazione della macchina virtuale VMware.	
vmware.vm.property	La proprietà della macchina virtuale VMware.	
vmware.vm.snapshot.get	Lo stato dello snapshot della macchina virtuale VMware.	
vmware.vm.state	Lo stato della macchina virtuale VMware.	
vmware.vm.storage.committed	Lo spazio di archiviazione impegnato della macchina virtuale VMware.	
vmware.vm.storage.readoio	Il numero medio di richieste di lettura in sospeso sul disco virtuale durante l'intervallo di raccolta.	
vmware.vm.storage.totalreadlatency	Il tempo medio impiegato da una lettura dal disco virtuale.	
vmware.vm.storage.totalwritelatency	Il tempo medio impiegato da una scrittura sul disco virtuale.	
vmware.vm.storage.uncommitted	Lo spazio di archiviazione non impegnato della macchina virtuale VMware.	
vmware.vm.storage.unshared	Lo spazio di archiviazione non condiviso della macchina virtuale VMware.	
vmware.vm.storage.writeoio	Il numero medio di richieste di scrittura in sospeso sul disco virtuale durante l'intervallo di raccolta.	
vmware.vm.tags.get	L'array dei tag della macchina virtuale VMware.	
vmware.vm.tools	Lo stato o la versione degli strumenti guest della macchina virtuale VMware.	
vmware.vm.uptime	L'uptime della macchina virtuale VMware.	
vmware.vm.vfs.dev.discovery	La discovery dei dispositivi disco della macchina virtuale VMware.	
vmware.vm.vfs.dev.read	Le statistiche di lettura del dispositivo disco della macchina virtuale VMware.	
vmware.vm.vfs.dev.write	Le statistiche di scrittura del dispositivo disco della macchina virtuale VMware.	
vmware.vm.vfs.fs.discovery	La discovery dei file system della macchina virtuale VMware.	
vmware.vm.vfs.fs.size	Le statistiche del file system della macchina virtuale VMware.	

Dettagli della chiave item I parametri senza parentesi angolari sono obbligatori. I parametri contrassegnati da parentesi angolari **< >** sono facoltativi.

vmware.eventlog[url,<mode>,<severity>]

 Il log degli eventi VMware.
 Valore restituito: *Log*.

Parametri:

- **url** - l'URL del servizio VMware;
- **mode** - *all* (predefinito) oppure *skip* - salta l'elaborazione dei dati meno recenti;
- **severity** - filtro per gravità: *error*, *warning*, *info* oppure *user*. Questo parametro deve essere racchiuso tra virgolette se viene specificata più di una gravità in un elenco delimitato da virgole (ad es.`"error,warning,info,user"`). Disabilitato per impostazione predefinita.

Commenti:

- Deve esserci una sola chiave di item `vmware.eventlog` per URL;
- Vedere anche l'esempio di filtraggio dei record del log eventi VMware.

vmware.fullnameurl

 Il nome completo del servizio VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.versionurl

 La versione del servizio VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.cl.perfcounter[url,id,path,<instance>]

 Le metriche del contatore delle prestazioni del cluster VMware.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **id** - l'ID del cluster VMware. `id` può essere ottenuto da `vmware.cluster.discovery[]` come {#CLUSTER.ID}.
- **path** - il percorso del contatore delle prestazioni[1];
- **instance** - l'istanza del contatore delle prestazioni.

vmware.cluster.alarms.get[url,id]

 I dati degli allarmi del cluster VMware.
 Valore restituito: JSON object.

Parametri:

- **url** - l'URL del servizio VMware;
- **id** - l'ID del cluster VMware.

vmware.cluster.discoveryurl

 La scoperta dei cluster VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.cluster.property[url,id,prop]

 La proprietà del cluster VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **id** - l'ID del cluster VMware;
- **prop** - il percorso della proprietà che corrisponde al nome di una proprietà dell'oggetto VM come definita nel VMware SDK.

Esempi:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 Lo stato del cluster VMware.
 Valore restituito: *0* - grigio; *1* - verde; *2* - giallo; *3* - rosso.

Parametri:

- **url** - l'URL del servizio VMware;

- **name** - il nome del cluster VMware.

vmware.cluster.tags.get[url,id]

 L'array dei tag del cluster VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **id** - l'ID del cluster VMware.

Questo item funziona con vSphere 6.5 e versioni successive.

vmware.datastore.alarms.get[url,uuid]

 I dati degli allarmi del datastore VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale del datastore VMware.

vmware.datastore.discovery[url,<filter_uuid>]

 La discovery dei datastore VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **filter_uuid** - un identificatore univoco di un singolo HV o VM (predefinito: vuoto). Se specificato, verranno rilevati solo i datastore collegati.

vmware.datastore.hv.list[url,datastore]

 L'elenco degli hypervisor del datastore.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **datastore** - l'UUID o il nome del datastore.

Esempio di output:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 Il valore del contatore di prestazioni del datastore VMware.
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale del datastore VMware;
- **path** - il percorso del contatore di prestazioni[1];
- **instance** - l'istanza del contatore di prestazioni. Utilizzare un'istanza vuota per i valori aggregati (predefinito). `instance` può essere ottenuto da `vmware.datastore.discovery[]` come parte dell'array {#DATASTORE.EXTENT}.

vmware.datastore.property[url,uuid,prop]

 La proprietà del datastore VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale del datastore VMware;
- **prop** - il percorso della proprietà.

vmware.datastore.read[url,datastore,<mode>]

 Il tempo impiegato per un'operazione di lettura dal datastore (millisecondi).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **datastore** - l'UUID o il nome del datastore;
- **mode** - *latency* (valore medio, predefinito) oppure *maxlatency* (valore massimo).

vmware.datastore.size[url,datastore,<mode>]

 Lo spazio del datastore VMware in byte o in percentuale del totale.
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.

Parametri:

- **url** - l'URL del servizio VMware;
- **datastore** - l'UUID o il nome del datastore;
- **mode** - valori possibili: *total* (predefinito), *free*, *pfree* (percentuale libera), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 L'array dei tag del datastore VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale del datastore VMware.

Questo item funziona con vSphere 6.5 e versioni successive.

vmware.datastore.write[url,datastore,<mode>]

 Il tempo impiegato per un'operazione di scrittura nel datastore (millisecondi).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **datastore** - l'UUID o il nome del datastore;
- **mode** - *latency* (valore medio, predefinito) oppure *maxlatency* (valore massimo).

vmware.dc.alarms.get[url,id]

 I dati degli allarmi del datacenter VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **id** - l'ID del datacenter VMware.

vmware.dc.discoveryurl

 La discovery dei datacenter VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.dc.tags.get[url,id]

 L'array dei tag del datacenter VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **id** - l'ID del datacenter VMware.

Questo item funziona con vSphere 6.5 e versioni successive.

vmware.dvswitch.discoveryurl

 Il rilevamento degli switch distribuiti VMware vSphere.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 I dati delle porte di VMware vSphere Distributed Switch.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale del DVSwitch VMware.
- **filter** - una singola stringa con criteri separati da virgole per selezionare le porte;
- **mode** - *state* (tutto l'XML senza i nodi XML "config", predefinito) oppure *full*.

Il parametro **filter** supporta i criteri disponibili nell'oggetto dati VMware DistributedVirtualSwitchPortCriteria.

Esempio:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 I dati degli allarmi dell'hypervisor VMware.
 Valore restituito: JSON object.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.cluster.name[url,uuid]

 Il nome del cluster dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.connectionstate[url,uuid]

 Lo stato della connessione dell'hypervisor VMware.
 Valore restituito: *String*: *connected*, *disconnected* oppure *notResponding*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.cpu.usage[url,uuid]

 L'utilizzo del processore dell'hypervisor VMware (Hz).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.cpu.usage.perf[url,uuid]

 L'utilizzo del processore dell'hypervisor VMware come percentuale durante l'intervallo.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.cpu.utilization[url,uuid]

 L'utilizzo del processore dell'hypervisor VMware come percentuale durante l'intervallo; dipende dalla gestione dell'alimentazione o da HT.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.datacenter.name[url,uuid]

 Il nome del datacenter dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.datastore.discovery[url,uuid]

 La discovery dei datastore dell'hypervisor VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.datastore.list[url,uuid]

 L'elenco dei datastore dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Esempio di output:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 Il numero di percorsi datastore disponibili.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **datastore** - l'UUID o il nome del datastore;
- **partitionid** - l'ID interno del dispositivo fisico da `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 Il tempo medio per un'operazione di lettura dal datastore (millisecondi).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **datastore** - l'UUID o il nome del datastore;
- **mode** - *latency* (predefinito).

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 Lo spazio del datastore VMware in byte o in percentuale del totale.
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **datastore** - l'UUID o il nome del datastore;
- **mode** - valori possibili: *total* (predefinito), *free*, *pfree* (percentuale libera), *uncommitted*.

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 Il tempo medio per un'operazione di scrittura nel datastore (millisecondi).
 Valore restituito: *Intero* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **datastore** - l'UUID o il nome del datastore;
- **mode** - *latency* (predefinito).

vmware.hv.discoveryurl

 Il rilevamento degli hypervisor VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.hv.diskinfo.get[url,uuid]

 I dati del disco dell'hypervisor VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.fullname[url,uuid]

 Il nome completo del prodotto, incluse le informazioni sulla versione.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.cpu.freq[url,uuid]

 La frequenza del processore dell'hypervisor VMware (Hz).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.cpu.model[url,uuid]

 Il modello del processore dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.cpu.num[url,uuid]

 Il numero di core del processore sull'hypervisor VMware.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.cpu.threads[url,uuid]

 Il numero di thread del processore sull'hypervisor VMware.
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.memory[url,uuid]

 La dimensione totale della memoria dell'hypervisor VMware (byte).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.model[url,uuid]

 Il modello dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.sensors.get[url,uuid]

 Il valore dei sensori hardware dell'hypervisor VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.hw.serialnumber[url,uuid]

 Il numero di serie dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Questo item funziona con vSphere API 6.7 e versioni successive.

vmware.hv.hw.uuid[url,uuid]

 Il BIOS UUID dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Questo item funziona con vSphere API 6.7 e versioni successive.

vmware.hv.hw.vendor[url,uuid]

 Il nome del fornitore dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Questo item funziona con vSphere API 6.7 e versioni successive.

vmware.hv.maintenance[url,uuid]

 Lo stato di manutenzione dell'hypervisor VMware.
 Valore restituito: *0* - non in manutenzione; *1* - in manutenzione.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.memory.size.ballooned[url,uuid]

 La dimensione della memoria ballooned dell'hypervisor VMware (byte).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.memory.used[url,uuid]

 La dimensione della memoria utilizzata dall'hypervisor VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.net.if.discovery[url,uuid]

 La discovery delle interfacce di rete dell'hypervisor VMware.
 Valore restituito: *oggetto JSON*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.network.in[url,uuid,<mode>]

 Le statistiche di input della rete dell'hypervisor VMware (byte al secondo).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **mode** - *bps* (predefinito), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 La velocità dell'interfaccia di rete dell'hypervisor VMware.
 Valore restituito: *Integer*. Restituisce *0* se l'interfaccia di rete è inattiva, altrimenti il valore della velocità dell'interfaccia.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **ifname** - il nome dell'interfaccia.

vmware.hv.network.out[url,uuid,<mode>]

 Le statistiche di output della rete dell'hypervisor VMware (byte al secondo).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **mode** - *bps* (predefinito), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,path,<instance>]

 Il valore del contatore di prestazioni dell'hypervisor VMware.
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **path** - il percorso del contatore di prestazioni [1];
- **instance** - l'istanza del contatore di prestazioni. Utilizzare un'istanza vuota per i valori aggregati (predefinito).

vmware.hv.property[url,uuid,prop]

 La proprietà dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **prop** - il percorso della proprietà.

vmware.hv.power[url,uuid,<max>]

 Il consumo energetico dell'hypervisor VMware (W).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware;
- **max** - il consumo energetico massimo consentito.

vmware.hv.sensor.health.state[url,uuid]

 Il sensore di riepilogo dello stato di integrità dell'hypervisor VMware.
 Valore restituito: *Integer*: *0* - grigio; *1* - verde; *2* - giallo; *3* - rosso.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Si noti che l'item potrebbe non funzionare in VMware vSphere 6.5 e versioni successive, perché VMware ha deprecato il sensore *VMware Rollup Health State*.

vmware.hv.sensors.get[url,uuid]

 I sensori di stato del vendor HW dell'hypervisor VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.status[url,uuid]

 Lo stato dell'hypervisor VMware.
 Valore restituito: *Integer*: *0* - grigio; *1* - verde; *2* - giallo; *3* - rosso.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Questo item utilizza la proprietà di stato complessivo del sistema host.

vmware.hv.tags.get[url,uuid]

 L'array dei tag dell'hypervisor VMware.
 Valore restituito: JSON object.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Questo item funziona con vSphere 6.5 e versioni successive.

vmware.hv.uptime[url,uuid]

 Il tempo di attività dell'hypervisor VMware (secondi).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

Questo item utilizza la proprietà dello stato complessivo del sistema host.

vmware.hv.version[url,uuid]

 La versione dell'hypervisor VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.hv.vm.num[url,uuid]

 Il numero di macchine virtuali sull'hypervisor VMware.
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale dell'hypervisor VMware.

vmware.rp.cpu.usage[url,rpid]

 L'utilizzo della CPU in hertz durante l'intervallo nel VMware Resource Pool.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **rpid** - l'ID del resource pool VMware.

vmware.rp.memory[url,rpid,<mode>]

 Le metriche di memoria del resource pool VMware.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **rpid** - l'ID del resource pool VMware;
- **mode** - valori possibili:
consumed (predefinito) - la quantità di memoria fisica dell'host consumata per supportare le pagine di memoria fisica guest
 ballooned - la quantità di memoria fisica guest recuperata dalla macchina virtuale dal driver balloon nel guest
overhead - la memoria fisica dell'host consumata dalle strutture dati ESXi per eseguire le macchine virtuali

vmware.alarms.geturl

 I dati degli allarmi del centro virtuale VMware.
 Valore restituito: JSON object.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.vm.alarms.get[url,uuid]

 I dati degli allarmi della macchina virtuale VMware.
 Valore restituito: JSON object.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.attribute[url,uuid,name]

 Il valore dell'attributo personalizzato della macchina virtuale VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;

- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **name** - il nome dell'attributo personalizzato.

vmware.vm.cluster.name[url,uuid]

 Il nome della macchina virtuale VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.consolidationneeded[url,uuid]

 Il disco della macchina virtuale VMware richiede il consolidamento.
 Valore restituito: *String*: *true* - è necessario il consolidamento; *false* - il consolidamento non è necessario.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.cpu.latency[url,uuid]

 La percentuale di tempo in cui la macchina virtuale non è in grado di essere eseguita perché è in contesa per l'accesso alla/e CPU fisica/he.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.cpu.num[url,uuid]

 Il numero di processori sulla macchina virtuale VMware.
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.cpu.readiness[url,uuid,<instance>]

 La percentuale di tempo in cui la macchina virtuale era pronta, ma non ha potuto essere pianificata per l'esecuzione sulla CPU fisica.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza della CPU.

vmware.vm.cpu.ready[url,uuid]

 Il tempo (in millisecondi) durante il quale la macchina virtuale era pronta, ma non ha potuto essere pianificata per l'esecuzione sulla CPU fisica. Il tempo di CPU ready dipende dal numero di macchine virtuali sul host e dai loro carichi CPU (%).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 La percentuale di tempo della CPU trascorsa in attesa dello swap-in.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza della CPU.

vmware.vm.cpu.usage[url,uuid]

 L'utilizzo del processore (Hz) della macchina virtuale VMware.
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.cpu.usage.perf[url,uuid]

 L'utilizzo del processore della macchina virtuale VMware come percentuale durante l'intervallo.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.datacenter.name[url,uuid]

 Il nome del datacenter della macchina virtuale VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.discoveryurl

 La discovery delle macchine virtuali VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.vm.guest.memory.size.swapped[url,uuid]

 La quantità di memoria fisica del guest trasferita nello spazio di swap (KB).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.guest.osuptime[url,uuid]

 Il tempo totale trascorso dall'ultimo avvio del sistema operativo (in secondi).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.hv.maintenance[url,uuid]

 Lo stato di manutenzione dell'hypervisor della macchina virtuale VMware.
 Valore restituito: *0* - non in manutenzione; *1* - in manutenzione.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.hv.name[url,uuid]

 Il nome dell'hypervisor della macchina virtuale VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size[url,uuid]

 La dimensione totale della memoria della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.ballooned[url,uuid]

 La dimensione della memoria ballooned della macchina virtuale VMware (byte).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.compressed[url,uuid]

 La dimensione della memoria compressa della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.consumed[url,uuid]

 La quantità di memoria fisica dell'host consumata per il backup delle pagine di memoria fisica del guest (KB).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.private[url,uuid]

 La dimensione della memoria privata della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.shared[url,uuid]

 La dimensione della memoria condivisa della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.swapped[url,uuid]

 La dimensione della memoria di swap della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.usage.guest[url,uuid]

 L'utilizzo della memoria guest (byte) della macchina virtuale VMware.
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.size.usage.host[url,uuid]

 L'utilizzo della memoria del host della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.memory.usage[url,uuid]

 La percentuale di memoria fisica del host che è stata consumata.
 Valore restituito: *Float*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.net.if.discovery[url,uuid]

 La discovery delle interfacce di rete delle macchine virtuali VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 Le statistiche di input dell'interfaccia di rete della macchina virtuale VMware (byte/pacchetti al secondo).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza dell'interfaccia di rete;
- **mode** - *bps* (predefinito) o *pps* - byte o pacchetti al secondo.

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 Le statistiche di output dell'interfaccia di rete della macchina virtuale VMware (byte/pacchetti al secondo).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza dell'interfaccia di rete;
- **mode** - *bps* (predefinito) oppure *pps* - byte o pacchetti al secondo.

vmware.vm.net.if.usage[url,uuid,<instance>]

 L'utilizzo della rete della macchina virtuale VMware (somma delle velocità di trasmissione e ricezione) durante l'intervallo (KBps).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza dell'interfaccia di rete.

vmware.vm.perfcounter[url,uuid,path,<instance>]

 Il valore del contatore delle prestazioni della macchina virtuale VMware.
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **path** - il percorso del contatore delle prestazioni [1];
- **instance** - l'istanza del contatore delle prestazioni. Utilizzare un'istanza vuota per i valori aggregati (predefinito).

vmware.vm.powerstate[url,uuid]

 Lo stato di alimentazione della macchina virtuale VMware.
 Valore restituito: *0* - poweredOff; *1* - poweredOn; *2* - suspended.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.property[url,uuid,prop]

 La proprietà della macchina virtuale VMware.
 Valore restituito: *String*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **prop** - il percorso della proprietà.

vmware.vm.snapshot.get[url,uuid]

 Lo stato dello snapshot della macchina virtuale VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.state[url,uuid]

 Lo stato della macchina virtuale VMware.
 Valore restituito: *String*: *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* oppure *unknown*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.storage.committed[url,uuid]

 Lo spazio di archiviazione committed della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.storage.readoio[url,uuid,instance]

 Il numero medio di richieste di lettura in sospeso verso il disco virtuale durante l'intervallo di raccolta.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza del dispositivo disco.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 Il tempo medio impiegato da una lettura dal disco virtuale (millisecondi).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza del dispositivo disco.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 Il tempo medio necessario per una scrittura sul disco virtuale (millisecondi).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza del dispositivo disco.

vmware.vm.storage.uncommitted[url,uuid]

 Lo spazio di archiviazione non impegnato della macchina virtuale VMware (byte).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.storage.unshared[url,uuid]

 Lo spazio di archiviazione non condiviso della macchina virtuale VMware (byte).
 Valore restituito: *Intero*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.storage.writeoio[url,uuid,instance]

 Il numero medio di richieste di scrittura in sospeso verso il disco virtuale durante l'intervallo di raccolta.
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;

- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza del dispositivo disco.

vmware.vm.tags.get[url,uuid]

 L'array dei tag della macchina virtuale VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

Questo item funziona con vSphere 6.5 e versioni successive.

vmware.vm.tools[url,uuid,<mode>]

 Lo stato degli strumenti guest della macchina virtuale VMware.
 Valore restituito: *String*.
 In modalità *status*: *guestToolsExecutingScripts* - VMware Tools si sta avviando; *guestToolsNotRunning* - VMware Tools non è in esecuzione; *guestToolsRunning* - VMware Tools è in esecuzione.
 In modalità *version*: versione.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **mode** - *version*, *status* (predefinito).

vmware.vm.uptime[url,uuid]

 Il tempo di attività della macchina virtuale VMware (secondi).
 Valore restituito: *Integer*.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.vfs.dev.discovery[url,uuid]

 La discovery dei dispositivi disco delle macchine virtuali VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 Le statistiche di lettura del dispositivo disco della macchina virtuale VMware (byte/operazioni al secondo).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza del dispositivo disco;
- **mode** - *bps* (predefinito) oppure *ops* - byte o operazioni al secondo.

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 Le statistiche di scrittura del dispositivo disco della macchina virtuale VMware (byte/operazioni al secondo).
 Valore restituito: *Integer* [2].

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **instance** - l'istanza del dispositivo disco;
- **mode** - *bps* (predefinito) oppure *ops* - byte o operazioni al secondo.

vmware.vm.vfs.fs.discovery[url,uuid]

 Il rilevamento dei file system della macchina virtuale VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware.

Affinché questo item funzioni, VMware Tools deve essere installato sulla macchina virtuale guest.

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 Le statistiche del file system della macchina virtuale VMware (byte/percentuali).
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.

Parametri:

- **url** - l'URL del servizio VMware;
- **uuid** - l'identificatore univoco globale della macchina virtuale VMware;
- **fsname** - il nome del file system;
- **mode** - *total*, *free*, *used*, *pfree* o *pused*.

Per il funzionamento di questo item, VMware Tools deve essere installato sulla macchina virtuale guest.

Note a piè di pagina

[1] Vedere Creazione di nomi personalizzati per i contatori delle prestazioni per VMware.

[2] Il valore di questi item viene ottenuto dai contatori delle prestazioni VMware e il parametro VMwarePerfFrequency viene utilizzato per aggiornare i loro dati nella cache VMware di Zabbix:

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write
- vmware.hv.network.in
- vmware.hv.network.out
- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

Ulteriori informazioni

Vedere Monitoraggio delle macchine virtuali per informazioni dettagliate su come configurare Zabbix per monitorare gli ambienti VMware.

Campi chiave di discovery delle macchine virtuali

La seguente tabella elenca i campi restituiti dalle chiavi di discovery relative alle macchine virtuali.

Item key		
Description	**Field**	**Retrieved content**
vmware.cluster.discovery		
Esegue la discovery del cluster.	{#CLUSTER.ID}	Identificatore del cluster.
	{#CLUSTER.NAME}	Nome del cluster.

	"resource_pool"	Un array contenente i dati del resource pool, incluso l'ID del gruppo di risorse, l'array dei tag, il percorso del resource pool, il numero di macchine virtuali.
		
		Struttura dell'array:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Per la struttura dell'array `"tags"`, vedere il campo "tags".

Item key		

	”tags”	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell’array:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Esegue la discovery del datastore.	{#DATASTORE}	Nome del datastore.

	{#DATASTORE.EXTENT}	Un array contenente l’ID della partizione dell’extent del datastore e il nome dell’istanza.
		
		Struttura dell’array:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Tipo di datastore. Esempi di valori: `VMFS`, `NFS`, `vsan`, ecc.
	{#DATASTORE.UUID}	Identificatore del datastore.

	”tags”	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell’array:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dc.discovery		
Esegue la discovery del datacenter.	{#DATACENTER}	Nome del datacenter.
	{#DATACENTERID}	Identificatore del datacenter.

	”tags”	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell’array:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dvswitch.discovery		
Esegue la discovery degli switch distribuiti di vSphere.	{#DVS.NAME}	Nome dello switch.
	{#DVS.UUID}	Identificatore dello switch.
vmware.hv.discovery		
Esegue la discovery dell’hypervisor.	{#HV.UUID}	Identificatore univoco dell’hypervisor.
	{#HV.ID}	Identificatore dell’hypervisor (nome dell’oggetto gestito HostSystem).
	{#HV.NAME}	Nome dell’hypervisor.
	{#HV.NETNAME}	Nome host di rete dell’hypervisor.
	{#HV.IP}	Indirizzo IP dell’hypervisor, potrebbe essere vuoto. Nel caso di una configurazione HA con più interfacce di rete, viene osservata la seguente priorità di selezione per l’interfaccia: - preferire l’IP che condivide la sottorete IP con l’IP di vCenter; - preferire l’IP della sottorete IP con gateway predefinito; - preferire l’IP dell’interfaccia con l’ID più basso.

Item key		
	{#CLUSTER.NAME}	Nome del cluster, potrebbe essere vuoto.
	{#DATACENTER.NAME}	Nome del datacenter.
	{#PARENT.NAME}	Nome del contenitore che ospita l'hypervisor.
	{#PARENT.TYPE}	Tipo di contenitore in cui è archiviato l'hypervisor. I valori possono essere `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, dove "VMware" indica un tipo di contenitore sconosciuto.

	"resource_pool"	Un array contenente i dati del resource pool, incluso l'ID del gruppo di risorse, l'array dei tag, il percorso del resource pool, il numero di macchine virtuali.
		
		Struttura dell'array:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Per la struttura dell'array `"tags"`, vedere il campo "tags".

	"tags"	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell'array:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.datastore.discovery		
Esegue la discovery del datastore dell'hypervisor. Si noti che più hypervisor possono usare lo stesso datastore.	{#DATASTORE}	Nome del datastore.
	{#DATASTORE.TYPE}	Tipo di datastore. Esempi di valori: `VMFS`, `NFS`, `vsan`, ecc.
	{#DATASTORE.UUID}	Identificatore del datastore.
	{#MULTIPATH.COUNT}	Numero registrato di percorsi del datastore.
	{#MULTIPATH.PARTITION.COUNT}	Numero di partizioni disco disponibili.

	"datastore_extent"	Un array contenente il nome dell'istanza dell'extent del datastore e l'ID della partizione.
		
		Struttura dell'array:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	"tags"	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell'array:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.net.if.discovery		
Esegue la discovery delle interfacce di rete dell'hypervisor.	{#IFNAME}	Nome dell'interfaccia.
	{#IFDRIVER}	Driver dell'interfaccia.

Item key		
	{#IFDUPLEX}	Impostazioni duplex dell'interfaccia.
	{#IFSPEED}	Velocità dell'interfaccia.
	{#IFMAC}	Indirizzo MAC dell'interfaccia.
vmware.vm.discovery		
Esegue la discovery della macchina virtuale.	{#VM.UUID}	Identificatore univoco della macchina virtuale.
	{#VM.ID}	Identificatore della macchina virtuale (nome dell'oggetto gestito VirtualMachine).
	{#VM.NAME}	Nome della macchina virtuale.
	{#HV.NAME}	Nome dell'hypervisor.
	{#HV.UUID}	Identificatore univoco dell'hypervisor.
	{#HV.ID}	Identificatore dell'hypervisor (nome dell'oggetto gestito HostSystem).
	{#CLUSTER.NAME}	Nome del cluster, potrebbe essere vuoto.
	{#DATACENTER.NAME}	Nome del datacenter.
	{#DATASTORE.NAME}	Nome del datastore.
	{#DATASTORE.UUID}	Identificatore del datastore.
	{#VM.IP}	Indirizzo IP della macchina virtuale, potrebbe essere vuoto.
	{#VM.DNS}	Nome DNS della macchina virtuale, potrebbe essere vuoto.
	{#VM.GUESTFAMILY}	Famiglia del sistema operativo guest della macchina virtuale, potrebbe essere vuoto.
	{#VM.GUESTFULLNAME}	Nome completo del sistema operativo guest della macchina virtuale, potrebbe essere vuoto.
	{#VM.FOLDER}	La catena delle cartelle padre della macchina virtuale, può essere usata come valore per gruppi nidificati; i nomi delle cartelle sono combinati con "/". Potrebbe essere vuoto.
	{#VM.TOOLS.STATUS}	Stato degli strumenti della macchina virtuale VMware.
	{#VM.POWERSTATE}	Stato di alimentazione della macchina virtuale VMware (`poweredOff`, `poweredOn`, o `suspended`).
	{#VM.RPOOL.ID}	Identificatore del resource pool.
	{#VM.RPOOL.PATH}	Percorso completo del resource pool escludendo il nome "root" "Resources". I nomi delle cartelle sono combinati con "/".
	{#VM.SNAPSHOT.COUNT}	Numero di snapshot della VM.

	"tags"	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell'array:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

	"vm_customattribute"	Un array di attributi personalizzati della macchina virtuale (se definiti).
		
		Struttura dell'array:
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

Item key		

	”net_if”	Un array di interfacce di rete della macchina virtuale.
		
		Struttura dell’array:
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		Per la descrizione dei dati restituiti, vedere la chiave item ”vmware.vm.net.if.discovery”.

vmware.vm.net.if.discovery		
Esegue la discovery delle interfacce di rete della macchina virtuale.	{#IFNAME}	Nome dell’interfaccia di rete.
	{#IFDESC}	Descrizione dell’interfaccia.
	{#IFMAC}	Indirizzo MAC dell’interfaccia.
	{#IFCONNECTED}	Stato di connessione dell’interfaccia (`false` - disconnessa; `true` - connessa).
	{#IFTYPE}	Tipo di interfaccia.
	{#IFBACKINGDEVICE}	Nome del dispositivo di backing.
	{#IFDVSWITCH.UUID}	Identificatore univoco di vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Port group distribuito.
	{#IFDVSWITCH.PORT}	Porta di vSphere Distributed Switch.
	”ifip”	Un array di indirizzi dell’interfaccia.
vmware.vm.vfs.dev.discovery		
Esegue la discovery del dispositivo disco della macchina virtuale.	{#DISKNAME}	Nome del dispositivo disco.
vmware.vm.vfs.fs.discovery		
Esegue la discovery del file system della macchina virtuale.	{#FSNAME}	Nome del file system.

Esempi JSON per gli item VMware

Panoramica Questa sezione fornisce informazioni aggiuntive sugli oggetti JSON restituiti da vari item VMware.

vmware.*.alarms.get Gli item **vmware.alarms.get[]**, **vmware.cluster.alarms.get[]**, **vmware.datastore.alarms.get[]**, **vmware.dc.alarms.get[]**, **vmware.hv.alarms.get[]**, **vmware.vm.alarms.get[]** restituiscono oggetti JSON con la seguente struttura (i valori sono forniti come esempio):

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
```

```
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get Gli item **vmware.cluster.tags.get[]**, **vmware.datastore.tags.get[]**, **vmware.dc.tags.get[]**, **vmware.hv.tags.get[]**, **vmware.vm.tags.get[]** restituiscono oggetti JSON con la seguente struttura (i valori sono forniti come esempio):

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "tag per la categoria tipo di OS",
        "category": "tipo di OS"
    },
    {
            "name": "SQL Server",
        "description": "tag per la categoria nome applicazione",
        "category": "nome applicazione"
        },
    {
            // altri tag
        }
    ]
}
```

vmware.hv.diskinfo.get The item **vmware.hv.diskinfo.get[]** returns JSON objects with the following structure (values are provided as an example):

```
[
  {
    "instance": "mpx.vmhba32:C0:T0:L0",
    "hv_uuid": "8002299e-d7b9-8728-d224-76004bbb6100",
```

```
    "datastore_uuid": "",
    "operational_state": [
      "ok"
    ],
    "lun_type": "disk",
    "queue_depth": 1,
    "model": "USB DISK",
    "vendor": "SMI Corp",
    "revision": "1100",
    "serial_number": "CCYYMMDDHHmmSS9S62CK",
    "vsan": {}
  },
  {
    // other instances
  }
]
```

vmware.dvswitch.fetchports.get L'item **vmware.dvswitch.fetchports.get[]** restituisce oggetti JSON con la seguente struttura (i valori sono forniti come esempio):

```
{
    "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"
            },
            {
                //altre chiavi
            }
        ]
    }
}
```

vmware.hv.hw.sensors.get L'item **vmware.hv.hw.sensors.get[]** restituisce oggetti JSON con la seguente struttura (i valori sono forniti come esempio):

```
{
    "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"
                 }
            },
            {
                    // altri sensori hw dell'hv
                }
        ]
        }
}
```

vmware.hv.sensors.get L'item **vmware.hv.sensors.get[]** restituisce oggetti JSON con la seguente struttura (i valori sono forniti come esempio):


```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10500",
                "unitModifier": "-2",
                "baseUnits": "Watts",
                "sensorType": "other"
            },
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "Power Supply 1 PS 1 Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                    "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10000",
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                    // altri sensori hv
                }
        ]
        }
}
```


vmware.vm.snapshot.get Se esistono snapshot, l'item **vmware.snapshot.get[]** restituisce un oggetto JSON con la seguente struttura (i valori sono forniti come esempio):


```
{
  "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
}
```


Se non esiste alcuno snapshot, l'item **vmware.snapshot.get[]** restituisce un oggetto JSON con valori vuoti:


```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```


Esempio di configurazione del monitoraggio VMware

Panoramica

L'esempio seguente descrive come configurare Zabbix per monitorare le macchine virtuali VMware. Ciò comporta:

- la creazione di un host che rappresenti il tuo ambiente VMware;
- la creazione di una regola di low-level discovery che rilevi le macchine virtuali nel tuo ambiente VMware;
- la creazione di un prototipo di host, sulla base del quale Zabbix genererà host reali per le macchine virtuali rilevate dalla regola di low-level discovery.

Prerequisiti

> **Note:**
> Questo esempio non copre la configurazione di VMware. Si presume che VMware sia già configurato.

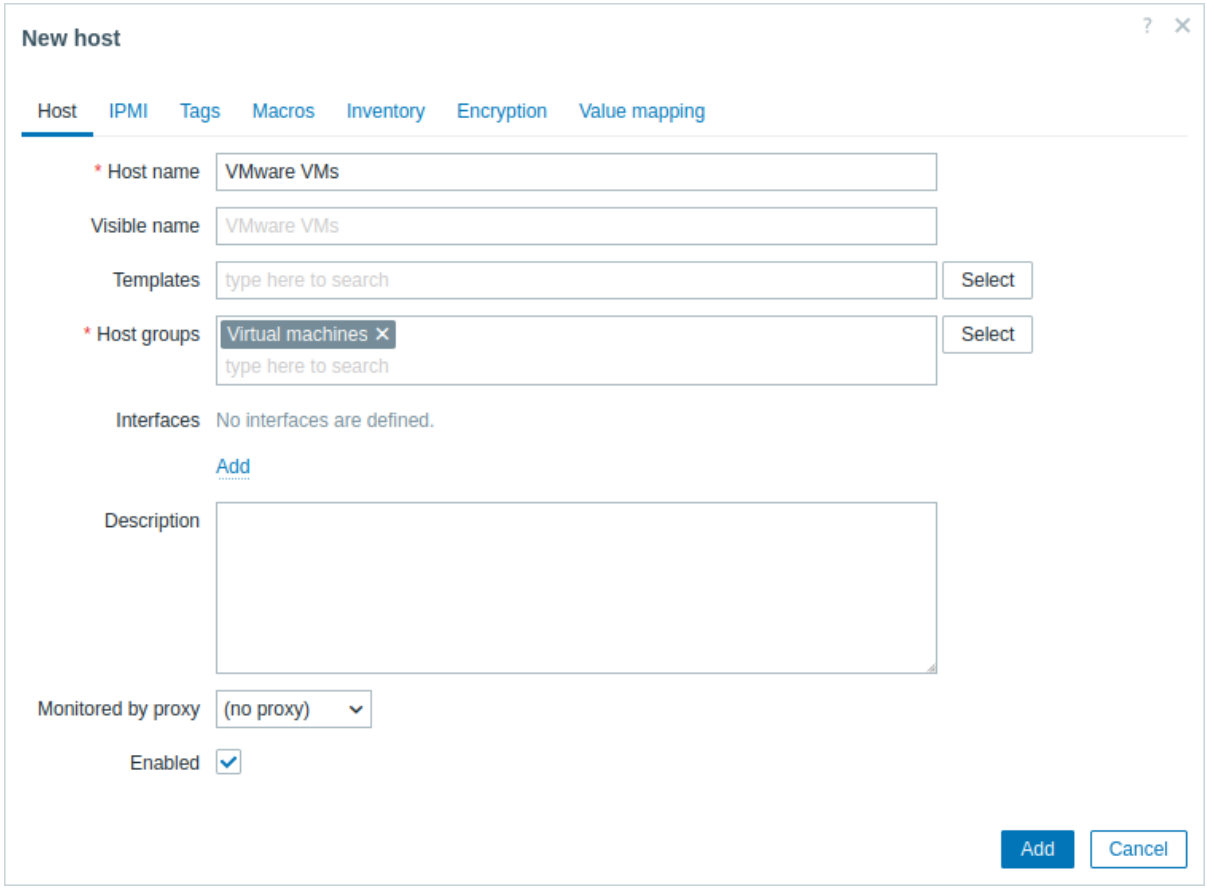
Prima di procedere, impostare il parametro `StartVMwareCollectors` nel file di configurazione di Zabbix server su 2 o più (il valore predefinito è 0).

Creare un host

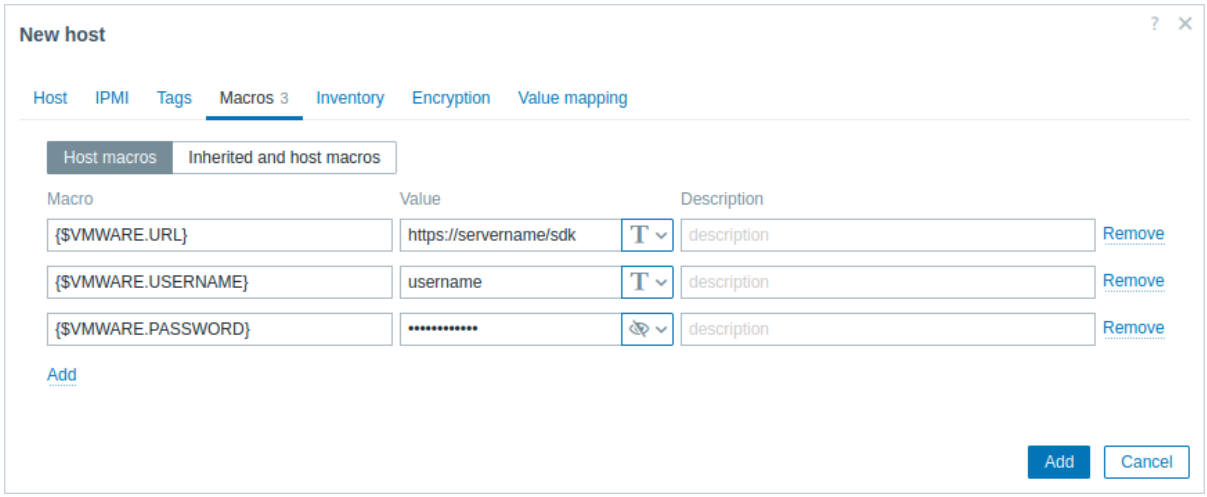
1. Vai a *Data collection* > *Hosts*.

2. Crea un host:

- Nel campo *Host name*, inserisci un nome host (ad esempio, "VMware VMs").
- Nel campo *Host groups*, digita o seleziona un gruppo host (ad esempio, "Virtual machines").



- Nella scheda *Macros*, imposta le seguenti macro host:
  - {$VMWARE.URL} - URL SDK del servizio VMware (hypervisor ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - nome utente del servizio VMware
  - {$VMWARE.PASSWORD} - password dell'utente {$VMWARE.USERNAME} del servizio VMware



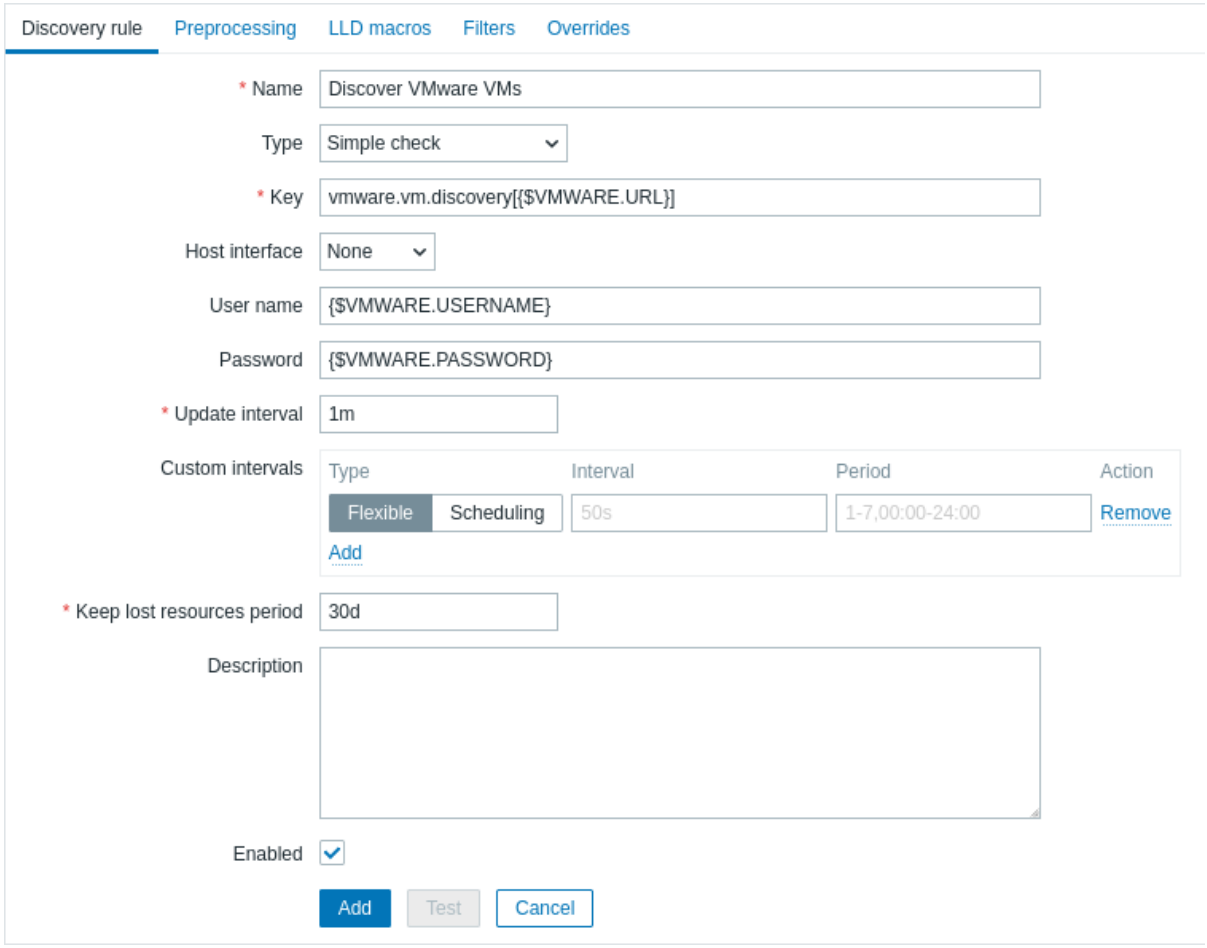
3. Fai clic sul pulsante *Add* per creare l'host. Questo host rappresenterà il tuo ambiente VMware.

Creare una regola di low-level discovery

1. Fare clic su *Discovery* per accedere all'elenco delle regole di low-level discovery per il host creato.

2. Creare una regola di low-level discovery:

- Nel campo *Name*, inserire un nome per la regola di low-level discovery (ad esempio, "Discover VMware VMs").
- Nel campo *Type*, selezionare "Simple check".
- Nel campo *Key*, inserire la chiave item integrata per il rilevamento delle macchine virtuali VMware: `vmware.vm.discovery[{$VMWARE`
- Nei campi *User name* e *Password*, inserire le macro corrispondenti precedentemente configurate sul host.



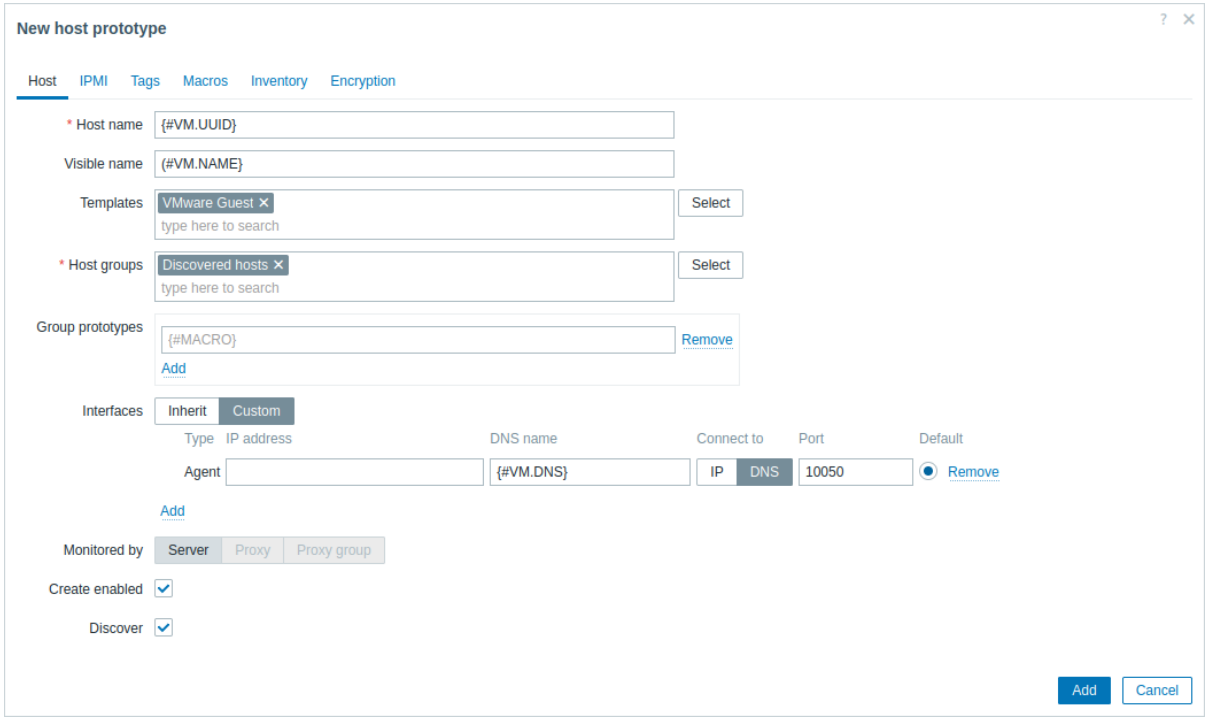
3. Fare clic sul pulsante *Add* per creare la regola di low-level discovery. Questa regola di discovery rileverà le macchine virtuali nel proprio ambiente VMware.

Creare un prototipo di host

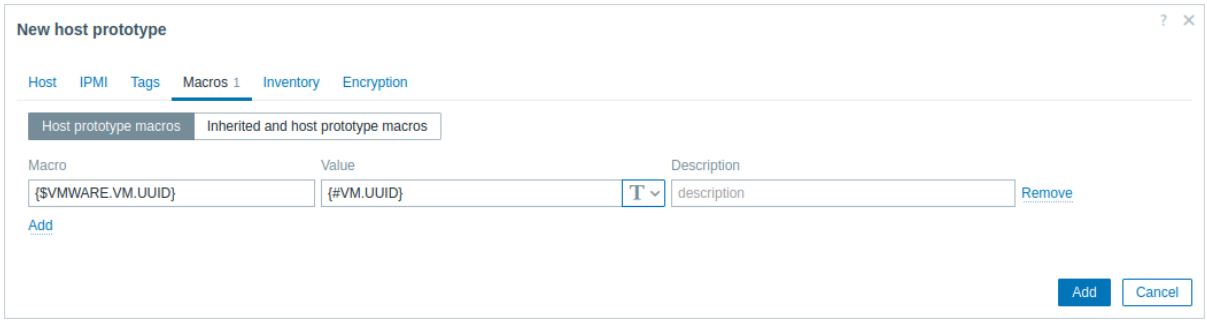
1. Nell'elenco delle regole di low-level discovery, fai clic su *Host prototypes* per la regola di low-level discovery creata in precedenza.

2. Crea un prototipo di host. Poiché i prototipi di host sono modelli per la creazione di host tramite regole di low-level discovery, la maggior parte dei campi conterrà macro di low-level discovery. Questo garantisce che gli host vengano creati con proprietà basate sul contenuto recuperato dalla regola di low-level discovery creata in precedenza.

- Nel campo *Host name*, inserisci la macro {#VM.UUID}.
- Nel campo *Visible name*, inserisci la macro {#VM.NAME}.
- Nel campo *Templates*, inserisci o seleziona il template "VMware Guest". Questo template contiene item VMware e regole di discovery per monitorare lo stato di alimentazione di una macchina virtuale, l'utilizzo della CPU, l'utilizzo della memoria, i dispositivi di rete, ecc.
- Nel campo *Host groups*, inserisci o seleziona un gruppo host (ad esempio, "Discovered hosts").
- Nel campo *Interfaces*, aggiungi un'interfaccia host personalizzata. Quindi, inserisci la macro {#VM.DNS} nel campo *DNS name* oppure inserisci la macro {#VM.IP} nel campo *IP address*. In alternativa, se le macchine virtuali del tuo ambiente VMware hanno più interfacce, passa alla sezione *Configurazione avanzata dell'interfaccia host*. La configurazione di un'interfaccia host personalizzata è necessaria per il corretto funzionamento del template *VMware Guest*.



- Nella scheda *Macros*, imposta la macro {$VMWARE.VM.UUID} con il valore {#VM.UUID}. Questo è necessario per il corretto funzionamento del template *VMware Guest*, che utilizza questa macro come macro utente a livello di host nei parametri degli item (ad esempio, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

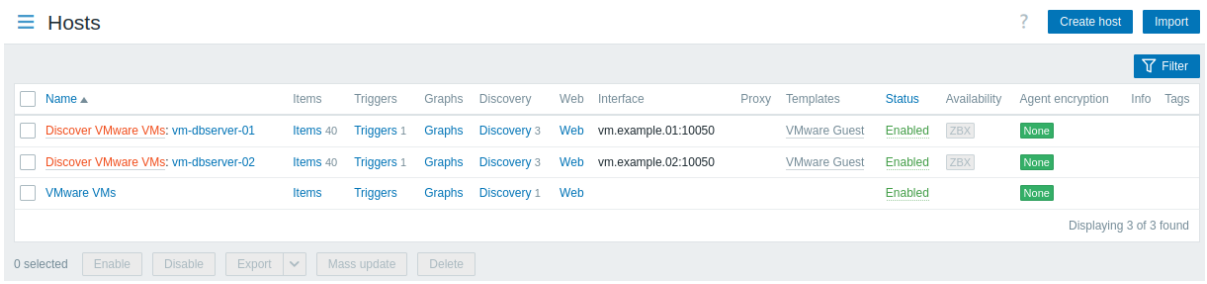


3. Fai clic sul pulsante *Add* per creare il prototipo di host. Questo prototipo di host verrà utilizzato per creare host per le macchine virtuali rilevate dalla regola di low-level discovery creata in precedenza.

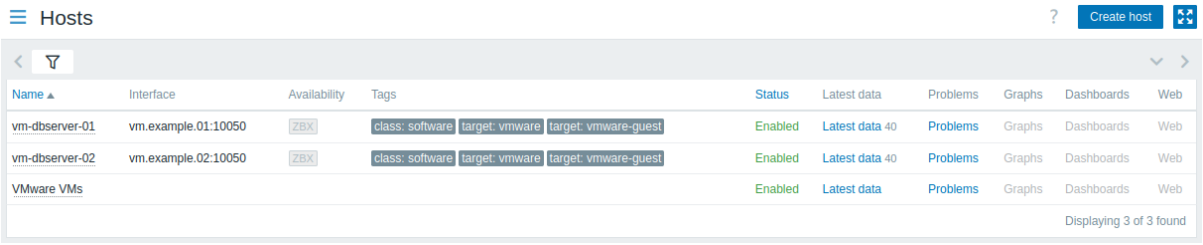
Visualizzare host e metriche

Dopo che il host prototype è stato creato, la regola di discovery a basso livello creerà host per le macchine virtuali VMware rilevate e Zabbix inizierà a monitorarle. Si noti che la discovery e la creazione degli host possono anche essere eseguite manualmente, se necessario.

Per visualizzare gli host creati, andare alla sezione di menu *Raccolta dati > Host*.



Per visualizzare le metriche raccolte, andare alla sezione di menu *Monitoraggio > Host* e fare clic su *Latest data* per uno degli host.



Configurazione avanzata dell'interfaccia host

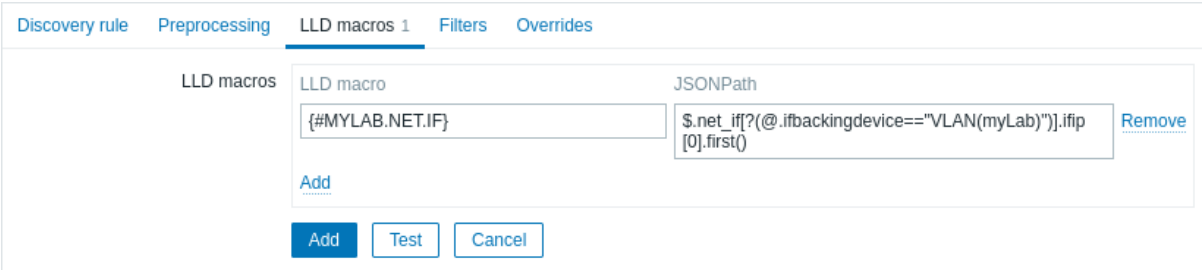
La chiave item `vmware.vm.discovery[{$VMWARE.URL}]`, configurata nella sezione *Creare una regola di low-level discovery*, restituisce i dati delle interfacce di rete nel campo "net_if":

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

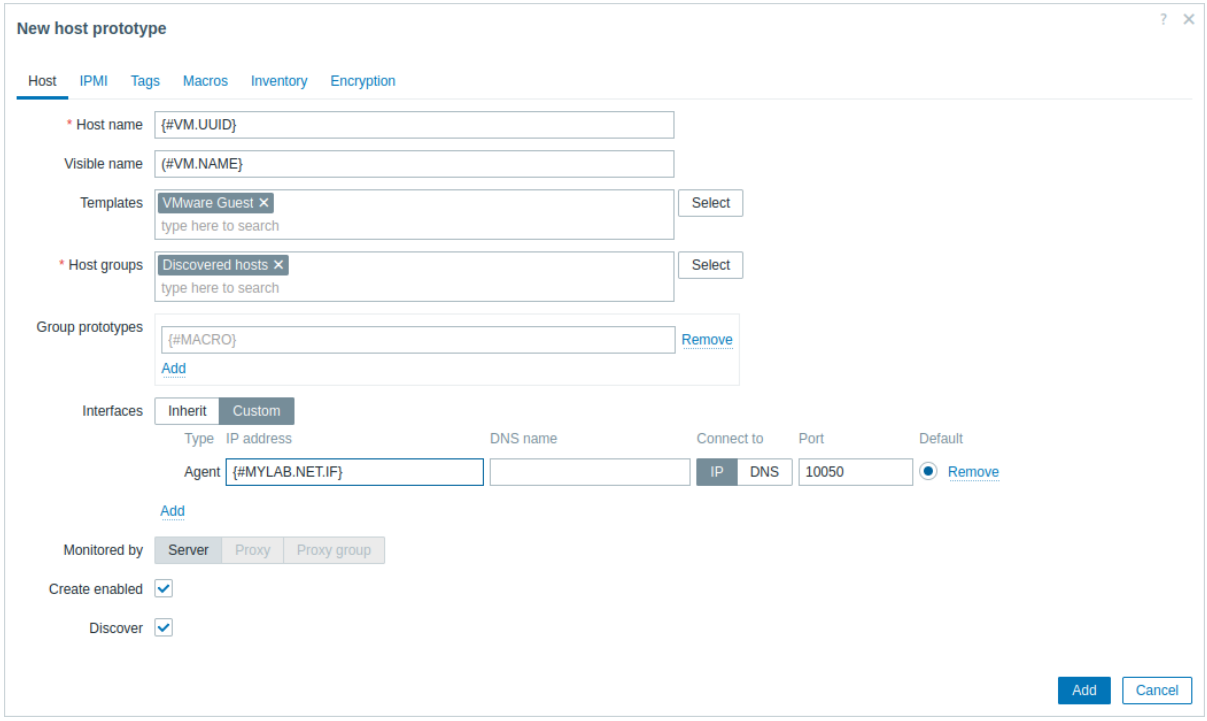
Questi dati possono essere utilizzati per configurare un'interfaccia host personalizzata.

1. Quando si crea una regola di low-level discovery, configurare inoltre una macro di low-level discovery. Nella scheda *LLD macros*, creare una macro LLD personalizzata con un valore JSONPath. Ad esempio:

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. Quando si crea un prototipo di host, aggiungere un'interfaccia host personalizzata e inserire la macro LLD nel campo *DNS name* o *IP address*.



Monitoraggio web

Panoramica Con Zabbix puoi verificare diversi aspetti di disponibilità dei siti web.

Attention:
Per eseguire il monitoraggio web, il server Zabbix deve essere inizialmente configurato con il supporto cURL (libcurl).

Per attivare il monitoraggio web devi definire degli scenari web. Uno scenario web è composto da una o più richieste HTTP o "step". Gli step vengono eseguiti periodicamente dal server Zabbix in un ordine predefinito. Se un host è monitorato da proxy, gli step vengono eseguiti dal proxy.

Gli scenari web sono associati a host/template nello stesso modo di item, trigger, ecc. Ciò significa che gli scenari web possono anche essere creati a livello di template e poi applicati a più host in un'unica operazione.

Le seguenti informazioni vengono raccolte in qualsiasi scenario web:

- velocità media di download al secondo per tutti gli step dell'intero scenario
- numero dello step che è fallito
- ultimo messaggio di errore

Le seguenti informazioni vengono raccolte in qualsiasi step dello scenario web:

- velocità di download al secondo
- tempo di risposta
- codice di risposta

Per ulteriori dettagli, consulta gli item di monitoraggio web.

I dati raccolti dall'esecuzione degli scenari web vengono conservati nel database. I dati vengono utilizzati automaticamente per grafici, trigger e notifiche.

Zabbix può anche verificare se una pagina HTML recuperata contiene una stringa predefinita. Può eseguire un accesso simulato e seguire un percorso di clic del mouse simulati sulla pagina.

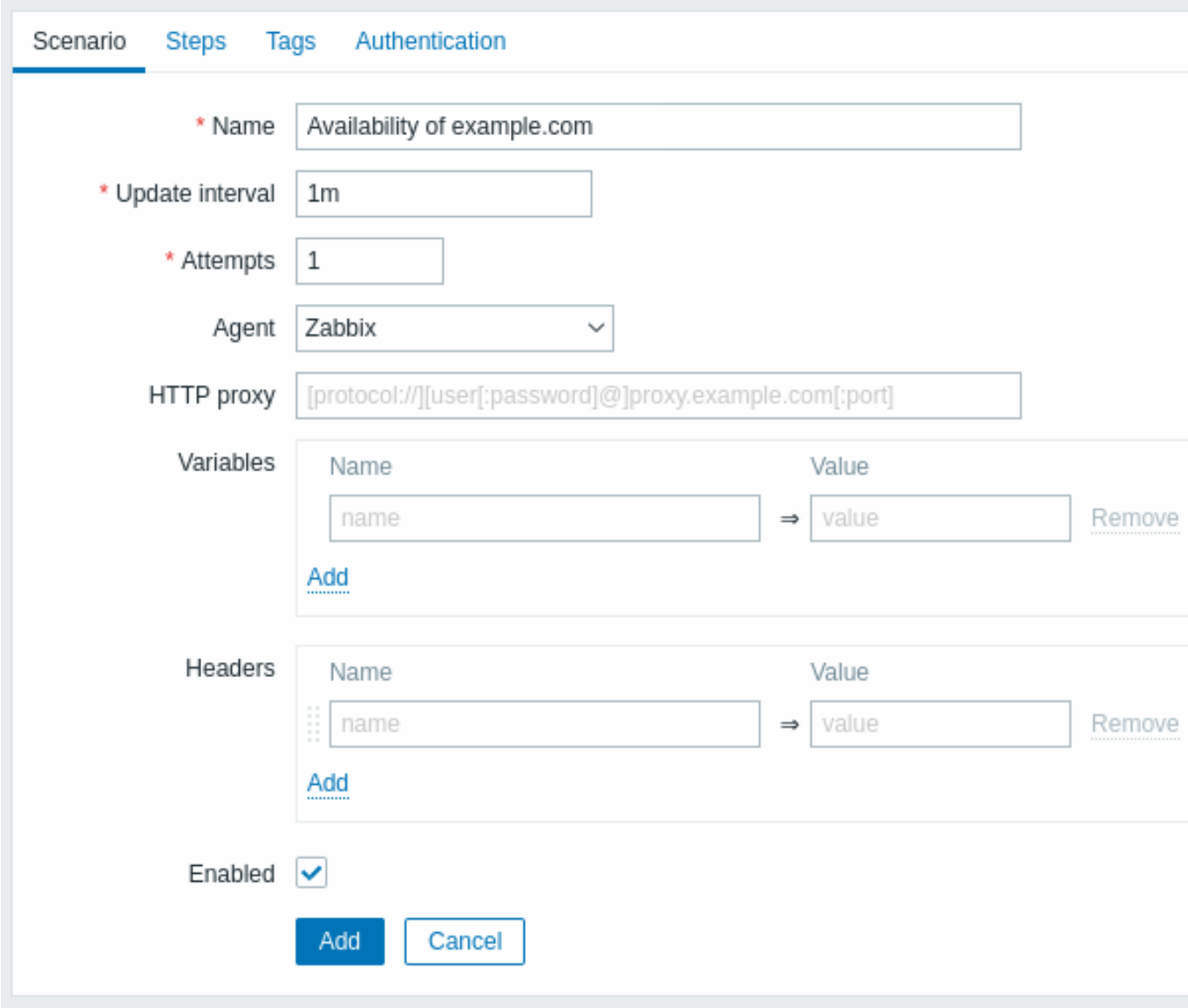
Il monitoraggio web di Zabbix supporta sia HTTP che HTTPS. Quando esegue uno scenario web, Zabbix può facoltativamente seguire i reindirizzamenti (vedi l'opzione *Follow redirects* sotto). Il numero massimo di reindirizzamenti è impostato in modo fisso a 10 (usando l'opzione cURL CURLOPT_MAXREDIRS). Tutti i cookie vengono mantenuti durante l'esecuzione di un singolo scenario.

Configurazione di uno scenario web Per configurare uno scenario web:

1. Vai a: *Raccolta dati* > *Host* (o *Template*).

2. Fai clic su *Web* nella riga dell'host/template.
3. Fai clic su *Crea scenario web* a destra (oppure sul nome dello scenario per modificare uno scenario esistente).
4. Inserisci i parametri dello scenario nel modulo.

La scheda **Scenario** consente di configurare i parametri generali di uno scenario web.



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametri dello scenario:

Parametro	Descrizione
Nome	Nome univoco dello scenario. Le macro utente sono supportate. *Nota*: se vengono utilizzate macro utente, queste macro rimarranno irrisolte nei nomi degli item di monitoraggio web.
Intervallo di aggiornamento	Frequenza di esecuzione dello scenario. I suffissi temporali sono supportati, ad esempio 30s, 1m, 2h, 1d. Le macro utente sono supportate. *Nota*: se viene utilizzata una macro utente e il suo valore viene modificato (ad esempio, 5m → 30s), il controllo successivo verrà eseguito in base al valore precedente (più avanti nel futuro con i valori dell'esempio). I nuovi scenari web verranno controllati entro 60 secondi dalla loro creazione.
Tentativi	Numero di tentativi per l'esecuzione dei passaggi dello scenario web. In caso di problemi di rete (timeout, assenza di connettività, ecc.), Zabbix può ripetere l'esecuzione di un passaggio diverse volte. Il valore impostato verrà applicato allo stesso modo a ogni passaggio dello scenario. È possibile specificare fino a 10 tentativi; il valore predefinito è 1. *Nota*: Zabbix non ripeterà un passaggio a causa di un codice di risposta errato o della mancata corrispondenza di una stringa richiesta.

Parametro	Descrizione
Agent	Seleziona un agent client. Zabbix simulerà il browser selezionato. Questa opzione è utile quando un sito web restituisce contenuti diversi per browser diversi. In questo campo è possibile utilizzare macro utente.
Proxy HTTP	È possibile specificare un proxy HTTP da utilizzare, nel formato `[protocol://][username[:password]@]proxy.example.com[:port]`. Questa impostazione definisce l'opzione cURL CURLOPT_PROXY. Il prefisso facoltativo `protocol://` può essere utilizzato per specificare protocolli proxy alternativi (il supporto per il prefisso del protocollo è stato aggiunto in cURL 7.21.7). Se non viene specificato alcun protocollo, il proxy verrà trattato come un proxy HTTP. Per impostazione predefinita verrà utilizzata la porta 1080. Se specificato, il proxy sovrascriverà le variabili d'ambiente relative al proxy, come http_proxy e HTTPS_PROXY. Se non specificato, il proxy non sovrascriverà le variabili d'ambiente relative al proxy. Il valore inserito viene passato "così com'è", senza alcun controllo di validità. È inoltre possibile inserire l'indirizzo di un proxy SOCKS. Se si specifica il protocollo errato, la connessione avrà esito negativo e l'item diventerà non supportato. *Nota*: con un proxy HTTP è supportata solo l'autenticazione semplice. In questo campo è possibile utilizzare macro utente.
Variabili	Variabili che possono essere utilizzate nei passaggi dello scenario (URL, variabili post). Hanno il formato seguente: **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (vedere funzioni delle macro) Ad esempio: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} Le macro possono quindi essere referenziate nei passaggi come {username}, {password}, {hostid}, ecc. Zabbix le sostituirà automaticamente con i valori effettivi. Nota: le variabili con `regex:` richiedono un passaggio per ottenere il valore dell'espressione regolare, pertanto il valore estratto può essere applicato solo al passaggio successivo. Se la parte del valore inizia con `regex:`, la parte successiva viene trattata come un'espressione regolare che cerca nella pagina web e, se trovata, memorizza la corrispondenza nella variabile. Deve essere presente almeno un sottogruppo affinché il valore corrispondente possa essere estratto. Le macro utente e le macro `{HOST.*}` sono supportate. Le variabili vengono codificate automaticamente in formato URL quando utilizzate nei campi di query o nei dati del modulo per le variabili post, ma devono essere codificate manualmente in formato URL quando vengono utilizzate nei dati post non elaborati o direttamente nell'URL.

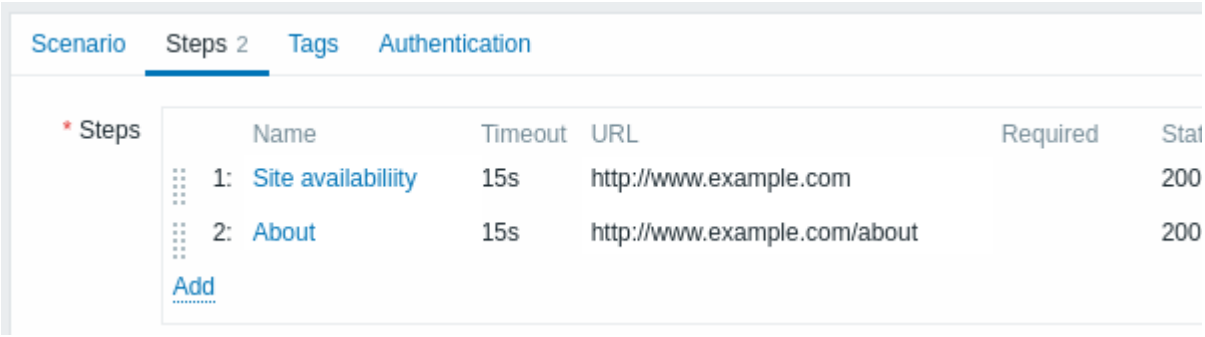
Parametro	Descrizione
Intestazioni	Le intestazioni HTTP vengono utilizzate durante l'esecuzione di una richiesta. È possibile utilizzare intestazioni predefinite e personalizzate. Le intestazioni verranno assegnate utilizzando le impostazioni predefinite in base al tipo di Agent selezionato dall'elenco a discesa a livello di scenario e verranno applicate a tutti i passaggi, a meno che non siano definite personalizzazioni a livello di passaggio. **È importante notare che la definizione dell'intestazione a livello di passaggio elimina automaticamente tutte le intestazioni definite in precedenza, ad eccezione di un'intestazione predefinita assegnata selezionando 'User-Agent' dall'elenco a discesa a livello di scenario.** Tuttavia, anche l'intestazione predefinita 'User-Agent' può essere sovrascritta specificandola a livello di passaggio. Per annullare l'impostazione dell'intestazione a livello di scenario, l'intestazione deve essere denominata e impostata senza alcun valore a livello di passaggio. Le intestazioni devono essere elencate utilizzando la stessa sintassi con cui apparirebbero nel protocollo HTTP, utilizzando facoltativamente alcune funzionalità aggiuntive supportate dall'opzione cURL CURLOPT_HTTPHEADER. Ad esempio: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Le macro utente e le macro `{HOST.*}` sono supportate.
Abilitato	Lo scenario è attivo se questa casella è selezionata; in caso contrario, è disabilitato.

Quando si modifica uno scenario esistente, nel modulo sono disponibili anche due pulsanti aggiuntivi:

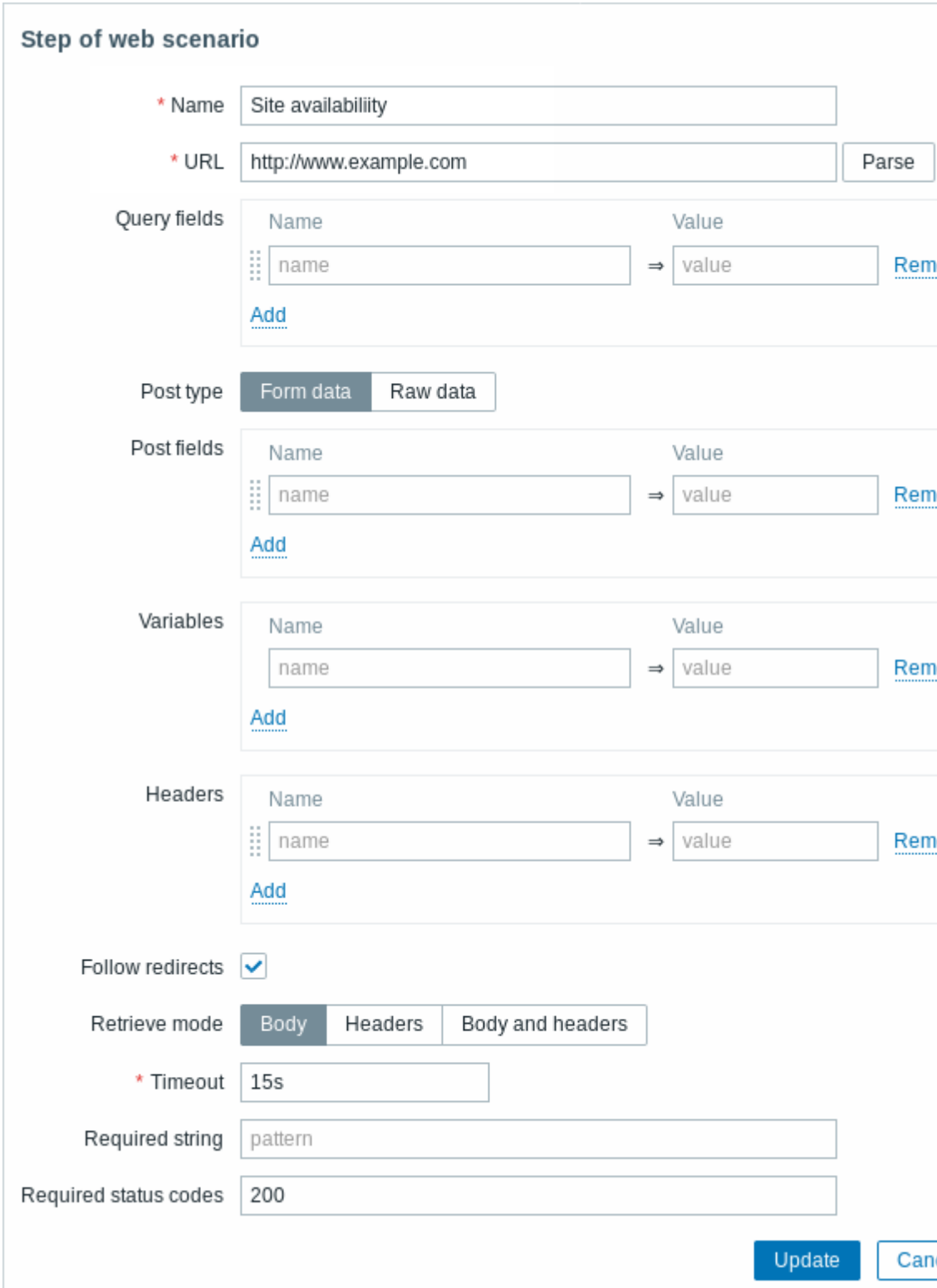
Clone	Crea un altro scenario basato sulle proprietà di quello esistente.
Clear history and trends	Elimina i dati storici e di trend dello scenario. In questo modo il server eseguirà lo scenario immediatamente dopo l'eliminazione dei dati.

Note:
Se il campo *Proxy HTTP* viene lasciato vuoto, un altro modo per utilizzare un proxy HTTP consiste nell'impostare le variabili d'ambiente relative al proxy.
Per i controlli HTTP, imposta la variabile d'ambiente `http_proxy` per l'utente del server Zabbix. Ad esempio, `http_proxy=http://proxy_ip:proxy_port`.
Per i controlli HTTPS, imposta la variabile d'ambiente `HTTPS_PROXY`. Ad esempio, `HTTPS_PROXY=http://proxy_ip:proxy_port`. Ulteriori dettagli sono disponibili eseguendo il comando shell: `# man curl`.

La scheda **Passaggi** consente di configurare i passaggi dello scenario web. Per aggiungere un passaggio allo scenario web, fai clic su *Aggiungi* nel blocco *Passaggi*.



Note:
Le macro utente segrete non devono essere utilizzate negli URL, poiché verranno risolte in "******".



Configurazione dei passaggi

Parametri del passaggio:

Parametro	Descrizione
Nome	Nome univoco del passaggio. Le macro utente sono supportate. *Nota*: se vengono utilizzate macro utente, queste macro rimarranno irrisolte nei nomi degli item di monitoraggio web.

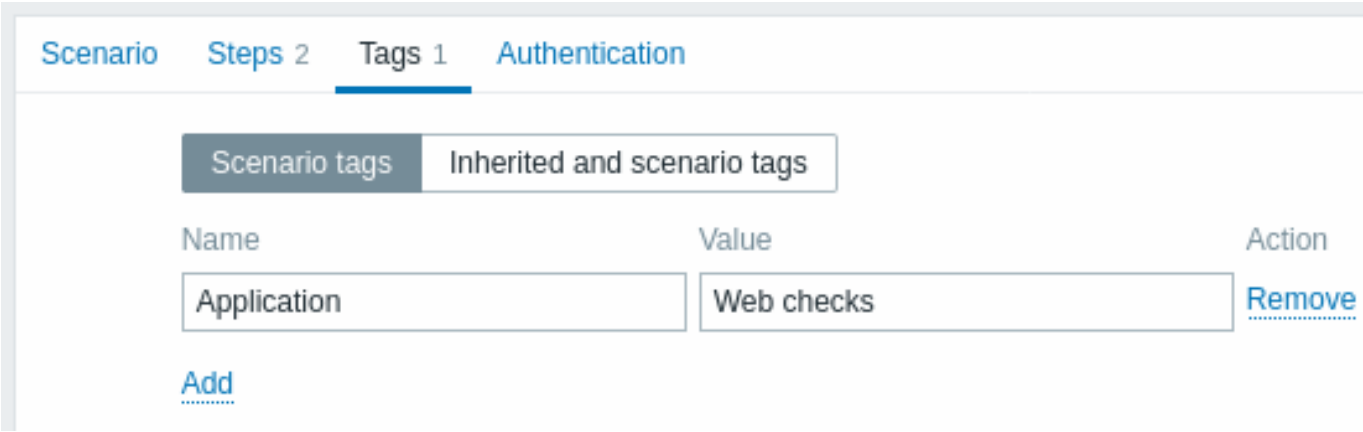
Parametro	Descrizione
URL	URL a cui connettersi e da cui recuperare i dati. Ad esempio: https://www.example.com http://www.example.com/download I nomi di dominio possono essere specificati utilizzando caratteri Unicode. Vengono automaticamente convertiti in punycode ASCII durante l'esecuzione del passaggio dello scenario web. Il pulsante *Analizza* può essere utilizzato per separare i campi di query facoltativi (come `?key=value&limit=10`) dall'URL, spostando attributi e valori in *Campi di query* per la codifica automatica dell'URL. Nell'URL è possibile utilizzare variabili mediante la sintassi {macro}. Le variabili possono essere codificate manualmente nell'URL utilizzando la sintassi {{macro}.urlencode()}. Sono supportate le macro utente e le macro `{HOST.*}`. Per motivi di sicurezza, non includere nomi utente o password reali negli URL. Utilizzare invece le macro utente per evitare di esporre le credenziali. Limitato a 2048 caratteri.
Campi di query	Variabili HTTP GET per l'URL. Specificate come coppie attributo e valore. I valori vengono codificati automaticamente nell'URL. I valori provenienti da variabili dello scenario, macro utente o macro `{HOST.*}` vengono risolti e quindi codificati automaticamente nell'URL. L'utilizzo della sintassi {{macro}.urlencode()} li codificherà due volte nell'URL. Sono supportate le macro utente e le macro `{HOST.*}`.
Post	Variabili HTTP POST. Nella modalità **Dati del modulo**, specificate come coppie attributo e valore. I valori vengono codificati automaticamente nell'URL. I valori provenienti da variabili dello scenario, macro utente o macro `{HOST.*}` vengono risolti e quindi codificati automaticamente nell'URL. Nella modalità **Dati grezzi**, attributi e valori vengono visualizzati su una singola riga e concatenati con il simbolo **&**. I valori grezzi possono essere codificati o decodificati manualmente nell'URL utilizzando la sintassi {{macro}.urlencode()} o {{macro}.urldecode()}. Ad esempio: id=2345&userid={user} Se {user} è definita come variabile dello scenario web, verrà sostituita dal relativo valore durante l'esecuzione del passaggio. Se si desidera codificare la variabile nell'URL, sostituire {user} con {{user}.urlencode()}. Sono supportate le macro utente e le macro `{HOST.*}`.
Variabili	Variabili a livello di passaggio che possono essere utilizzate per le funzioni GET e POST. Specificate come coppie attributo e valore. Le variabili a livello di passaggio hanno la precedenza sulle variabili a livello di scenario o sulle variabili del passaggio precedente. Tuttavia, il valore di una variabile a livello di passaggio influisce solo sul passaggio successivo (e non su quello corrente). Hanno il seguente formato: **{macro}**=value **{macro}**=regex:<regular expression> Per ulteriori informazioni, vedere la descrizione delle variabili a livello di scenario. Le variabili vengono codificate automaticamente nell'URL quando utilizzate nei campi di query o nei dati del modulo per le variabili POST, ma devono essere codificate manualmente nell'URL quando utilizzate nei dati POST grezzi o direttamente nell'URL.
Intestazioni	Intestazioni HTTP personalizzate che verranno inviate durante l'esecuzione di una richiesta. Specificate come coppie attributo e valore. Un'intestazione definita a livello di passaggio verrà utilizzata per quel particolare passaggio. **È importante notare che la definizione dell'intestazione a livello di passaggio elimina automaticamente tutte le intestazioni definite in precedenza, ad eccezione di un'intestazione predefinita assegnata selezionando 'User-Agent' dall'elenco a discesa a livello di scenario.** Tuttavia, anche l'intestazione predefinita 'User-Agent' può essere sovrascritta specificandola a livello di passaggio. Ad esempio, assegnare un nome a un'intestazione senza impostare alcun valore annullerà l'intestazione predefinita a livello di scenario. Sono supportate le macro utente e le macro `{HOST.*}`. Questa impostazione definisce l'opzione cURL CURLOPT_HTTPHEADER.
Segui reindirizzamenti	Selezionare la casella di controllo per seguire i reindirizzamenti HTTP. Questa impostazione definisce l'opzione cURL CURLOPT_FOLLOWLOCATION.

Parametro	Descrizione
Modalità di recupero	Selezionare la modalità di recupero: **Corpo** - recupera solo il corpo della risposta HTTP **Intestazioni** - recupera solo le intestazioni della risposta HTTP **Corpo e intestazioni** - recupera il corpo e le intestazioni della risposta HTTP
Timeout	Zabbix non dedicherà all'elaborazione dell'URL più del tempo impostato (da un secondo a un massimo di 1 ora). In realtà, questo parametro definisce il tempo massimo per stabilire la connessione all'URL e il tempo massimo per eseguire una richiesta HTTP. Pertanto, Zabbix non dedicherà al passaggio più di **2 x Timeout** secondi. Sono supportati i suffissi temporali, ad esempio 30s, 1m, 1h. Sono supportate le macro utente.
Stringa richiesta	Pattern di espressione regolare richiesto. Se il contenuto recuperato (HTML) non corrisponde al pattern richiesto, il passaggio avrà esito negativo. Se vuoto, non viene eseguito alcun controllo sulla stringa richiesta. Ad esempio: Homepage of Zabbix Welcome.*admin *Nota*: in questo campo non è supportato il riferimento alle espressioni regolari create nel frontend di Zabbix. Sono supportate le macro utente e le macro `{HOST.*}`.
Codici di stato richiesti	Elenco dei codici di stato HTTP previsti. Se Zabbix riceve un codice non incluso nell'elenco, il passaggio avrà esito negativo. Se vuoto, non viene eseguito alcun controllo sui codici di stato. Ad esempio: 200,201,210-299 Sono supportate le macro utente.

Note:
Le modifiche ai passaggi dello scenario web verranno salvate solo quando viene salvato l'intero scenario.

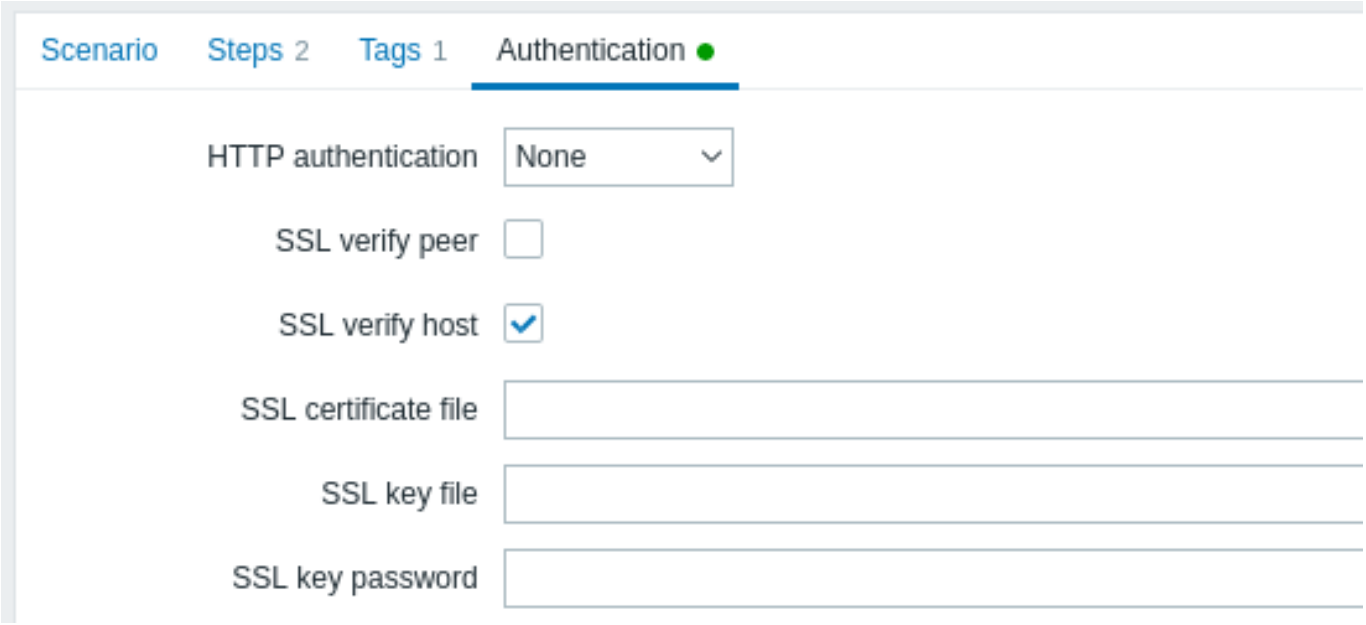
Vedere anche un esempio reale di come configurare i passaggi del monitoraggio web.

Configurazione dei tag La scheda **Tag** consente di definire i tag a livello di scenario.



L'uso dei tag consente di filtrare gli scenari web e gli item di monitoraggio web.

Configurazione dell'autenticazione La scheda **Autenticazione** consente di configurare le opzioni di autenticazione dello scenario. Un punto verde accanto al nome della scheda indica che è abilitato un tipo di autenticazione HTTP.



Parametri di autenticazione:

Parametro	Descrizione
Autenticazione HTTP	Selezionare l'opzione di autenticazione: **Nessuna** - non viene utilizzata alcuna autenticazione; **Basic** - viene utilizzata l'autenticazione di base; **NTLM** - viene utilizzata l'autenticazione NTLM (Windows NT LAN Manager); **Kerberos** - viene utilizzata l'autenticazione Kerberos (vedere anche: Configurazione di Kerberos con Zabbix); **Digest** - viene utilizzata l'autenticazione Digest.
Utente	Inserire il nome utente (fino a 255 caratteri). Questo campo è disponibile se *Autenticazione HTTP* è impostata su Basic, NTLM, Kerberos o Digest. Sono supportate le macro utente.
Password	Inserire la password dell'utente (fino a 255 caratteri). Questo campo è disponibile se *Autenticazione HTTP* è impostata su Basic, NTLM, Kerberos o Digest. Sono supportate le macro utente.
Verifica peer SSL	Selezionare la casella per verificare il certificato SSL del web server. Il certificato del server viene verificato rispetto ai certificati delle autorità di certificazione (CA) attendibili nella posizione CA a livello di sistema. È possibile sovrascrivere la posizione dei file CA utilizzando il parametro di configurazione del server Zabbix o del proxy SSLCALocation. Imposta l'opzione cURL CURLOPT_SSL_VERIFYPEER.
Verifica host SSL	Selezionare la casella per verificare che il campo *Common Name* o il campo *Subject Alternate Name* del certificato del web server corrisponda. Imposta l'opzione cURL CURLOPT_SSL_VERIFYHOST.
File del certificato SSL	Nome del file del certificato SSL utilizzato per l'autenticazione del client. Il file del certificato deve essere in formato PEM[1]. Se il file del certificato contiene anche la chiave privata, lasciare vuoto il campo *File della chiave SSL*. Se la chiave è crittografata, specificare la password nel campo *Password della chiave SSL*. La directory contenente questo file è specificata dal parametro di configurazione del server Zabbix o del proxy SSLCertLocation. In questo campo è possibile utilizzare le macro `HOST.*` e le macro utente. Imposta l'opzione cURL CURLOPT_SSLCERT.
File della chiave SSL	Nome del file della chiave privata SSL utilizzato per l'autenticazione del client. Il file della chiave privata deve essere in formato PEM[1]. La directory contenente questo file è specificata dal parametro di configurazione del server Zabbix o del proxy SSLKeyLocation. In questo campo è possibile utilizzare le macro `HOST.*` e le macro utente. Imposta l'opzione cURL CURLOPT_SSLKEY.
Password della chiave SSL	Password del file della chiave privata SSL. In questo campo è possibile utilizzare le macro utente. Imposta l'opzione cURL CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix supporta solo i file di certificato e di chiave privata in formato PEM. Se i dati del certificato e della chiave privata si trovano in un file in formato PKCS #12 (solitamente con estensione *.p12 o *.pfx), è possibile generare il file PEM utilizzando i seguenti comandi:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

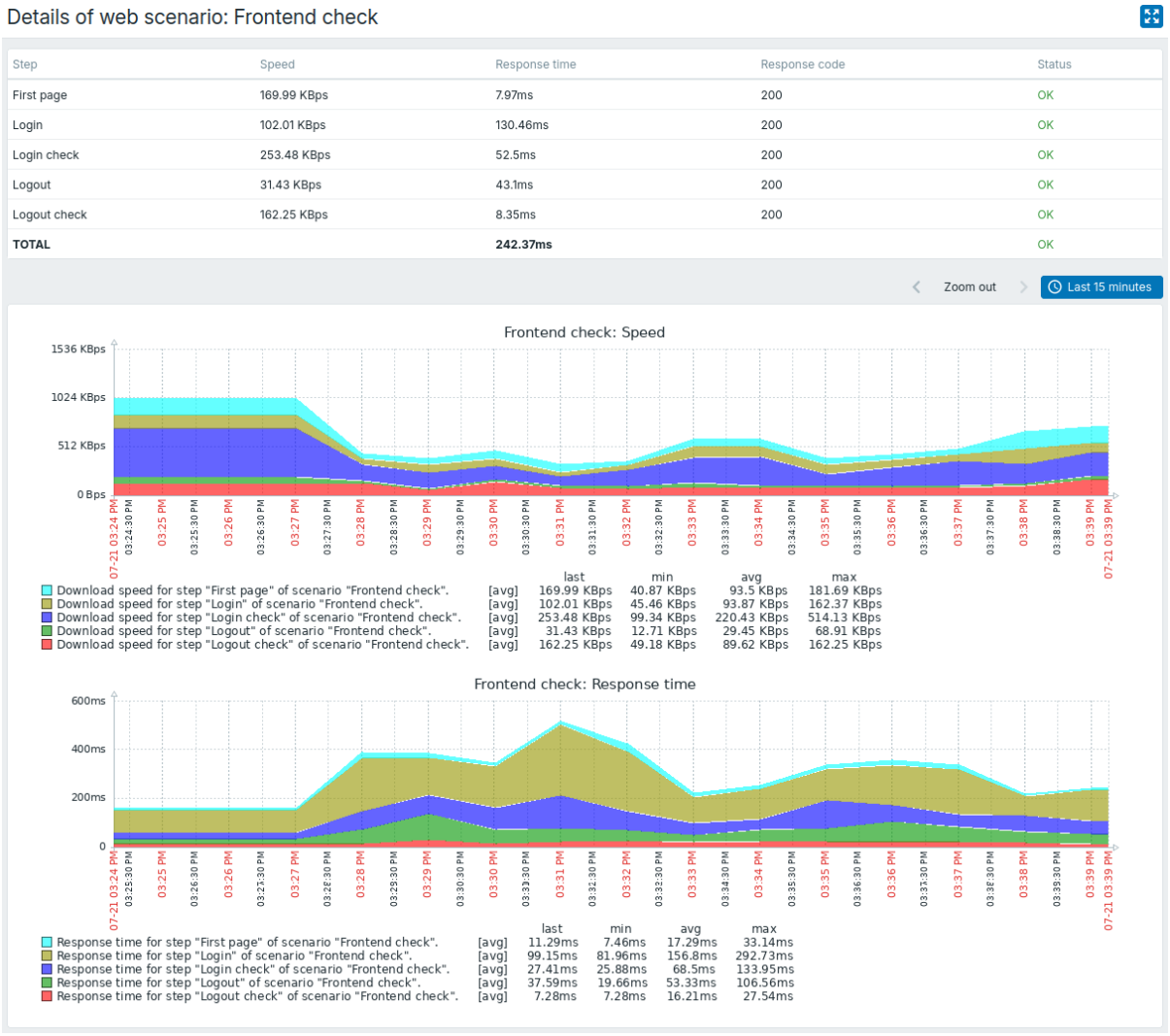
Note:
Il server Zabbix rileva le modifiche ai certificati senza riavviarsi.

Note:
Se il certificato client e la chiave privata si trovano in un unico file, specificarlo nel campo "File del certificato SSL" e lasciare vuoto il campo "File della chiave SSL". Il certificato e la chiave devono comunque essere in formato PEM. È facile combinare certificato e chiave:

```
cat client.crt client.key > client.pem
```

Visualizzazione Per visualizzare gli scenari web configurati per un host:

1. Vai a *Monitoraggio* > *Host*.
2. Individua l'host nell'elenco e fai clic sul collegamento ipertestuale *Web* nell'ultima colonna.
3. Fai clic sul nome dello scenario per visualizzare informazioni dettagliate.



Una panoramica degli scenari web può essere visualizzata anche in *Dashboard* tramite il widget Monitoraggio web.

I risultati recenti dell'esecuzione dello scenario web sono disponibili nella sezione *Monitoraggio* > *Ultimi dati*.

Monitoraggio esteso A volte è necessario registrare il contenuto della pagina HTML ricevuta.
Questo è particolarmente utile se un passaggio dello scenario web fallisce.

Il livello di debug 5 (trace) serve a questo scopo.
Questo livello può essere impostato nei file di configurazione di server e proxy oppure usando un'opzione di runtime control (`-R log_level_increase="http poller,N"`, dove N è il numero del processo).
I seguenti esempi mostrano come avviare il monitoraggio esteso, a condizione che il livello di debug 4 sia già impostato:

```
# Increase log level of all http pollers:
zabbix_server -R log_level_increase="http poller"

# Increase log level of second http poller:
zabbix_server -R log_level_increase="http poller,2"
```

Se il monitoraggio web esteso non è richiesto, può essere interrotto usando l'opzione `-R log_level_decrease`.

Elementi di monitoraggio web

Panoramica

Alcuni nuovi elementi vengono aggiunti automaticamente per il monitoraggio quando gli scenari web sono creati.

Tutti gli elementi ereditano i tag dallo scenario Web.

Item dello scenario

Non appena viene creato uno scenario, Zabbix aggiunge automaticamente i seguenti item per il monitoraggio.

Item	Descrizione
Velocità di download per lo scenario <Scenario>	Questo item raccoglie informazioni sulla velocità di download (byte al secondo) dell'intero scenario, cioè la media di tutti gli step. Chiave item: web.test.in[Scenario,,bps] Tipo: *Numeric(float)*
Step non riuscito dello scenario <Scenario>	Questo item visualizza il numero dello step che non è riuscito nello scenario. Se tutti gli step vengono eseguiti correttamente, viene restituito 0. Chiave item: web.test.fail[Scenario] Tipo: *Numeric(unsigned)*
Ultimo messaggio di errore dello scenario <Scenario>	Questo item restituisce il testo dell'ultimo messaggio di errore dello scenario. Un nuovo valore viene memorizzato solo se lo scenario ha uno step non riuscito. Se tutti gli step sono corretti, non viene raccolto alcun nuovo valore. Chiave item: web.test.error[Scenario] Tipo: *Character*

Al posto di "Scenario" verrà utilizzato il nome effettivo dello scenario.

> **Note:**
> Se il nome dello scenario contiene macro utente, queste macro verranno lasciate non risolte nei nomi degli item di monitoraggio web.

 Se il nome dello scenario inizia con un doppio apice o contiene una virgola o una parentesi quadra, verrà correttamente racchiuso tra virgolette nelle chiavi degli item. Negli altri casi non verrà eseguita alcuna quotatura aggiuntiva.

> **Note:**
> Gli item di monitoraggio web vengono aggiunti con uno storico di 30 giorni e un periodo di conservazione dei trend di 90 giorni.

Questi item possono essere utilizzati per creare trigger e definire condizioni di notifica.

Esempio 1

Per creare un trigger "Scenario Web non riuscito", è possibile definire un espressione trigger:

```
last(/host/web.test.fail[Scenario])<>0
```

Assicurati di sostituire "Scenario" con il vero nome del tuo scenario.

Esempio 2

Per creare un trigger ”Scenario Web non riuscito” con un problema utile nella descrizione nel nome del trigger, è possibile definire un trigger con il nome:

```
Scenario web "Scenario" non riuscito: {ITEM.VALUE}
```

ed espressione trigger:

```
length(last(/host/web.test.error[Scenario]))>0 e last(/host/web.test.fail[Scenario])>0
```

Assicurati di sostituire ”Scenario” con il vero nome del tuo scenario.

Esempio 3

Per creare un trigger ”L’applicazione Web è lenta”, puoi definire un espressione trigger:

```
last(/host/web.test.in[Scenario,,bps])<10000
```

Assicurati di sostituire ”Scenario” con il vero nome del tuo scenario.

Elementi Passi dello scenario

Non appena viene creato un passaggio, Zabbix aggiunge automaticamente quanto segue elementi per il monitoraggio.

Articolo	Descrizione
Velocità di download per il passaggio <Passaggio> dello scenario <Scenario>	Questo elemento raccoglierà informazioni sulla velocità di download (byte al secondo) del passaggio. Chiave oggetto: web.test.in[Scenario,Passaggio ,bps] Tipo: *Numeric(float)*
Tempo di risposta per il passaggio <Passo> dello scenario <Scenario>	Questo elemento raccoglierà informazioni sul tempo di risposta del passaggio in secondi. Il tempo di risposta viene conteggiato dall’inizio della richiesta fino al trasferimento di tutte le informazioni. Chiave elemento: web.test.time[Scenario,Step,resp] Tipo: *Numeric(float)*
Codice di risposta per il passaggio <Passaggio> dello scenario <Scenario>	Questo elemento raccoglierà i codici di risposta del passaggio. Chiave oggetto: web.test.rspcode[Scenario,Passaggio] Tipo: * Numerico(senza segno)*

Al posto di ”Scenario” e verranno utilizzati i nomi degli scenari e dei passaggi effettivi rispettivamente ”Passo”.

> **Note:**
> Gli elementi di monitoraggio Web vengono aggiunti con una cronologia di 30 giorni e a Periodo di conservazione del trend di 90 giorni.

> **Note:**
> Se il nome dello scenario inizia con virgolette o contiene una virgola o parentesi quadre, sarà opportunamente quotato nelle chiavi dell’oggetto. In altro casi non verranno eseguiti preventivi aggiuntivi.

Questi elementi possono essere utilizzati per creare trigger e definire la notifica condizioni. Ad esempio, per creare un ”accesso alla GUI Zabbix è troppo lento” trigger, è possibile definire un’espressione trigger:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

Scenario reale

Panoramica

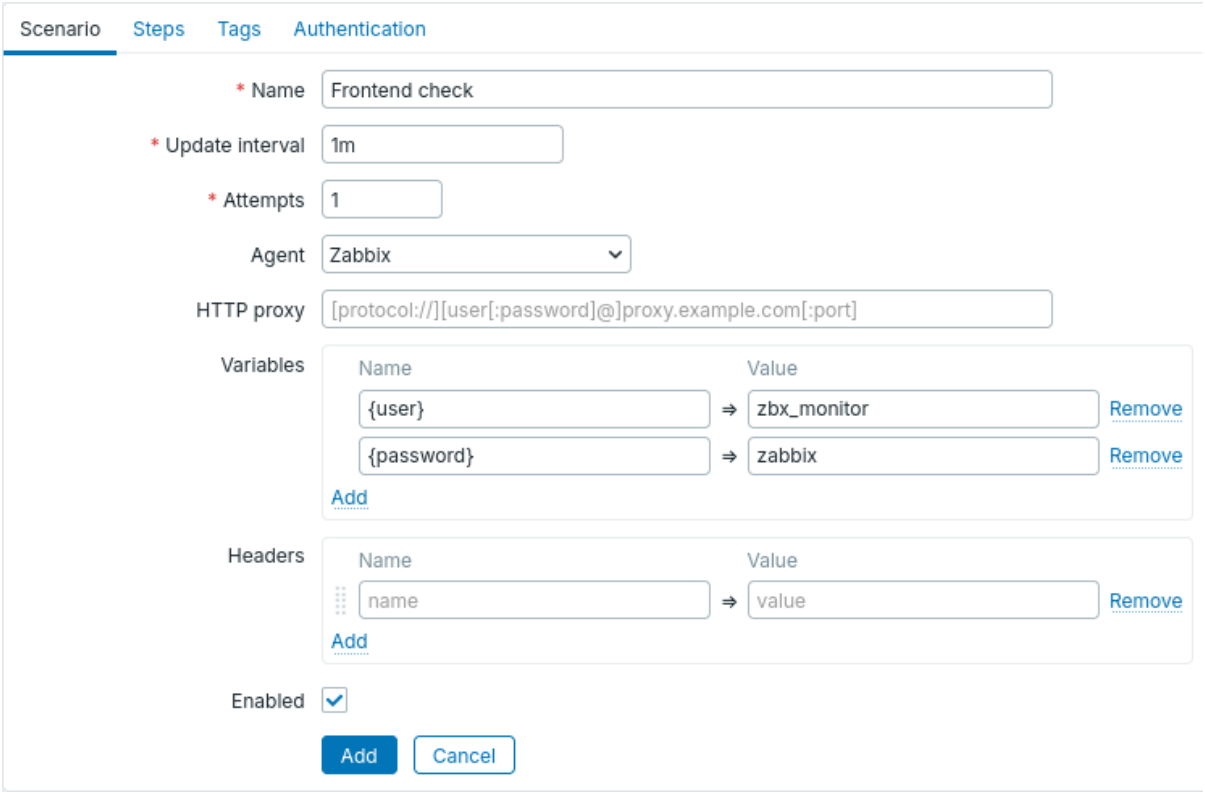
Questa sezione presenta un esempio pratico e realistico, passo dopo passo, di come può essere utilizzato il monitoraggio web.

Il monitoraggio web di Zabbix verrà utilizzato per monitorare il frontend di Zabbix. L’obiettivo è determinare se è disponibile, se fornisce il contenuto corretto e con quale rapidità funziona. Per raggiungere questo obiettivo, sono necessari diversi passaggi, tra cui verificare la disponibilità della prima pagina, effettuare l’accesso con nome utente e password, verificare il successo dell’accesso, effettuare il logout e confermare il logout.

Scenario

Aggiungere un nuovo scenario web

1. Vai a *Data collection* > *Hosts*.
2. Scegli un host e fai clic su *Web* nella riga di quell'host.
3. Fai clic su *Create web scenario*.



4. Nel nuovo modulo dello scenario, nella scheda *Scenario*, compila i seguenti campi:
    - *Name*: Frontend check.
    - *Update interval*: 1m.
    - *Attempts*: 1.
    - *Agent*: Zabbix.
 Nella sezione *Variables*, aggiungi le seguenti coppie nome-valore:
    - `{user}`: le credenziali utente Zabbix esistenti;
    - `{password}`: le credenziali utente Zabbix esistenti.

Note:
Per motivi di sicurezza, si consiglia di creare un utente separato con privilegi minimi da usare per le attività di monitoraggio.

Facoltativamente, passa alla scheda *Tags* e aggiungi i tag dello scenario web. Puoi usare i tag dello scenario web per identificare rapidamente gli item e i trigger correlati oppure per cercare tra i dati raccolti. Ad esempio, i tag adatti per questo tutorial sono `component: web-scenario` e/o `target: frontend`.

Configurare i passaggi dello scenario web

Passa alla scheda *Steps* e definisci i passaggi per lo scenario.
 Dopo aver configurato ogni passaggio, fai clic sul pulsante *Add*.

Attention:
Tieni presente che il frontend di Zabbix usa un reindirizzamento JavaScript durante l'accesso, quindi il login deve avvenire per primo e le funzionalità disponibili dopo l'accesso possono essere verificate solo nei passaggi successivi. Inoltre, il passaggio di login deve usare l'URL completo del file **index.php**.

Campi comuni a tutti i passaggi

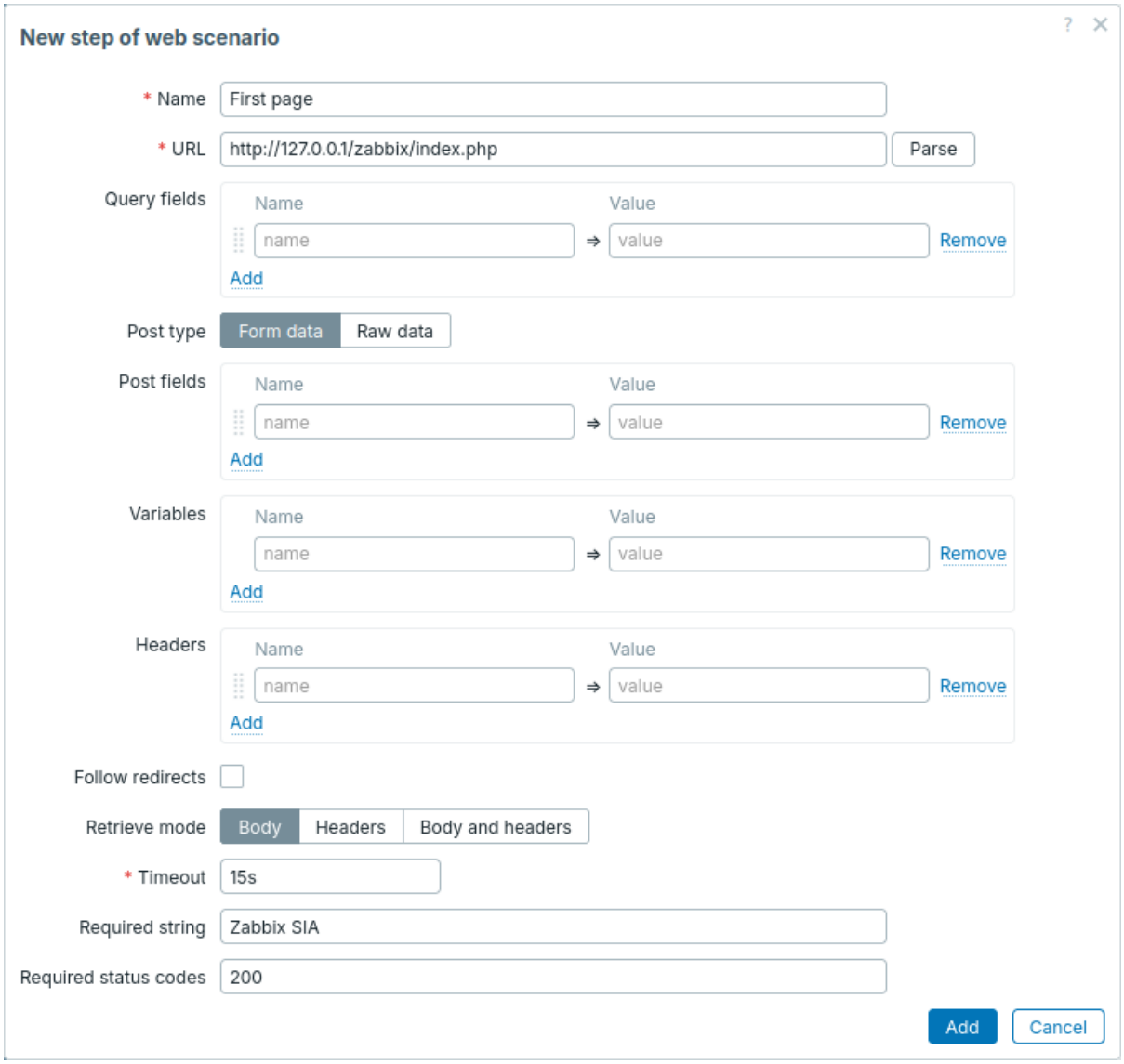
Oltre ai campi specifici del passaggio, compilare i seguenti campi comuni per ogni passaggio descritto di seguito:

- *URL*: l'URL del frontend Zabbix (`http://<server_ip_or_name>/zabbix/index.php` (per Apache) o `http://<server_ip_or_na`
  (per NGINX)). Per il passaggio 3 dello scenario Web, utilizzare un URL diverso, come descritto di seguito.
- *Timeout*: 15s.
- *Codici di stato richiesti*: 200.

Passo 1 dello scenario web

Verificare che la prima pagina risponda correttamente, restituisca il codice di stato HTTP 200 e contenga il testo *Zabbix SIA*.
 Compilare i seguenti campi:

- *Name*: Prima pagina.
- *Timeout*: 15s.
- *Required string*: Zabbix SIA.
- I campi comuni per tutti i passi.

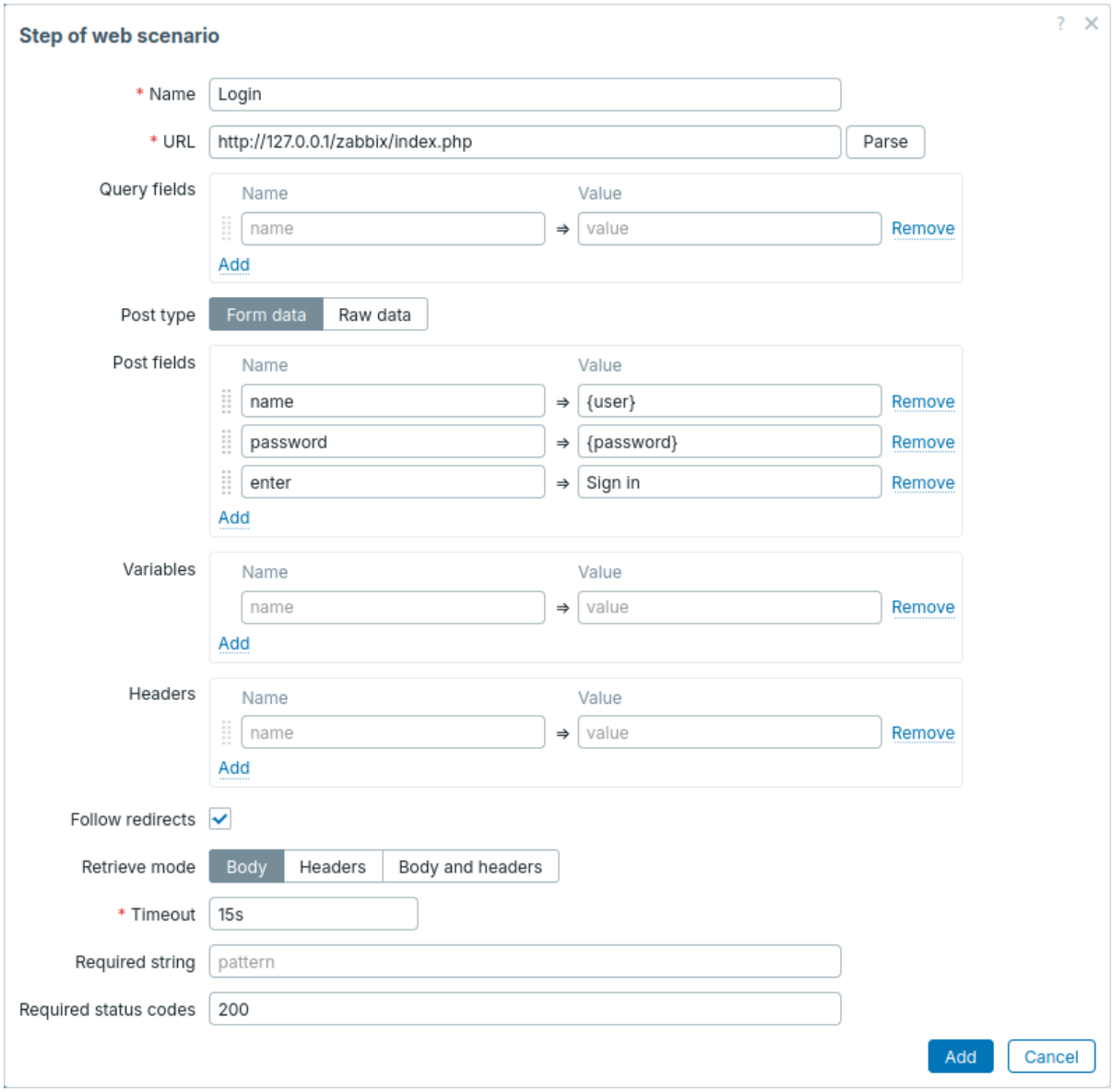


Passo 2 dello scenario web

Accedi al frontend di Zabbix utilizzando le macro (variabili) definite nella scheda *Scenario*: *{user}* e *{password}*.
 Compila i seguenti campi:

- *Name*: Login.
- Nella sezione *Post fields*, aggiungi le seguenti coppie nome-valore:
  - name: `{user}`;
  - password: `{password}`;
  - enter: Sign in.
- I campi comuni per tutti i passaggi.

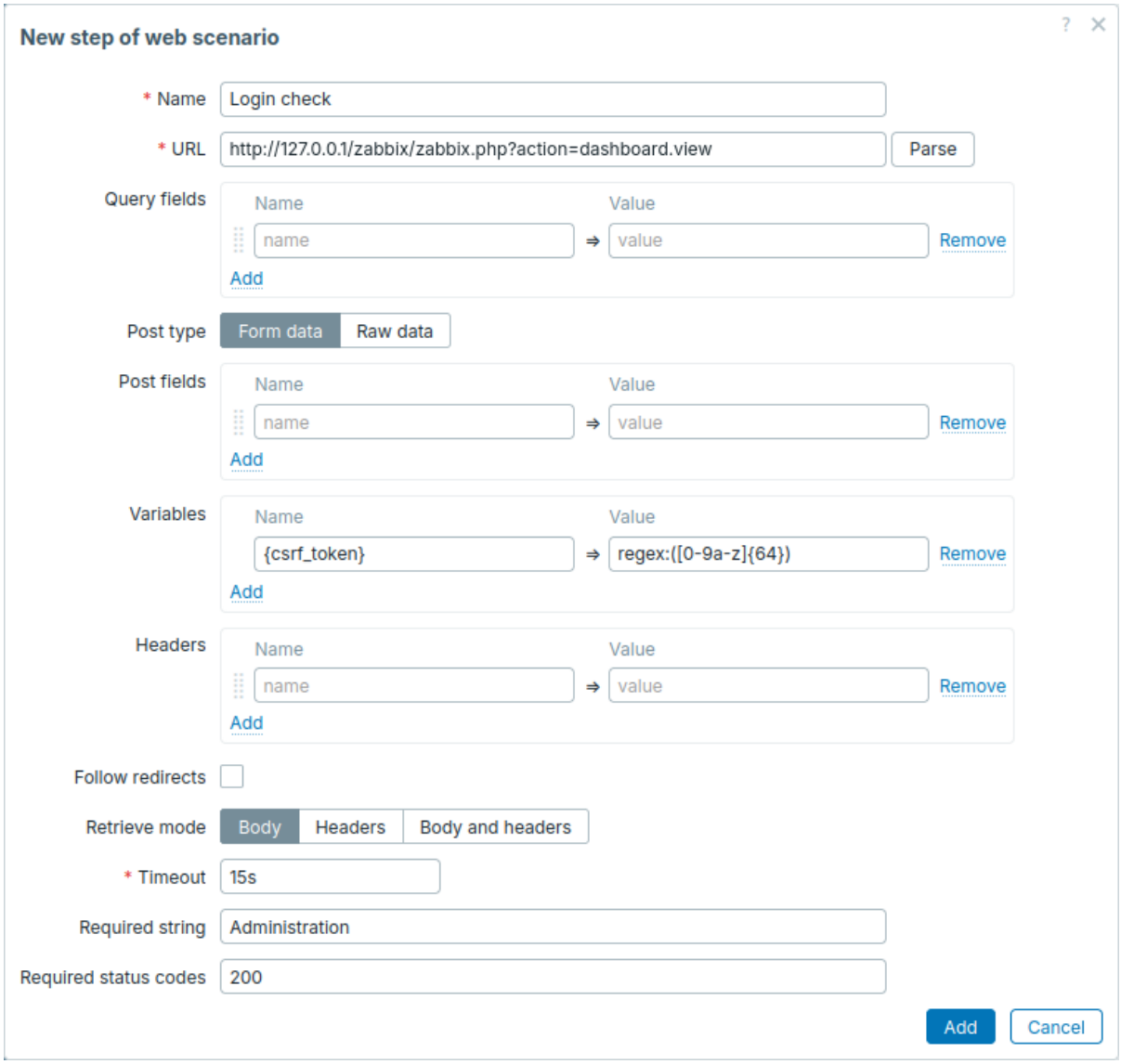
Seleziona la casella di controllo *Follow redirects*.



Passo 3 dello scenario web

Dopo aver effettuato l'accesso, verifica il successo controllando una stringa visibile solo quando si è autenticati, ad esempio *Administration*.
 Compila i seguenti campi:

- *Name*: Login check.
- *URL*: l'URL a cui l'utente viene reindirizzato dopo il login (ad esempio, http://127.0.0.1/zabbix/zabbix.php?action=dashboard.view).
- Nella sezione *Variables*, aggiungi la coppia nome-valore `{csrf_token}: regex:([0-9a-z]{64})`. Questa variabile acquisirà il valore del token CSRF assegnato da riutilizzare nel passo 4 dello scenario web.
- *Required string*: Administration.
- I campi comuni per tutti i passi.



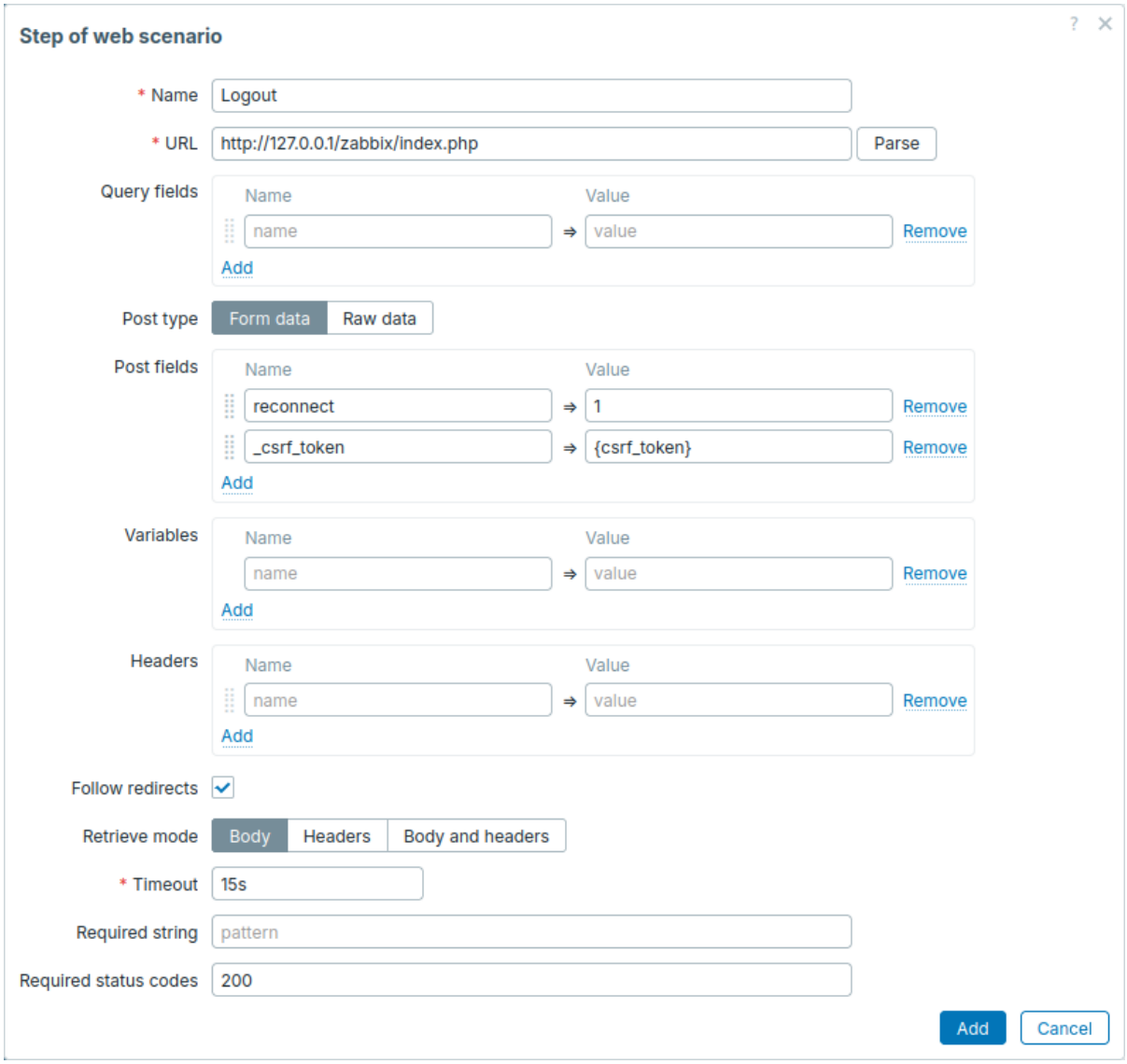
Passaggio 4 dello scenario web

Dopo aver verificato l'accessibilità e il login del frontend, aggiungere un passaggio di logout; in caso contrario, il database di Zabbix si riempirà di numerosi record di sessione aperti.
 Compilare i seguenti campi:

- *Nome*: Logout.
- Nella sezione *Campi POST*, aggiungere le seguenti coppie nome-valore:
  - reconnect: 1;
  - _csrf_token: `{csrf_token}`.
- I campi comuni a tutti i passaggi.

Selezionare la casella di controllo *Segui reindirizzamenti*.

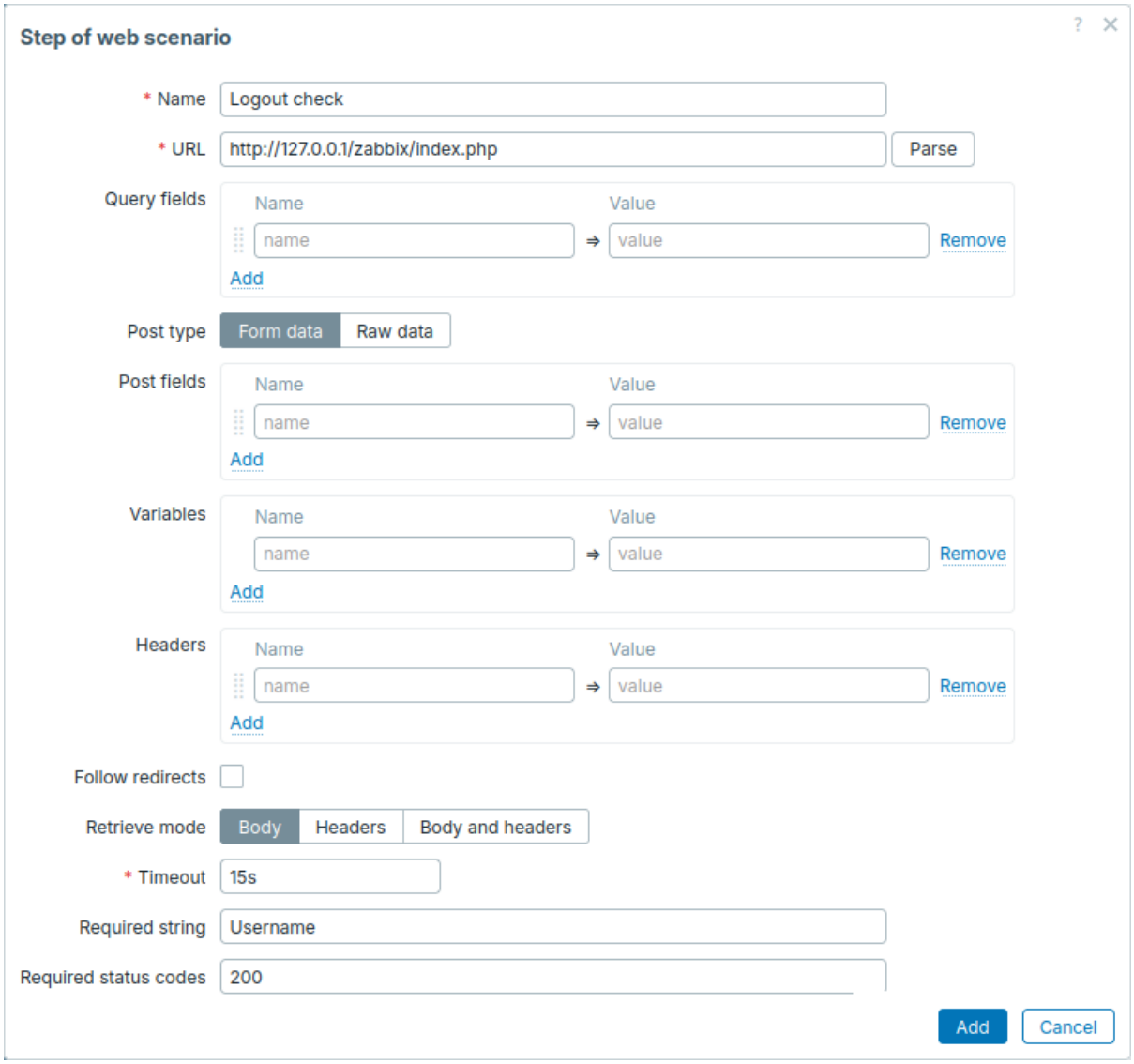
Questo passaggio utilizza la variabile `{csrf_token}` ottenuta nel passaggio 3 dello scenario web.



Passo 5 dello scenario web

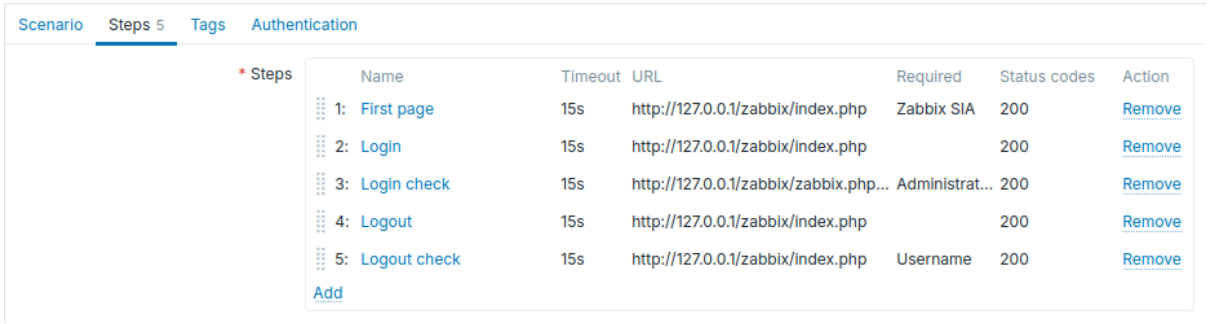
Per confermare il logout, controlla la stringa *Username*.
 Compila i seguenti campi:

- *Name*: Controllo logout.
- *Required string*: Username.
- I campi comuni per tutti i passaggi.



Configurazione completa dei passaggi

Una configurazione completa dei passaggi dello scenario web dovrebbe apparire così:

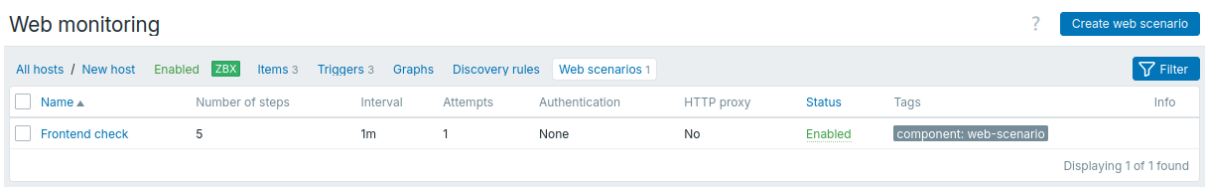


Controlla i risultati

Salva lo scenario di web monitoring completato.

Quando lo scenario web viene creato, vengono aggiunti automaticamente nuovi item di web monitoring per il monitoraggio.

Lo scenario verrà aggiunto all'host. Per visualizzare le informazioni sullo scenario web, vai su *Monitoring* > *Hosts*, individua l'host nell'elenco e fai clic sul collegamento ipertestuale *Web* nell'ultima colonna.



Fai clic sul nome dello scenario per vedere le statistiche dettagliate:



Monitoraggio dei servizi

Panoramica Il monitoraggio dei servizi è un monitoraggio a livello aziendale che può essere utilizzato per ottenere una panoramica dell'intero albero dei servizi dell'infrastruttura IT, identificare i punti deboli dell'infrastruttura, calcolare lo SLA di vari servizi IT e verificare altre informazioni a un livello superiore. Il monitoraggio dei servizi si concentra sulla disponibilità complessiva di un servizio invece che sui dettagli di basso livello, come la mancanza di spazio su disco, un carico elevato del processore e così via. Il monitoraggio dei servizi fornisce inoltre funzionalità per individuare la causa principale di un problema se un servizio non funziona come previsto.

Il monitoraggio dei servizi consente di creare una rappresentazione gerarchica dei dati monitorati.

Una struttura di servizi molto semplice può essere simile alla seguente:

```
Servizio
|
|-Workstation
| |
| |-Workstation1
| |
| |-Workstation2
|
|-Server
```

Ogni nodo della struttura ha un attributo di stato. Lo stato viene calcolato e propagato ai livelli superiori in base all'algoritmo selezionato. Lo stato dei singoli nodi è influenzato dallo stato dei problemi mappati. La mappatura dei problemi viene eseguita tramite il tagging.

Zabbix può inviare notifiche o eseguire automaticamente uno script sul server Zabbix nel caso venga rilevata una modifica dello stato del servizio. È possibile definire regole flessibili per stabilire se un servizio padre debba entrare nello "Stato problema" in base agli stati dei servizi figlio. I dati relativi ai problemi dei servizi possono quindi essere utilizzati per calcolare lo SLA e inviare report SLA sulla base di un insieme flessibile di condizioni.

Il monitoraggio dei servizi viene configurato nel menu Servizi, che comprende le seguenti sezioni:

- Servizi

La sezione Servizi consente di creare una gerarchia dell'infrastruttura monitorata aggiungendo servizi padre e quindi servizi figlio ai servizi padre.

Oltre a configurare l'albero dei servizi, questa sezione fornisce una panoramica dell'intera infrastruttura e consente di identificare rapidamente i problemi che hanno causato una modifica dello stato del servizio.

- SLA

In questa sezione è possibile definire gli accordi sul livello di servizio e impostare gli obiettivi del livello di servizio per servizi specifici.

- Report SLA

In questa sezione è possibile visualizzare i report SLA.

Azioni dei servizi

È inoltre possibile configurare le azioni dei servizi.

Le azioni dei servizi sono facoltative e consentono di:

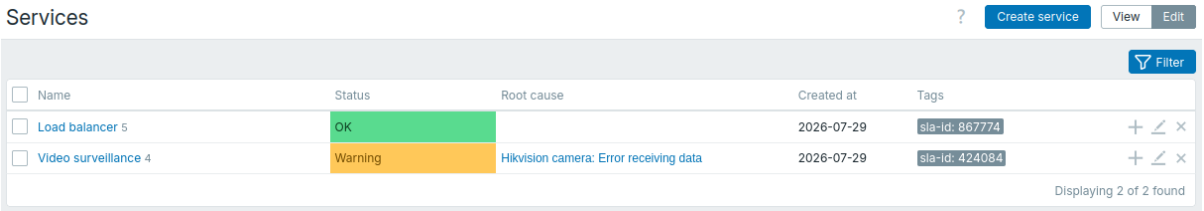
- inviare una notifica quando un servizio non è disponibile
- eseguire un comando remoto sul server Zabbix in seguito a una modifica dello stato del servizio
- inviare una notifica di ripristino quando un servizio torna disponibile.

Vedere anche:

- Esempio di configurazione del monitoraggio SLA
- Note sull'aggiornamento dei servizi dalle versioni di Zabbix precedenti alla 6.0

Albero dei servizi

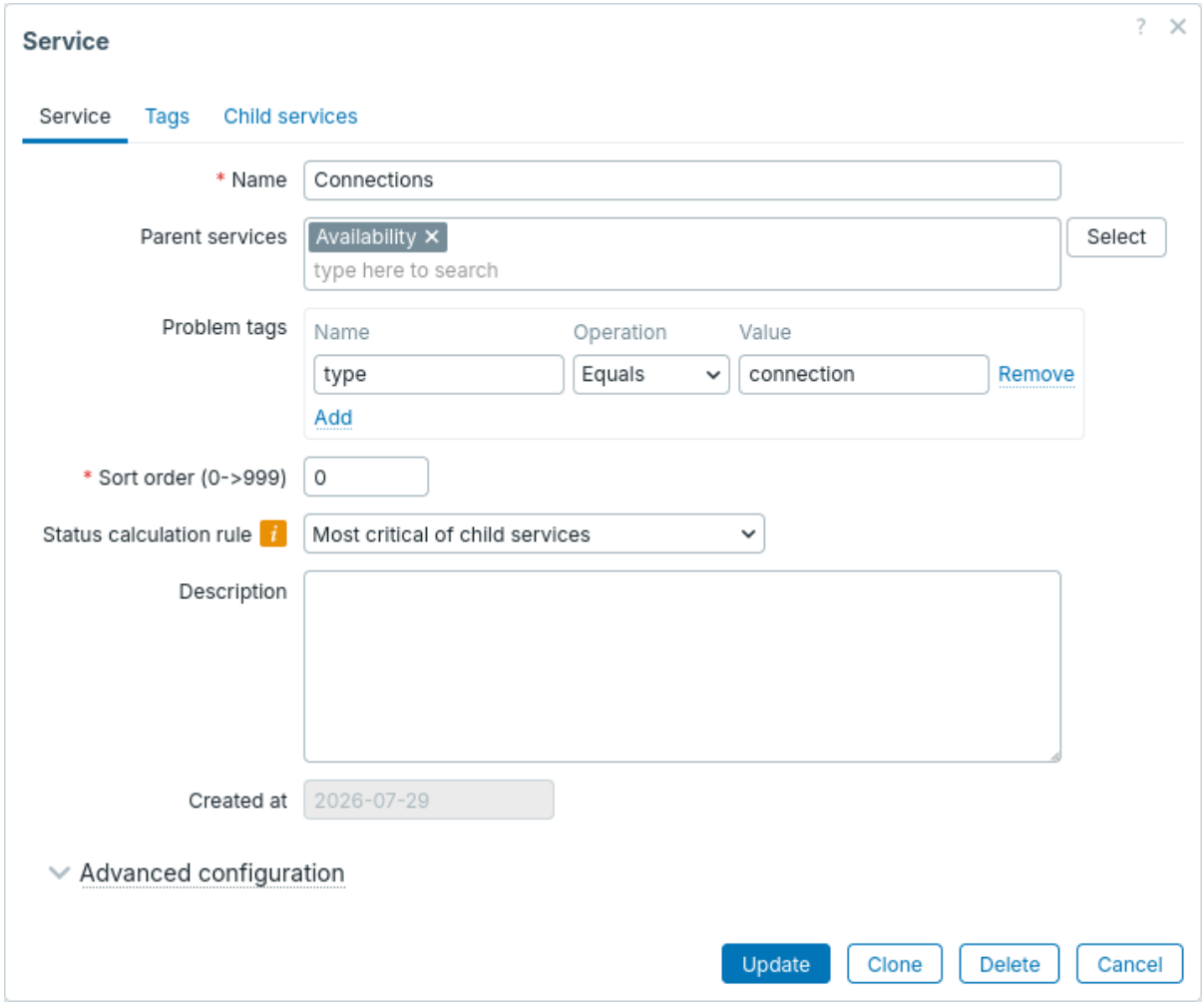
L'albero dei servizi viene configurato nella sezione di menu *Services* > *Services*. Nell'angolo in alto a destra, passa dalla modalità view alla modalità di modifica.



Per configurare un nuovo servizio, fai clic sul pulsante *Create service* nell'angolo in alto a destra.

Per aggiungere rapidamente un servizio figlio, puoi in alternativa fare clic sull'icona più accanto al servizio padre. Questo aprirà lo stesso modulo di configurazione del servizio, ma il parametro *Parent services* sarà precompilato.

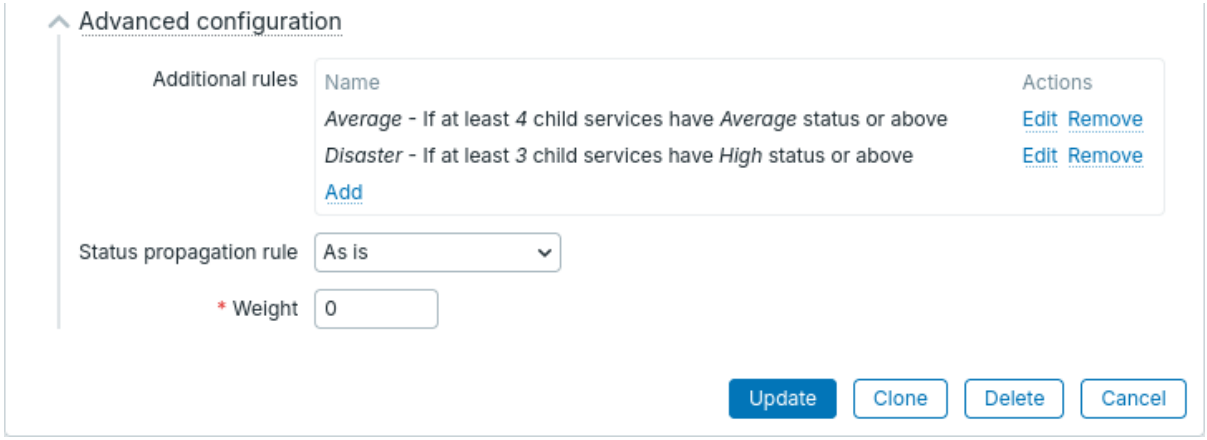
Configurazione del service Nella scheda **Service**, specificare i parametri richiesti del service:



Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

Parameter	Description
Name	Nome del service.
Parent services	Service padre a cui appartiene il service. Lasciare questo campo vuoto se si sta aggiungendo un service di livello più alto. Un service può avere più service padre. In questo caso, verrà visualizzato nell'albero dei service sotto ciascuno dei service padre.
Problem tags	Specificare i tag per associare i dati dei problemi al service. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole/minuscole. Per ogni condizione sono disponibili due operatori: **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, sensibile alle maiuscole/minuscole)
Sort order	Ordine di visualizzazione; il valore più basso viene mostrato per primo.
Status calculation rule	Regola per il calcolo dello stato del service: **Most critical if all children have problems** - il problema più grave nei service figlio viene usato per colorare lo stato del service, se tutti i service figlio hanno problemi **Most critical of child services** - il problema più grave nei service figlio viene usato per colorare lo stato del service **Set status to OK** - non calcolare lo stato del service Ulteriori regole di calcolo dello stato possono essere configurate nelle opzioni di advanced configuration.
Description	Descrizione del service.
Created at	L'ora in cui il service è stato creato; visualizzata durante la modifica di un service esistente.
Advanced configuration	Fare clic sull'intestazione *Advanced configuration* per visualizzare le opzioni di advanced configuration.

Configurazione avanzata



Parametro		Descrizione
Regole aggiuntive		Fai clic su *Aggiungi* per configurare regole aggiuntive di calcolo dello stato.
	Imposta stato su	Imposta lo stato del servizio su *OK* (predefinito), *Not classified*, *Information*, *Warning*, *Average*, *High* o *Disaster* in caso di corrispondenza con una condizione.
	Condizione	Seleziona la condizione per i servizi figlio diretti: **se almeno (N) servizi figlio hanno stato (Status) o superiore** **se almeno (N%) dei servizi figlio hanno stato (Status) o superiore** **se meno di (N) servizi figlio hanno stato (Status) o inferiore** **se meno di (N%) dei servizi figlio hanno stato (Status) o inferiore** **se il peso dei servizi figlio con stato (Status) o superiore è almeno (W)** **se il peso dei servizi figlio con stato (Status) o superiore è almeno (N%)** **se il peso dei servizi figlio con stato (Status) o inferiore è inferiore a (W)** **se il peso dei servizi figlio con stato (Status) o inferiore è inferiore a (N%)** Se vengono specificate più condizioni e la situazione corrisponde a più di una condizione, verrà impostata la gravità più alta.
	N (W)	Imposta il valore di N o W (1-100000), oppure N% (1-100) nella condizione.
	Status	Seleziona il valore di *Status* nella condizione: *OK* (predefinito), *Not classified*, *Information*, *Warning*, *Average*, *High* o *Disaster*.
Regola di propagazione dello stato		Regola per propagare lo stato del servizio al servizio padre: **As is** - lo stato viene propagato senza modifiche **Increase by** - puoi aumentare lo stato propagato da 1 a 5 livelli di gravità **Decrease by** - puoi diminuire lo stato propagato da 1 a 5 livelli di gravità **Ignore this service** - lo stato non viene propagato affatto al servizio padre **Fixed status** - lo stato viene propagato in modo statico, cioè sempre uguale
Peso		Peso del servizio (intero nell'intervallo da 0 (predefinito) a 1000000).

Note:
Le regole aggiuntive di calcolo dello stato possono essere usate solo per aumentare il livello di gravità rispetto al livello calcolato in base al parametro principale *Status calculation rule*. Se, in base alle regole aggiuntive, lo stato dovrebbe essere Warning, ma in base a *Status calculation rule* lo stato è Disaster, il servizio avrà stato Disaster.

La scheda *Tag* contiene tag a livello di servizio. I tag a livello di servizio vengono utilizzati per identificare un servizio. I tag di questo tipo non vengono utilizzati per associare i problemi al servizio (a questo scopo, utilizzare i *tag dei problemi* nella prima scheda).

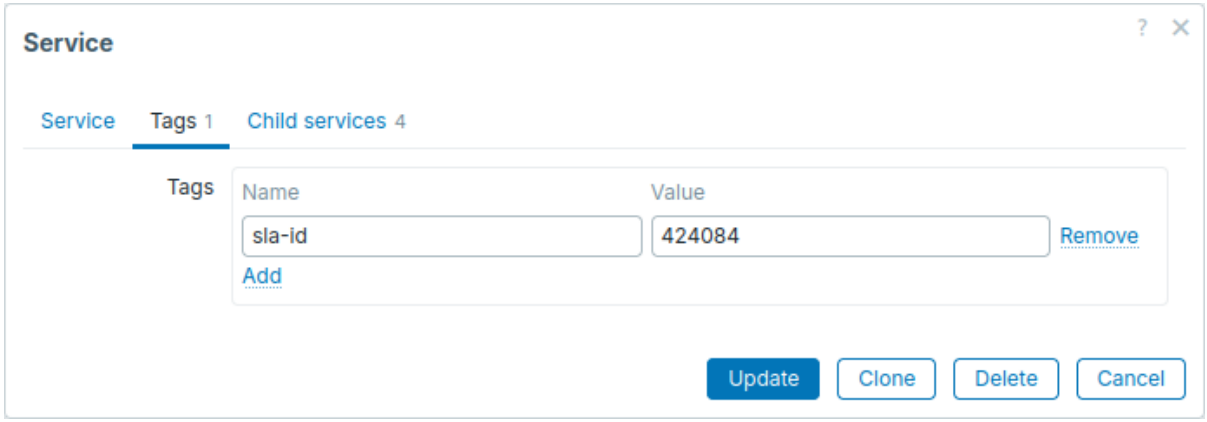
La scheda *Servizi figlio* consente di specificare i servizi dipendenti. Fare clic su *Aggiungi* per aggiungere un servizio dall'elenco dei servizi esistenti. Se si desidera aggiungere un nuovo servizio figlio, salvare prima questo servizio, quindi fare clic sull'icona più accanto al servizio appena creato.

Tag Esistono due diversi tipi di tag nei servizi:

- Tag del servizio
- Tag del problema

Tag del servizio

I tag del servizio vengono usati per associare i servizi alle azioni del servizio e agli SLA. Questi tag sono specificati nella scheda di configurazione del servizio *Tag*. Per il mapping degli SLA, viene usata la logica *OR*: un servizio verrà associato a un SLA se ha almeno un tag corrispondente. Nelle azioni del servizio, le regole di mapping sono configurabili e possono usare la logica *AND*, *OR* oppure *AND/OR*.

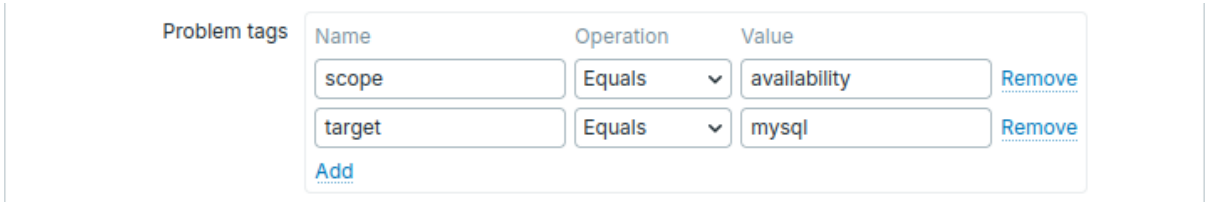


Tag dei problemi

I tag dei problemi vengono utilizzati per associare i problemi ai servizi. Questi tag sono specificati nella scheda di configurazione del servizio principale.

Solo i servizi figlio del livello gerarchico più basso possono avere tag dei problemi definiti ed essere correlati direttamente ai problemi. Se i tag dei problemi corrispondono, lo stato del servizio cambierà nello stesso stato del problema. In caso di più problemi, un servizio avrà lo stato del problema più grave. Lo stato di un servizio padre viene quindi calcolato in base agli stati dei servizi figlio secondo le regole di calcolo dello stato.

Se sono specificati più tag, viene utilizzata la logica *AND*: un problema deve avere tutti i tag specificati nella configurazione del servizio per essere associato al servizio.

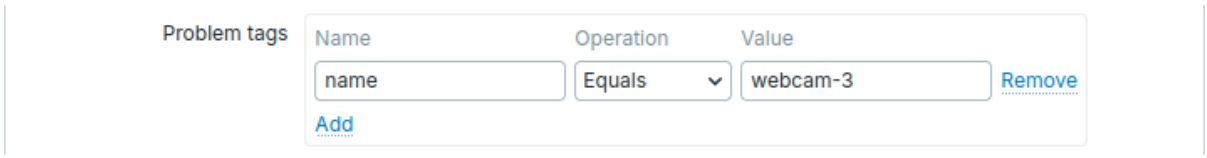


Note:
Un problema in Zabbix eredita i tag dall'intera catena di template, host, item, scenari web e trigger. Uno qualsiasi di questi tag può essere utilizzato per associare i problemi ai servizi.

Esempio:

Il problema *Web camera 3 is down* ha i tag `type:video-surveillance`, `floor:1` e `name:webcam-3` e stato *Warning*

Il service **Web camera 3** ha solo il tag del problema specificato: `name:webcam-3`



Lo stato del service cambierà da *OK* a *Warning* quando questo problema viene rilevato.

Se il service **Web camera 3** avesse i tag del problema `name:webcam-3` e `floor:2`, il suo stato non verrebbe modificato quando il problema viene rilevato, perché le condizioni sono soddisfatte solo parzialmente.

Modifica dei servizi esistenti

Note:
I pulsanti descritti di seguito sono visibili solo quando la sezione *Services* è in modalità di modifica.

Per modificare un servizio esistente, fai clic sull'icona a forma di matita accanto al servizio.

Per clonare un servizio esistente, fai clic sull'icona a forma di matita per aprirne la configurazione e quindi fai clic sul pulsante *Clone*. Quando un servizio viene clonato, i collegamenti al relativo parent vengono mantenuti, mentre i collegamenti ai child non vengono mantenuti.

Per eliminare un servizio, fai clic sull'icona x accanto ad esso. Quando elimini un servizio parent, i suoi servizi child non verranno eliminati e passeranno di un livello più in alto nell'albero dei servizi (i child di 1° livello avranno lo stesso livello del servizio parent eliminato).

Due pulsanti sotto l'elenco dei servizi offrono alcune opzioni di modifica massiva:

- *Mass update* - aggiornamento massivo delle proprietà del servizio
- *Delete* - elimina i servizi

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi servizi, quindi fai clic sul pulsante richiesto.

SLA

Panoramica Una volta creati i servizi, è possibile iniziare a monitorare se le loro prestazioni rispettano l'accordo sul livello di servizio (SLA).

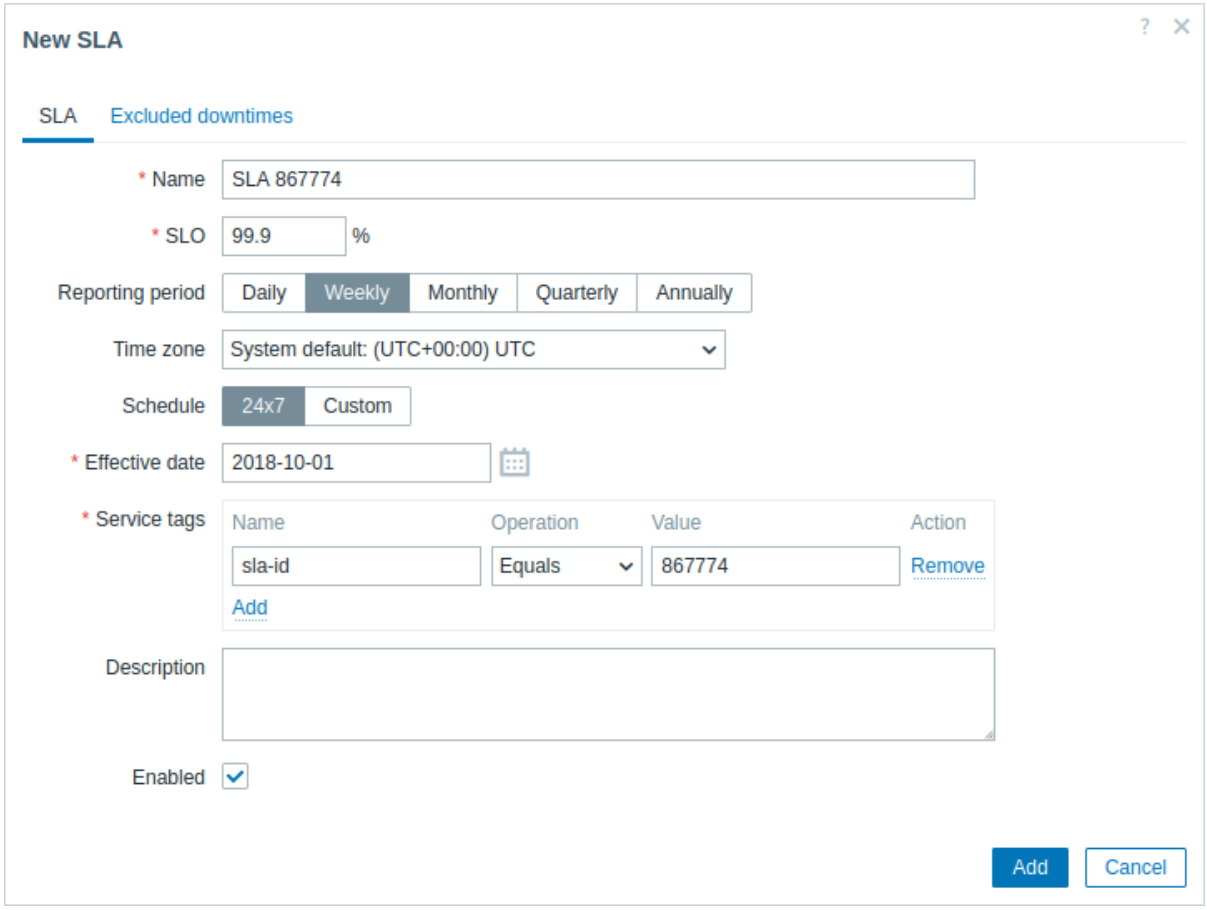
La sezione di menu *Servizi* > *SLA* consente di configurare gli SLA per vari servizi. In Zabbix, uno SLA definisce l'obiettivo del livello di servizio (SLO), il programma di disponibilità previsto e i periodi di inattività pianificati.

Gli SLA e i servizi vengono associati tramite i tag dei servizi. Lo stesso SLA può essere applicato a più servizi: le prestazioni verranno misurate separatamente per ciascun servizio corrispondente. A un singolo servizio possono essere assegnati più SLA: i dati di ciascuno SLA verranno visualizzati separatamente.

Nei report SLA, Zabbix fornisce i dati dell'indicatore del livello di servizio (SLI), che misura la disponibilità effettiva del servizio. Per determinare se un servizio soddisfa gli obiettivi dello SLA, lo SLO (disponibilità prevista in %) viene confrontato con lo SLI (disponibilità reale in %).

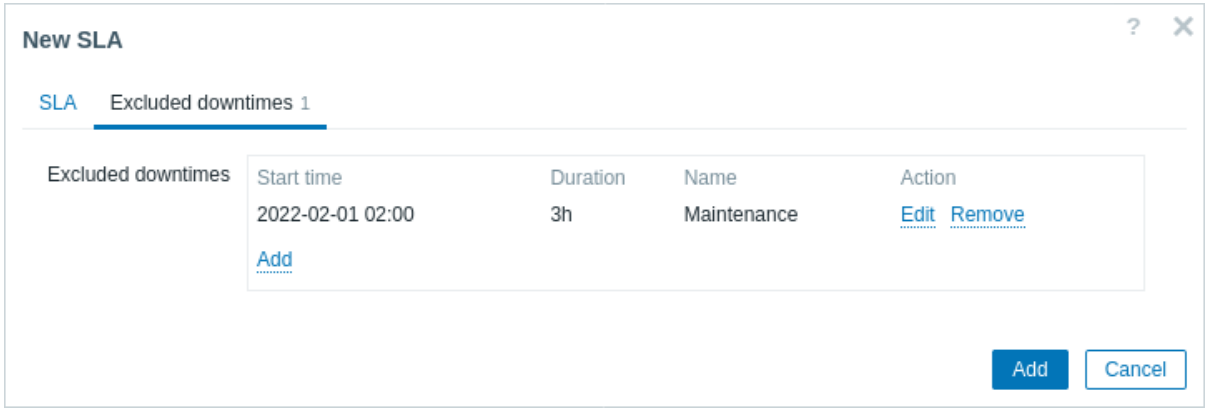
Configurazione Per creare un nuovo SLA, fare clic sul pulsante *Crea SLA*.

La scheda *SLA* consente di specificare i parametri generali dello SLA.



Parametro	Descrizione
Nome	Inserire il nome dello SLA.
SLO	Inserire l'obiettivo del livello di servizio (SLO) in percentuale.
Periodo di reporting	Selezionare il periodo di reporting dell'SLI utilizzato per calcolare i dati SLI nel report SLA: **Giornaliero** - ogni giorno, dalle 00:00:00 alle 23:59:59. **Settimanale** - ogni settimana, da domenica 00:00:00 a sabato 23:59:59. **Mensile** - ogni mese, dal primo giorno alle 00:00:00 all'ultimo giorno alle 23:59:59. **Trimestrale** - ogni trimestre solare: Q1 (gen-mar), Q2 (apr-giu), Q3 (lug-set), Q4 (ott-dic). **Annuale** - ogni anno solare, dal 1° gennaio alle 00:00:00 al 31 dicembre alle 23:59:59.
Fuso orario	Selezionare il fuso orario dello SLA.
Pianificazione	Selezionare la pianificazione dello SLA: 24x7 o personalizzata.
Data di decorrenza	Selezionare la data di inizio del calcolo dello SLA.
Tag del servizio	Aggiungere tag del servizio per identificare i servizi a cui applicare questo SLA. **Nome** - nome del tag del servizio; deve corrispondere esattamente e la distinzione tra maiuscole e minuscole è significativa. **Operazione** - selezionare *Uguale a* se il valore del tag deve corrispondere esattamente (con distinzione tra maiuscole e minuscole) oppure *Contiene* se deve corrispondere una parte del valore del tag (senza distinzione tra maiuscole e minuscole). **Valore** - valore del tag del servizio da cercare in base all'operazione selezionata. Lo SLA viene applicato a un servizio se corrisponde almeno un tag del servizio.
Descrizione	Aggiungere una descrizione per lo SLA.
Abilitato	Selezionare la casella di controllo per abilitare il calcolo dello SLA.

La scheda *Periodi di inattività esclusi* consente di specificare i periodi di inattività da escludere dal calcolo dello SLA.



Fare clic su *Aggiungi* per configurare i periodi di inattività esclusi, quindi inserire il nome del periodo, la data di inizio e la durata.

Report SLA Le prestazioni di un service rispetto a un SLA sono visibili nel report SLA. I report SLA possono essere visualizzati:

- dalla sezione *SLA* facendo clic sul collegamento ipertestuale del report SLA;
- dalla sezione *Services* facendo clic sul nome dell'SLA nella scheda info;
- nel widget *SLA report* della Dashboard.

Una volta configurato un SLA, la scheda *Info* nella sezione dei services mostrerà anche alcune informazioni sulle prestazioni del service.

Esempio di configurazione

Panoramica Questa sezione descrive una semplice configurazione per il monitoraggio del cluster ad alta affidabilità Zabbix come servizio.

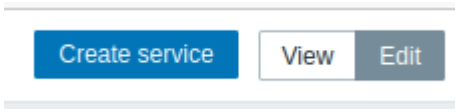
Prerequisiti Prima di configurare il monitoraggio dei servizi, è necessario che gli host siano configurati:

- *Nodo HA 1* con almeno un trigger e un tag (preferibilmente impostato a livello di trigger) `component:ha-node-1`
- *Nodo HA 2* con almeno un trigger e un tag (preferibilmente impostato a livello di trigger) `component:ha-node-2`

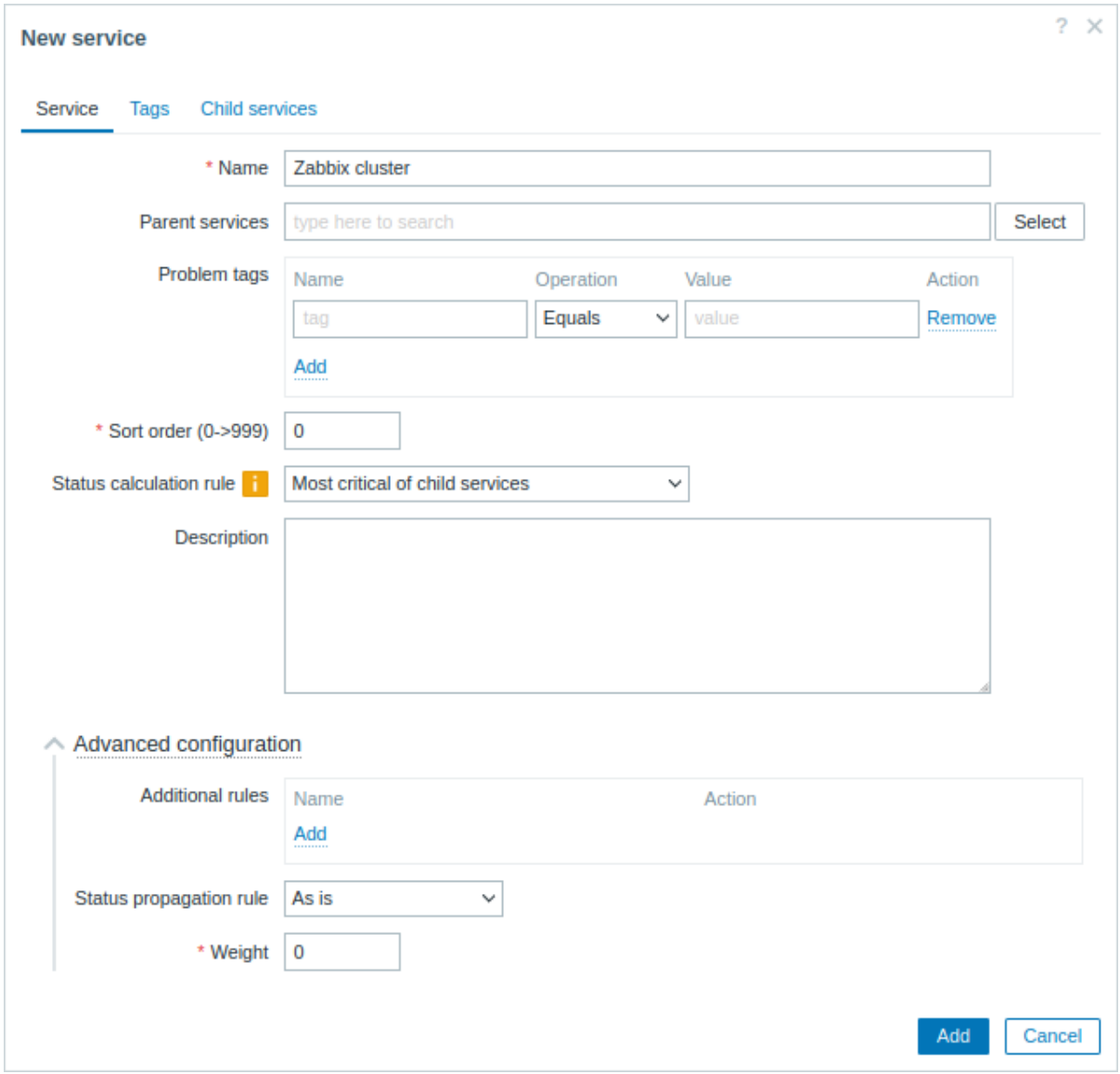
Albero dei servizi Il passo successivo è costruire l'albero dei servizi. In questo esempio, l'infrastruttura è molto semplice e consiste in tre servizi: *Zabbix cluster* (padre) e due servizi figlio *Zabbix server node 1* e *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

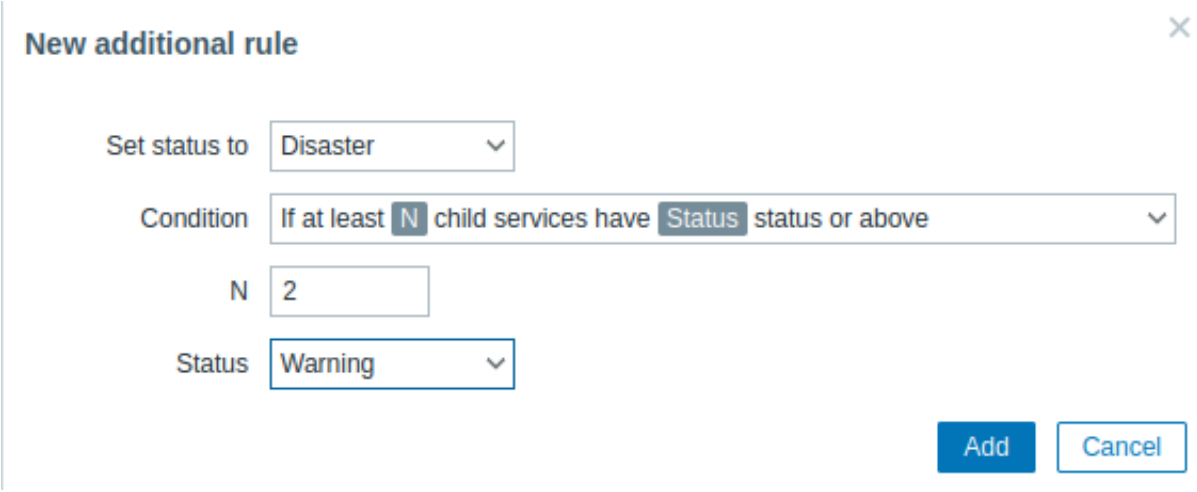
Nella pagina Services, attiva la modalità *Edit* e fai clic su *Create service*:



Nella finestra di configurazione del servizio, inserisci il nome *Zabbix cluster* e fai clic sull'intestazione *Advanced configuration* per visualizzare le opzioni di configurazione avanzate.



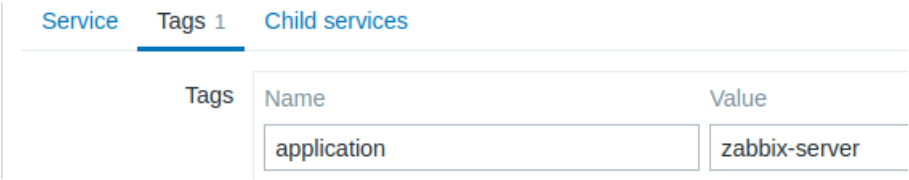
Configura una regola aggiuntiva:



Zabbix cluster avrà due servizi figlio, uno per ciascun nodo HA. Se entrambi i nodi HA presentano problemi con stato almeno *Warning*, lo stato del servizio padre deve essere impostato su *Disaster*. Per ottenere questo risultato, la regola aggiuntiva deve essere configurata come segue:

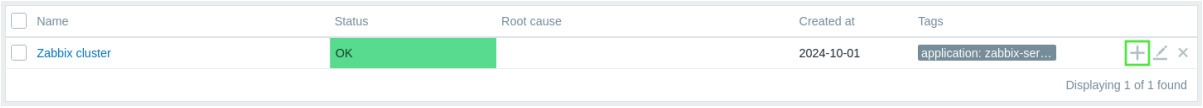
- Imposta stato su: Disaster
- Condizione: Se almeno `N` servizi figlio hanno stato `Status` o superiore
- N: 2
- Stato: Warning

Passa alla scheda *Tags* e aggiungi il tag `application:zabbix-server`. Questo tag verrà utilizzato in seguito per le azioni sui servizi e per i report SLA.



Salva il nuovo servizio.

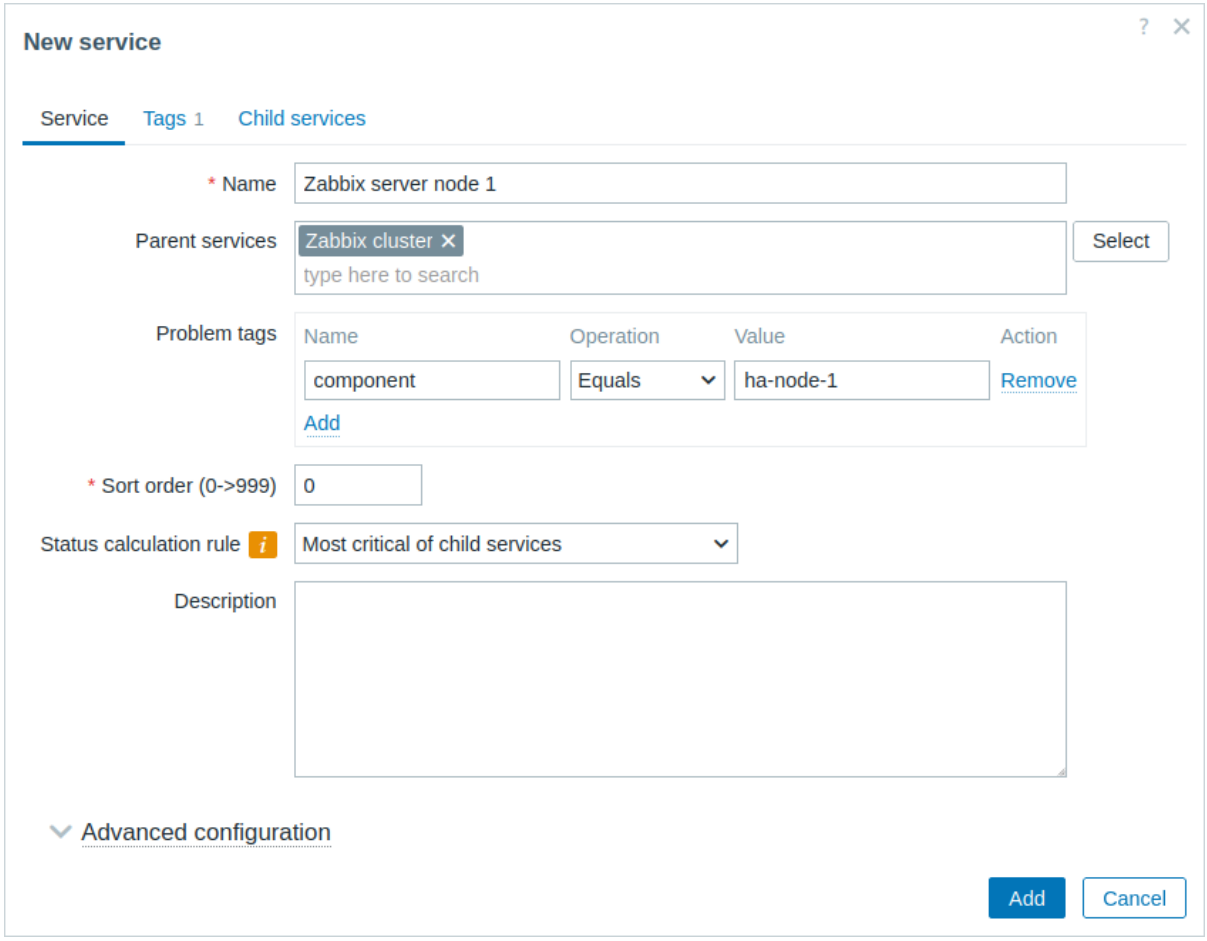
Per aggiungere un servizio figlio, fai clic sull'icona più accanto al servizio del cluster Zabbix (l'icona è visibile solo in modalità *Edit*).



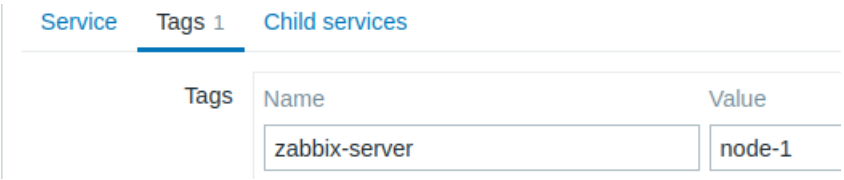
Nella finestra di configurazione del servizio, inserisci il nome *Zabbix server node 1*. Nota che il parametro Parent services è già precompilato con *Zabbix cluster*.

La disponibilità di questo servizio è influenzata dai problemi sull'host *HA node 1*, contrassegnati dal tag del problema `component:ha-node-1`. Nel parametro Problem tags, inserisci:

- Name: component
- Operation: Equals
- Value: ha-node-1



Passa alla scheda *Tags* e aggiungi un tag del servizio: `zabbix-server:node-1`. Questo tag verrà usato in seguito per le azioni del servizio e i report SLA.



Salva il nuovo servizio.

Creare un altro servizio figlio del cluster Zabbix con il nome ”Zabbix server node 2”.

Impostare i tag del problema come segue:

- Nome: component
- Operazione: Uguale
- Valore: ha-node-2

Passare alla scheda *Tag* e aggiungere un tag di servizio: `zabbix-server:node-2`.

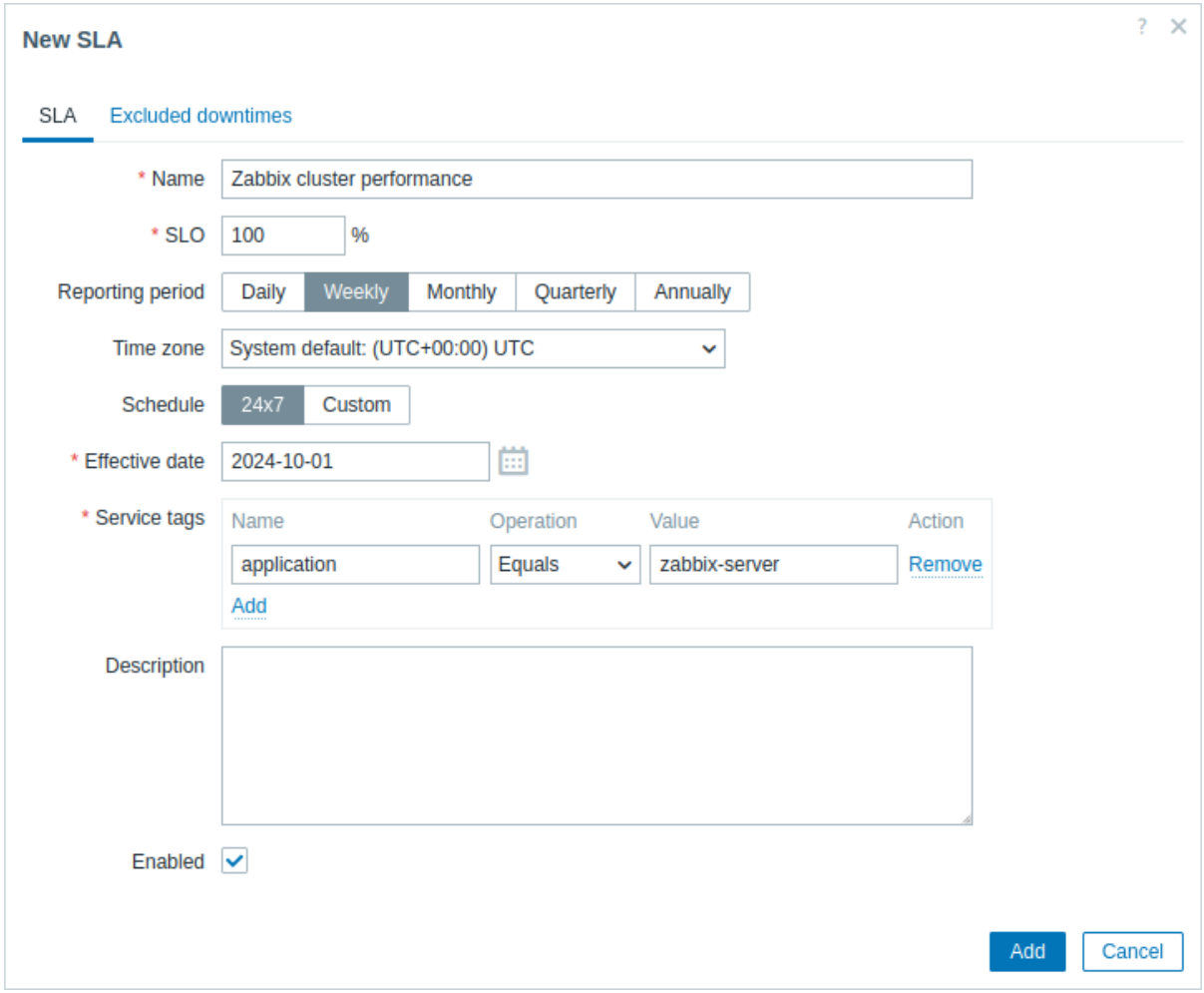
Salvare il nuovo servizio.

SLA In questo esempio, la disponibilità prevista del cluster Zabbix è del 100%, escludendo un periodo di manutenzione semestrale di un’ora.

Per prima cosa, è necessario aggiungere un nuovo SLA:

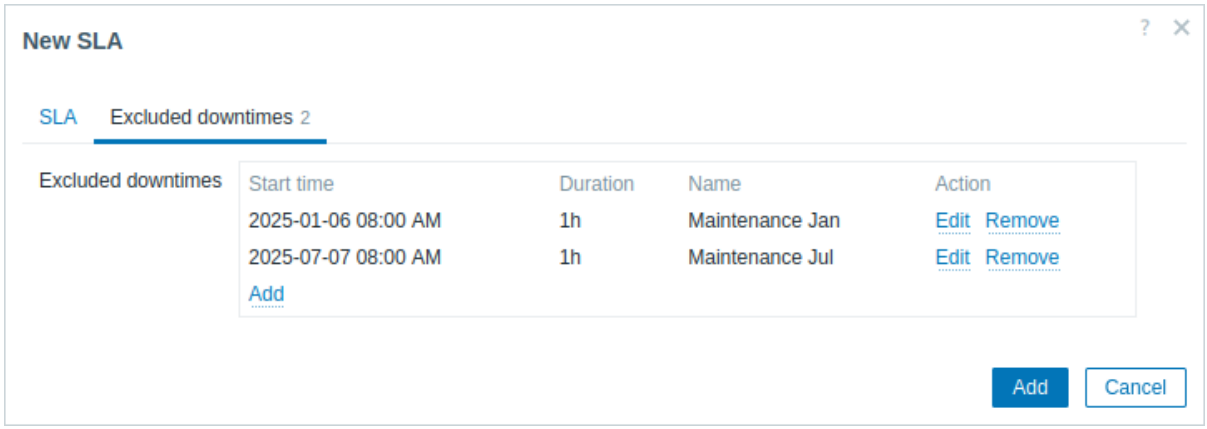
1. Andare a *Services* > *SLA*.
2. Fare clic su *Create SLA*.
3. Inserire il nome *Zabbix cluster performance* e impostare l’SLO al 100%. Il servizio Zabbix cluster ha il service tag `application:zabbix-server`.
4. Per utilizzare questo SLA per misurare le prestazioni del cluster Zabbix, nel parametro *Service tags* specificare:

- Name: application
- Operation: Equals
- Value: zabbix-server



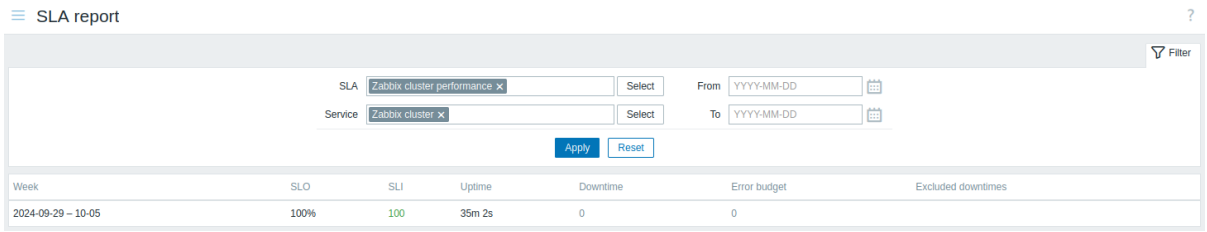
In un ambiente reale, è inoltre possibile aggiornare il periodo di reporting desiderato, il fuso orario e la data di inizio, oppure modificare la pianificazione da 24/7 a personalizzata. Per questo esempio, le impostazioni predefinite sono sufficienti.

5. Passare alla scheda *Excluded downtimes* e aggiungere i periodi di inattività relativi agli intervalli di manutenzione pianificati, in modo da escludere tali periodi dal calcolo dello SLA.
6. Nella sezione *Excluded downtimes*, fare clic sul collegamento *Add*, quindi inserire il nome del periodo di inattività, l'ora di inizio pianificata e la durata.

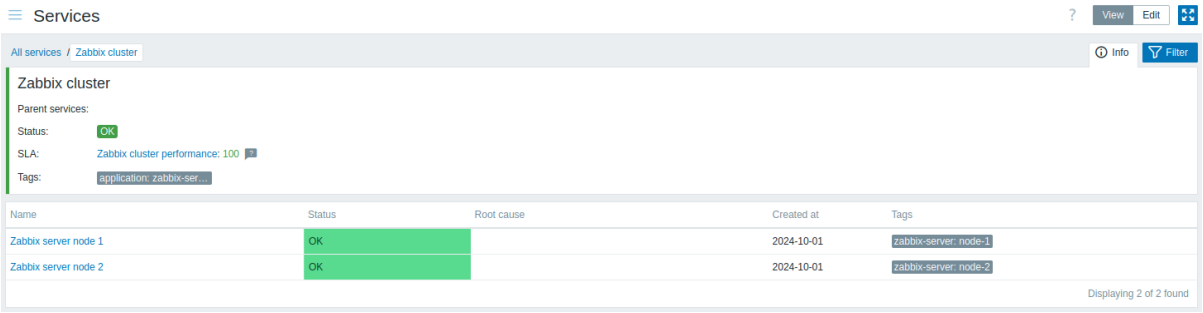


7. Fare clic su *Add* per salvare il nuovo SLA.

Passare alla sezione dei report SLA per visualizzare il report SLA del cluster Zabbix.



Le informazioni sullo SLA possono essere verificate anche nella sezione *Services*.



Aggiornamento del monitoraggio dei servizi

Panoramica In Zabbix 6.0, la funzionalità di monitoraggio dei servizi è stata notevolmente rielaborata (vedere Novità in Zabbix 6.0.0 per l'elenco delle modifiche).

Questa pagina descrive come i servizi e gli SLA, definiti nelle versioni precedenti di Zabbix, vengono modificati durante un aggiornamento a Zabbix 6.0 o versioni successive.

Servizi Nelle versioni precedenti di Zabbix, i servizi avevano due tipi di dipendenze: soft e hard. Dopo un aggiornamento, tutte le dipendenze diventeranno equivalenti.

Se un servizio "Child service" è stato precedentemente collegato a "Parent service 1" tramite una dipendenza hard e inoltre a "Parent service 2" tramite una dipendenza soft, dopo un aggiornamento il "Child service" avrà due servizi padre: "Parent service 1" e "Parent service 2".

La mappatura tra problemi e servizi basata su trigger è stata sostituita dalla mappatura basata su tag. In Zabbix 6.0 e versioni successive, il modulo di configurazione del servizio ha un nuovo parametro *Problem tags*, che consente di specificare una o più coppie nome e valore del tag per la corrispondenza dei problemi. I trigger che sono stati collegati a un servizio riceveranno un nuovo tag `ServiceLink : <trigger ID>:<trigger name>` (il valore del tag verrà troncato a 32 caratteri). I servizi collegati riceveranno `ServiceLink` problem tag con lo stesso valore.

Regole di calcolo dello stato

L'algoritmo di calcolo dello stato verrà aggiornato in base alle seguenti regole:

- Non calcolare → Imposta lo stato su OK
- Problema, se almeno un elemento figlio presenta un problema → Il più critico tra i servizi figli
- Problema, se tutti gli elementi figli presentano problemi → Il più critico se tutti gli elementi figli presentano problemi

SLA In precedenza, gli obiettivi SLA dovevano essere definiti separatamente per ciascun servizio. Da Zabbix 6.0, l'SLA è diventato un'entità separata, che contiene informazioni sulla pianificazione del servizio, sul livello di servizio obiettivo previsto (SLO) e sui periodi di inattività da escludere dal calcolo. Una volta configurato, un SLA può essere assegnato a più servizi tramite i tag del servizio.

Durante un aggiornamento:

- Gli SLA identici definiti per ciascun servizio verranno raggruppati e verrà creato un SLA per ogni gruppo.
- Ogni servizio interessato riceverà un tag speciale `SLA:<ID>` e lo stesso tag verrà specificato nel parametro *Tag del servizio* dell'SLA corrispondente.
- L'ora di creazione del servizio, una nuova metrica nei report SLA, verrà impostata su 01/01/2000 00:00 per i servizi esistenti.

Crittografia e sicurezza

Encryption Zabbix supporta comunicazioni cifrate tra i componenti Zabbix usando il protocollo Transport Layer Security (TLS) v.1.2 e 1.3 (a seconda della libreria crittografica). È supportata la cifratura basata su certificati e quella basata su chiavi precondivise.

La cifratura può essere configurata per le connessioni:

- Tra Zabbix server, Zabbix proxy, Zabbix agent, Zabbix web service, zabbix_sender e le utility zabbix_get
- Verso il database Zabbix da Zabbix frontend e server/proxy
- Tra Zabbix frontend e Zabbix server

La cifratura è opzionale e configurabile per i singoli componenti:

- Alcuni proxy e agent possono essere configurati per usare la cifratura basata su certificati con il server, mentre altri possono usare la cifratura basata su chiavi precondivise, e altri ancora possono continuare a usare comunicazioni non cifrate (come in precedenza).
- Il server (proxy) può usare configurazioni di cifratura diverse per host diversi.

I programmi daemon di Zabbix usano una sola porta di ascolto per connessioni in ingresso cifrate e non cifrate. L'aggiunta della cifratura non richiede l'apertura di nuove porte sui firewall.

Limitazioni

- Le chiavi private sono memorizzate in testo semplice in file leggibili dai componenti di Zabbix durante l'avvio.
- Le chiavi precondivise vengono inserite nel frontend di Zabbix e memorizzate nel database di Zabbix in testo semplice.
- La crittografia integrata non protegge le comunicazioni tra il web server che esegue il frontend di Zabbix e il browser web dell'utente.
- Attualmente ogni connessione crittografata si apre con un handshake TLS completo; non sono implementati né il caching delle sessioni né i ticket.
- L'aggiunta della crittografia aumenta il tempo necessario per i controlli degli item e le azioni, a seconda della latenza di rete:
  - Ad esempio, se il ritardo dei pacchetti è di 100 ms, l'apertura di una connessione TCP e l'invio di una richiesta non crittografata richiedono circa 200 ms. Con la crittografia, vengono aggiunti circa 1000 ms per stabilire la connessione TLS.
  - Potrebbe essere necessario aumentare i timeout; in caso contrario, alcuni item e azioni che eseguono script remoti sugli agent potrebbero funzionare con connessioni non crittografate, ma andare in timeout con connessioni crittografate.
- La crittografia non è supportata da network discovery. I controlli di Zabbix agent eseguiti da network discovery saranno non crittografati e, se Zabbix agent è configurato per rifiutare le connessioni non crittografate, tali controlli non avranno esito positivo.

Compilazione di Zabbix con supporto alla crittografia Per supportare la crittografia, Zabbix deve essere compilato e collegato a una delle librerie crittografiche supportate:

- GnuTLS - dalla versione 3.1.18
- OpenSSL - versioni 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x - 3.5.x
- LibreSSL - testato con le versioni 2.7.4, 2.8.2:
  - LibreSSL 2.6.x non è supportato
  - LibreSSL è supportato come sostituto compatibile di OpenSSL; le nuove funzioni API `tls_*()` specifiche di LibreSSL non vengono utilizzate. I componenti Zabbix compilati con LibreSSL non potranno utilizzare PSK; sarà possibile utilizzare solo i certificati.

> **Note:**
> Per ulteriori informazioni sulla configurazione di SSL per il frontend Zabbix, consultare queste best practice.

La libreria viene selezionata specificando l'opzione corrispondente nello script "configure":

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (utilizzata anche per LibreSSL)

Ad esempio, per configurare i sorgenti per server e agent con *OpenSSL*, è possibile utilizzare un comando simile al seguente:

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

Diversi componenti Zabbix possono essere compilati con librerie crittografiche diverse (ad esempio, un server con *OpenSSL* e un agent con *GnuTLS*).

> **Attention:**
> Se si prevede di utilizzare chiavi precondivise (PSK), valutare l'utilizzo delle librerie *GnuTLS* o *OpenSSL 1.1.0* (o versioni successive) nei componenti Zabbix che utilizzano PSK. Le librerie *GnuTLS* e *OpenSSL 1.1.0* supportano suite di cifratura PSK con Perfect Forward Secrecy. Anche le versioni precedenti della libreria *OpenSSL* (1.0.1, 1.0.2c) supportano le PSK, ma le suite di cifratura PSK disponibili non forniscono Perfect Forward Secrecy.

Gestione della crittografia delle connessioni Le connessioni in Zabbix possono utilizzare:

- nessuna crittografia (impostazione predefinita)

- crittografia basata su certificati RSA
- crittografia basata su PSK

Esistono due parametri importanti utilizzati per specificare la crittografia tra i componenti Zabbix:

- TLSConnect - specifica quale crittografia utilizzare per le connessioni in uscita (non crittografata, PSK o certificato)
- TLSAccept - specifica quali tipi di connessione sono consentiti per le connessioni in entrata (non crittografata, PSK o certificato). È possibile specificare uno o più valori.

`TLSConnect` viene utilizzato nei file di configurazione del proxy Zabbix (in modalità attiva, specifica solo le connessioni al server) e dell'agent Zabbix (per i controlli attivi). Nel frontend Zabbix, l'equivalente di `TLSConnect` è il campo *Connessioni all'host* in *Raccolta dati* > *Host* > <some host> > scheda *Crittografia* e il campo *Connessioni al proxy* in *Amministrazione* > *Proxy* > <some proxy> > scheda *Crittografia*. Se il tipo di crittografia configurato per la connessione non riesce, non verranno provati altri tipi di crittografia.

`TLSAccept` viene utilizzato nei file di configurazione del proxy Zabbix (in modalità passiva, specifica solo le connessioni dal server) e dell'agent Zabbix (per i controlli passivi). Nel frontend Zabbix, l'equivalente di `TLSAccept` è il campo *Connessioni dall'host* in *Raccolta dati* > *Host* > <some host> > scheda *Crittografia* e il campo *Connessioni dal proxy* in *Amministrazione* > *Proxy* > <some proxy> > scheda *Crittografia*.

Normalmente si configura un solo tipo di crittografia per le connessioni in entrata. Tuttavia, potrebbe essere necessario cambiare il tipo di crittografia, ad esempio da non crittografata a basata su certificato, con tempi di inattività minimi e possibilità di rollback. Per ottenere questo risultato:

- Impostare `TLSAccept=unencrypted,cert` nel file di configurazione dell'agent e riavviare l'agent Zabbix.
- Testare la connessione con `zabbix_get` all'agent utilizzando un certificato. Se funziona, è possibile riconfigurare la crittografia per quell'agent nel frontend Zabbix, in *Raccolta dati* > *Host* > <some host> > scheda *Crittografia*, impostando *Connessioni all'host* su *Certificato*.
- Quando la cache di configurazione del server viene aggiornata (e la configurazione del proxy viene aggiornata se l'host è monitorato da un proxy), le connessioni a quell'agent saranno crittografate.
- Se tutto funziona come previsto, è possibile impostare `TLSAccept=cert` nel file di configurazione dell'agent e riavviare l'agent Zabbix. A questo punto l'agent accetterà solo connessioni crittografate basate su certificato. Le connessioni non crittografate e basate su PSK verranno rifiutate.

In modo analogo, il funzionamento è lo stesso per server e proxy. Se nel frontend Zabbix, nella configurazione dell'host, *Connessioni dall'host* è impostato su "Certificato", dall'agent (controlli attivi) e da zabbix_sender (item trapper) verranno accettate solo connessioni crittografate basate su certificato.

Molto probabilmente si configureranno le connessioni in entrata e in uscita utilizzando lo stesso tipo di crittografia oppure nessuna crittografia. Tuttavia, tecnicamente è possibile configurarle in modo asimmetrico, ad esempio con crittografia basata su certificato per le connessioni in entrata e basata su PSK per quelle in uscita.

La configurazione della crittografia per ogni host viene visualizzata nel frontend Zabbix, in *Raccolta dati* > *Host*, nella colonna *Crittografia dell'agent*. Ad esempio:

Esempio	Connessioni all'host	Connessioni consentite dall'host	Connessioni rifiutate dall'host
NONE	Non crittografate	Non crittografate	Crittografate, basate su certificato e basate su PSK
CERT NONE PSK CERT	Crittografate, basate su certificato	Crittografate, basate su certificato	Non crittografate e crittografate basate su PSK
PSK NONE PSK CERT	Crittografate, basate su PSK	Crittografate, basate su PSK	Non crittografate e crittografate basate su certificato
PSK NONE PSK CERT	Crittografate, basate su PSK	Non crittografate e crittografate basate su PSK	Crittografate basate su certificato
CERT NONE PSK CERT	Crittografate, basate su certificato	Non crittografate, crittografate basate su PSK o certificato	-

Attention:
Le connessioni non sono crittografate per impostazione predefinita. La crittografia deve essere configurata individualmente per ogni host e proxy.

zabbix_get e zabbix_sender con crittografia Vedere le pagine man di zabbix_get e zabbix_sender per usarli con crittografia.

Ciphersuites Le ciphersuite sono configurate internamente per impostazione predefinita durante l'avvio di Zabbix.

Sono inoltre supportate ciphersuite configurate dall'utente per GnuTLS e OpenSSL. Gli utenti possono configurare le ciphersuite in base alle proprie policy di sicurezza. L'uso di questa funzionalità è facoltativo (le ciphersuite predefinite integrate continuano a funzionare).

Per le librerie crittografiche compilate con le impostazioni predefinite, le regole integrate di Zabbix in genere producono le seguenti ciphersuite (in ordine di priorità dalla più alta alla più bassa):

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Ciphersuite definite dall'utente I criteri di selezione delle ciphersuite integrati possono essere sostituiti con ciphersuite definite dall'utente.

> **Attention:**
> Le ciphersuite definite dall'utente sono una funzionalità pensata per utenti esperti che comprendono le ciphersuite TLS, la loro sicurezza e le conseguenze di eventuali errori, e che hanno dimestichezza con la risoluzione dei problemi TLS.

I criteri di selezione delle ciphersuite integrati possono essere sostituiti usando i seguenti parametri:

Override scope	Parameter	Value	Description
Selezione delle ciphersuite per i certificati	TLSCipherCert13	Stringhe di cifratura OpenSSL 1.1.1 valide cipher strings per il protocollo TLS 1.3 (i loro valori vengono passati alla funzione OpenSSL `SSL_CTX_set_ciphersuites()`).	Criteri di selezione delle ciphersuite basati su certificato per TLS 1.3 Solo OpenSSL 1.1.1 o versioni successive.

Override scope	Parameter	Value	Description
	TLSCipherCert	Stringhe di cifratura OpenSSL valide cipher strings per TLS 1.2 o stringhe di priorità GnuTLS valide priority strings. I loro valori vengono passati rispettivamente alle funzioni `SSL_CTX_set_cipher_list()` o `gnutls_priority_init()`.	Criteri di selezione delle ciphersuite basati su certificato per TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Selezione delle ciphersuite per PSK	TLSCipherPSK13	Stringhe di cifratura OpenSSL 1.1.1 valide cipher strings per il protocollo TLS 1.3 (i loro valori vengono passati alla funzione OpenSSL `SSL_CTX_set_ciphersuites()`).	Criteri di selezione delle ciphersuite basati su PSK per TLS 1.3 Solo OpenSSL 1.1.1 o versioni successive.
	TLSCipherPSK	Stringhe di cifratura OpenSSL valide cipher strings per TLS 1.2 o stringhe di priorità GnuTLS valide priority strings. I loro valori vengono passati rispettivamente alle funzioni `SSL_CTX_set_cipher_list()` o `gnutls_priority_init()`.	Criteri di selezione delle ciphersuite basati su PSK per TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Elenco combinato delle ciphersuite per certificato e PSK	TLSCipherAll13	Stringhe di cifratura OpenSSL 1.1.1 valide cipher strings per il protocollo TLS 1.3 (i loro valori vengono passati alla funzione OpenSSL `SSL_CTX_set_ciphersuites()`).	Criteri di selezione delle ciphersuite per TLS 1.3 Solo OpenSSL 1.1.1 o versioni successive.
	TLSCipherAll	Stringhe di cifratura OpenSSL valide cipher strings per TLS 1.2 o stringhe di priorità GnuTLS valide priority strings. I loro valori vengono passati rispettivamente alle funzioni `SSL_CTX_set_cipher_list()` o `gnutls_priority_init()`.	Criteri di selezione delle ciphersuite per TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Per sostituire la selezione delle ciphersuite nelle utility zabbix_get e zabbix_sender, usare i parametri da riga di comando:

- `--tls-cipher13`
- `--tls-cipher`

I nuovi parametri sono facoltativi. Se un parametro non è specificato, viene usato il valore predefinito interno. Se un parametro è definito, non può essere vuoto.

Se l'impostazione di un valore TLSCipher* nella libreria crittografica non riesce, il server, proxy o agent non si avvierà e verrà registrato un errore.

È importante capire quando ciascun parametro è applicabile.

Connessioni in uscita

Il caso più semplice è quello delle connessioni in uscita:

- Per le connessioni in uscita con certificato - usare TLSCipherCert13 o TLSCipherCert
- Per le connessioni in uscita con PSK - usare TLSCipherPSK13 o TLSCipherPSK
- Nel caso delle utility zabbix_get e zabbix_sender, è possibile usare i parametri da riga di comando `--tls-cipher13` o `--tls-cipher` (la crittografia è specificata in modo univoco con il parametro `--tls-connect`)

Connessioni in ingresso

La situazione è un po' più complessa per le connessioni in ingresso, perché le regole sono specifiche per i componenti e per la configurazione.

Per Zabbix **agent**:

Configurazione connessione agent	Configurazione cifrari
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13

Configurazione connessione agent	Configurazione cifrari
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Per Zabbix **server** e **proxy**:

Configurazione connessione	Configurazione cifrari
Connessioni in uscita tramite PSK	TLSCipherPSK, TLSCipherPSK13
Connessioni in ingresso tramite certificati	TLSCipherAll, TLSCipherAll13
Connessioni in ingresso tramite PSK se il server non ha un certificato	TLSCipherPSK, TLSCipherPSK13
Connessioni in ingresso tramite PSK se il server ha un certificato	TLSCipherAll, TLSCipherAll13

Dalle due tabelle precedenti si può osservare uno schema:

- TLSCipherAll e TLSCipherAll13 possono essere specificati solo se viene usata una lista combinata di suite di cifratura basate su certificati **e** su PSK. Ci sono due casi in cui ciò avviene: server (proxy) con un certificato configurato (le suite di cifratura PSK sono sempre configurate su server, proxy se la libreria crittografica supporta PSK), agent configurato per accettare connessioni in ingresso basate sia su certificati sia su PSK
- negli altri casi, TLSCipherCert* e/o TLSCipherPSK* sono sufficienti

Le tabelle seguenti mostrano i valori predefiniti incorporati di `TLSCipher*`. Possono essere un buon punto di partenza per i tuoi valori personalizzati.

Parametro	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

Parametro	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] I valori predefiniti sono diversi per le versioni precedenti di OpenSSL (1.0.1, 1.0.2, 1.1.0), per LibreSSL e se OpenSSL è compilato senza supporto PSK.

Esempi di suite di cifratura configurate dall'utente

Di seguito sono riportati i seguenti esempi di suite di cifratura configurate dall'utente:

- Test delle stringhe di cifratura e autorizzazione delle sole suite di cifratura PFS
- Passaggio da AES128 ad AES256

Test delle stringhe dei cifrari e autorizzazione delle sole suite di cifratura PFS

Per visualizzare quali suite di cifratura sono state selezionate, è necessario impostare `DebugLevel=4` nel file di configurazione oppure utilizzare l'opzione `-vv` per zabbix_sender.

Potrebbe essere necessario fare alcuni tentativi con i parametri `TLSCipher*` prima di ottenere le suite di cifratura desiderate. È scomodo riavviare più volte Zabbix server, proxy o agent solo per modificare i parametri `TLSCipher*`. È più pratico utilizzare zabbix_sender o il comando `openssl`. Vediamo entrambi i metodi.

1. Utilizzo di zabbix_sender.

Creiamo un file di configurazione di test, ad esempio `/home/zabbix/test.conf`, con la sintassi di un file `zabbix_agentd.conf`:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

Per questo esempio sono necessari certificati CA e dell'agent validi, oltre a una PSK. Modificare i percorsi e i nomi dei file dei certificati e della PSK in base al proprio ambiente.

Se non si utilizzano certificati, ma solo una PSK, è possibile creare un file di test più semplice:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Le suite di cifratura selezionate possono essere visualizzate eseguendo zabbix_sender (esempio compilato con 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
```

Qui sono visualizzate le suite di cifratura selezionate per impostazione predefinita. Questi valori predefiniti sono scelti per garantire l'interoperabilità con gli agent Zabbix in esecuzione su sistemi con versioni precedenti di OpenSSL (dalla 1.0.1).

Nei sistemi più recenti è possibile aumentare il livello di sicurezza consentendo solo alcune suite di cifratura, ad esempio solo le suite di cifratura con PFS (Perfect Forward Secrecy). Configuriamo i parametri `TLSCipher*` per consentire solo le suite di cifratura che supportano PFS.

> **Attention:**
> Il risultato non sarà interoperabile con i sistemi che utilizzano OpenSSL 1.0.1 e 1.0.2, se viene utilizzata una PSK. La cifratura basata su certificati dovrebbe funzionare.

Aggiungere due righe al file di configurazione `test.conf`:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

ed eseguire nuovamente il test:

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

Gli elenchi "certificate ciphersuites" e "PSK ciphersuites" sono cambiati e sono più brevi rispetto a prima: contengono solo le suite di cifratura TLS 1.3 e le suite di cifratura TLS 1.2 ECDHE-*, come previsto.

2. TLSCipherAll e TLSCipherAll13 non possono essere testati con zabbix_sender; non influiscono sul valore "certificate and PSK ciphersuites" mostrato nell'esempio precedente. Per modificare TLSCipherAll e TLSCipherAll13 è necessario fare dei tentativi con l'agent, il proxy o il server.

Pertanto, per consentire solo le suite di cifratura PFS, potrebbe essere necessario aggiungere fino a tre parametri

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

a `zabbix_agentd.conf`, `zabbix_proxy.conf` e `zabbix_server.conf` se ciascuno di essi ha un certificato configurato e l'agent dispone anche di una PSK.

Se l'ambiente Zabbix utilizza solo la cifratura basata su PSK e nessun certificato, è sufficiente un solo parametro:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Ora che si è compreso il funzionamento, è possibile testare la selezione delle suite di cifratura anche al di fuori di Zabbix, utilizzando il comando `openssl`. Testiamo tutti e tre i valori dei parametri `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
```

È possibile preferire `openssl ciphers` con l'opzione `-V` per ottenere un output più dettagliato:

```
$ 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
```

Analogamente, è possibile testare le stringhe di priorità per GnuTLS:

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

Passaggio da AES128 ad AES256

Zabbix usa AES128 come impostazione predefinita integrata per i dati. Supponiamo che tu stia usando certificati e voglia passare ad AES256, su OpenSSL 1.1.1.

Questo può essere ottenuto aggiungendo i parametri corrispondenti in `zabbix_server.conf`:

```
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
Anche se verranno utilizzate solo suite di cifratura relative ai certificati, i parametri `TLSCipherPSK*` sono definiti comunque per evitare i loro valori predefiniti, che includono cifrari meno sicuri per una maggiore interoperabilità. Le suite di cifratura PSK non possono essere completamente disabilitate su server/proxy.

E 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
```

Risoluzione dei problemi di crittografia Raccomandazioni generali:

- Iniziare comprendendo quale componente agisce come client TLS e quale come server TLS nel caso problematico.
 Zabbix server, proxy e agent, a seconda dell'interazione tra loro, possono tutti funzionare sia come server TLS sia come client TLS.
 Ad esempio, Zabbix server che si connette a un agent per un controllo passivo agisce come client TLS. L'agent svolge il ruolo di server TLS.
 Zabbix agent, che richiede un elenco di controlli attivi da un proxy, agisce come client TLS. Il proxy svolge il ruolo di server TLS.
 Le utility `zabbix_get` e `zabbix_sender` agiscono sempre come client TLS.
- Zabbix utilizza l'autenticazione reciproca.
 Ciascuna parte verifica la controparte e può rifiutare la connessione.
 Ad esempio, Zabbix server che si connette a un agent può chiudere immediatamente la connessione se il certificato dell'agent non è valido. E viceversa -
 Zabbix agent che accetta una connessione dal server può chiudere la connessione se il server non è considerato attendibile dall'agent.
- Esaminare i file di log su entrambe le parti - nel client TLS e nel server TLS.
 La parte che rifiuta la connessione può registrare un motivo preciso del rifiuto. L'altra parte spesso riporta un errore piuttosto generico (ad esempio,
 "Connection closed by peer", "connection was non-properly
 terminated").
- A volte una crittografia configurata in modo errato genera messaggi di errore fuorvianti che non indicano in alcun modo la causa reale.
 Nelle sottosezioni seguenti cerchiamo di fornire una raccolta, tutt'altro che esaustiva, di messaggi e possibili cause che possono aiutare nella risoluzione dei problemi.
 Si noti che diversi toolkit crittografici (OpenSSL, GnuTLS) spesso producono messaggi di errore diversi nelle stesse situazioni problematiche.
 A volte i messaggi di errore dipendono persino dalla particolare combinazione di toolkit crittografici su entrambe le parti.

Risoluzione dei problemi relativi al tipo di connessione o ai problemi di autorizzazione Server è configurato per connettersi con PSK a agent ma agent accetta solo connessioni non cifrate

Nel log di server o proxy (con *GnuTLS* 3.3.16)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

Nel log di server o proxy (con *OpenSSL* 1.0.2c)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

Un lato si connette con certificato ma l'altro lato accetta solo PSK o viceversa

In qualsiasi log (con *GnuTLS*):

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21 Could not negotiate a supported cipher suite.
```

In qualsiasi log (con *OpenSSL* 1.0.2c):

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

Tentativo di utilizzare Zabbix sender compilato con supporto TLS per inviare dati a Zabbix server/proxy compilato senza TLS Nel log del lato di connessione:

Linux:

```
...In zbx_tls_init_child()
...Libreria OpenSSL (versione OpenSSL 1.1.1  11 Sep 2018) inizializzata
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...Fine di zbx_tls_connect():FAIL errore:'connection closed by peer'
...errore di invio valore: TCP riuscito, impossibile stabilire TLS con [[localhost]:10051]: connection clo
```

Windows:

```
...Libreria OpenSSL (versione OpenSSL 1.1.1a  20 Nov 2018) inizializzata
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() ha richiesto l'identità PSK "PSK test sender"
...Fine di zbx_tls_connect():FAIL errore:'SSL_connect() I/O error: [0x00000000] The operation completed su
...errore di invio valore: TCP riuscito, impossibile stabilire TLS con [[192.168.1.2]:10051]: SSL_connect(
```

Nel log del lato di accettazione:

```
...failed to accept an incoming connection: from 127.0.0.1: support for TLS was not compiled in
```

Un lato si connette con PSK ma l'altro lato usa LibreSSL o è stato compilato senza supporto per la crittografia LibreSSL non supporta PSK.

Nel log del lato di connessione:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

Nel log del lato di accettazione:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Nel frontend di Zabbix:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

Un lato si connette con PSK ma l'altro lato usa OpenSSL con il supporto PSK disabilitato Nel log del lato che si connette:

```
...TCP riuscito, impossibile stabilire TLS con [[192.168.1.2]:10050]: SSL_connect() ha impostato il codice
```

Nel log del lato che accetta:

```
...impossibile accettare una connessione in ingresso: da 192.168.1.2: TLS handshake ha impostato il codice
```

Crittografia tramite certificati

Panoramica Zabbix può utilizzare certificati RSA in formato PEM, firmati da un'autorità di certificazione (CA) pubblica o interna.

La verifica del certificato viene eseguita rispetto a un certificato CA preconfigurato. Facoltativamente, è possibile utilizzare le Certificate Revocation Lists (CRL).

Ogni componente Zabbix può avere configurato un solo certificato.

Per ulteriori informazioni sulla configurazione e gestione di una CA interna, sulla generazione e firma delle richieste di certificato e sulla revoca dei certificati, fare riferimento a tutorial come l'OpenSSL PKI Tutorial v2.0.

Valutare e testare con attenzione le estensioni del certificato. Per ulteriori dettagli, vedere Limitations on using X.509 v3 certificate extensions.

Parametri di configurazione dei certificati I seguenti parametri di configurazione sono supportati per la configurazione dei certificati sui componenti di Zabbix.

Parametro	Obbligatorio	Descrizione
TLSCAFile	sì	Percorso completo di un file contenente i certificati della CA di livello superiore per la verifica del certificato del peer. Se si utilizza una catena di certificati con più membri, ordinare i certificati con i certificati delle CA di livello inferiore per primi, seguiti dai certificati delle CA di livello superiore. I certificati di più CA possono essere inclusi in un singolo file.
TLSCRLFile	no	Percorso completo di un file contenente le Certificate Revocation Lists (CRL).
TLSCertFile	sì	Percorso completo di un file contenente il certificato. Se si utilizza una catena di certificati con più membri, ordinare i certificati con il certificato del server, proxy o agent per primo, seguiti dai certificati delle CA di livello inferiore e conclusi dai certificati delle CA di livello superiore.
TLSKeyFile	sì	Percorso completo di un file contenente la chiave privata. Assicurarsi che questo file sia leggibile solo dall'utente Zabbix impostando i diritti di accesso appropriati.
TLSServerCertIssuer	no	Emittente consentito del certificato del server.
TLSServerCertSubject	no	Soggetto consentito del certificato del server.

Esempi di configurazione Dopo aver configurato i certificati necessari, configura i componenti di Zabbix per usare la crittografia basata su certificati.

Di seguito sono riportati i passaggi dettagliati per configurare:

- Zabbix server
- Zabbix proxy

- Zabbix agent

Zabbix server

1. Preparare il file del certificato CA.

Per verificare i certificati peer, Zabbix server deve poter accedere al file contenente i certificati CA radice autofirmati di livello superiore. Ad esempio, se sono necessari certificati di due CA radice indipendenti, inserirli in un file in `/home/zabbix/zabbix_ca_file.crt`:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVm0
-----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. Inserire il certificato/la catena di certificati di Zabbix server in un file, ad esempio in `/home/zabbix/zabbix_server.crt`. Il primo certificato è il certificato di Zabbix server, seguito dal certificato della CA intermedia:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
```

```
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJM0Hw==
-----END CERTIFICATE-----
```

Note:
Utilizzare solo gli attributi menzionati sopra per i certificati client e server, per evitare di influire sul processo di verifica dei certificati. Ad esempio, OpenSSL potrebbe non riuscire a stabilire una connessione crittografata se vengono utilizzate le estensioni *X509v3 Subject Alternative Name* o *Netscape Cert Type*. Per ulteriori informazioni, vedere Limitazioni sull'utilizzo delle estensioni dei certificati X.509 v3.

3. Inserire la chiave privata di Zabbix server in un file, ad esempio in `/home/zabbix/zabbix_server.key`:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Modificare i parametri di configurazione TLS nel file di configurazione di server di Zabbix:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

proxy Zabbix

1. Preparare i file contenenti i certificati CA di livello superiore, il certificato/la catena di certificati del proxy Zabbix e la chiave privata, come descritto nella sezione server Zabbix. Quindi, modificare di conseguenza i parametri `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` nel file di configurazione del proxy Zabbix.

2. Modificare i parametri TLS aggiuntivi nel file di configurazione del proxy Zabbix:

- Per il proxy attivo: `TLSConnect=cert`
- Per il proxy passivo: `TLSAccept=cert`

> **Note:**
> Per migliorare la sicurezza del proxy, è inoltre possibile impostare i parametri `TLSServerCertIssuer` e `TLSServerCertSubject`. Per ulteriori informazioni, vedere Limitazione dell'autorità di certificazione e del subject consentiti.

I parametri TLS nel file di configurazione finale del proxy possono essere simili ai seguenti:

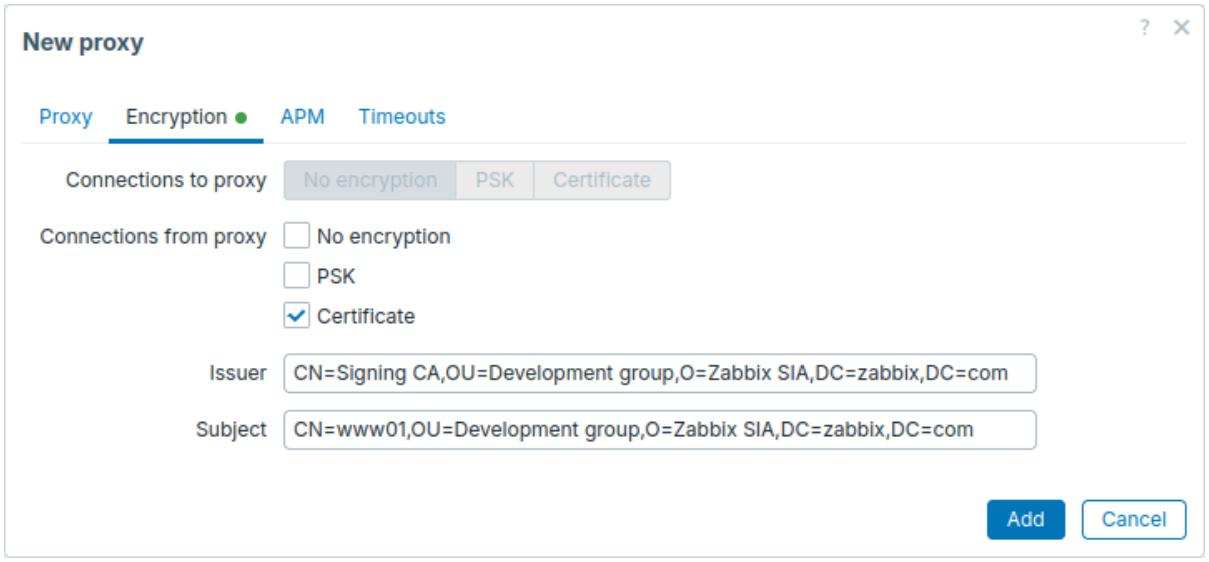
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Configurare la crittografia per questo proxy nel frontend Zabbix:

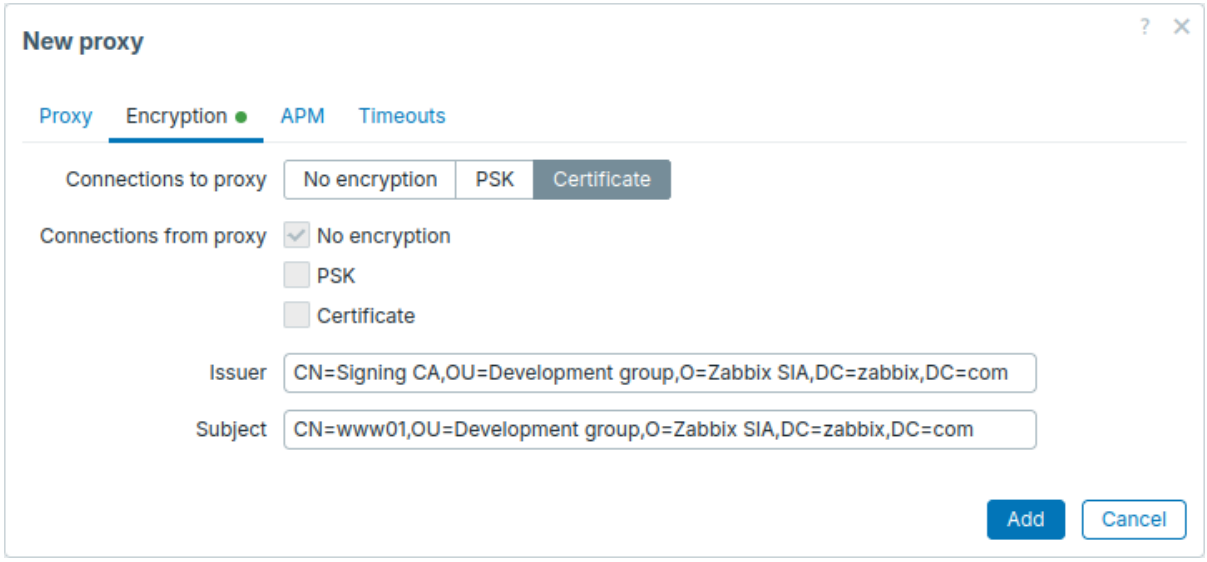
- Andare a: *Amministrazione* > *Proxy*.
- Selezionare il proxy e fare clic sulla scheda *Crittografia*.

Negli esempi seguenti, i campi *Emittente* e *Subject* sono compilati. Per ulteriori informazioni sul motivo e sulle modalità di utilizzo di questi campi, vedere Limitazione dell'autorità di certificazione e del subject consentiti.

Per il proxy attivo:



Per il proxy passivo:



Zabbix agent

1. Preparare i file con i certificati CA di livello superiore, il certificato/la catena di certificati del Zabbix agent e la chiave privata, come descritto nella sezione Zabbix server. Quindi, modificare di conseguenza i parametri `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` nel file di configurazione dell'agent Zabbix.

2. Modificare i parametri TLS aggiuntivi nel file di configurazione dell'agent Zabbix:

- Per l'agent attivo: `TLSConnect=cert`
- Per l'agent passivo: `TLSAccept=cert`

Note:
Per migliorare la sicurezza dell'agent, è possibile impostare i parametri `TLSServerCertIssuer` e `TLSServerCertSubject`. Per ulteriori informazioni, vedere Limitazione dell'emittente e del subject del certificato consentiti.

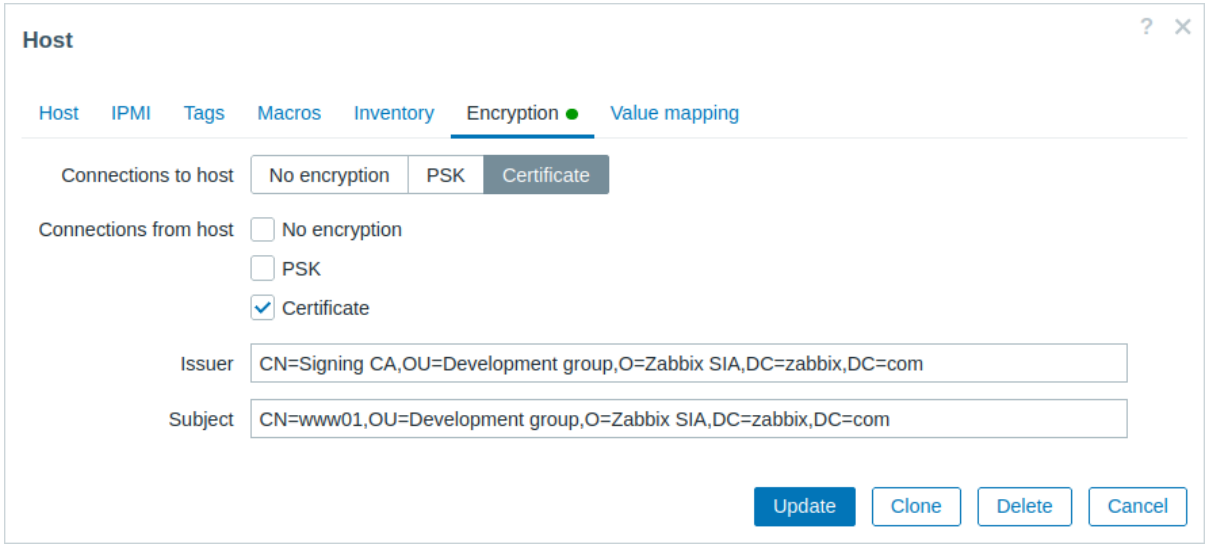
I parametri TLS nel file di configurazione finale dell'agent possono essere simili ai seguenti. Si noti che l'esempio presuppone che il host sia monitorato da un proxy; per questo motivo viene specificato come Subject del certificato:

```
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. Configurare la crittografia nel frontend Zabbix per il host monitorato da questo agent.

- Andare a: *Raccolta dati* > *Host*.
- Selezionare il host e fare clic sulla scheda *Crittografia*.

Nell'esempio seguente, i campi *Issuer* e *Subject* sono compilati. Per ulteriori informazioni sul motivo e sulle modalità di utilizzo di questi campi, vedere Limitazione dell'emittente e del subject del certificato consentiti.



Zabbix web service

1. Preparare i file con i certificati CA di livello superiore, il certificato/la catena di certificati del Zabbix web service e la chiave privata, come descritto nella sezione Zabbix server. Quindi, modificare di conseguenza i parametri `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` nel file di configurazione del web service di Zabbix.

2. Modificare un parametro TLS aggiuntivo nel file di configurazione del web service di Zabbix: `TLSAccept=cert`

I parametri TLS nel file di configurazione finale del web service possono essere simili ai seguenti:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Configurare Zabbix server affinché si connetta al Zabbix web service configurato per TLS modificando il parametro `WebServiceURL` nel file di configurazione del server di Zabbix:

```
WebServiceURL=https://example.com:443/report
```

Limitare l'autorità di certificazione e il subject consentiti Quando due componenti Zabbix (ad esempio, server e agent) stabiliscono una connessione TLS, convalidano reciprocamente i propri certificati. Se il certificato del peer è firmato da una CA attendibile (con un certificato di livello superiore preconfigurato in `TLSCAFile`), è valido, non è scaduto e supera gli altri controlli, la comunicazione tra i componenti può procedere. In questo caso più semplice, l'autorità di certificazione e il subject del certificato non vengono verificati.

Tuttavia, ciò comporta un rischio: chiunque disponga di un certificato valido può impersonare chiunque altro (ad esempio, un certificato host potrebbe essere utilizzato per impersonare un server). Sebbene ciò possa essere accettabile in ambienti di piccole dimensioni, in cui i certificati sono firmati da una CA interna dedicata e il rischio di impersonificazione è basso, potrebbe non essere sufficiente in ambienti più grandi o con requisiti di sicurezza più elevati.

Se la CA di livello superiore emette certificati che non devono essere accettati da Zabbix o se si desidera ridurre il rischio di impersonificazione, è possibile limitare i certificati consentiti specificandone l'autorità di certificazione e il subject.

Ad esempio, nel file di configurazione del proxy di Zabbix, è possibile specificare:

```
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
```

Con queste impostazioni, un proxy attivo non comunicherà con un server Zabbix il cui certificato abbia un'autorità di certificazione o un subject diverso. Analogamente, un proxy passivo non accetterà richieste da tale server.

Regole per la corrispondenza delle stringhe `Issuer` e `Subject`

Le regole per la corrispondenza delle stringhe `Issuer` e `Subject` sono le seguenti:

- Le stringhe `Issuer` e `Subject` vengono verificate in modo indipendente. Entrambe sono facoltative.
- Una stringa non specificata significa che viene accettata qualsiasi stringa.
- Le stringhe vengono confrontate *così come sono* e devono corrispondere esattamente.
- Sono supportati i caratteri UTF-8. Tuttavia, non sono supportati i caratteri jolly (*) né le espressioni regolari.
- Sono implementati i seguenti requisiti di RFC 4514 - i caratteri che richiedono l'escape (con una barra rovesciata '\', U+005C):

    - in qualsiasi punto della stringa: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
    - all'inizio della stringa: spazio (' ', U+0020) o cancelletto ('#', U+0023);
    - alla fine della stringa: spazio (' ', U+0020).
- I caratteri nulli (U+0000) non sono supportati. Se viene rilevato un carattere nullo, la corrispondenza fallirà.
- Gli standard RFC 4517 e RFC 4518 non sono supportati.

Ad esempio, se le stringhe dell'organizzazione (O) di `Issuer` e `Subject` contengono spazi finali e la stringa dell'unità organizzativa (OU) di `Subject` contiene virgolette doppie, questi caratteri devono essere escapati:

```
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
```

Ordine e formattazione dei campi

Zabbix segue le raccomandazioni di RFC 4514, che specifica un ordine "inverso" per questi campi, iniziando dai campi di livello più basso (CN), proseguendo con i campi di livello intermedio (OU, O) e concludendo con i campi di livello più alto (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
```

Al contrario, OpenSSL per impostazione predefinita visualizza le stringhe `Issuer` e `Subject` in ordine dal livello più alto a quello più basso. Nell'esempio seguente, i campi `Issuer` e `Subject` iniziano con il livello più alto (DC) e terminano con il campo di livello più basso (CN). Anche la formattazione con spazi e separatori di campo varia in base alle opzioni utilizzate e quindi non corrisponderà al formato richiesto da Zabbix.

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Per formattare correttamente le stringhe *Issuer* e *Subject* per Zabbix, invoca OpenSSL con le seguenti opzioni:

```
$ 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
```

L'output sarà quindi in ordine inverso, separato da virgole e utilizzabile nei file di configurazione di Zabbix e nel frontend:

```
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
```

Limitazioni nell'uso delle estensioni dei certificati X.509 v3 Quando si implementano certificati X.509 v3 in Zabbix, alcune estensioni potrebbero non essere completamente supportate oppure potrebbero causare un comportamento incoerente.

Estensione Subject Alternative Name

Zabbix non supporta l'estensione *Subject Alternative Name*, che viene usata per specificare nomi DNS alternativi come indirizzi IP o indirizzi email. Zabbix può convalidare solo il valore del campo *Subject* del certificato (vedere Restricting Allowed Certificate Issuer and Subject). Se i certificati includono il campo `subjectAltName`, l'esito della convalida del certificato può variare a seconda dei toolkit crittografici specifici usati per compilare i componenti di Zabbix. Di conseguenza, Zabbix può accettare o rifiutare i certificati in base a queste combinazioni.

Estensione Extended Key Usage

Zabbix supporta l'estensione *Extended Key Usage*. Tuttavia, se viene usata, in genere è necessario specificare sia l'attributo *clientAuth* (per l'autenticazione TLS WWW client) sia l'attributo *serverAuth* (per l'autenticazione TLS WWW server). Ad esempio:

- Nei controlli passivi, in cui l'agent Zabbix opera come server TLS, l'attributo *serverAuth* deve essere incluso nel certificato dell'agent.
- Nei controlli attivi, in cui l'agent opera come client TLS, l'attributo *clientAuth* deve essere incluso nel certificato dell'agent.

Sebbene GnuTLS possa generare un avviso per violazioni dell'uso delle chiavi, in genere consente comunque alla comunicazione di proseguire nonostante questi avvisi.

Estensione Name Constraints

Il supporto per l'estensione *Name Constraints* varia tra i toolkit crittografici. Assicurarsi che il toolkit scelto supporti questa estensione. Questa estensione può impedire a Zabbix di caricare i certificati CA se questa sezione è contrassegnata come critica, a seconda del toolkit specifico in uso.

Liste di revoca dei certificati (CRL) Se un certificato viene compromesso, la Certificate Authority (CA) può revocarlo includendo il certificato in una Certificate Revocation List (CRL). Le CRL sono gestite tramite file di configurazione e possono essere specificate usando il parametro `TLSCRLFile` nei file di configurazione di server, proxy e agent. Ad esempio:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

In questo caso, `zabbix_crl_file.crt` può contenere CRL di più CA e potrebbe avere un aspetto simile al seguente:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

Il file CRL viene caricato solo all'avvio di Zabbix. Per aggiornare la CRL, riavvia Zabbix.

Attention:
Se i componenti di Zabbix sono compilati con OpenSSL e vengono utilizzate CRL, assicurati che ogni CA di livello superiore e intermedia nelle catene di certificati abbia una CRL corrispondente (anche se vuota) inclusa nel `TLSCRLFile`.

Risoluzione dei problemi relativi ai certificati OpenSSL usato con CRL e per alcuni CA nella catena di certificati la relativa CRL non è inclusa in `TLSCRLFile`

Nel log del server TLS nel caso di peer *OpenSSL*:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

Nel log del server TLS nel caso di peer *GnuTLS*:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL scaduto o che scade durante il funzionamento del server

OpenSSL, nel log del server:

- prima della scadenza:

```
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"
```

- dopo la scadenza:

```
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"
```

Il punto qui è che con un CRL valido un certificato revocato viene segnalato come "certificate revoked". Quando il CRL scade, il messaggio di errore cambia in "certificate expired", il che è piuttosto fuorviante.

GnuTLS, nel log del server:

- prima e dopo la scadenza lo stesso:

```
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.
```

Certificato autofirmato, CA sconosciuta

OpenSSL, nel 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"'
```

Questo è stato osservato quando il certificato del server, per errore, aveva la stessa stringa Issuer e Subject, anche se era firmato da una CA. Issuer e Subject sono uguali nel certificato CA di livello superiore, ma non possono essere uguali nel certificato del server. (Lo stesso vale per i certificati di proxy e agent.)

Per verificare se un certificato contiene le stesse voci Issuer e Subject, eseguire:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

È accettabile che il certificato root (di livello superiore) abbia valori identici per Issuer e Subject.

Crittografia tramite chiavi pre-condivise

Panoramica Ogni chiave precondivisa (PSK) in Zabbix è in realtà costituita da una coppia di:

- stringa identificativa PSK non segreta;
- valore stringa PSK segreto.

La stringa identificativa PSK è una stringa UTF-8 non vuota. Ad esempio, "PSK ID 001 Zabbix agentd". È un nome univoco con cui questo specifico PSK viene identificato dai componenti Zabbix. Non inserire informazioni sensibili nella stringa identificativa PSK: viene trasmessa sulla rete senza crittografia.

Il valore PSK è una stringa di cifre esadecimali difficile da indovinare, ad esempio, e560cb0d918d26d31b4f642181f5f570ad89a390931102

Warning:
Non utilizzare mai in ambienti di produzione i valori PSK di esempio forniti nella documentazione. Genera invece un PSK univoco e crittograficamente sicuro.

Limiti di dimensione In Zabbix esistono limiti di dimensione per l'identità e il valore PSK; in alcuni casi, una libreria crittografica può avere un limite inferiore:

Componente	Dimensione massima dell'identità PSK	Dimensione minima del valore PSK	Dimensione massima del valore PSK
Zabbix	128 caratteri UTF-8	128 bit (PSK di 16 byte, inserita come 32 cifre esadecimali)	2048 bit (PSK di 256 byte, inserita come 512 cifre esadecimali)
GnuTLS	128 byte (può includere caratteri UTF-8)	-	2048 bit (PSK di 256 byte, inserita come 512 cifre esadecimali)
OpenSSL 1.0.x, 1.1.0	127 byte (può includere caratteri UTF-8)	-	2048 bit (PSK di 256 byte, inserita come 512 cifre esadecimali)
OpenSSL 1.1.1	127 byte (può includere caratteri UTF-8)	-	512 bit (PSK di 64 byte, inserita come 128 cifre esadecimali)
OpenSSL 1.1.1a e versioni successive	127 byte (può includere caratteri UTF-8)	-	2048 bit (PSK di 256 byte, inserita come 512 cifre esadecimali)

Attention:
Il frontend Zabbix consente di configurare una stringa di identità PSK lunga fino a 128 caratteri e una PSK lunga 2048 bit, indipendentemente dalle librerie crittografiche utilizzate.
 Se alcuni componenti Zabbix supportano limiti inferiori, è responsabilità dell'utente configurare l'identità e il valore PSK con una lunghezza consentita da tali componenti.
 Il superamento dei limiti di lunghezza causa errori di comunicazione tra i componenti Zabbix.

Prima che il server Zabbix si connetta all'agent utilizzando una PSK, il server cerca nel database l'identità e il valore PSK configurati per tale agent (in realtà nella cache di configurazione). Quando riceve una connessione, l'agent utilizza l'identità e il valore PSK dal proprio file di configurazione. Se entrambe le parti hanno la stessa stringa di identità PSK e lo stesso valore PSK, la connessione può avere esito positivo.

Attention:
Ogni identità PSK deve essere associata a un solo valore. È responsabilità dell'utente assicurarsi che non esistano due PSK con la stessa stringa di identità ma con valori diversi. In caso contrario, potrebbero verificarsi errori imprevedibili o interruzioni della comunicazione tra i componenti Zabbix che utilizzano PSK con questa stringa di identità PSK.

Generazione della PSK Ad esempio, è possibile generare una PSK a 256 bit (32 byte) utilizzando i seguenti comandi:

- con *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- con *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
```

Si noti che "psktool" genera un file di database contenente un'identità PSK e la PSK associata. Zabbix si aspetta solo una PSK nel file PSK, pertanto la stringa dell'identità e i due punti (':') devono essere rimossi dal file.

Configurazione del PSK per la comunicazione server-agent (esempio) Sul host dell'agent, scrivere il valore PSK generato in precedenza in un file, ad esempio `/home/zabbix/zabbix_agentd.psk`. Il file deve contenere il valore PSK nella prima riga.

Impostare i diritti di accesso al file PSK: deve essere leggibile solo dall'utente Zabbix.

Modificare i parametri TLS nel file di configurazione dell'agent di Zabbix (`zabbix_agentd.conf`). Ad esempio, impostare:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

L'agent si connetterà al server (controlli attivi) e accetterà dal server e da `zabbix_get` solo connessioni che utilizzano PSK. L'identità PSK sarà "PSK 001".

Riavviare l'agent. Ora è possibile testare la connessione utilizzando `zabbix_get`, ad esempio:

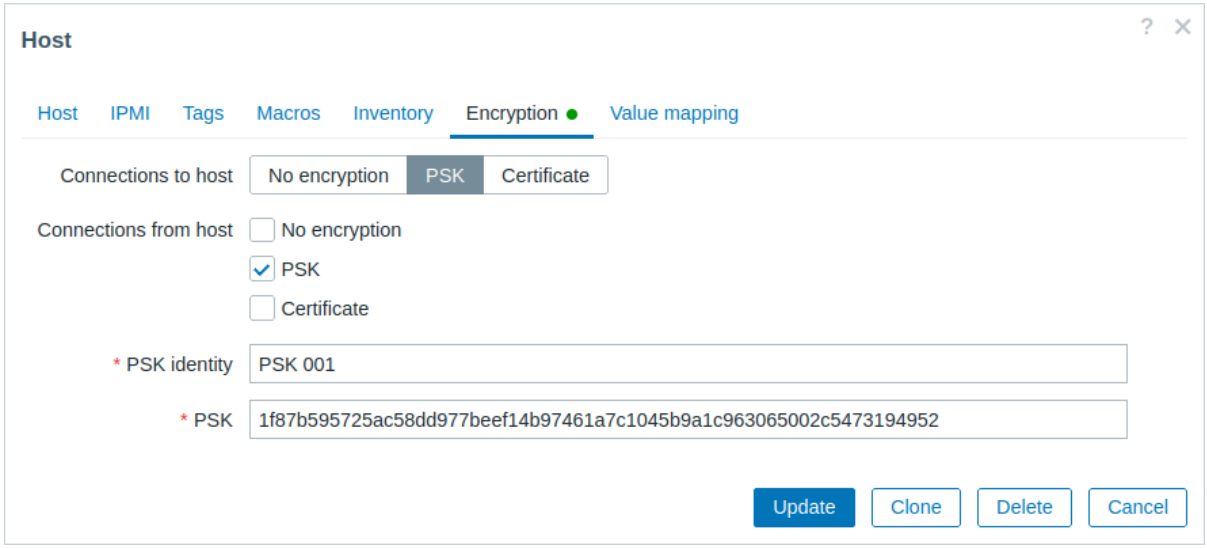
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Per ridurre al minimo i tempi di inattività, vedere come modificare il tipo di connessione in Gestione della crittografia delle connessioni).

Configurare la crittografia PSK per questo agent nel frontend di Zabbix:

1. Andare a: *Raccolta dati* > *Host*.
2. Selezionare l'host e fare clic sulla scheda *Crittografia*.
3. Configurare i dettagli della crittografia PSK.

Esempio:



Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Quando la cache di configurazione viene sincronizzata con il database, le nuove connessioni utilizzeranno PSK. Controllare i file di log del server e dell'agent per verificare la presenza di messaggi di errore.

Configurazione del PSK per la comunicazione tra server e proxy attivo (esempio) Sul proxy, scrivere il valore PSK in un file, ad esempio `/home/zabbix/zabbix_proxy.psk`. Il file deve contenere il PSK nella prima riga di testo, ad esempio:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Impostare i diritti di accesso al file PSK: deve essere leggibile solo dall'utente Zabbix.

Modificare i parametri TLS nel file di configurazione del proxy di Zabbix (`zabbix_proxy.conf`). Ad esempio, impostare:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Il proxy si connetterà al server utilizzando il PSK. L'identità PSK sarà "PSK 002".

(Per ridurre al minimo i tempi di inattività, vedere come modificare il tipo di connessione in Gestione della crittografia delle connessioni).

Configurare il PSK per questo proxy nel frontend di Zabbix. Andare a *Administration* > *Proxies*, selezionare il proxy e aprire la scheda *Encryption*. In "Connections from proxy", selezionare PSK. Incollare "PSK 002" nel campo "PSK identity" e "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9" nel campo "PSK". Fare clic su "Update".

Riavviare il proxy. Inizierà a utilizzare connessioni crittografate basate su PSK verso il server. Controllare i file di log del server e del proxy per verificare la presenza di messaggi di errore.

Per un proxy passivo, la procedura è molto simile. L'unica differenza consiste nell'impostare `TLSAccept=psk` nel file di configurazione del proxy di Zabbix e impostare *Connections to proxy* su PSK nel frontend di Zabbix.

Risoluzione dei problemi relativi a PSK PSK contiene un numero dispari di cifre esadecimali

Il proxy o l'agent non si avvia, messaggio nel log del proxy o dell'agent:

```
PSK non valida nel file "/home/zabbix/zabbix_proxy.psk"
```

La stringa di identità PSK più lunga di 128 byte viene passata a GnuTLS

Nel log del lato client TLS:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

Nel log del lato server TLS:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Valore PSK troppo lungo usato con OpenSSL 1.1.1

Nel log del lato di connessione:

```
...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
```

Nel log del lato di accettazione:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Questo problema si verifica in genere durante l'aggiornamento di OpenSSL da 1.0.x o 1.1.0 a 1.1.1 e se il valore PSK è più lungo di 512 bit (PSK da 64 byte, inserito come 128 cifre esadecimali).

Vedi anche: Limiti di dimensione dei valori

Connessione sicura al database

Panoramica

Questa sezione fornisce i passaggi di configurazione di Zabbix e esempi di configurazione per connessioni TLS sicure tra:

Database	Componenti Zabbix
MySQL	frontend Zabbix, server Zabbix, proxy Zabbix
PostgreSQL	frontend Zabbix, server Zabbix, proxy Zabbix

Per configurare la crittografia della connessione all'interno del DBMS, consultare la documentazione ufficiale del fornitore per i dettagli:

- MySQL: server di database di replica source e replica.
- MySQL: server di database per group replication, ecc.
- PostgreSQL opzioni di crittografia.

Tutti gli esempi si basano sulle release GA di MySQL CE (8.0) e PgSQL (13) disponibili tramite i repository ufficiali usando CentOS 8.

Requisiti

Per configurare la crittografia è necessario quanto segue:

- Sistema operativo supportato dal produttore con OpenSSL >=1.1.X o alternativa.

> **Note:**
> Si consiglia di evitare sistemi operativi in stato di fine vita, soprattutto nel caso di nuove installazioni

- Motore di database (RDBMS) installato e mantenuto dal repository ufficiale fornito dal produttore. I sistemi operativi spesso vengono distribuiti con versioni obsolete del software di database per le quali il supporto alla crittografia non è implementato, ad esempio sistemi basati su RHEL 7 e PostgreSQL 9.2, MariaDB 5.5 senza supporto alla crittografia.

Terminologia

L'impostazione di questa opzione impone l'uso di una connessione TLS al database da parte di server/proxy Zabbix e frontend verso il database:

- `required` - connessione tramite TLS come modalità di trasporto senza controlli dell'identità
- `verify_ca` - connessione tramite TLS e verifica del certificato
- `verify_full` - connessione tramite TLS, verifica del certificato e verifica che l'identità del database (CN) specificata da DBHost corrisponda al relativo certificato

Configurazione di Zabbix

Frontend al database

È possibile configurare una connessione sicura al database durante l'installazione del frontend:

- Selezionare la casella di controllo *Crittografia TLS del database* nel passaggio Configura connessione al DB per abilitare la crittografia del trasporto.
- Selezionare la casella di controllo *Verifica certificato del database*, visualizzata quando il campo *Crittografia TLS* è selezionato, per abilitare la crittografia con i certificati.

Note:
Per MySQL, la casella di controllo *Crittografia TLS del database* è disabilitata se *Host del database* è impostato su localhost, perché una connessione che utilizza un file socket (su Unix) o la memoria condivisa (su Windows) non può essere crittografata.
 Per PostgreSQL, la casella di controllo *Crittografia TLS* è disabilitata se il valore del campo *Host del database* inizia con una barra o se il campo è vuoto.

Le seguenti impostazioni diventano disponibili nella modalità di crittografia TLS con certificati (se entrambe le caselle di controllo sono selezionate):

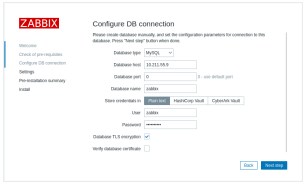
Parametro	Descrizione
File CA TLS del database	Specificare il percorso completo di un file valido dell'autorità di certificazione (CA) TLS.
File della chiave TLS del database	Specificare il percorso completo di un file valido della chiave TLS.
File del certificato TLS del database	Specificare il percorso completo di un file valido del certificato TLS.
Verifica dell'host del database	Selezionare questa casella di controllo per attivare la verifica dell'host. Disabilitata per MySQL perché la libreria PHP MySQL non consente di ignorare il passaggio di convalida del certificato peer.
Elenco di cifrari TLS del database	Specificare un elenco personalizzato di cifrari validi. Il formato dell'elenco dei cifrari deve essere conforme allo standard OpenSSL. Disponibile solo per MySQL.

Attention:
I parametri TLS devono puntare a file validi. Se puntano a file inesistenti o non validi, si verificherà un errore di autorizzazione.
 Se i file dei certificati sono scrivibili, il frontend genera un avviso nel report Informazioni di sistema: "I file dei certificati TLS devono essere di sola lettura." (visualizzato solo se l'utente PHP è il proprietario del certificato).
I certificati protetti da password non sono supportati.

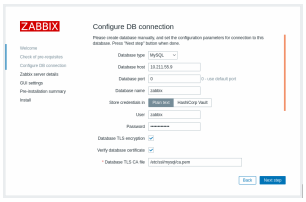
Casi d'uso

Il frontend di Zabbix utilizza un'interfaccia GUI per definire le possibili opzioni: `required`, `verify_ca`, `verify_full`. Specificare le opzioni richieste nel passaggio della procedura guidata di installazione *Configura connessioni DB*. Queste opzioni vengono mappate nel file di configurazione del frontend (`zabbix.conf.php`) nel modo seguente:

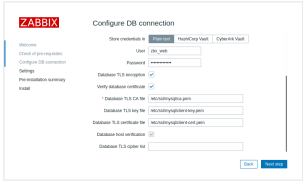
Impostazioni GUI	File di configurazione	Descrizione	Risultato

	...
	// Utilizzato per la connessione TLS.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Selezionare *Crittografia TLS del database*
		2. Lasciare deselezionato *Verifica certificato del database*
			Abilita la modalità `required`.

Impostazioni GUI	File di configurazione	Descrizione	Risultato

	...
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
		2. Specificare il percorso del *file CA TLS del database*
			Abilita la modalità `verify_ca`.

Impostazioni GUI	File di configurazione	Descrizione	Risultato

	...
	// Utilizzato per la connessione TLS con un elenco Cipher definito rigorosamente.
	`$DB['ENCRYPTION'] = true;`
	`$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>';`
	...
	
	Oppure:
	
	...
	// Utilizzato per la connessione TLS senza un elenco Cipher definito, selezionato dal server MySQL
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
		2. Specificare il percorso del *file chiave TLS del database*
		3. Specificare il percorso del *file CA TLS del database*
		4. Specificare il percorso del *file certificato TLS del database*
		5. Specificare l'*elenco Cipher TLS del database* (facoltativo)
			Abilita la modalità `verify_full` per MySQL.

Impostazioni GUI	File di configurazione	Descrizione	Risultato

	...
	`$DB['ENCRYPTION'] = true;`
	`$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. Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
		2. Specificare il percorso del *file chiave TLS del database*
		3. Specificare il percorso del *file CA TLS del database*
		4. Specificare il percorso del *file certificato TLS del database*
		5. Selezionare *Verifica host del database*
			Abilita la modalità `verify_full` per PostgreSQL.

Vedere anche: Esempi di configurazione della crittografia per MySQL, Esempi di configurazione della crittografia per PostgreSQL.

Configurazione di Zabbix server/proxy

Le connessioni sicure al database possono essere configurate con i rispettivi parametri nel file di configurazione di Zabbix server e/o proxy.

Configurazione	Risultato
Nessuna	Connessione al database senza crittografia.
Impostare `DBTLSConnect=required`	Il server/proxy stabilisce una connessione TLS al database. Non è consentita una connessione non crittografata.
1. Impostare `DBTLSConnect=verify_ca` 2. Impostare `DBTLSCAFile` - specificare il file dell'autorità di certificazione TLS	Il server/proxy stabilisce una connessione TLS al database dopo aver verificato il certificato del database.
1. Impostare `DBTLSConnect=verify_full` 2. Impostare `DBTLSCAFile` - specificare il file dell'autorità di certificazione TLS	Il server/proxy stabilisce una connessione TLS al database dopo aver verificato il certificato del database e l'identità dell'host del database.
1. Impostare `DBTLSCAFile` - specificare il file dell'autorità di certificazione TLS 2. Impostare `DBTLSCertFile` - specificare il file del certificato della chiave pubblica del client 3. Impostare `DBTLSKeyFile` - specificare il file della chiave privata del client	Il server/proxy fornisce un certificato client durante la connessione al database.
Impostare `DBTLSCipher` - l'elenco dei cifrari di crittografia consentiti dal client per le connessioni che utilizzano protocolli TLS fino a TLS 1.2 oppure `DBTLSCipher13` - l'elenco dei cifrari di crittografia consentiti dal client per le connessioni che utilizzano il protocollo TLS 1.3	(MySQL) La connessione TLS viene stabilita utilizzando un cifrario dell'elenco fornito. (PostgreSQL) L'impostazione di questa opzione verrà considerata un errore.

Configurazione della crittografia MySQL

Panoramica

Questa sezione fornisce diversi esempi di configurazione della crittografia per CentOS 8.2 e MySQL 8.4.0 e può essere utilizzata come guida rapida per crittografare la connessione al database.

Attention:
Se l'host MySQL è impostato su localhost, le opzioni di crittografia non saranno disponibili. In questo caso, una connessione tra il frontend di Zabbix e il database utilizza un file socket (su Unix) o la memoria condivisa (su Windows) e non può essere crittografata.

Note:
L'elenco delle combinazioni di crittografia non è limitato a quelle elencate in questa pagina. Sono disponibili molte altre combinazioni.

Prerequisiti

Installa il database MySQL dal repository ufficiale.

Consulta la documentazione MySQL per i dettagli su come utilizzare il repository MySQL.

Il server MySQL è pronto ad accettare connessioni sicure utilizzando un certificato autofirmato.

Per vedere quali utenti stanno utilizzando una connessione crittografata, esegui la seguente query (Performance Schema deve essere impostato su ON):

```
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;
```

Crittografia solo del trasporto

Configurazione di MySQL

Le versioni moderne del database sono pronte all'uso per la modalità di crittografia `required`. Dopo la configurazione iniziale e l'avvio verrà creato un certificato lato server.

Creare utenti e ruoli per i componenti principali (sostituire <strong_password> con una password complessa e univoca):

```
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'@'%';
```

Si noti che il protocollo X.509 non viene utilizzato per verificare l'identità, ma l'utente è configurato per utilizzare solo connessioni crittografate. Per ulteriori informazioni sulla configurazione degli utenti, consultare la documentazione di MySQL.

Eseguire il comando seguente per verificare la connessione (non è possibile utilizzare una connessione socket per testare le connessioni sicure):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Verificare lo stato corrente e le suite di cifratura disponibili:

```
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

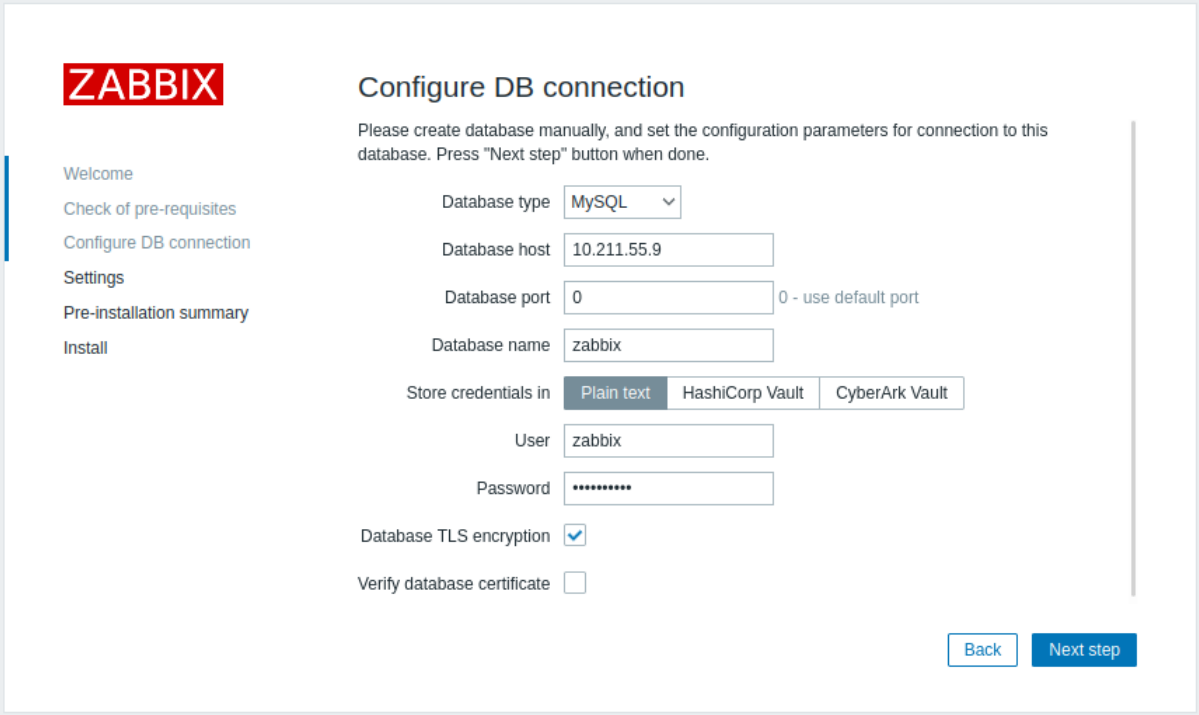

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Frontend

Per abilitare la crittografia solo del trasporto per le connessioni tra il frontend di Zabbix e il database:

- Selezionare *Database TLS encryption*
- Lasciare deselezionato *Verify database certificate*



Server

Per abilitare la crittografia solo del trasporto per le connessioni tra il server e il database, modificare il file di configurazione del server:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Crittografia con verifica dell'autorità di certificazione

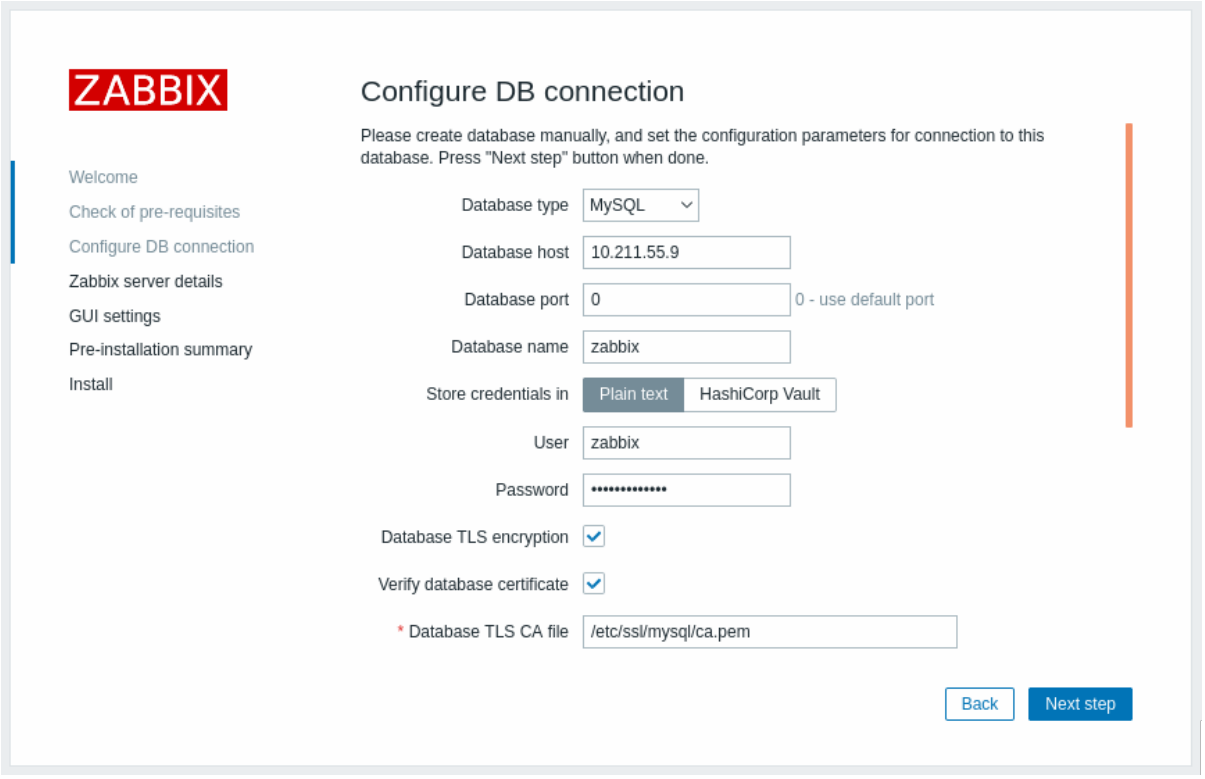
Copia il CA MySQL richiesto sul server frontend di Zabbix e assegna le autorizzazioni appropriate per consentire al webserver di leggere questo file.

> **Note:**
> Questa modalità non funziona su RHEL 7 a causa delle versioni più datate delle librerie MySQL.

frontend

Per abilitare la crittografia con verifica del certificato per le connessioni tra il frontend di Zabbix e il database:

- Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
- Specificare il percorso del file CA TLS del database



In alternativa, è possibile configurare questa impostazione nel file di configurazione del frontend (`zabbix.conf.php`):

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Risoluzione dei problemi: utilizzare lo strumento da riga di comando per verificare se la connessione è possibile per l'utente richiesto:

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

Server

Per abilitare la crittografia con verifica del certificato per le connessioni tra Zabbix server e il database, modificare il file di configurazione del server:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Crittografia con verifica completa

Configurazione MySQL

Impostare l'opzione di configurazione del server MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) su:

```
[mysqld]
...
#### in questi esempi le chiavi si trovano nella directory datadir di MySQL CE
ssl_ca=ca.pem
ssl_cert=server-cert.pem
```

```
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
...
```

Le chiavi per il server MySQL CE e per il client (frontend di Zabbix) devono essere create manualmente in base alla documentazione di MySQL CE: Creating SSL and RSA certificates and keys using MySQL oppure Creating SSL certificates and keys using openssl

Attention:
Il certificato del server MySQL deve contenere il campo Common Name impostato al nome FQDN, poiché il frontend di Zabbix utilizzerà il nome DNS per comunicare con il database o l'indirizzo IP dell'host del database.

Creare l'utente MySQL:

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

Verificare se è possibile accedere con quell'utente:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

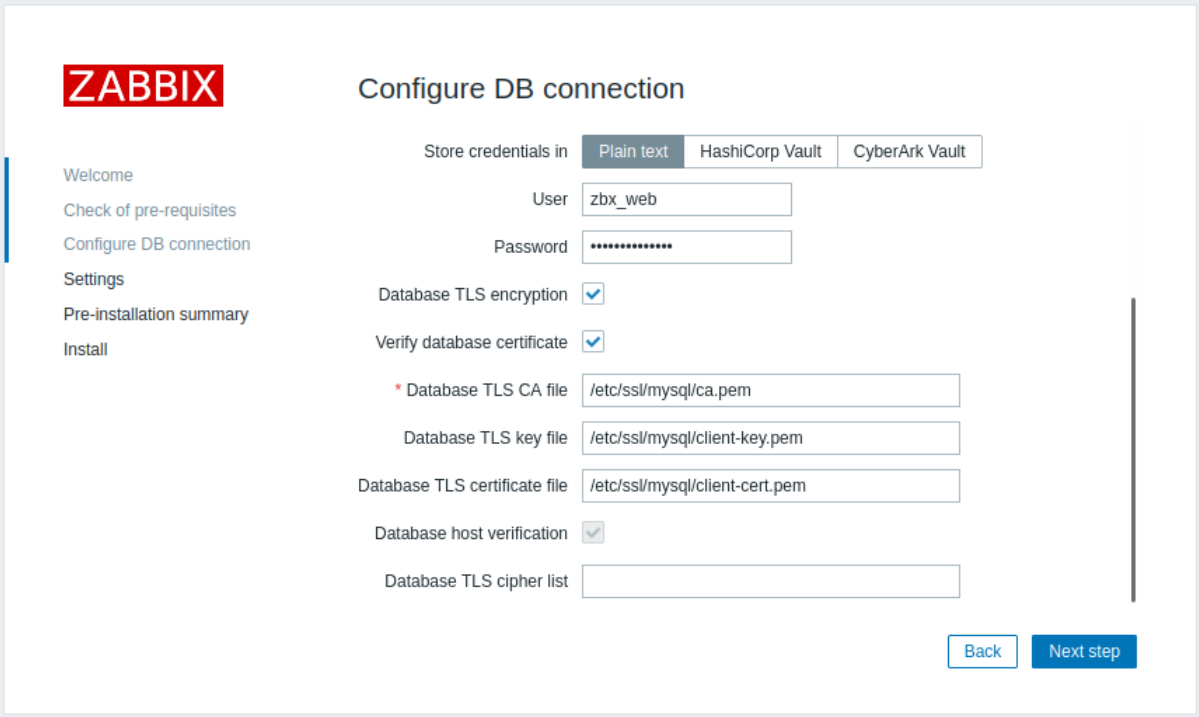
frontend

Per abilitare la crittografia con verifica completa per le connessioni tra il frontend di Zabbix e il database:

- Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
- Specificare il percorso del *file della chiave TLS del database*
- Specificare il percorso del *file CA TLS del database*
- Specificare il percorso del *file del certificato TLS del database*

Si noti che *Verifica host del database* è selezionato e disabilitato: questo passaggio non può essere ignorato per MySQL.

Warning:
Se il campo *Elenco di cifrari TLS del database* viene lasciato vuoto, verranno abilitati i cifrari comuni consentiti sia dal frontend (client) sia dal server. In alternativa, i cifrari possono essere impostati esplicitamente, in conformità con i requisiti di configurazione dei cifrari.



In alternativa, questa impostazione può essere configurata nel file di configurazione del frontend (`zabbix.conf.php`):

```
...
// Utilizzato per la connessione TLS con un elenco di cifrari definito esplicitamente.
$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
...
// oppure
...
// Utilizzato per la connessione TLS senza un elenco di cifrari definito: selezionato dal server MySQL
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Per abilitare la crittografia con verifica completa per le connessioni tra il server Zabbix e il database, modificare il file di configurazione del server:

```
...
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
...
```

Configurazione della crittografia PostgreSQL

Panoramica

Questa sezione fornisce diversi esempi di configurazione della crittografia per CentOS 8.2 e PostgreSQL 13.

> **Note:**
> La connessione tra frontend di Zabbix e PostgreSQL non può essere crittografata (i parametri nell'interfaccia grafica sono disabilitati) se il valore del campo *Database host* inizia con una barra obliqua oppure se il campo è vuoto.

Prerequisiti

Installa il database PostgreSQL utilizzando il repository ufficiale.

PostgreSQL non è configurato per accettare connessioni TLS per impostazione predefinita. Segui le istruzioni della documentazione di PostgreSQL per la preparazione dei certificati con postgresql.conf e anche per il controllo degli accessi degli utenti tramite `ph_hba.conf`.

Per impostazione predefinita, il socket di PostgreSQL è associato a localhost; per consentire connessioni remote di rete, deve ascoltare sulla reale interfaccia di rete.

Le impostazioni di PostgreSQL per tutte le modalità possono essere simili a queste:

`/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'
...
```

Per il controllo degli accessi, modifica `/var/lib/pgsql/13/data/pg_hba.conf`:

```
...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

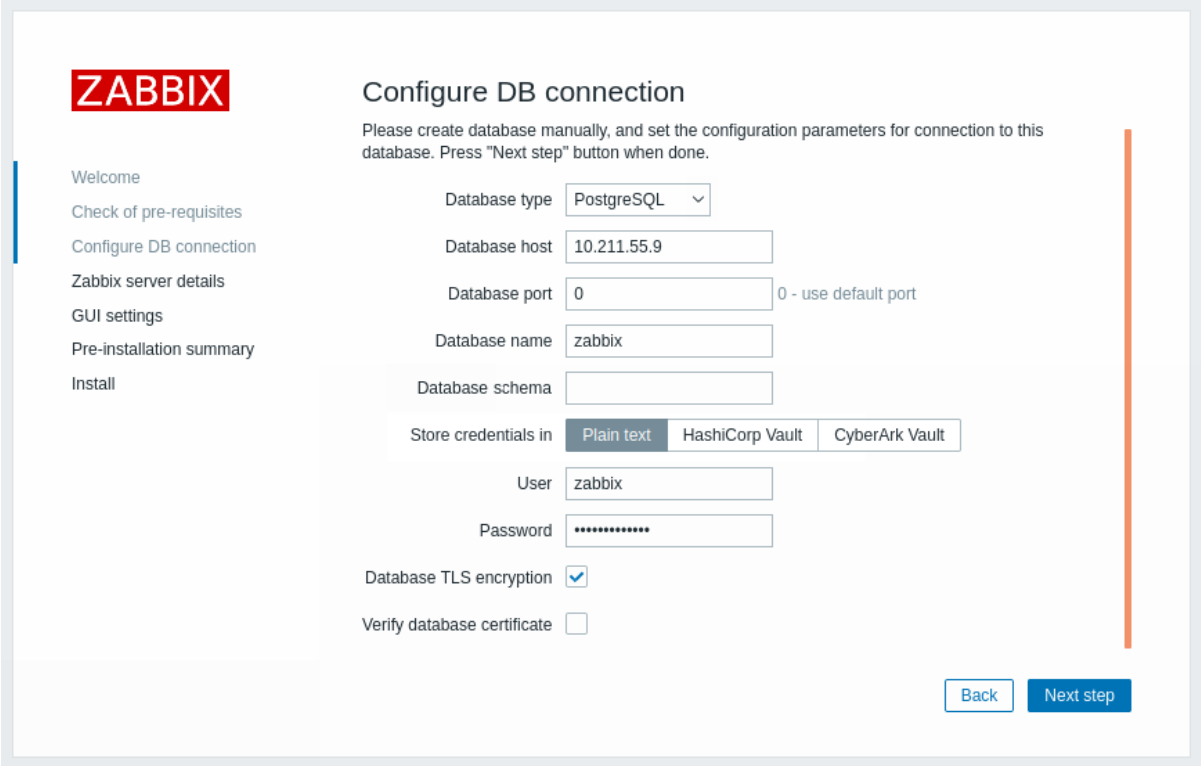
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Crittografia solo del trasporto

Frontend

Per abilitare la crittografia solo del trasporto per le connessioni tra il frontend di Zabbix e il database:

- Seleziona *Database TLS encryption*
- Lascia deselezionato *Verify database certificate*



Server

Per abilitare la crittografia solo del trasporto per le connessioni tra il server e il database, modificare il file di configurazione del server:

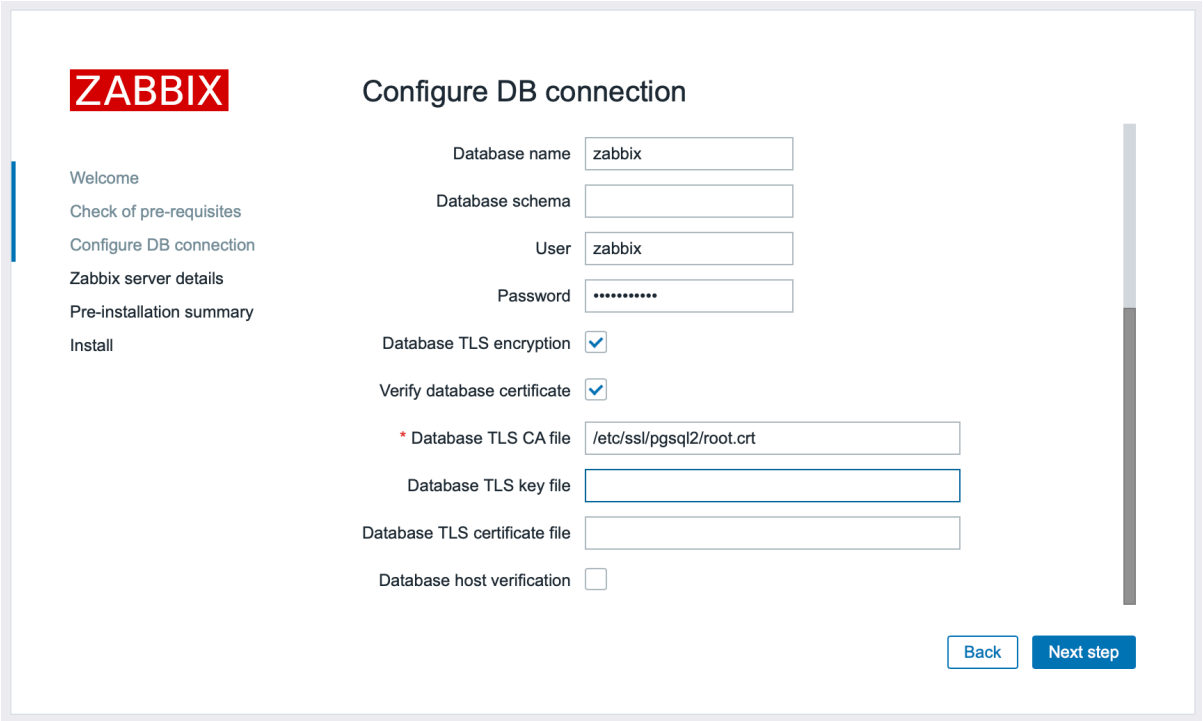
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Crittografia con verifica dell'autorità di certificazione

frontend

Per abilitare la crittografia con verifica dell'autorità di certificazione per le connessioni tra il frontend di Zabbix e il database:

- Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
- Specificare il percorso del *file CA TLS del database*



In alternativa, è possibile impostare questa configurazione nel file di configurazione del frontend (`zabbix.conf.php`):

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Server

Per abilitare la crittografia con verifica del certificato per le connessioni tra il server Zabbix e il database, modifica il file di configurazione del server (sostituisci `<strong_password>` con una password complessa e univoca):

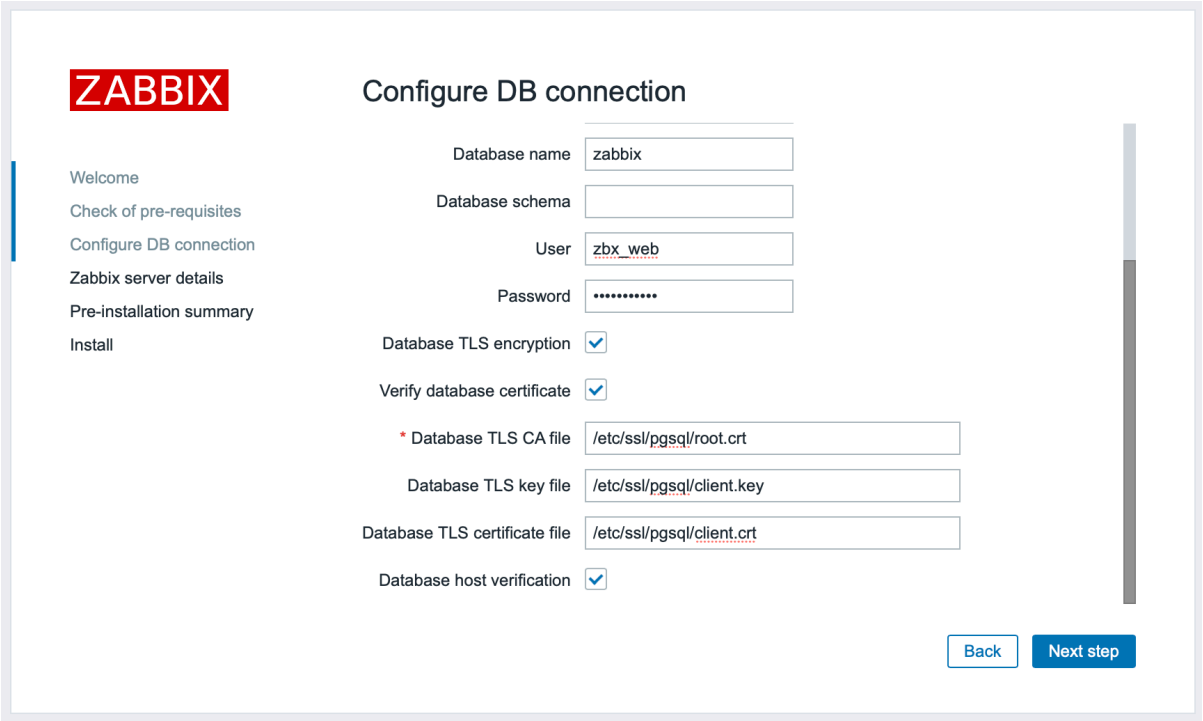
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Crittografia con verifica completa

frontend

Per abilitare la crittografia con il certificato e la verifica dell'identità dell'host del database per le connessioni tra il frontend di Zabbix e il database:

- Selezionare *Crittografia TLS del database* e *Verifica certificato del database*
- Specificare il percorso del *file della chiave TLS del database*
- Specificare il percorso del *file CA TLS del database*
- Specificare il percorso del *file del certificato TLS del database*
- Selezionare *Verifica dell'host del database*



In alternativa, è possibile configurare questa impostazione nel file di configurazione del frontend (`zabbix.conf.php`):

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Per abilitare la crittografia con la verifica del certificato e dell'identità dell'host del database per le connessioni tra il server Zabbix e il database, modificare il file di configurazione del server:

```
...
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
...
```

Connessione sicura tra frontend e server

Panoramica

Questa sezione fornisce i passaggi di configurazione di Zabbix e alcuni esempi di configurazione per connessioni TLS sicure tra il frontend di Zabbix e il server Zabbix.

Configurazione

Per impostazione predefinita, la comunicazione tra il frontend Zabbix e il server Zabbix non è crittografata.
Per una maggiore sicurezza, abilita TLS su entrambi i lati.
Di seguito è riportato un esempio del modo più semplice per farlo.

1. Genera certificati e chiavi.

Crea una directory di lavoro:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Crea un certificato CA (modifica il valore `MyZabbixCA` in modo che corrisponda al nome comune effettivo):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Genera una chiave privata e un certificato per il server Zabbix (modifica il valore `zabbix-server.example.com` in modo che corrisponda al nome comune effettivo):

```
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
```

Genera una chiave privata e un certificato per il frontend Zabbix (modifica il valore `zabbix-frontend-node` in modo che corrisponda al nome comune effettivo):

```
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. Imposta le autorizzazioni appropriate.

Per il server Zabbix (modifica il proprietario/gruppo in base all'utente del demone del server Zabbix della tua distribuzione):

```
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
```

Per il frontend (modifica il proprietario/gruppo in base all'utente del webserver della tua distribuzione):

```
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. Configura il server Zabbix.

Nel file di configurazione del server (`zabbix_server.conf`) aggiungi:

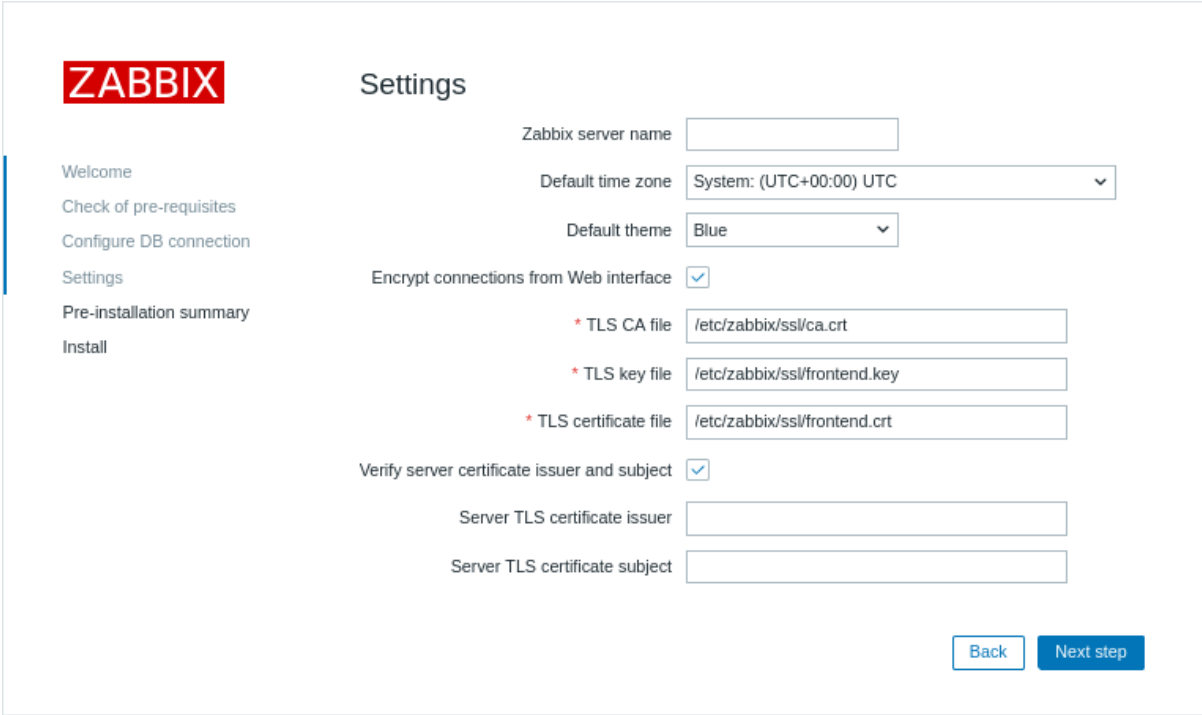
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
### Facoltativo:
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

Quindi riavvia il server:

```
sudo systemctl restart zabbix-server
```

4. Configura il frontend Zabbix.

Durante l'installazione dell'interfaccia web, abilita l'opzione *Crittografa le connessioni dall'interfaccia web* (e l'opzione *Verifica l'emittente e il soggetto del certificato del server*, se necessario) e compila i campi *File CA TLS*, *File della chiave TLS*, *File del certificato TLS* (e i campi *Emittente del certificato TLS del server* e *Soggetto del certificato TLS del server*, se necessario):



Parametro	Descrizione
File CA TLS	Specifica il percorso completo del file del certificato dell'Autorità di certificazione (CA) utilizzato per verificare il certificato del server.
File della chiave TLS	Specifica il percorso completo del file della chiave privata client corrispondente al certificato client.
File del certificato TLS	Specifica il percorso completo del file del certificato client se è richiesta l'autenticazione TLS reciproca.
Emittente del certificato TLS del server	Specifica un Distinguished Name (DN) dell'emittente da confrontare con il certificato del server.
Soggetto del certificato TLS del server	Specifica un Distinguished Name (DN) del soggetto da confrontare con il certificato del server.

Nelle installazioni esistenti, modifica i seguenti parametri nel file di configurazione del frontend:

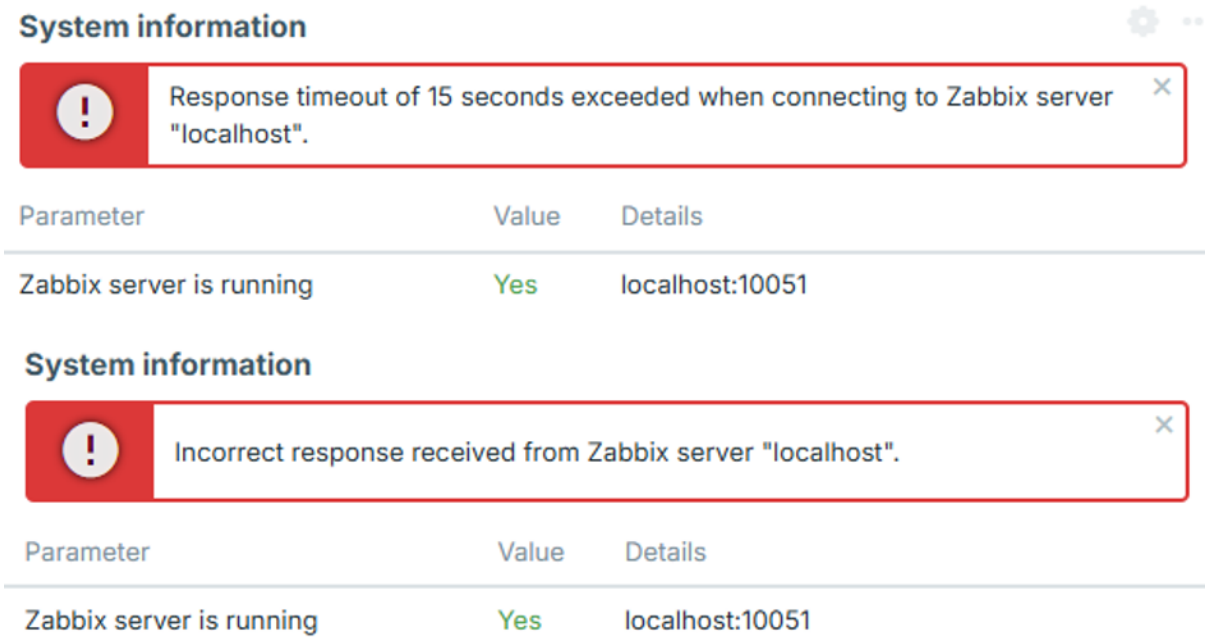
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Facoltativo:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

5. Verifica la crittografia confermando che non siano presenti messaggi di errore nel frontend Zabbix o nel file di log del server Zabbix:

```
tail -f /var/log/zabbix/zabbix_server.log
```

Risoluzione dei problemi

Una configurazione TLS errata o incompleta può causare errori nel frontend di Zabbix oppure indicare che il server Zabbix è in esecuzione, ma che la connessione non viene stabilita correttamente:



Per identificare il problema di configurazione esatto, impostare il livello di log del server Zabbix su 4, riprodurre l'errore nel frontend e controllare il log del server Zabbix per individuare il messaggio di errore corrispondente.

I casi seguenti descrivono gli errori comuni e le relative possibili cause.

L'indirizzo IP del frontend non è consentito

Il log del server Zabbix contiene un errore simile a:

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

La connessione viene interrotta dopo l'handshake TCP perché l'indirizzo IP del frontend non è autorizzato.

Aggiungere l'indirizzo IP del frontend al parametro `FrontendAllowedIP` in `zabbix_server.conf` e riavviare il server Zabbix.

La connessione non crittografata non è consentita

Il log del server Zabbix contiene un errore simile a:

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

Il server Zabbix è configurato per accettare solo connessioni TLS, ad esempio con `TLSFrontendAccept=cert`, ma il frontend sta tentando di connettersi senza crittografia.

Abilitare la crittografia del frontend e configurare i file TLS richiesti in `zabbix.conf.php`, come descritto nel passaggio 4 della Configurazione.

Il frontend tenta TLS mentre il server si aspetta una connessione non crittografata

Il log del server Zabbix contiene un errore simile a:

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

Il messaggio di errore esatto dipende dalla versione di OpenSSL.

Il server Zabbix è configurato per accettare solo connessioni non crittografate, ad esempio con `TLSFrontendAccept=unencrypted`, mentre il frontend tenta di stabilire una connessione TLS.

Configurare il server Zabbix affinché accetti connessioni TLS impostando `TLSFrontendAccept=cert` e definendo `TLSCertFile` e `TLSKeyFile` in `zabbix_server.conf`. In alternativa, disabilitare la crittografia del frontend in `zabbix.conf.php`.

Il certificato del frontend non è considerato attendibile

Il log del server Zabbix contiene un errore simile a:

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

Il frontend ha presentato un certificato, ma il server Zabbix non considera attendibile l'Autorità di certificazione (CA) che lo ha firmato.

Lo stesso problema può comparire anche nelle informazioni di sistema del frontend come:

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

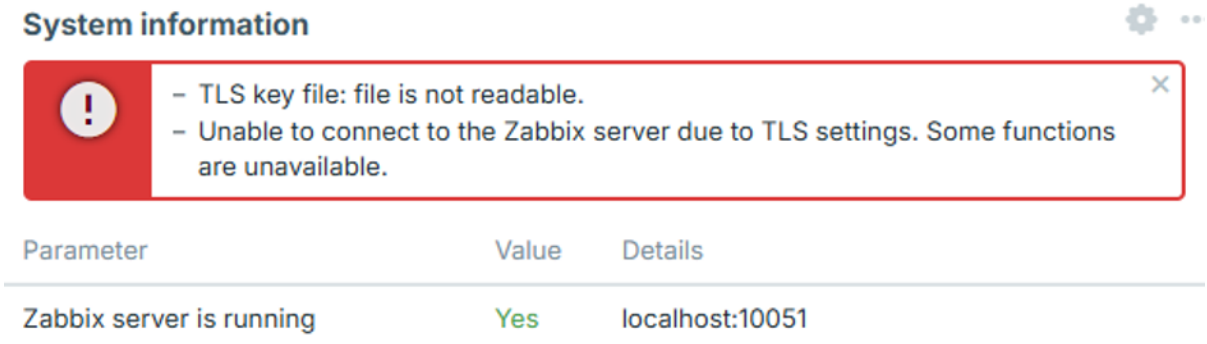
Il frontend potrebbe inoltre visualizzare il messaggio ”Incorrect response received from Zabbix server ”localhost””.

Verificare che il certificato del frontend sia firmato dalla CA specificata da `TLSCAFile` in `zabbix_server.conf`.

Il file della chiave TLS non è leggibile

Il frontend potrebbe visualizzare ”File della chiave TLS: il file non è leggibile.”

Potrebbe inoltre visualizzare ”Impossibile connettersi al server Zabbix a causa delle impostazioni TLS. Alcune funzioni non sono disponibili.”.

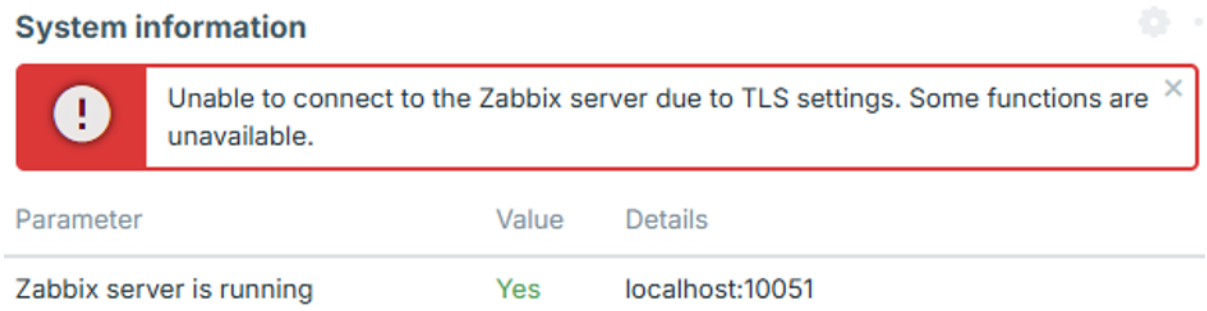


L’utente del web server non dispone di autorizzazioni sufficienti per leggere uno o più file di certificato, chiave o CA configurati in `zabbix.conf.php`. Questo può verificarsi quando i file dei certificati sono archiviati in una directory con accesso limitato, come /root/ o la home directory di un utente.

Spostare i file dei certificati in una directory accessibile al web server e impostare le autorizzazioni appropriate come descritto nel passaggio 2 della Configurazione.

I file TLS non sono validi o danneggiati

Il frontend potrebbe visualizzare ”Impossibile connettersi al server Zabbix a causa delle impostazioni TLS. Alcune funzioni non sono disponibili.”.



Questo può verificarsi quando i file del certificato o della chiave configurati esistono e sono leggibili, ma il loro contenuto è vuoto, danneggiato o comunque non valido. In questo caso, il frontend non stabilisce una connessione al server Zabbix e nel log del server Zabbix non viene visualizzato alcun tentativo di connessione corrispondente.

Verificare che i file del certificato e della chiave contengano dati validi e formattati correttamente e che la chiave privata corrisponda al certificato.

Controllo degli accessi

Panoramica

Questa sezione contiene le best practice per configurare il controllo degli accessi in modo sicuro.

Principio del privilegio minimo

Gli account utente, in ogni momento, dovrebbero operare con il minor numero possibile di privilegi. Ciò significa che gli account utente nel frontend di Zabbix, gli utenti del database o l’utente per i processi di Zabbix server/proxy/agent dovrebbero avere solo i privilegi essenziali per svolgere le funzioni previste.

Attention:
Concedere privilegi aggiuntivi all'utente 'zabbix' gli consentirà di accedere ai file di configurazione ed eseguire operazioni che possono compromettere la sicurezza dell'infrastruttura.

Quando si configurano i privilegi degli account utente, è necessario considerare i tipi di utente del frontend di Zabbix. Si noti che, sebbene il tipo di utente *Admin* abbia meno privilegi del tipo di utente *Super Admin*, può comunque gestire la configurazione ed eseguire script personalizzati.

Note:
Alcune informazioni sono disponibili anche per gli utenti non privilegiati. Ad esempio, mentre *Alerts* → *Scripts* è disponibile solo per gli utenti *Super Admin*, gli script possono essere recuperati anche tramite Zabbix API. In questo caso, limitare i permessi degli script ed escludere le informazioni sensibili dagli script (ad esempio, le credenziali di accesso) può aiutare a evitare l'esposizione di informazioni sensibili disponibili negli script globali.

Utente sicuro per Zabbix agent

Per impostazione predefinita, i processi di Zabbix server, proxy e agent (o agent 2) condividono un unico utente `zabbix`. Per impedire a Zabbix agent/agent 2 (in esecuzione sulla stessa macchina di server/proxy) di accedere a dettagli sensibili nella configurazione di server/proxy (ad esempio, le credenziali del database), l'agent dovrebbe essere eseguito con un utente diverso:

Per Zabbix agent:

1. Creare un gruppo e utente sicuri (ad esempio, `zabbix-agent`).
2. Impostare questo utente nel parametro User del file di configurazione dell'agent.
3. Riavviare l'agent per ridurre i privilegi al nuovo utente.

Per Zabbix agent 2, la configurazione deve essere applicata a livello di service, perché il file di configurazione di agent 2 non supporta il parametro `User`. Per un esempio, vedere ZBX-26442.

Revocare l'accesso in scrittura alla configurazione SSL (Windows)

Se hai compilato Zabbix agent su Windows, con OpenSSL situato in una directory non protetta (ad esempio, `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static` o `C:\dev\openssl`), assicurati di revocare l'accesso in scrittura a questa directory per gli utenti non amministratori. In caso contrario, l'agent carica le impostazioni SSL da un percorso che può essere modificato da utenti non privilegiati, con conseguente potenziale vulnerabilità di sicurezza.

Rafforzamento della sicurezza dei componenti Zabbix

Alcune funzionalità possono essere disattivate per rafforzare la sicurezza dei componenti Zabbix:

- L'esecuzione globale degli script sul server Zabbix può essere disabilitata impostando `EnableGlobalScripts=0` nella configurazione del server.
- L'esecuzione globale degli script sul proxy Zabbix è disabilitata per impostazione predefinita (può essere abilitata impostando `EnableRemoteCommands=1` nella configurazione del proxy).
- L'esecuzione globale degli script sugli agent Zabbix è disabilitata per impostazione predefinita (può essere abilitata aggiungendo un parametro `AllowKey=system.run[<command>,*]` o `AllowKeyRegexp=^system\.run\[.*\]$` per ogni comando consentito nella configurazione dell'agent).
- L'accesso alla configurazione dell'autenticazione HTTP degli utenti è abilitato per impostazione predefinita; può essere disabilitato impostando `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` nella configurazione del frontend.
- L'accesso alla configurazione dei moduli è abilitato per impostazione predefinita; può essere disabilitato impostando `$ZBX_FEATURE_FLAGS['modules_config_enabled'] = false` nella configurazione del frontend.
- La modifica di specifici tipi di media può essere limitata impostando `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]` nella configurazione del frontend. Valori possibili: `'email'`, `'script'`, `'sms'`, `'webhook'` e `'push'`. È possibile specificare uno o più valori.

La reinstallazione del frontend eseguendo `setup.php` è limitata se esiste il file di configurazione `zabbix.conf.php`.

Accesso ai percorsi UNC su Windows tramite Zabbix agent

Gli Zabbix agent su Windows seguono i percorsi UNC (condivisioni SMB come `\\server\share\file.txt`) in item come `vfs.file.*`, `vfs.dir.*`, `modbus.get` e `perf_counter*`. In alcuni contesti, ciò può rappresentare un rischio per la sicurezza.

Quando a Windows viene richiesto di accedere a un percorso UNC, tenta di autenticarsi su quel server. Ciò significa che una richiesta dannosa a Zabbix agent può esporre l'hash NTLM al server del richiedente. Gli utenti possono mitigare questo rischio configurando, secondo necessità, i parametri di configurazione AllowKey, AllowKeyRegexp, DenyKey e DenyKeyRegexp.

Protezione di MySQL/MariaDB

Panoramica

Questa sezione contiene le best practice per proteggere un database MySQL/MariaDB.

> **Note:**
> Per una configurazione di base, vedere le istruzioni predefinite per la creazione del database MySQL/MariaDB, che includono la creazione dell'utente 'zabbix' con privilegi completi sul database Zabbix. Questo utente è il proprietario del database e dispone anche dei privilegi necessari per modificare la struttura del database durante l'aggiornamento di Zabbix.

Per migliorare la sicurezza, è consigliato creare ulteriori ruoli e utenti del database con privilegi minimi. Questi ruoli e utenti devono essere configurati in base al principio del privilegio minimo, ossia devono avere solo i privilegi essenziali per svolgere le funzioni previste.

Creazione dei ruoli utente

Creare i seguenti ruoli con i privilegi corrispondenti:

- **zbx_srv** - ruolo per l'esecuzione di Zabbix server e proxy:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - ruolo per l'esecuzione di Zabbix frontend e API:

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - ruolo per il backup delle tabelle:

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

> **Note:**
> Il ripristino e l'aggiornamento delle tabelle devono essere eseguiti dal proprietario del database.

- **zbx_part** - ruolo con un insieme ridotto di privilegi per il partizionamento del database; si noti che questo ruolo può essere creato solo dopo la creazione del database, poiché concede privilegi su tabelle specifiche del database:

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

Dopo aver creato i ruoli, questi possono essere assegnati agli utenti.

Assegnazione dei ruoli utente

Per assegnare i ruoli utente creati, creare gli utenti e assegnare loro i ruoli pertinenti. Sostituire <user>, <host>, <role> e <password> con i valori appropriati. Assicurarsi che <password> sia complessa e univoca.

```
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;
```

Ad esempio, per creare e assegnare il ruolo per l'esecuzione di Zabbix server e proxy:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

Proteggere PostgreSQL/TimescaleDB

Panoramica

Questa sezione contiene le best practice per proteggere un database PostgreSQL.

> **Note:**
> Per una configurazione di base, vedere le istruzioni predefinite per la creazione del database PostgreSQL, che includono la creazione dell'utente 'zabbix' con privilegi completi sul database Zabbix. Questo utente è il proprietario del database e dispone anche dei privilegi necessari per modificare la struttura del database durante l'aggiornamento di Zabbix.

Per migliorare la sicurezza, si consiglia di configurare un uso sicuro dello schema, oltre a creare ruoli e utenti aggiuntivi del database con privilegi minimi. Questi ruoli e utenti devono essere configurati in base al principio del privilegio minimo, ossia devono avere solo i privilegi essenziali per svolgere le funzioni previste.

Configurazione del database

Creare l'utente che sarà il proprietario del database e creare il database Zabbix; il proprietario del database è l'utente specificato durante la creazione del database:

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

> **Note:**
> Un'installazione pulita o un aggiornamento del database deve essere eseguito dal proprietario del database. Questo perché il diritto di eliminare un oggetto del database o modificarne la definizione è un privilegio intrinseco del proprietario del database e non può essere concesso né revocato.

> **Attention:**
> I seguenti comandi in questa pagina devono essere eseguiti mentre la connessione a PostgreSQL è effettuata specificamente al database `zabbix`.

Creare lo schema `zabbix` e impostare il proprietario del database (`usr_owner`) come proprietario di questo schema:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Configurare un modello di utilizzo sicuro dello schema usage pattern:

```
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";
```

Dopo aver configurato il database, procedere alla creazione dei ruoli utente.

Creazione dei ruoli utente

Creare i seguenti ruoli con i privilegi corrispondenti:

- **zbx_srv** - ruolo per l'esecuzione di Zabbix server e 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** - ruolo per l'esecuzione di Zabbix frontend e 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** - ruolo per il backup delle tabelle:

```
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:**
> Il ripristino delle tabelle è possibile solo da parte del proprietario del database.

Dopo aver creato i ruoli, è possibile assegnarli agli utenti.

Assegnazione dei ruoli utente

Per assegnare i ruoli utente creati, creare gli utenti e assegnare loro i ruoli pertinenti. Sostituire <user>, <role> e <password> con i valori appropriati. Assicurarsi che <password> sia complessa e univoca.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Ad esempio, per creare e assegnare il ruolo per l'esecuzione di Zabbix server e proxy:

```
CREATE USER usr_srv WITH ENCRYPTED password 'strong-password';
GRANT zbx_srv TO usr_srv;
```

Partizionamento del database con TimescaleDB

Il partizionamento del database è facilitato da TimescaleDB. Per utilizzare TimescaleDB, il server Zabbix richiede i privilegi del proprietario del database.

Se lo schema PostgreSQL zabbix è già stato creato nel database zabbix, è possibile abilitare TimescaleDB con il seguente comando:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

Crittografia

Panoramica

Questa sezione contiene le best practice per configurare la crittografia in modo sicuro.

Configurazione di SSL per il frontend di Zabbix

Sui sistemi basati su RHEL, installare il pacchetto mod_ssl:

```
dnf install mod_ssl
```

Creare una directory per le chiavi SSL:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Creare il certificato SSL:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Compilare i prompt in modo appropriato. La riga più importante è quella che richiede il Common Name. È necessario inserire il nome di dominio che si desidera associare al proprio server. In alternativa, se non si dispone di un nome di dominio, è possibile inserire l'indirizzo IP pubblico.

```
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 []:
```

Modificare il file di configurazione SSL di Apache (`/etc/httpd/conf.d/ssl.conf`):

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

Riavviare il servizio Apache per applicare le modifiche:

```
systemctl restart httpd.service
```

Proteggere il server web

Panoramica

Questa sezione contiene le best practice per configurare il frontend in modo sicuro.

Forzare il reindirizzamento dell'URL root a Zabbix SSL

Sui sistemi basati su RHEL, aggiungere un virtual host alla configurazione di Apache (`/etc/httpd/conf/httpd.conf`) e impostare un reindirizzamento permanente della document root all'URL SSL di Zabbix. Si noti che `example.com` deve essere sostituito con il nome effettivo del server.

```
### Aggiungere le righe:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Riavviare il servizio Apache per applicare le modifiche:

```
systemctl restart httpd.service
```

Abilitazione di HTTP Strict Transport Security (HSTS) sul server web

Per proteggere il frontend di Zabbix dagli attacchi di downgrade del protocollo, consigliamo di abilitare la policy HSTS sul server web.

Per abilitare la policy HSTS per il frontend di Zabbix nella configurazione di Apache, seguire questi passaggi:

1. Individuare il file di configurazione del virtual host:

- `/etc/httpd/conf/httpd.conf` sui sistemi basati su RHEL
- `/etc/apache2/sites-available/000-default.conf` su Debian/Ubuntu

2. Aggiungere la seguente direttiva al file di configurazione del virtual host:

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Riavviare il servizio Apache per applicare le modifiche:

```
### Su sistemi basati su RHEL:
systemctl restart httpd.service

### Su Debian/Ubuntu
systemctl restart apache2.service
```

Applicazione di cookie di sessione Secure e SameSite in Zabbix

Quando si configura Zabbix, è essenziale applicare gli attributi secure e SameSite ai cookie di sessione per migliorare la sicurezza e prevenire attacchi di cross-site request forgery (CSRF). Tuttavia, l'impostazione di `SameSite=Strict` può causare problemi in alcuni scenari, ad esempio:

- I widget URL della dashboard mostrano "user not logged in" quando si incorporano iframe dello stesso dominio.
- Gli utenti che accedono alla dashboard tramite HTTP invece di HTTPS possono riscontrare problemi di accesso.
- Impossibilità di condividere URL verso sezioni specifiche del menu di Zabbix o verso host.

Per attenuare questi problemi, gli utenti dovrebbero avere un modo per modificare la policy SameSite.

1. Cookie Secure

L'impostazione del flag `secure` garantisce che i cookie vengano trasmessi solo tramite HTTPS, evitando l'esposizione su connessioni non cifrate.

Per abilitare i cookie secure in Zabbix, aggiungere o modificare la seguente impostazione nella configurazione del web server:

Per Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Per Nginx:

```
proxy_cookie_path / "/; Secure";
```

Assicurarsi che il frontend di Zabbix sia accessibile tramite HTTPS; in caso contrario, i cookie con il flag `Secure` non verranno inviati.

2. Configurazione dell'attributo SameSite

Le impostazioni del web server possono anche applicare l'attributo SameSite:

Per Apache:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Per Nginx (versione 1.19.3+):

```
proxy_cookie_flags ~ samesite=Strict; # Replace ~ with 'zbx_session' for specificity
```

Abilitare Content Security Policy (CSP) sul web server

Per proteggere il frontend di Zabbix da Cross Site Scripting (XSS), injection di dati e attacchi simili, consigliamo di abilitare Content Security Policy sul web server. Per farlo, configurare il web server in modo che restituisca l'intestazione HTTP.

> **Attention:**
> La seguente configurazione dell'intestazione CSP è valida solo per l'installazione predefinita del frontend di Zabbix e per i casi in cui tutto il contenuto proviene dal dominio del sito (esclusi i sottodomini). Potrebbe essere necessaria una configurazione diversa dell'intestazione CSP se, ad esempio, si sta configurando il widget *URL* per visualizzare contenuti dai sottodomini del sito o da domini esterni, passando da *OpenStreetMap* a un altro motore di mappe, oppure aggiungendo CSS o widget esterni. Se si utilizza il metodo di autenticazione multi-factor authentication Duo Universal Prompt, assicurarsi di aggiungere "duo.com" alla direttiva CSP nel file di configurazione del virtual host.

Per abilitare CSP per il frontend di Zabbix nella configurazione di Apache, seguire questi passaggi:

1. Individuare il file di configurazione del virtual host:

- `/etc/httpd/conf/httpd.conf` sui sistemi basati su RHEL
- `/etc/apache2/sites-available/000-default.conf` su Debian/Ubuntu

2. Aggiungere la seguente direttiva al file di configurazione del virtual host:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Riavviare il servizio Apache per applicare le modifiche:

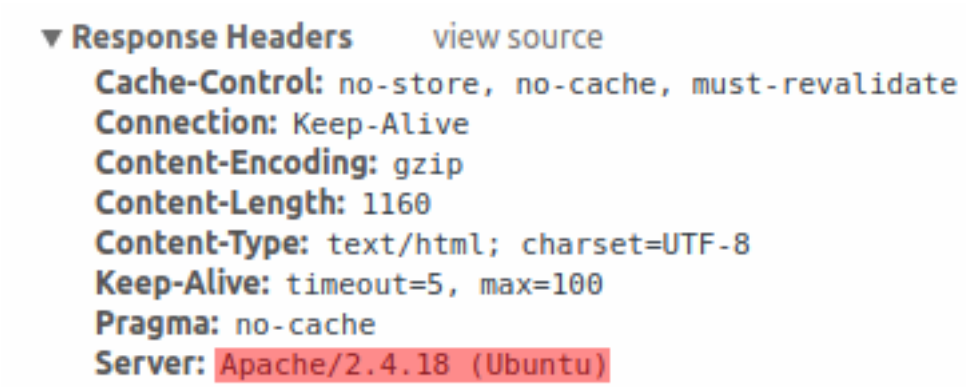
```
### Sui sistemi basati su RHEL:
systemctl restart httpd.service

### Su Debian/Ubuntu
systemctl restart apache2.service
```

Disabilitazione dell'esposizione delle informazioni del web server

Per migliorare la sicurezza, è consigliabile disabilitare tutte le firme del web server.

Per impostazione predefinita, il web server espone la firma del software:



La firma può essere disabilitata aggiungendo i seguenti parametri al file di configurazione di Apache:

```
ServerSignature Off
ServerTokens Prod
```

La firma PHP (intestazione HTTP X-Powered-By) può essere disabilitata modificando il file di configurazione `php.ini` (per impostazione predefinita, la firma è disabilitata):

```
expose_php = Off
```

È necessario riavviare il web server affinché le modifiche al file di configurazione vengano applicate.

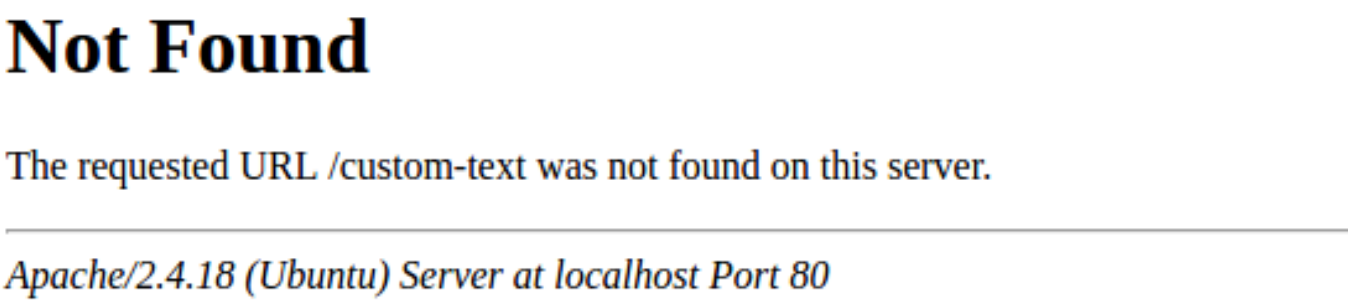
Per una maggiore sicurezza, è possibile utilizzare lo strumento *mod_security* con Apache (pacchetto *libapache2-mod-security2*). Questo strumento consente di rimuovere la firma del server, invece di rimuovere solo la versione dalla firma del server. La firma del server può essere modificata impostando "SecServerSignature" su qualsiasi valore desiderato dopo l'installazione di *mod_security*.

Consultare la documentazione del proprio web server per informazioni su come rimuovere o modificare le firme del software.

Disabilitazione delle pagine di errore predefinite del web server

Per evitare l'esposizione di informazioni, si consiglia di disabilitare le pagine di errore predefinite.

Per impostazione predefinita, un web server utilizza pagine di errore integrate:



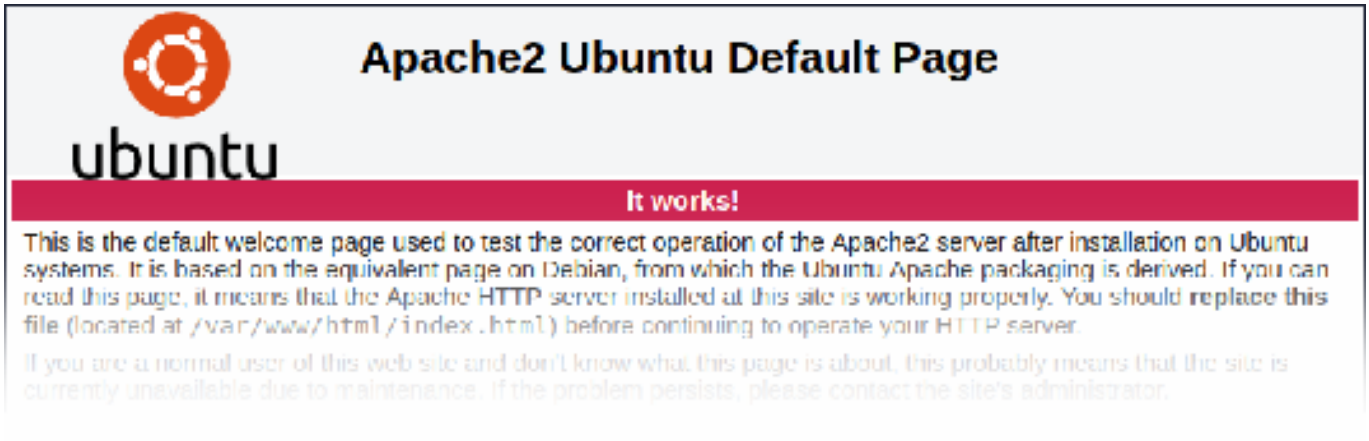
Queste pagine di errore predefinite devono essere sostituite/rimosse. Ad esempio, la direttiva "ErrorDocument" può essere utilizzata per definire una pagina/un testo di errore personalizzato per il web server Apache.

Consultare la documentazione del proprio web server per informazioni su come sostituire/rimuovere le pagine di errore predefinite.

Rimozione della pagina di test del web server

Per evitare l'esposizione di informazioni, si consiglia di rimuovere la pagina di test del web server.

Per impostazione predefinita, la webroot del web server Apache contiene la pagina di test `index.html`:



Consultare la documentazione del proprio web server per informazioni su come rimuovere le pagine di test predefinite.

Impostare l'intestazione HTTP X-Frame-Options

Per impostazione predefinita, Zabbix è configurato con il parametro *Use X-Frame-Options HTTP header* impostato su `SAMEORIGIN`. Ciò significa che il contenuto può essere caricato solo in un frame che ha la stessa origine della pagina stessa.

Gli elementi del frontend di Zabbix che recuperano contenuti da URL esterni (in particolare, il widget dashboard widget URL) visualizzano il contenuto recuperato in una sandbox con tutte le restrizioni di sandboxing abilitate.

Queste impostazioni migliorano la sicurezza del frontend di Zabbix e offrono protezione contro gli attacchi XSS e clickjacking. Gli utenti *Super admin* possono modificare i parametri *Use iframe sandboxing* e *Use X-Frame-Options HTTP header* secondo necessità. Valutare attentamente rischi e benefici prima di modificare le impostazioni predefinite. Non è consigliato disattivare completamente il sandboxing iframe o l'intestazione HTTP X-Frame-Options.

Nascondere il file con l'elenco delle password comuni

Per aumentare la complessità degli attacchi di brute force sulle password, si consiglia di limitare l'accesso al file `ui/data/top_passwords.tx`
Questo file contiene un elenco delle password più comuni e specifiche del contesto e impedisce agli utenti di impostare tali password (se il parametro *Avoid easy-to-guess passwords* è abilitato nella password policy).

Per limitare l'accesso al file `top_passwords.txt`, modifica la configurazione del tuo web server.

Su Apache, l'accesso ai file può essere limitato usando il file `.htaccess`:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

Su NGINX, l'accesso ai file può essere limitato usando la direttiva `location`:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

Archiviazione dei segreti

Panoramica Zabbix può essere configurato per recuperare informazioni sensibili da un vault sicuro. Sono supportati i seguenti servizi di gestione dei secret: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

I secret possono essere utilizzati per recuperare:

- valori delle macro utente
- credenziali di accesso al database

Zabbix fornisce accesso in sola lettura ai secret in un vault, presupponendo che i secret siano gestiti da un'altra persona.

Per informazioni sulla configurazione specifica del provider vault, vedere:

- Configurazione HashiCorp
- Configurazione CyberArk

Memorizzazione nella cache dei valori segreti Per impostazione predefinita, i valori delle macro segrete del vault vengono recuperati dal server Zabbix a ogni aggiornamento dei dati di configurazione e quindi memorizzati nella cache di configurazione. Il proxy Zabbix riceve i valori delle macro segrete del vault dal server Zabbix a ogni sincronizzazione della configurazione e li memorizza nella propria cache di configurazione.

> **Attention:**
> La crittografia deve essere abilitata tra il server Zabbix e il proxy; in caso contrario, viene registrato un messaggio di avviso dal server.

È inoltre possibile configurare il recupero indipendente dei valori delle macro da parte del server Zabbix e del proxy Zabbix.

Per attivare manualmente l'aggiornamento dei valori segreti memorizzati nella cache da un vault, utilizzare l'opzione della riga di comando `secrets_reload`.

La memorizzazione nella cache delle credenziali del database del frontend Zabbix è disabilitata per impostazione predefinita, ma può essere abilitata impostando l'opzione `$DB['VAULT_CACHE'] = true` nel file di configurazione del frontend. Le credenziali verranno memorizzate in una cache locale utilizzando la directory dei file temporanei del filesystem. Il web server deve consentire la scrittura in una cartella temporanea privata (ad esempio, per Apache deve essere impostata l'opzione di configurazione `PrivateTmp=True`). Per controllare la frequenza di aggiornamento/invalidazione della cache dei dati, utilizzare la costante `ZBX_DATA_CACHE_TTL`.

Configurazione TLS Per configurare TLS per la comunicazione tra i componenti Zabbix e il vault, aggiungi un certificato firmato da un'autorità di certificazione (CA) all'archivio CA predefinito a livello di sistema.
Per usare un'altra posizione, specifica la directory nel parametro di configurazione SSLCALocation di Zabbix server/proxy, posiziona il file del certificato all'interno di quella directory, quindi esegui il comando CLI:

```
c_rehash .
```

Configurazione di CyberArk

Questa sezione spiega come configurare Zabbix per recuperare i segreti da CyberArk Vault CV12.

Il vault deve essere installato e configurato come descritto nella documentazione ufficiale di CyberArk.

Per informazioni sulla configurazione di TLS in Zabbix, vedere *Archiviazione dei segreti*.

Credenziali del database

L'accesso a un secret contenente le credenziali del database viene configurato separatamente per ogni componente Zabbix.

Server e proxy

Per ottenere le credenziali del database dal vault per server o proxy Zabbix, specificare i seguenti parametri di configurazione nel file di configurazione:

- `Vault` - quale provider del vault utilizzare;
- `VaultURL` - URL HTTP(S) del server del vault;
- `VaultDBPath` - query al secret del vault contenente le credenziali del database, che verranno recuperate tramite le chiavi "Content" e "UserName" (questa opzione può essere utilizzata solo se DBUser e DBPassword non sono specificati);
- `VaultTLSCertFile`, `VaultTLSKeyFile` - nomi dei file del certificato SSL e della chiave; la configurazione di queste opzioni non è obbligatoria, ma è fortemente consigliata;
- `VaultPrefix` - prefisso personalizzato per il percorso o la query del vault, a seconda del vault; se non specificato, verrà utilizzato il valore predefinito più appropriato.

> **Attention:**
> I parametri di configurazione `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` e `VaultPrefix` vengono utilizzati anche per l'autenticazione al vault durante l'elaborazione delle macro secret del vault da parte del server Zabbix (e del proxy Zabbix, se configurato). Il server Zabbix e i proxy Zabbix non apriranno le macro secret del vault che contengono le credenziali DB provenienti da VaultDBPath.

Il server Zabbix e il proxy Zabbix leggono i parametri di configurazione relativi al vault dai file `zabbix_server.conf` e `zabbix_proxy.conf` all'avvio.

Esempio

1. In `zabbix_server.conf`, specificare i seguenti parametri:

```
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 invierà la seguente richiesta API al 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. La risposta del vault conterrà le chiavi "Content" e "UserName":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

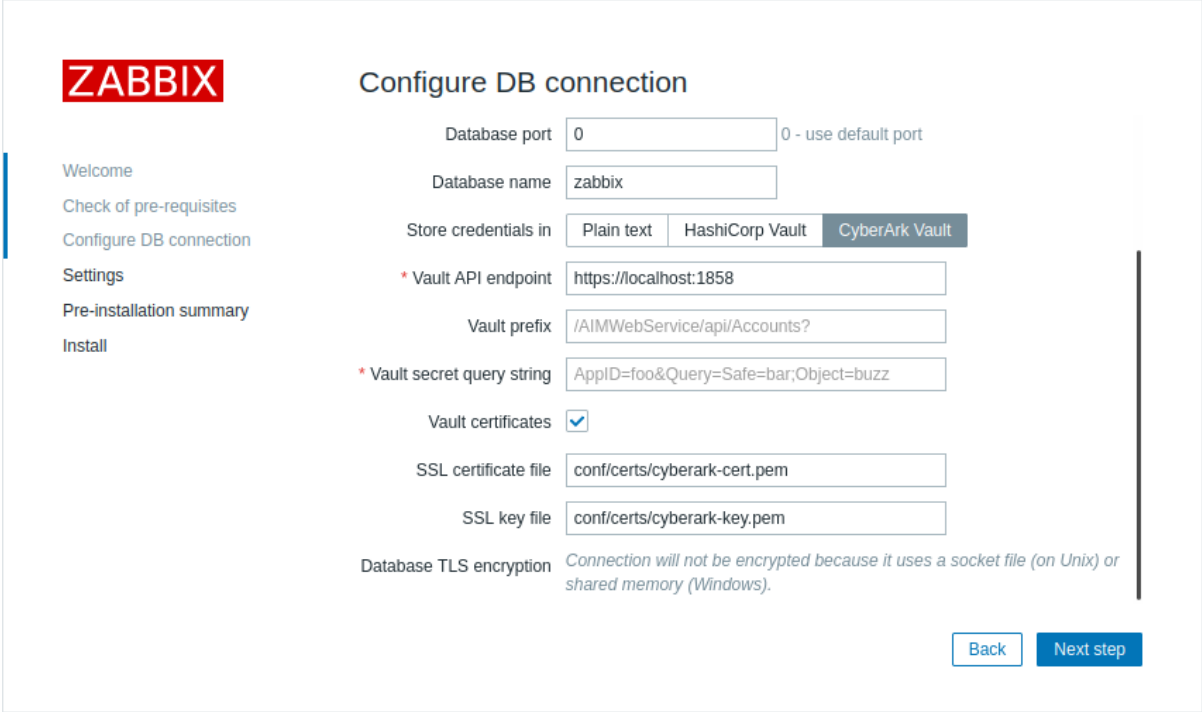
4. Di conseguenza, Zabbix utilizzerà le seguenti credenziali per l'autenticazione al database:

- Username: <username>
- Password: <password>

frontend

Per ottenere le credenziali del database dal vault per il frontend di Zabbix, specificare i seguenti parametri durante l'installazione del frontend.

1. Nel passaggio *Configure DB Connection*, impostare il parametro *Store credentials in* su *CyberArk Vault*.



2. Compilare quindi i parametri aggiuntivi:

Parametro	Obbligatorio	Valore predefinito	Descrizione
Vault API endpoint	sì	https://localhost:1858	Specificare l'URL per la connessione al vault nel formato `scheme://host:port`

Parametro	Obbligatorio	Valore predefinito	Descrizione
Vault prefix	no	/AIMWebService/api/Accounts?	Fornire un prefisso personalizzato per il percorso o la query del vault. Se non specificato, viene utilizzato il valore predefinito.
Vault secret query string	sì		Una query che specifica da dove devono essere recuperate le credenziali del database. Esempio: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		Dopo aver selezionato la casella di controllo, verranno visualizzati ulteriori parametri che consentono di configurare l'autenticazione del client. Sebbene questo parametro sia facoltativo, è fortemente consigliato abilitarlo per la comunicazione con CyberArk Vault.
SSL certificate file	no	conf/certs/cyberark-cert.pem	Percorso del file del certificato SSL. Il file deve essere in formato PEM. Se il file del certificato contiene anche la chiave privata, lasciare vuoto il parametro SSL key file.
SSL key file	no	conf/certs/cyberark-key.pem	Nome del file della chiave privata SSL utilizzato per l'autenticazione del client. Il file deve essere in formato PEM.

Valori delle macro utente

Per usare CyberArk Vault per memorizzare i valori delle macro utente *Vault secret*, assicurati che:

- Zabbix server sia configurato per funzionare con CyberArk Vault;
- il parametro *Vault provider* in *Administration > General > Other* sia impostato su "CyberArk Vault".



Note:
Zabbix server (e Zabbix proxy, se configurato) richiedono l'accesso ai valori delle macro *Vault secret* dal vault. Zabbix frontend non necessita di tale accesso.

Il valore della macro dovrebbe contenere una query (come `query:key`).

Vedi *Vault secret macros* per informazioni dettagliate sull'elaborazione dei valori delle macro da parte di Zabbix.

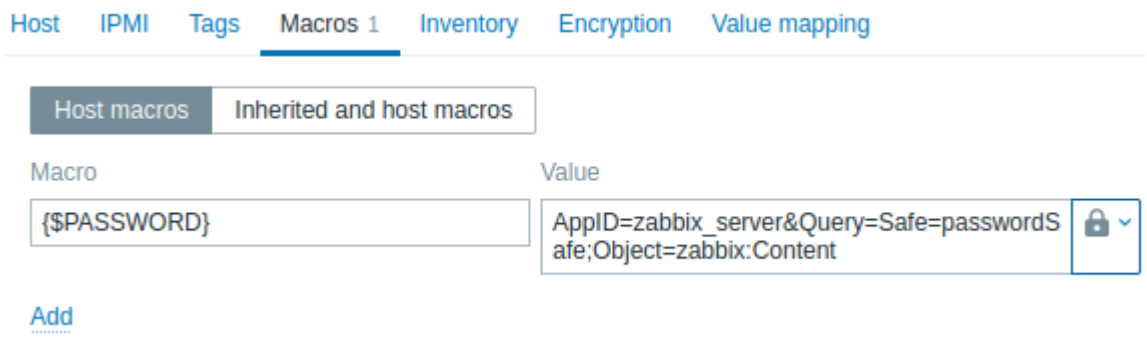
Sintassi delle query

Il simbolo dei due punti (`:`) è riservato alla separazione della query dalla chiave.

Se una query contiene una barra o i due punti, questi simboli devono essere codificati nell'URL (`/` viene codificato come `%2F`, `:` viene codificato come `%3A`).

Esempio

1. In Zabbix, aggiungere una macro utente `{$PASSWORD}` di tipo *Vault secret* e con il valore `AppID=zabbix_server&Query=Safe=pass`



2. Zabbix invierà la seguente richiesta API al 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. La risposta del vault conterrà la chiave `Content`:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Di conseguenza, Zabbix risolverà la macro `{$PASSWORD}` nel valore - `<password>`

Aggiornare la configurazione esistente

Per aggiornare una configurazione esistente per il recupero dei segreti da un CyberArk Vault:

1. Aggiornare i parametri del file di configurazione del server Zabbix o del proxy come descritto nella sezione *Credenziali del database*.
2. Aggiornare i parametri del file di configurazione del frontend di Zabbix:

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. Configurare le macro utente come descritto nella sezione *Valori delle macro utente*, se necessario.

Per aggiornare una configurazione esistente per il recupero dei segreti da un HashiCorp Vault, vedere *Configurazione di HashiCorp*.

Configurazione HashiCorp

Panoramica

Questa sezione spiega come configurare Zabbix per recuperare i secret da HashiCorp Vault KV Secrets Engine - Version 2.

Il vault deve essere distribuito e configurato come descritto nella documentazione ufficiale di HashiCorp.

Per informazioni sulla configurazione di TLS in Zabbix, vedere *Storage of secrets*.

Recupero delle credenziali del database

Per recuperare correttamente un secret contenente le credenziali del database, è necessario configurare entrambi:

- Zabbix server/proxy
- Zabbix frontend

Server/proxy

Per configurare il server o il proxy Zabbix, specificare i seguenti parametri di configurazione nel file di configurazione:

- `Vault` - provider vault da utilizzare.
- `VaultToken` - token di autenticazione vault (vedere il file di configurazione del server/proxy Zabbix per i dettagli); non può essere utilizzato con `VaultAppRoleID`.
- `VaultAppRoleID` e `VaultAppSecretID` - ID del ruolo e ID secret per l'autenticazione vault (vedere il file di configurazione del server/proxy Zabbix per i dettagli); non possono essere utilizzati con `VaultToken`.
- `VaultURL` - URL HTTP(S) del server vault.
- `VaultDBPath` - percorso del secret vault contenente le credenziali del database (questa opzione può essere utilizzata solo se DBUser e DBPassword non sono specificati). Il server o il proxy Zabbix recupererà le credenziali tramite le chiavi "password" e "username".
- `VaultPrefix` - prefisso personalizzato per il percorso o la query vault, a seconda del vault; se non specificato, verrà utilizzato il valore predefinito più appropriato. Il prefisso deve iniziare e terminare con `/`.

Attention:
I parametri di configurazione `Vault`, `VaultToken` (o `VaultAppRoleID`/`VaultAppSecretID`), `VaultURL` e `VaultPrefix` vengono utilizzati anche per l'autenticazione vault durante l'elaborazione delle macro secret vault da parte del server Zabbix (e del proxy Zabbix, se configurato). Il server Zabbix e i proxy non apriranno le macro secret vault che contengono le credenziali DB provenienti da VaultDBPath.

Si consiglia vivamente di utilizzare token diversi per proxy diversi.

Il server Zabbix e il proxy Zabbix leggono i parametri di configurazione relativi al vault da `zabbix_server.conf` e `zabbix_proxy.conf` all'avvio.

Inoltre, il server Zabbix e il proxy Zabbix leggono una volta la variabile d'ambiente `VAULT_TOKEN` durante l'avvio e la annullano, in modo che non sia disponibile tramite gli script forked; si verifica un errore se entrambi i parametri `VaultToken` e `VAULT_TOKEN` contengono un valore.

Esempio

1. In `zabbix_server.conf`, specificare i seguenti parametri.

Usando VaultToken:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

Oppure, usando VaultAppRoleID:

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Eseguire i seguenti comandi CLI per creare il secret richiesto nel vault:

```
#### Abilitare il punto di mount "secret/" se non è già abilitato; notare che deve essere usato "kv-v2".
vault secrets enable -path=secret/ kv-v2

#### Inserire nuovi secret con le chiavi username e password sotto il punto di mount "secret/" e il path '
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Verificare che il secret sia stato aggiunto correttamente.
vault kv get secret/zabbix/database

#### Infine testare con Curl; notare che "data" deve essere aggiunto manualmente dopo il punto di mount e
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Di conseguenza, il server Zabbix recupererà le seguenti credenziali per l'autenticazione al database:

- Username: zabbix
- Password: <password>

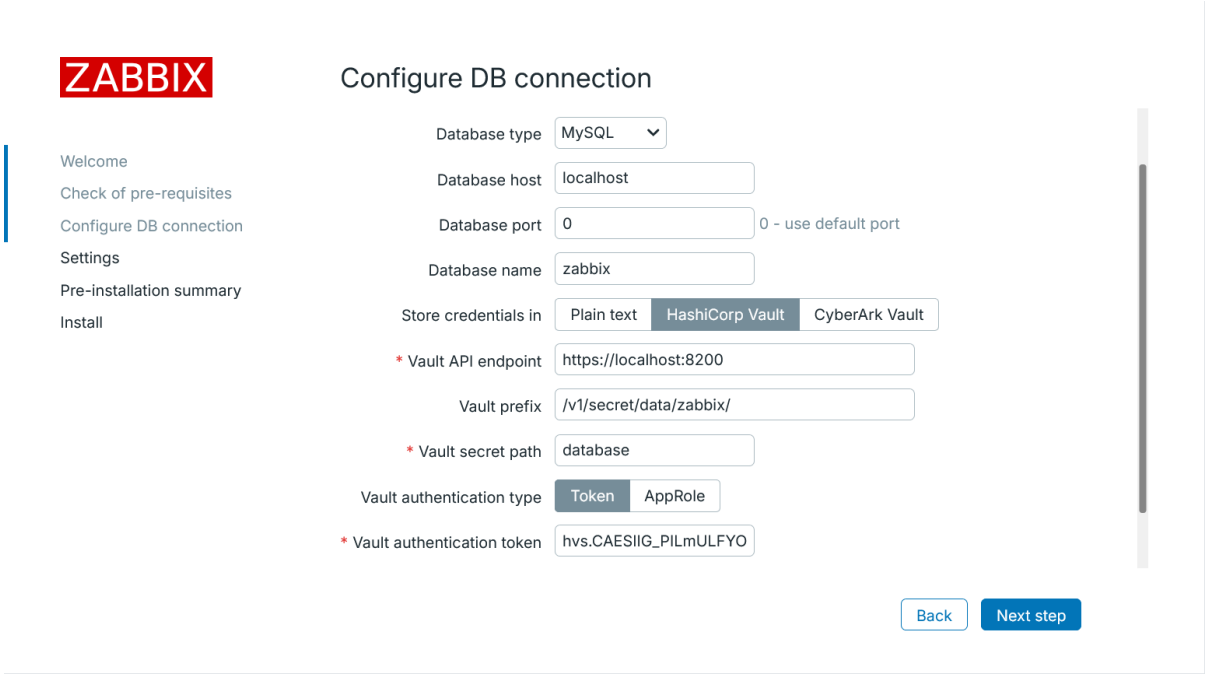
frontend

Il frontend di Zabbix può essere configurato per recuperare le credenziali del database dal vault durante l'installazione del frontend oppure aggiornando il file di configurazione del frontend (`zabbix.conf.php`).

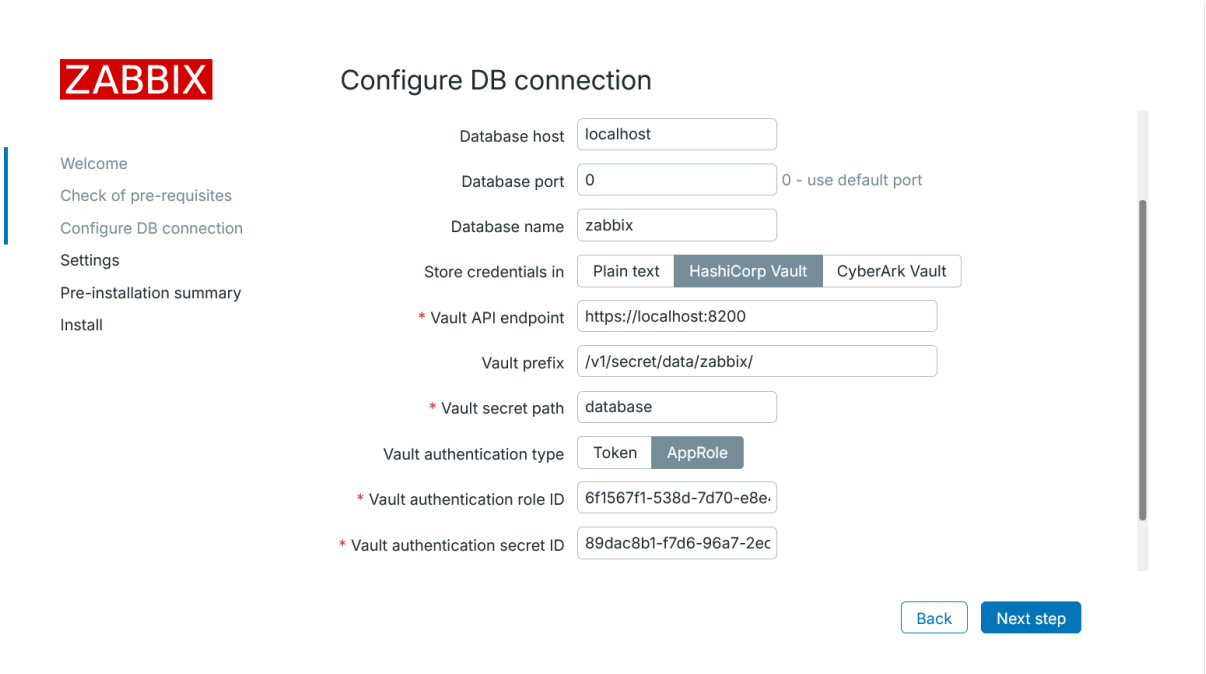
Attention:
Se le credenziali del vault sono cambiate dall'installazione precedente del frontend, aggiornare la configurazione esistente.

Durante l'**installazione del frontend**, i parametri di connessione devono essere specificati nel passaggio *Configure DB Connection*.

Esempio di configurazione con autenticazione tramite token:



Esempio di configurazione con autenticazione AppRole:



Parametro	Obbligatorio	Valore predefinito	Descrizione
Store credentials in	no		Selezionare **HashiCorp Vault**.
Vault API endpoint	sì	https://localhost:8200	Specificare l'URL per la connessione al vault nel formato `scheme://host:port`
Vault prefix	no	/v1/secret/data/	Specificare un prefisso personalizzato per il percorso o la query del vault. Se non specificato, viene utilizzato quello predefinito. Esempio: `/v1/secret/data/zabbix/`
Vault secret path	sì		Percorso del secret dal quale le credenziali del database devono essere recuperate tramite le chiavi "password" e "username". Esempio: `database`
Vault authentication type	no		Selezionare il tipo di autenticazione del vault: **Token** - utilizzare l'autenticazione tramite token; **AppRole** - utilizzare il metodo di autenticazione AppRole.

Parametro	Obbligatorio	Valore predefinito	Descrizione
Vault authentication token	sì		Fornire un token di autenticazione per l'accesso in sola lettura al percorso del secret. Per informazioni sulla creazione di token e policy del vault, consultare la documentazione di HashiCorp. Questo campo è disponibile solo se *Token* è selezionato come tipo di autenticazione del vault.
Vault authentication role ID	sì		Fornire l'ID del ruolo dell'AppRole. Per informazioni sulle AppRole, consultare la documentazione di HashiCorp. Questo campo è disponibile solo se *AppRole* è selezionato come tipo di autenticazione del vault.
Vault authentication secret ID	sì		Fornire il secret ID emesso per l'AppRole. Per informazioni sulle AppRole, consultare la documentazione di HashiCorp. Questo campo è disponibile solo se *AppRole* è selezionato come tipo di autenticazione del vault.

Recupero dei valori delle macro utente

Per utilizzare HashiCorp Vault per memorizzare i valori delle macro utente *Vault secret*, assicurarsi che:

- Zabbix server/proxy sia configurato per funzionare con HashiCorp Vault.
- Il parametro *Vault provider* in *Administration > General > Other* sia impostato su "HashiCorp Vault" (predefinito).



Note:
Zabbix server (e Zabbix proxy, se configurato) richiedono l'accesso ai valori delle macro *Vault secret* dal vault. Zabbix frontend non necessita di tale accesso.

Il valore della macro dovrebbe contenere un percorso di riferimento (come `path:key`, ad esempio `macros:password`). Il token di autenticazione specificato durante la configurazione di Zabbix server/proxy (tramite il parametro `VaultToken`) deve fornire accesso in sola lettura a questo percorso.

Per informazioni dettagliate sull'elaborazione dei valori delle macro da parte di Zabbix, vedere *Vault secret macros*.

Sintassi del percorso

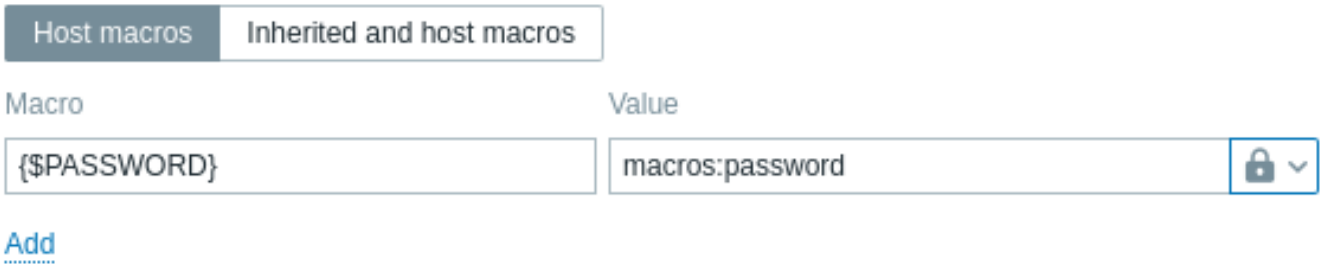
I simboli barra obliqua ("/") e due punti (":") sono riservati.

Una barra obliqua può essere usata solo per separare un mount point da un percorso (ad esempio, *secret/zabbix*, dove il mount point è "secret" e il percorso è "zabbix"). Nel caso delle macro di Vault, i due punti possono essere usati solo per separare un percorso/query da una chiave.

È possibile codificare in URL i simboli barra obliqua e due punti se è necessario creare un mount point con un nome separato da una barra obliqua (ad esempio, *foo/bar/zabbix*, dove il mount point è "foo/bar" e il percorso è "zabbix", può essere codificato come "foo%2Fbar/zabbix") e se il nome di un mount point o un percorso deve contenere due punti.

Esempio

1. In Zabbix, aggiungi una macro utente `{$PASSWORD}` di tipo "Vault secret" e con valore `macros:password`



2. Esegui i seguenti comandi CLI per creare il secret richiesto nel vault:

```
#### Abilita il punto di mount "secret/" se non è già abilitato; nota che deve essere usato "kv-v2".
vault secrets enable -path=secret/ kv-v2

#### Inserisci un nuovo secret con chiave "password" sotto il punto di mount "secret/" e il percorso "zabb
vault kv put -mount=secret zabbix/macros password=<password>

#### Verifica che il secret sia stato aggiunto correttamente.
vault kv get secret/zabbix/macros

#### Infine verifica con Curl; nota che "data" deve essere aggiunto manualmente dopo il punto di mount e '
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. Di conseguenza, Zabbix risolverà la macro {$PASSWORD} nel valore: <password>

Aggiornamento della configurazione esistente

Per aggiornare una configurazione esistente per il recupero dei segreti da HashiCorp Vault:

1. Aggiornare i parametri del file di configurazione del server o del proxy Zabbix come descritto nella sezione *Credenziali del database*.
2. Aggiornare i parametri del file di configurazione del frontend di Zabbix:

Utilizzando il token Vault:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

Oppure utilizzando il metodo Vault AppRole:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. Configurare le macro utente come descritto nella sezione *Valori delle macro utente*, se necessario.

Per aggiornare una configurazione esistente per il recupero dei segreti da CyberArk Vault, vedere *Configurazione di CyberArk*.

Risoluzione dei problemi

1. Nel file di log di Zabbix server/proxy:

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

Causa probabile: HashiCorp Vault è stato (ri)avviato, ma non è stato unsealed (richiesto nella configurazione predefinita, non richiesto se è configurato Auto Unseal).

2. Nel file di log di Zabbix server/proxy:

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

oppure

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

Causa probabile: VaultAppSecretID è scaduto o ha raggiunto il limite di utilizzo in HashiCorp vault.

Fiducia del certificato TLS di Google Chrome

Panoramica

Questa pagina fornisce i passaggi di configurazione di Zabbix e esempi di configurazione per l'uso di Google Chrome con frontend Zabbix o endpoint zabbix-web-service protetti da un certificato autofirmato o da una autorità di certificazione privata.

Queste istruzioni presuppongono che il server web di destinazione sia già configurato per HTTPS. Per configurare TLS sul frontend Zabbix, vedere Connessione sicura al frontend.

Configurazione

Google Chrome su Linux utilizza un database di certificati NSS per utente per i certificati attendibili. Per fare in modo che Chrome si fidi di un certificato autofirmato o di una autorità di certificazione privata, aggiungi i certificati richiesti al database NSS dell'account utente che esegue Chrome.

1. Installa i pacchetti richiesti.

Per Debian/Ubuntu:

```
sudo apt install ca-certificates libnss3-tools
```

Per sistemi basati su RHEL:

```
sudo dnf install ca-certificates nss-tools
```

Oppure:

```
sudo yum install ca-certificates nss-tools
```

2. Prepara i file del certificato.

Per ogni certificato necessario: crea un file `.crt` basato su PEM per il certificato autofirmato, oppure crea file `.crt` separati basati su PEM per il certificato dell'autorità di certificazione radice e, se richiesto, per ciascun certificato intermedio.

3. Crea la directory del database NSS.

Usa la directory home dell'utente che esegue Chrome.

Ad esempio:

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

Se Chrome sul sistema di destinazione usa una posizione diversa per il database NSS, usa invece quella directory.

Assicurati che i file del certificato siano leggibili dall'utente che esegue Google Chrome. Se necessario, copiali in una posizione accessibile a quell'utente.

4. Importa i certificati.

Se usi un certificato autofirmato:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix self-signed certificate" \
  -i /path/to/self-signed.crt
```

Se usi un'autorità di certificazione privata, importa il certificato dell'autorità di certificazione radice nel database:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix root certificate authority" \
  -i /path/to/root-ca.crt
```

Se vengono usati certificati intermedi, importa ciascuno separatamente:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix intermediate certificate" \
  -i /path/to/intermediate-ca.crt
```

5. Verifica il database dei certificati.

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Testa Chrome.

Google Chrome utilizza il database dei certificati dell'account utente che lo avvia. Se l'account utente non dispone di una directory home scrivibile, configura una home scrivibile e le directory XDG prima di eseguire il test.

Esempio (sostituisci `google-chrome` con `chromium` se viene usato Chromium.):

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

Se la configurazione è corretta, il comando restituisce l'HTML della pagina richiesta.

Se Chrome non si fida del certificato, il comando restituisce una pagina di errore del certificato contenente messaggi come `NET::ERR_CERT_AUTHORITY_INVALID`.

Risoluzione dei problemi

Errore	Possibile causa
`NET::ERR_CERT_AUTHORITY_INVALID`	Il certificato è stato importato nel profilo utente sbagliato, il certificato dell'autorità di certificazione radice manca oppure manca un certificato intermedio.
`NET::ERR_CERT_COMMON_NAME_INVALID`	Il nome host non corrisponde al nome host usato nell'URL del browser.
`NET::ERR_CERT_DATE_INVALID`	Il certificato è scaduto, non è ancora valido oppure l'orologio di sistema non è corretto.
Chrome continua a mostrare un avviso relativo al certificato dopo l'importazione	Chrome sta usando un database NSS diverso per l'utente corrente oppure i flag di attendibilità del certificato non sono corretti.

Best practice di sicurezza

Panoramica Questa sezione contiene le best practice per configurare Zabbix in modo sicuro.

Le pratiche descritte in questa sezione non sono necessarie per il funzionamento di Zabbix, ma sono consigliate per una maggiore sicurezza del sistema.

Per le best practice relative alla sicurezza dell'app mobile Zabbix e dell'adattatore bridge Zabbix, vedere Accesso mobile sicuro.

Codifica UTF-8 UTF-8 è l'unica codifica supportata da Zabbix. È noto che funziona senza alcuna vulnerabilità di sicurezza. Gli utenti devono essere consapevoli che esistono problemi di sicurezza noti se si utilizzano alcune delle altre codifiche.

Percorsi del programma di installazione di Windows Quando si utilizzano i programmi di installazione di Windows, si consiglia di usare i percorsi predefiniti forniti dal programma di installazione. L'uso di percorsi personalizzati senza le autorizzazioni appropriate potrebbe compromettere la sicurezza dell'installazione.

Macro negli script globali definiti dall'utente Per migliorare la sicurezza, si consiglia di usare funzioni macro invece delle semplici macro negli script globali definiti dall'utente, poiché le macro non vengono escape automaticamente.

Avvisi di sicurezza di Zabbix e database CVE Vedi *Avvisi di sicurezza di Zabbix e database CVE*.

Template email HTML Quando si creano o modificano template di messaggio utilizzati per le email HTML, racchiudere sempre ogni macro nella funzione macro `htmlencode()`. Ad esempio:

```
<b>Problem started</b> at {{EVENT.TIME}.htmlencode()} on {{EVENT.DATE}.htmlencode()}<br><b>Problem name:</
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Severity:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Operational
data:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>Original problem ID:</b> {{EVENT.ID}.htmlencode()}<br>{{TRIG
```

L'utilizzo di `htmlencode()` garantisce che eventuali caratteri HTML nei valori delle macro vengano sottoposti a escape e impedisce l'inserimento di codice HTML nelle notifiche (ad esempio, un utente malintenzionato potrebbe inserire un link dannoso o di phishing in una notifica).

Nota: i messaggi email HTML predefiniti forniti da Zabbix applicano già `htmlencode()` alle macro. Questa indicazione si applica quando si modificano template esistenti o se ne creano di nuovi: verificare che le macro siano codificate prima di utilizzare un template per inviare email HTML.

Interfaccia web

Panoramica Per accedere facilmente a Zabbix da qualsiasi luogo e piattaforma, è disponibile un'interfaccia basata sul web.

> **Note:**
> Se si utilizza più di un'istanza frontend, assicurarsi che le impostazioni locali e le librerie (LDAP, SAML ecc.) siano installate e configurate in modo identico per tutti i frontend.

Guida del frontend Un link alla guida è disponibile nelle sezioni e nei moduli del frontend di Zabbix, nell'angolo superiore destro, e rimanda direttamente alle parti corrispondenti della documentazione.

Per i campi dei moduli con un'icona della guida, passando il mouse sull'icona viene visualizzato il suggerimento corrispondente.

Modalità kiosk Per le sezioni del frontend con il pulsante *Modalità kiosk* (situato nell'angolo superiore destro), fare clic sul pulsante per passare la sezione alla modalità kiosk. In modalità kiosk viene visualizzato solo il contenuto della sezione.

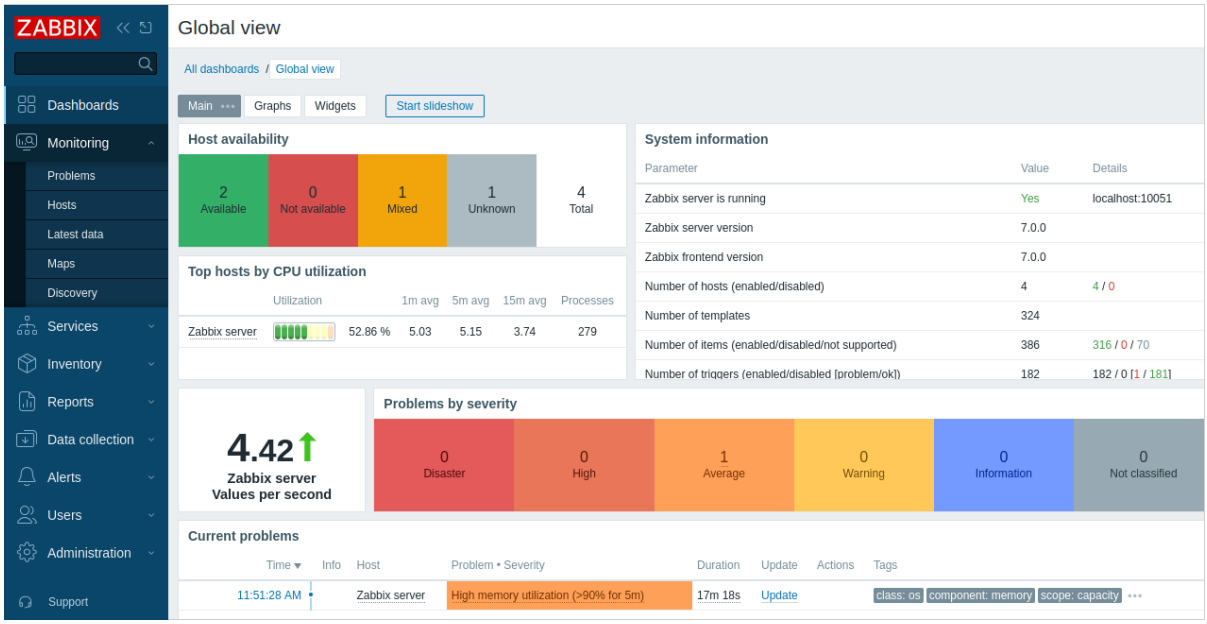
Per uscire dalla modalità kiosk, passare il cursore sul contenuto e fare clic sul pulsante *Visualizzazione normale* che appare nell'angolo superiore destro.

Menu

Panoramica

Un menu verticale in una barra laterale fornisce l'accesso a varie sezioni del frontend di Zabbix.

Il menu è blu scuro nel tema predefinito.

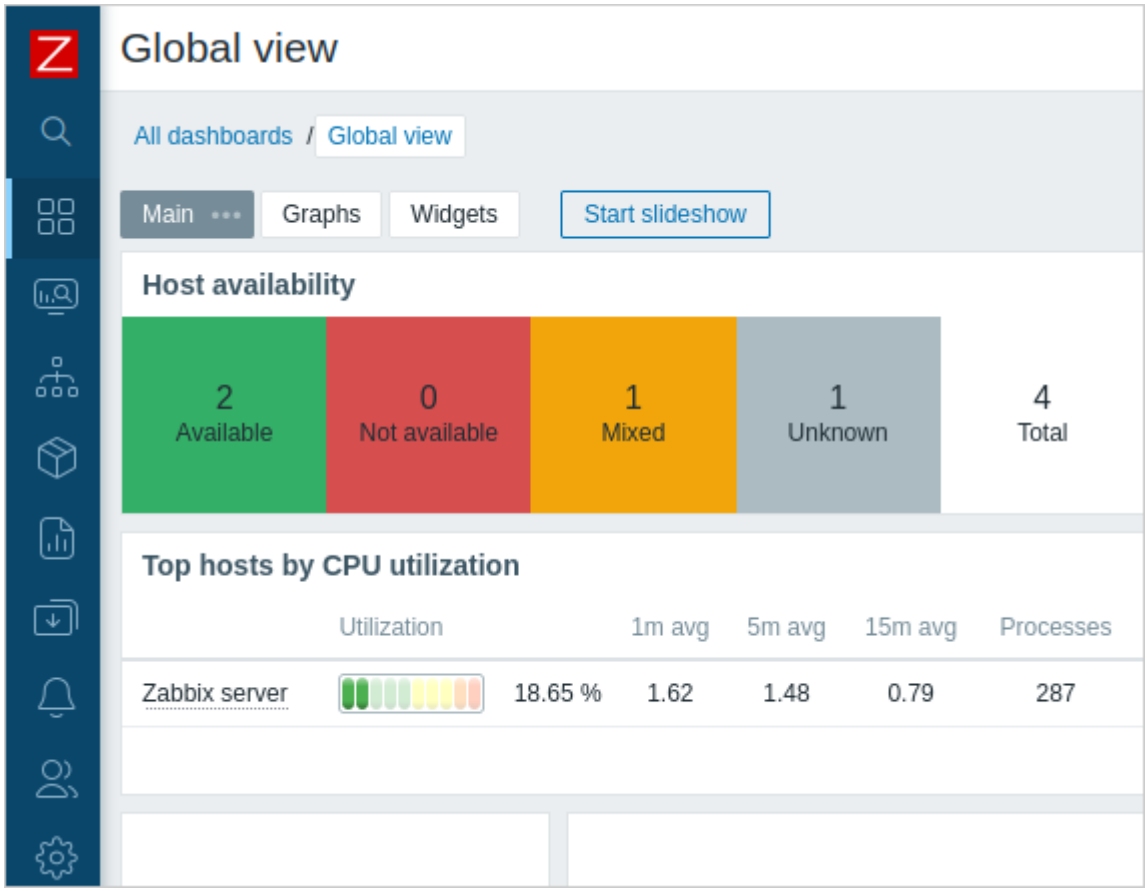


Lavorare con il menu

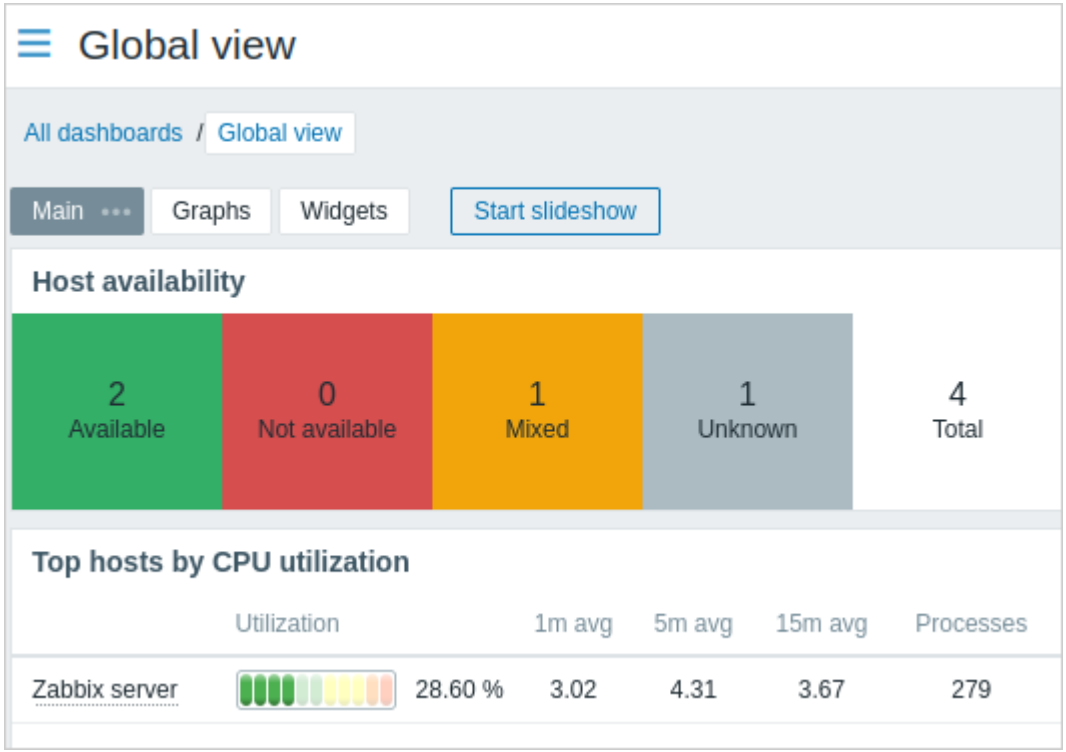
Una casella di ricerca globale si trova sotto il logo Zabbix.

Il menu può essere compresso o completamente nascosto:

- Per comprimerlo, fare clic su  accanto al logo Zabbix. Nel menu compresso sono visibili solo le icone.



- Per nasconderlo, fare clic su  accanto al logo Zabbix. Nel menu nascosto tutto è nascosto.



Menu compresso

Quando il menu è compresso e mostra solo le icone, il menu completo riappare non appena il cursore del mouse viene posizionato su di esso. Il menu riappare sopra il contenuto della pagina; per spostare il contenuto della pagina verso destra è necessario fare clic sul pulsante di espansione. Se il cursore del mouse viene nuovamente spostato fuori dal menu completo, il menu verrà compresso di nuovo dopo due secondi.

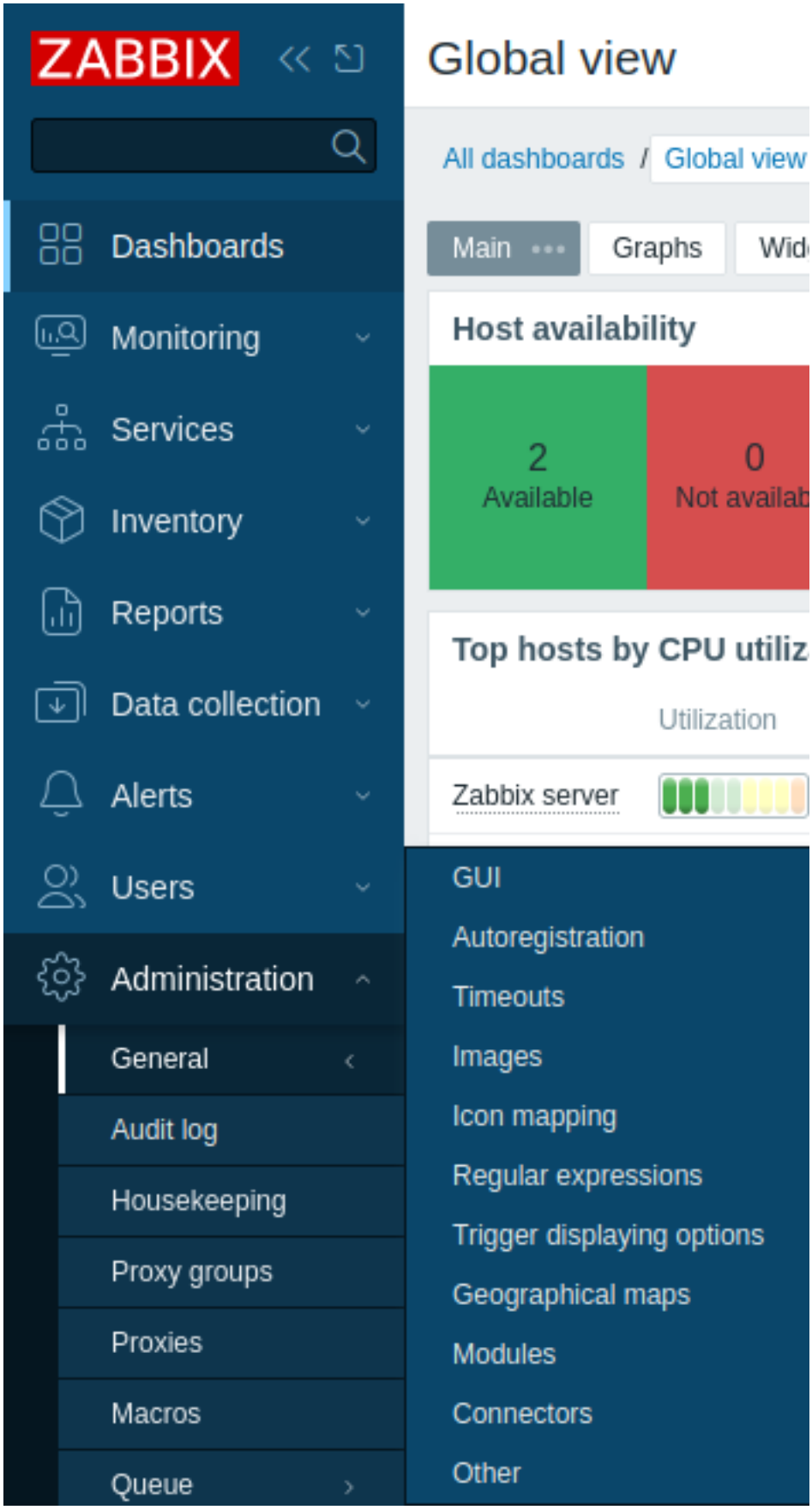
È inoltre possibile visualizzare completamente un menu compresso premendo il tasto Tab. Premendo ripetutamente il tasto Tab è possibile spostare il focus sull'elemento di menu successivo.

Menu nascosto

Anche quando il menu è completamente nascosto, un menu completo è a un solo clic del mouse di distanza, facendo clic sull'icona hamburger . Si noti che riappare sopra il contenuto della pagina; per spostare il contenuto della pagina verso destra è necessario rendere nuovamente visibile il menu facendo clic sul pulsante mostra barra laterale.

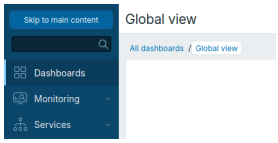
Livelli del menu

Nel menu sono presenti fino a tre livelli.



Accessibilità: Salta la navigazione

Il link *Salta al contenuto principale* consente agli utenti che utilizzano la tastiera e i lettori di schermo di saltare la navigazione ripetitiva della barra laterale e passare direttamente al contenuto principale.



Rimane nascosto durante l'uso normale e appare solo quando riceve il focus, ad esempio quando si preme il tasto Tab dopo il caricamento della pagina o quando viene utilizzato tramite un lettore di schermo.

Menu contestuali

Oltre al menu principale, Zabbix fornisce menu contestuali per host, item e event per un accesso rapido alle entità utilizzate frequentemente, come gli ultimi valori, il grafico semplice, il modulo di configurazione, gli script correlati o i collegamenti esterni.

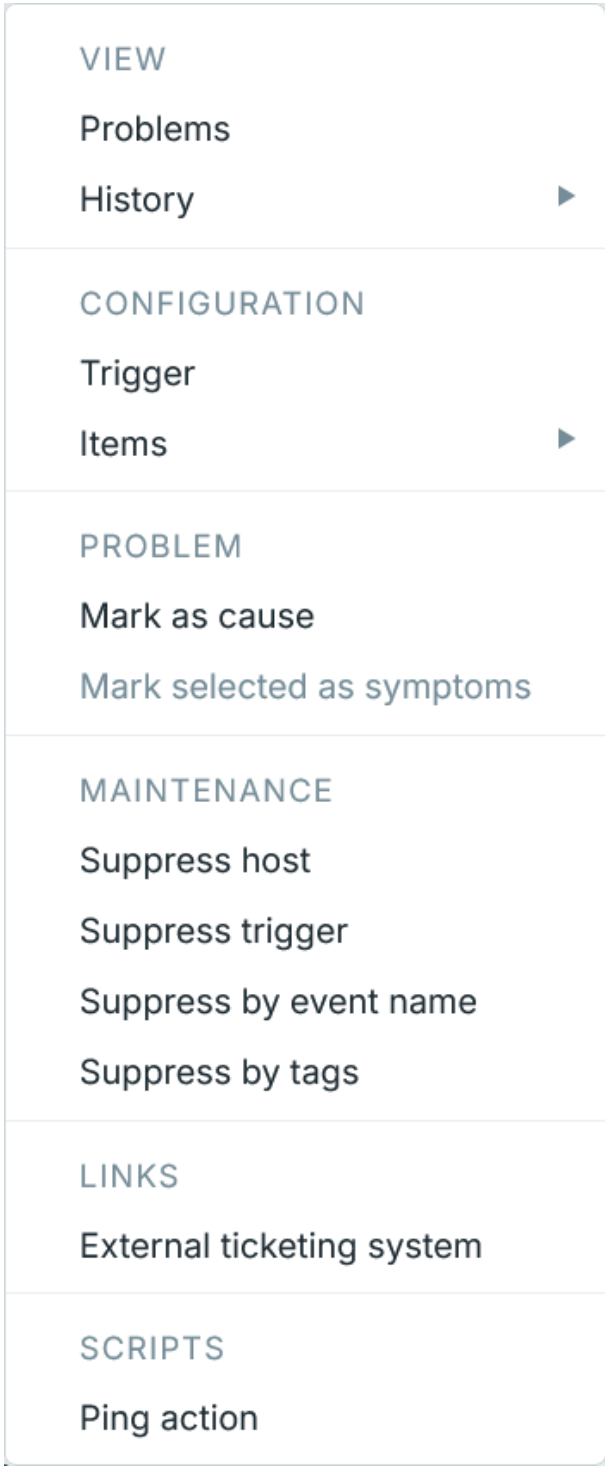
I menu contestuali sono accessibili facendo clic sul nome di host, item o problema/trigger nelle posizioni supportate.

Menu eventi

Panoramica

Il menu degli eventi contiene collegamenti rapidi alle azioni o alle sezioni del frontend che vengono richieste frequentemente per un evento.

Il menu degli eventi può essere aperto facendo clic sul nome dell'evento.



Contenuto

Il menu contestuale dell'evento contiene diverse sezioni: *Visualizzazione*, *Azioni*, *Configurazione*, *Problema*, *Manutenzione*, *Collegamenti* e *Script*. Per le entità non configurate, i collegamenti sono disabilitati e visualizzati in grigio. Le sezioni *Script* e *Collegamenti* vengono visualizzate se le relative entità sono configurate.

La sezione *Visualizzazione* contiene collegamenti a:

- **Problemi** - apre l'elenco dei problemi non risolti del trigger sottostante;
- **Cronologia** - apre il grafico/la cronologia degli item in *Ultimi dati* per gli item sottostanti. Se un trigger utilizza diversi item, saranno disponibili collegamenti per ciascuno di essi.

La sezione *Azioni* è disponibile solo nei widget *Panoramica dei trigger*. Contiene un collegamento a:

- **Aggiorna problema** - apre la schermata di aggiornamento del problema.

La sezione *Configurazione* contiene collegamenti alla configurazione di:

- **Trigger** che ha generato il problema;
- **Item** utilizzati nell'espressione del trigger.

Note:
La sezione di configurazione è disponibile solo per gli utenti Admin e Super admin.

La sezione *Problema* contiene le opzioni per:

- **Contrassegna come causa** - contrassegna il problema come causa;
- **Contrassegna i selezionati come sintomi** - contrassegna i problemi selezionati come sintomi di questo problema.

La sezione *Manutenzione* contiene le opzioni per avviare una manutenzione ad hoc:

- **Sopprimi host** - avvia un periodo di manutenzione per l'host;
- **Sopprimi trigger** - avvia un periodo di manutenzione per il trigger;
- **Sopprimi per nome evento** - avvia un periodo di manutenzione per il nome dell'evento;
- **Sopprimi per tag** - avvia un periodo di manutenzione per gli eventi con i tag corrispondenti.

Quando si seleziona l'opzione corrispondente, si apre il modulo di configurazione del periodo di manutenzione.

Verrà preselezionato un periodo di manutenzione una tantum di un'ora (configurabile nelle impostazioni e nel profilo dell'utente), con inizio immediato. Verranno preselezionati anche l'host/il trigger/il nome dell'evento/i tag corrispondenti. Il nome della manutenzione verrà precompilato nel formato `Ad-hoc: <event name>`.

La sezione *Collegamenti* contiene collegamenti per:

- accedere a un URL del trigger configurato;
- accedere ai collegamenti personalizzati configurati negli script globali (con ambito "Azione manuale sull'evento" e tipo "URL");
- accedere a un ticket esterno configurato per il problema (vedere l'opzione *Includi voce nel menu dell'evento* durante la configurazione dei webhook).

La sezione *Script* contiene collegamenti per eseguire uno script globale (con ambito *Azione manuale sull'evento*). Questa funzionalità può essere utile per eseguire script utilizzati per gestire i ticket dei problemi in sistemi esterni.

Posizioni supportate

Il menu contestuale dell'evento è accessibile facendo clic sul nome di un problema o di un evento in varie sezioni del frontend, ad esempio:

- Dashboard widgets, come il widget *Problems*, il widget *Trigger overview*, ecc.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Dettagli evento
- *Reports* > *Top 100 triggers* (gli script globali e l'accesso al ticket esterno non sono supportati in questa posizione)

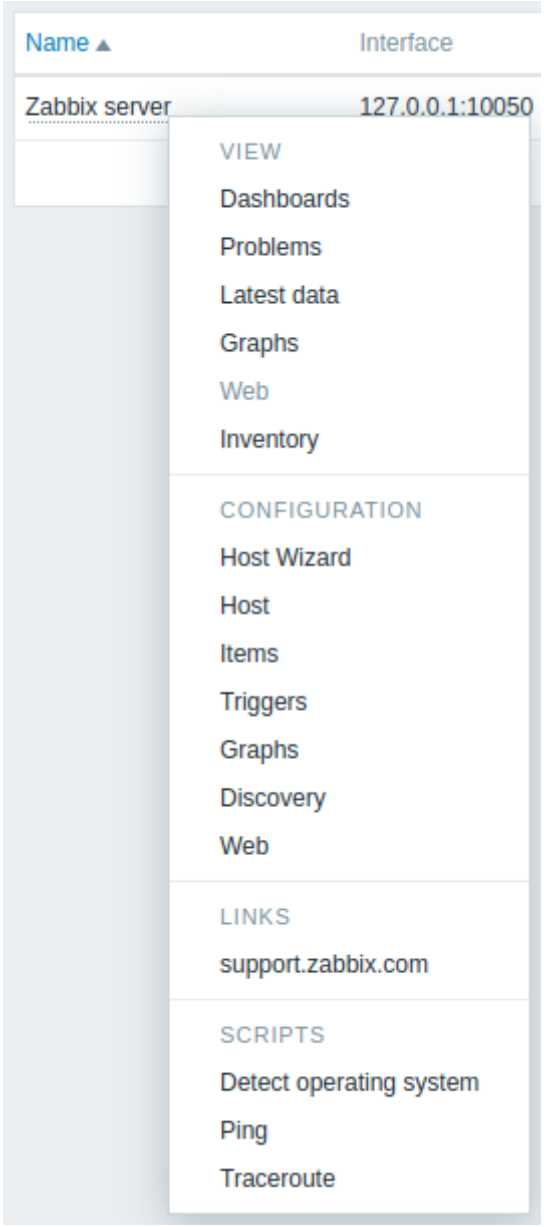
Menu host

Panoramica

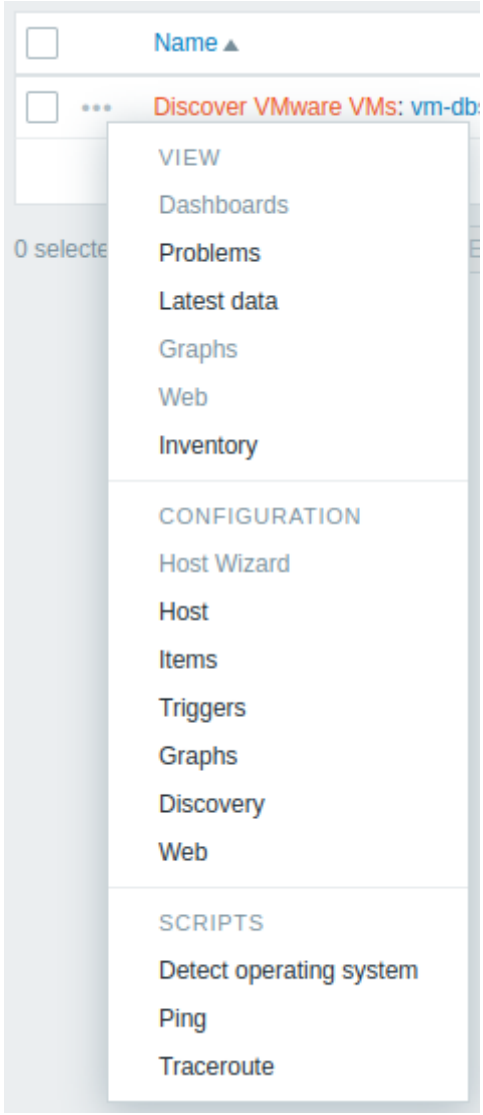
Il menu dell'host contiene collegamenti rapidi ad azioni o sezioni del frontend che sono frequentemente necessarie per un host.

Il menu dell'host può essere aperto facendo clic sul nome dell'host o sull'icona con i tre puntini, a seconda della sezione del frontend, ad esempio:

Monitoring > *Latest data*



Data collection > *Hosts*



Contenuto

Il menu contestuale dell'host contiene quattro sezioni: *Visualizzazione*, *Configurazione*, *Collegamenti* e *Script*. Per le entità non configurate, i collegamenti sono disabilitati e visualizzati in grigio. Le sezioni *Script* e *Collegamenti* vengono visualizzate se le relative entità sono configurate.

La sezione *Visualizzazione* contiene collegamenti a:

- **Dashboard** - apre widget e grafici.
- **Problemi** - apre la sezione *Problemi* con l'elenco dei problemi non risolti del trigger sottostante.
- **Ultimi dati** - apre la sezione *Ultimi dati* con l'elenco di tutti gli ultimi dati dell'host corrente.
- **Grafici** - apre i grafici semplici dell'host corrente.
- **Web** - apre il collegamento agli scenari web configurati.
- **Inventario** - apre il collegamento all'inventario dell'host corrente.

La sezione *Configurazione* contiene collegamenti a:

- **Host Wizard** - apre Host Wizard per l'host corrente (disabilitato per gli host rilevati).
- **Host** - modulo di configurazione dell'host corrente.
- **Item** - elenco degli item dell'host corrente.
- **Trigger** - elenco dei trigger dell'host corrente.
- **Grafici** - grafici semplici dell'host corrente.
- **Rilevamento** - elenco delle regole di rilevamento di basso livello dell'host corrente.

- **Web** - elenco degli scenari web dell'host corrente.

> **Note:**
> La sezione di configurazione è disponibile solo per gli utenti Admin e Super admin.

La sezione *Collegamenti* contiene collegamenti a:

- accedere a un URL del trigger configurato.
- accedere ai collegamenti personalizzati configurati in Script globali (con ambito *Azione manuale sull'host* e tipo 'URL').

La sezione *Script* consente di eseguire gli script globali configurati per l'host corrente. Per essere disponibili nel menu dell'host, questi script devono avere l'ambito definito come *Azione manuale sull'host*.

Posizioni supportate

Il menu dell'host è accessibile facendo clic sul nome di un host in varie sezioni del frontend, ad esempio:

- Dashboard widget, come *Problems*, *Top items*, *Trigger overview*, ecc.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Dettagli evento
- *Monitoring* > *Hosts*
- *Monitoring* > *Hosts* > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

Il menu dell'host è accessibile facendo clic sull'icona con i tre puntini in Data collection > Hosts.

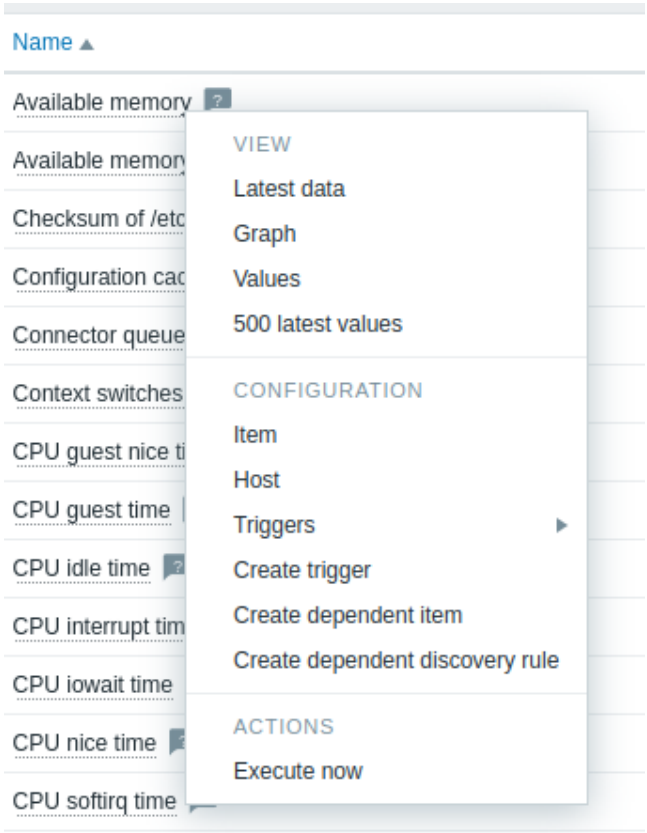
Menu item

Panoramica

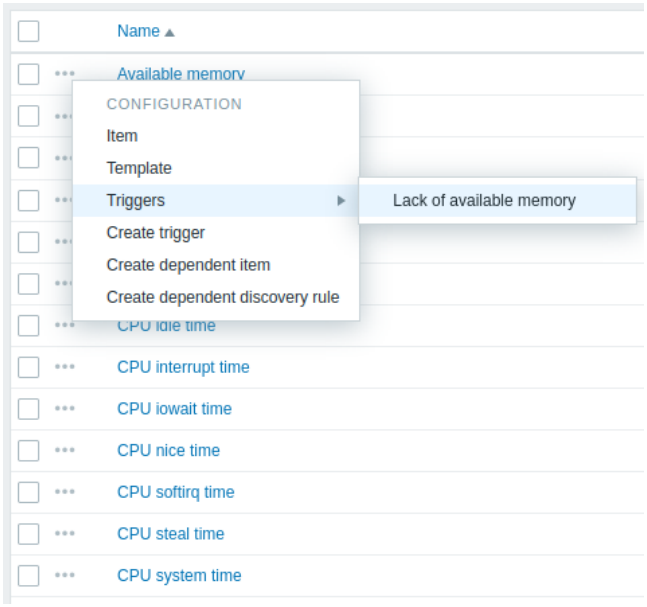
Il menu dell'item contiene scorciatoie verso azioni o sezioni del frontend che sono frequentemente necessarie per un item.

Il menu dell'item può essere aperto facendo clic sul nome dell'item o sull'icona con i tre puntini, a seconda della sezione del frontend, ad esempio:

Monitoring > *Latest data*



Data Collection > *Templates* > *Items*



Contenuto

Il menu dell'item è suddiviso in tre sezioni: *Visualizzazione*, *Configurazione* e *Azioni*.

La sezione *Visualizzazione* contiene le seguenti opzioni:

- **Ultimi dati** - apre la sezione *Ultimi dati* filtrata per l'host e l'item correnti;
- **Grafico** - apre un grafico semplice dell'item corrente;
- **Valori** - apre l'elenco di tutti i valori ricevuti per l'item corrente negli ultimi 60 minuti;
- **500 valori più recenti** - apre l'elenco dei 500 valori più recenti dell'item corrente.

La sezione *Configurazione* (disponibile solo per gli utenti di tipo *Admin* e *Super admin*) contiene le seguenti opzioni:

- **Item** - apre il modulo di configurazione dell'item corrente;
- **Template** - apre il modulo di configurazione del template a cui appartiene l'item (disponibile solo quando si accede al menu dell'item da *Raccolta dati* > *Template* > *Item*);
- **Host** - apre il modulo di configurazione dell'host a cui appartiene l'item;
- **Trigger** - al passaggio del mouse, apre un elenco dei trigger dell'item, se presenti; facendo clic su un trigger, si apre il relativo modulo di configurazione del trigger;
- **Crea trigger** - apre il modulo di configurazione del trigger per creare un trigger per questo item;
- **Crea item dipendente** - apre il modulo di configurazione dell'item per creare un item dipendente, con l'item corrente come item master;
- **Crea regola di discovery dipendente** - apre il modulo di configurazione della regola di discovery per creare una regola di discovery dipendente, con l'item corrente come item master.

La sezione *Azioni* contiene la seguente opzione:

- **Esegui ora** - esegue un controllo per ottenere immediatamente un nuovo valore dell'item.

Posizioni supportate

Il menu dell'item è accessibile facendo clic sul nome di un item in varie sezioni del frontend, ad esempio:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items*
- *Data collection* > *Hosts* > *Discovery rules* > *Item prototypes*

Il menu dell'item è accessibile facendo clic sul valore di un item nel widget della dashboard *Top items*.

Sezioni del frontend

Struttura del menu Il menu del frontend di Zabbix ha la seguente struttura:

- Dashboard
- Monitoraggio
  - Problemi
  - Host
  - Ultimi dati
  - Mappe
  - Discovery
- APM
  - Tracce
  - Metriche
  - Log
- Servizi
  - Servizi
  - SLA
  - Report SLA
- Inventario
  - Panoramica
  - Host
- Report
  - Informazioni di sistema
  - Report pianificati
  - Report sulla disponibilità
  - Primi 100 trigger
  - Registro di audit
  - Registro delle azioni
  - Notifiche
- Raccolta dati
  - Gruppi di template
  - Gruppi di host
  - Template
  - Host
  - Manutenzione
  - Correlazione degli eventi
  - Discovery
- Avvisi
  - Azioni
    - Azioni dei trigger
    - Azioni dei servizi
    - Azioni di discovery
    - Azioni di autoregistrazione
    - Azioni interne
  - Tipi di supporto
  - Script
- Utenti
  - Gruppi di utenti
  - Ruoli utente
  - Utenti
  - Token API
  - Autenticazione
  - Dispositivi
- Amministrazione
  - Generale
    - GUI
    - Autoregistrazione
    - Timeout
    - Immagini
    - Mappatura delle icone

    - Espressioni regolari
    - Opzioni di visualizzazione dei trigger
    - Mappe geografiche
    - Moduli
    - Connettori
    - Altro
- Registro di audit
- Housekeeping
- Gruppi di proxy
- Proxy
- Macro
- Coda
    - Panoramica della coda
    - Panoramica della coda per proxy
    - Dettagli della coda

Dashboard

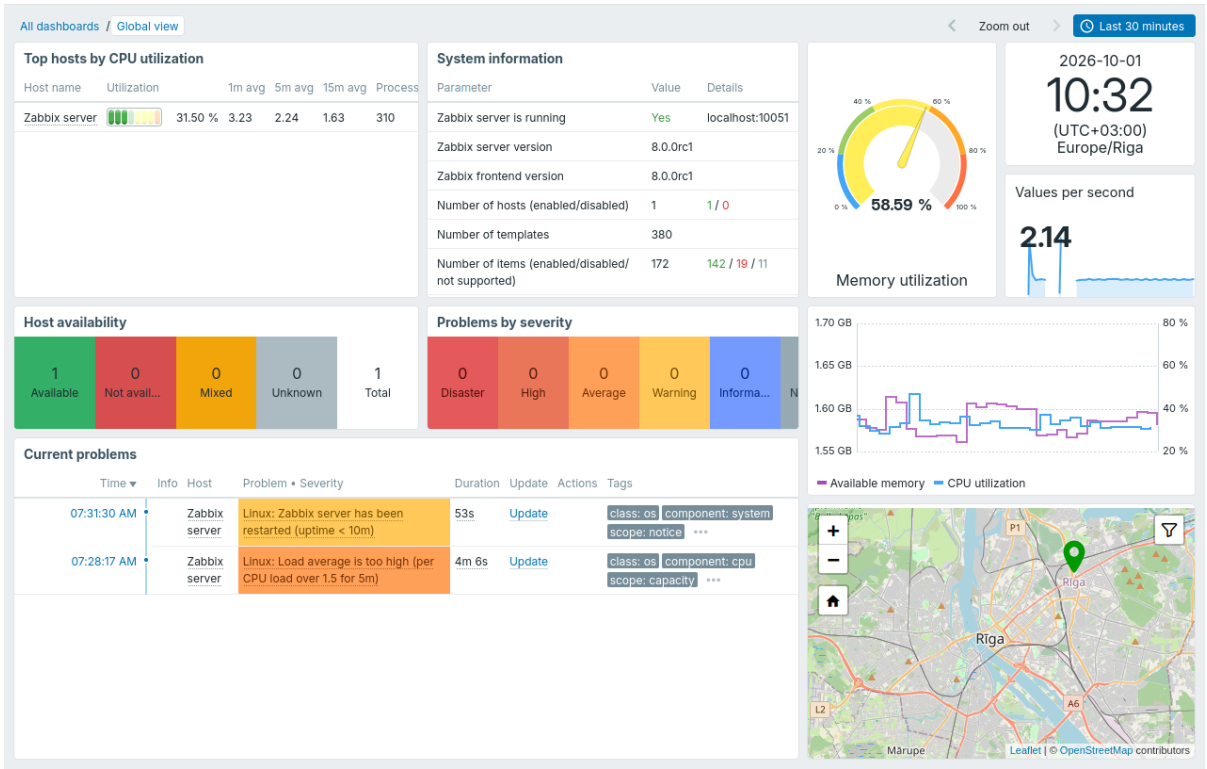
Le dashboard offrono una panoramica dell'ambiente monitorato. È possibile crearli e personalizzarli in base alle proprie esigenze.

Gli elementi costitutivi sono i widget: un grafico, una mappa, un elenco dei problemi recenti e così via, ciascuno progettato per visualizzare un tipo specifico di informazioni.

I widget possono essere organizzati in più pagine del dashboard. Tali pagine possono essere visualizzate come presentazione. È inoltre possibile accedere a qualsiasi pagina facendo clic sulla relativa scheda.

È possibile creare diversi dashboard, anche se ne viene visualizzato solo uno alla volta.

Zabbix include alcuni dashboard predefiniti. Uno di questi è il dashboard *Vista globale*, che mostra lo stato e le condizioni dei propri host monitorati, incluso il server Zabbix:

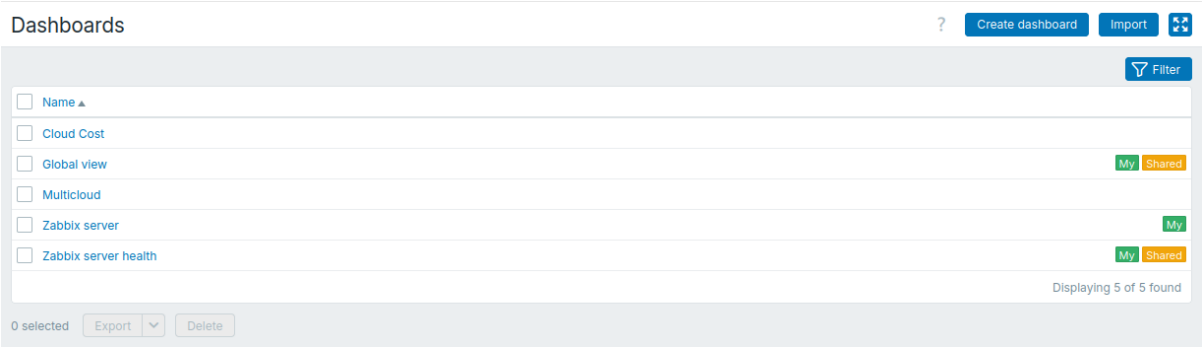


È possibile trovare altri dashboard nel repository dei dashboard. Questi dashboard sono progettati per funzionare con i template Zabbix esistenti e possono essere scaricati e importati nel proprio ambiente Zabbix. Una descrizione dettagliata di ogni dashboard, inclusi i prerequisiti, è disponibile nel file `README` del dashboard.

Elenco delle dashboard

Per aprire l'elenco delle dashboard, è possibile:

- Fare clic su *Dashboards* nel menu.
- Fare clic su *All dashboards* durante la visualizzazione di una dashboard per aprire l'elenco delle dashboard.



L'accesso al menu *Dashboards* dipende dal ruolo utente e dall'autorizzazione *Dashboard* nella sezione *Access to UI elements*.

Da qui è possibile: - Creare dashboard. - Eliminare dashboard. È possibile eliminare una dashboard selezionando la relativa casella di controllo e facendo clic sul pulsante *Delete*. - Importare dashboard. - Esportare dashboard.

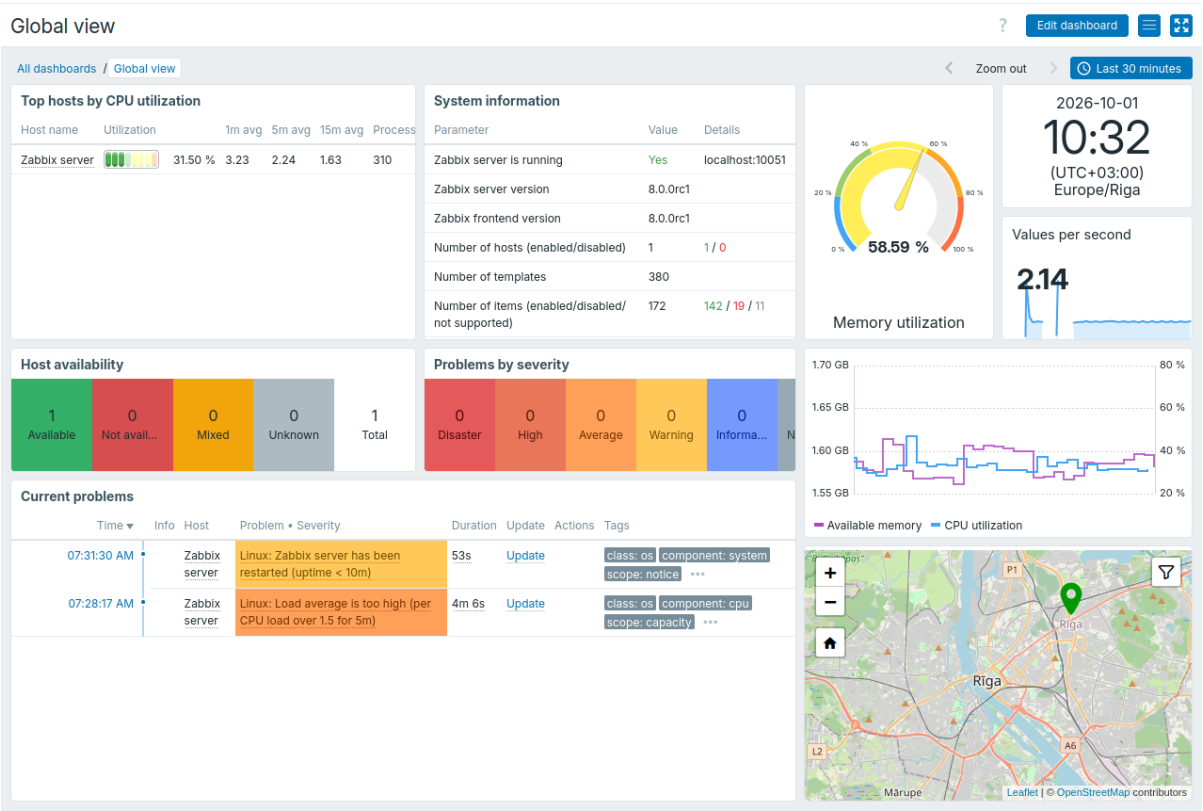
Le dashboard presentano tag che indicano la proprietà e le impostazioni di condivisione:

- *My* - la dashboard è di tua proprietà ed è visibile solo a te.
- *My + Shared* - la dashboard è di tua proprietà, ma è pubblica o condivisa con utenti o gruppi di utenti specifici.
- Le dashboard senza tag sono di proprietà di altri utenti, ma sono pubbliche o condivise con te.

È possibile filtrare l'elenco per nome e proprietario della dashboard.

Visualizzazione della dashboard

Fare clic sul nome di una dashboard nell'elenco delle dashboard per visualizzarla:



L'interfaccia utente di Zabbix ricorda l'ultima dashboard aperta per la sessione successiva.

Note:
I dati visualizzati in una dashboard dipendono dall'accesso agli host.

Per i widget che mostrano grafici, il selettore dell'intervallo di tempo sopra i widget imposta l'intervallo temporale. Mostra l'intervallo temporale corrente come etichetta. Fare clic sull'etichetta per aprire o chiudere il selettore.

Da qui è possibile gestire le dashboard come segue:

- Modificare la dashboard.
- Eseguire le seguenti azioni dal menu *Azioni* della dashboard :
  - *Condivisione* - modificare le impostazioni di condivisione della dashboard.

  - *Crea nuova* - creare una nuova dashboard.
  - *Clona* - creare una nuova dashboard clonando quella corrente. È possibile modificare le proprietà della dashboard prima di salvarla.
  - *Elimina* - eliminare la dashboard.
  - *Esporta* - esportare la dashboard.
  - *Crea nuovo report* - creare un report pianificato. Questa opzione è disabilitata se non si dispone dell'autorizzazione per gestire i report pianificati.
  - *Visualizza report correlati* - visualizzare un elenco dei report esistenti per la dashboard corrente. Questa opzione è disabilitata se non esistono report correlati o se non si dispone dell'autorizzazione per gestire i report pianificati.
- Passare alla modalità kiosk.
- Avviare o arrestare la presentazione della dashboard.

Nella modalità di visualizzazione della dashboard, per i widget sono disponibili le seguenti opzioni:

- Modificare il widget.
- Eseguire le seguenti azioni dal menu *Azioni* del widget ⋯:
  - *Copia* - copiare il widget.
  - *Scarica immagine* - scaricare il widget come immagine PNG. Questa opzione è disponibile per i widget che supportano questa funzionalità (vedere le singole pagine dei widget).
  - *Intervallo di aggiornamento* - selezionare la frequenza di aggiornamento dei contenuti del widget.

Per le dashboard con più pagine, nella modalità di visualizzazione della dashboard sono disponibili le seguenti azioni nel menu *Azioni* della pagina :

- *Copia* - copiare la pagina.
- *Proprietà* - personalizzare le proprietà della pagina, inclusi il nome e l'intervallo della presentazione.

Presentazione

Se la casella di controllo *Avvia presentazione* è selezionata nelle proprietà del dashboard, la presentazione si avvia automaticamente quando il dashboard contiene più di una pagina. In caso contrario, è possibile avviare la presentazione facendo clic sul pulsante *Avvia presentazione*.



Le pagine cambiano in base agli intervalli specificati nelle proprietà del dashboard e delle singole pagine. Se la proprietà *Intervallo presentazione* è definita per una singola pagina, viene applicato l'intervallo specificato per quella pagina. In caso contrario, viene applicato l'*Intervallo predefinito della presentazione* del dashboard.

Per interrompere la presentazione, fare clic su *Interrompi presentazione*.

In modalità kiosk, utilizzare i seguenti pulsanti:

-  per avviare la presentazione;
-  per interrompere la presentazione;
-  per passare alla pagina precedente;
-  per passare alla pagina successiva.

Condividere una dashboard

Le dashboard possono essere pubbliche o private.

Le dashboard pubbliche sono visibili a tutti gli utenti.
Le dashboard private sono visibili solo ai rispettivi proprietari e agli utenti o gruppi di utenti con cui sono condivise.

Per modificare le impostazioni di condivisione di una dashboard, fare clic su *Condivisione* nel menu *Azioni* della dashboard mentre si visualizza la dashboard:

Parametro	Descrizione
Tipo	Selezionare il tipo di dashboard: **Privata** - la dashboard è visibile solo ai gruppi di utenti e agli utenti selezionati. **Pubblica** - la dashboard è visibile a tutti gli utenti.
Elenco delle condivisioni con i gruppi di utenti	Selezionare i gruppi di utenti che possono accedere alla dashboard. È possibile concedere l'accesso in sola lettura o in lettura e scrittura.

Parametro	Descrizione
Elenco delle condivisioni con gli utenti	Selezionare gli utenti che possono accedere alla dashboard. È possibile concedere l'accesso in sola lettura o in lettura e scrittura.

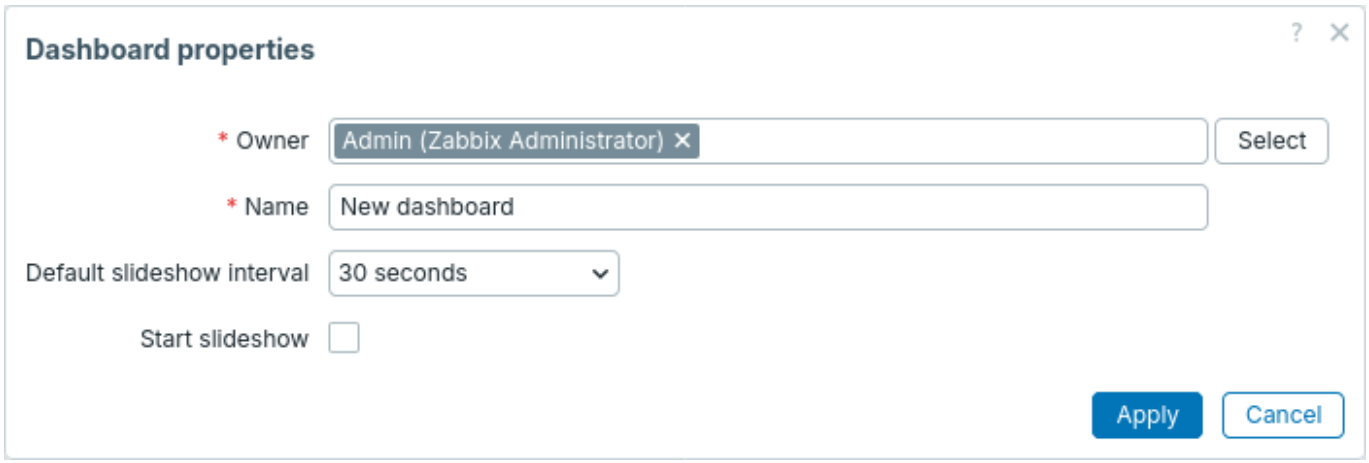
Consultare Autorizzazioni per le dashboard per scoprire in che modo le opzioni di condivisione limitano le azioni disponibili agli utenti.

Crea dashboard

Puoi creare un dashboard in due modi:

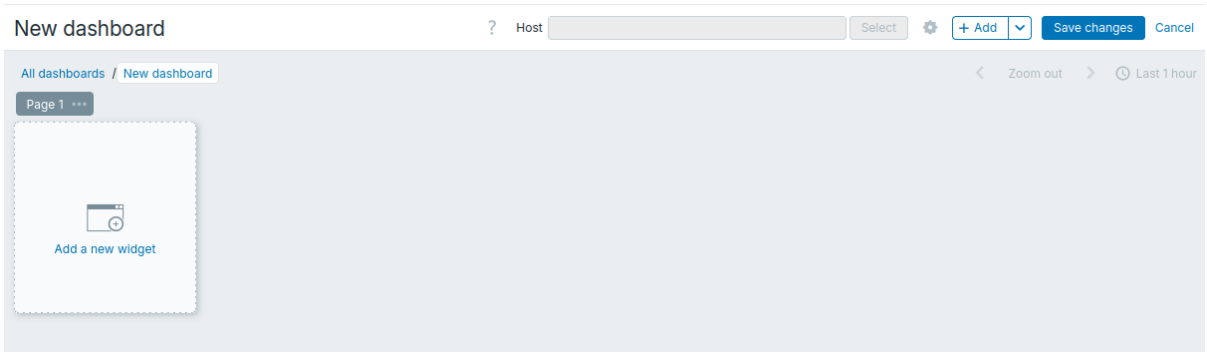
- Nell'elenco dei dashboard, fai clic sul pulsante *Crea dashboard*.
- Nella modalità di visualizzazione del dashboard, fai clic su *Crea nuovo* nel menu *Azioni* del dashboard.

Quindi, specifica le proprietà del dashboard:



Parametro	Descrizione
Proprietario	Seleziona l'utente che sarà il proprietario del dashboard.
Nome	Inserisci il nome del dashboard.
Intervallo predefinito della presentazione	Seleziona per quanto tempo visualizzare ogni pagina del dashboard prima che la presentazione passi alla pagina successiva.
Avvia presentazione	Seleziona questa casella di controllo per avviare automaticamente la presentazione quando il dashboard contiene più di una pagina.

Quindi, fai clic su *Applica*. Il nuovo dashboard si apre in modalità di modifica:



Da qui puoi aggiungere widget e pagine al dashboard.

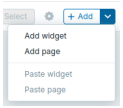
Tieni presente che la larghezza minima di un dashboard è di 1200 pixel; nelle finestre più strette viene visualizzata una barra di scorrimento orizzontale. Un dashboard più largo riempie sempre la finestra del browser; i relativi widget non possono estendersi oltre i bordi della finestra.

Fai clic sul pulsante *Salva modifiche* per salvare il dashboard oppure su *Annulla* per tornare alla schermata precedente.

Aggiunta di widget

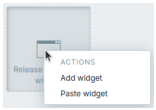
In modalità di modifica del dashboard, è possibile aggiungere i widget in due modi:

- Fare clic sul pulsante *Aggiungi*. In alternativa, fare clic sulla freccia a discesa del pulsante *Aggiungi* e selezionare *Aggiungi widget*.



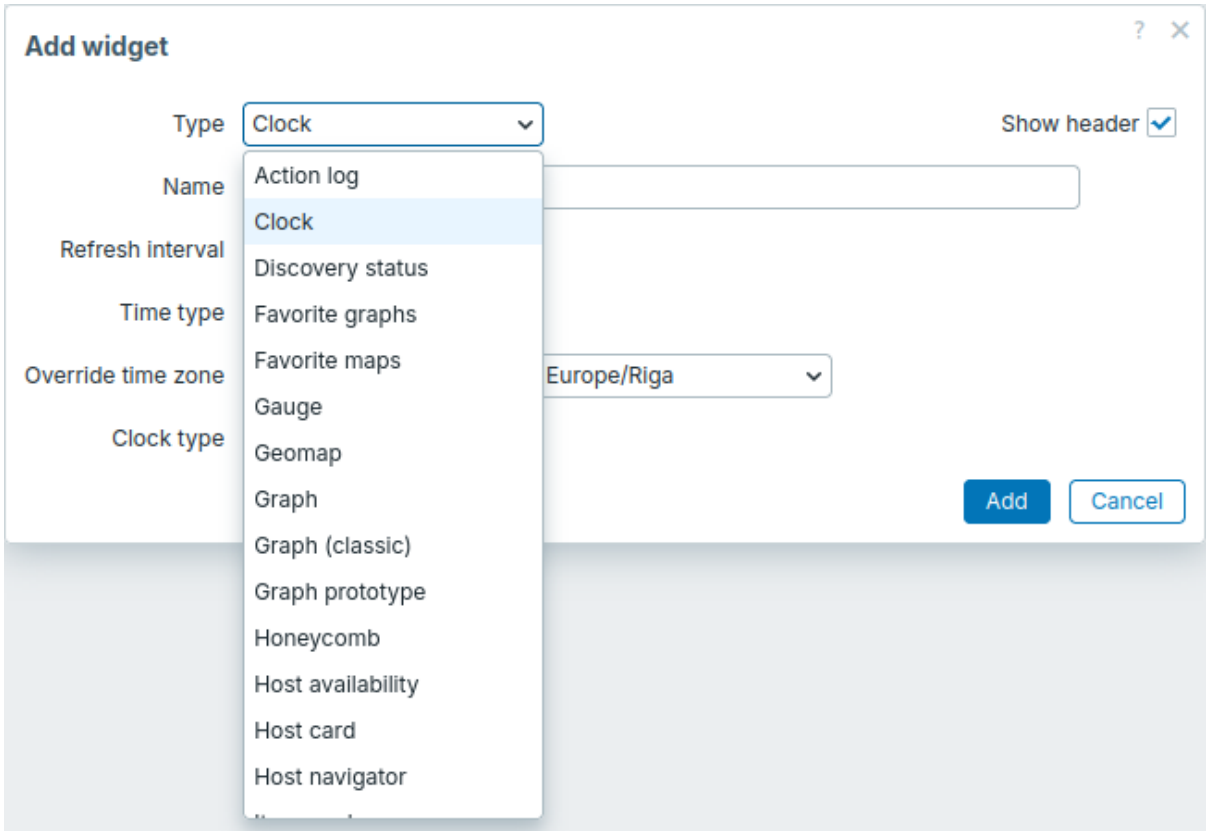
Il segnaposto del widget viene aggiunto con le dimensioni predefinite al primo spazio vuoto disponibile nel dashboard, con il modulo di configurazione del widget aperto.

- Spostare il puntatore del mouse su un'area vuota del dashboard. Quindi fare clic sul segnaposto del widget per aprire il modulo di configurazione del widget. In alternativa, fare clic sul segnaposto e trascinarlo per ridimensionarlo, quindi rilasciarlo per aprire il modulo di configurazione del widget; se un widget è stato copiato, verrà prima richiesto di scegliere *Aggiungi widget* o *Incolla widget*.



Quindi, seguire questi passaggi:

1. Nel modulo di configurazione del widget, selezionare il tipo di widget.
2. Configurare i parametri del widget.
3. Fare clic su *Aggiungi* per aggiungere il widget al dashboard.



Il widget viene aggiunto con le dimensioni predefinite oppure ridimensionato per adattarsi allo spazio disponibile. È possibile ridimensionare e riposizionare i widget come descritto nella sezione *Modifica dei widget*.

È inoltre possibile copiare e incollare i widget per creare nuovi widget con la stessa configurazione, all'interno dello stesso dashboard oppure tra dashboard aperti in schede diverse del browser.

Copiare/incollare i widget

Per copiare e incollare un widget:

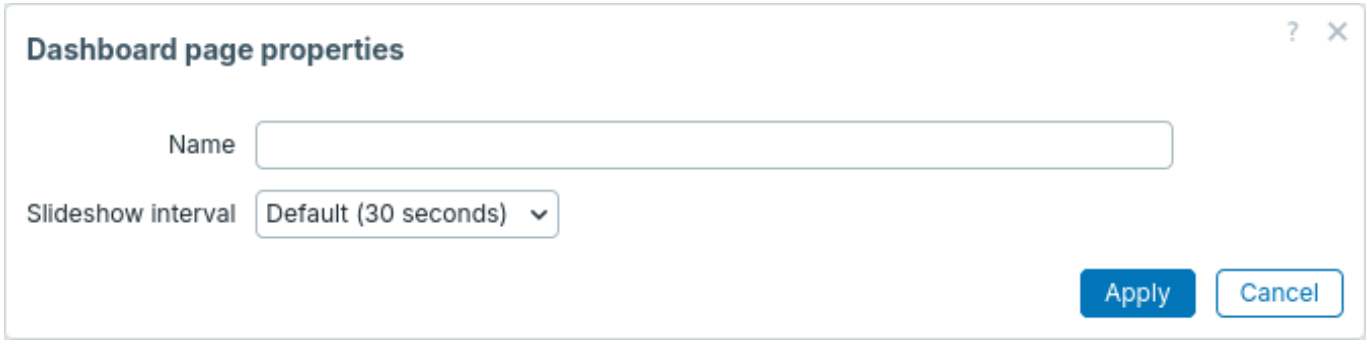
1. Nel menu *Azioni* del widget ⋮, selezionare *Copia*.
2. In modalità di modifica della dashboard, fare clic sulla freccia del menu a discesa del pulsante *Aggiungi* e selezionare *Incolla widget*. In alternativa, fare clic su un'area vuota della dashboard e selezionare *Incolla widget*.

È inoltre possibile incollare un widget copiato sopra uno esistente utilizzando l'opzione *Incolla* nel relativo menu *Azioni*.

Aggiunta di pagine

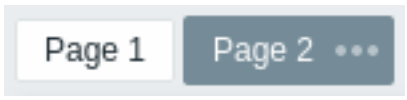
Per aggiungere una nuova pagina a un dashboard:

1. In modalità di modifica del dashboard, fare clic sulla freccia a discesa del pulsante *Add* e selezionare *Add page*.
2. Compilare le proprietà della pagina:
    - *Name* - inserire il nome della pagina. Se si lascia il nome vuoto, la pagina viene aggiunta con il nome *Page N*, dove "N" è il numero progressivo della pagina.
    - *Slideshow interval* - selezionare per quanto tempo visualizzare la pagina prima che lo slideshow passi alla pagina successiva.



3. Fare clic su *Apply*.

Per la nuova pagina viene aggiunta una nuova scheda (ad esempio, *Page 2*).



È inoltre possibile copiare e incollare le pagine per creare nuove pagine con le stesse proprietà, all'interno dello stesso dashboard o tra dashboard aperti in schede diverse.

Copiare/incollare le pagine

Per copiare e incollare una pagina:

1. Nel menu *Azioni* della pagina , selezionare *Copia*.
2. In modalità di modifica della dashboard, fare clic sulla freccia del menu a discesa del pulsante *Aggiungi* e selezionare *Incolla pagina*.

Modifica della dashboard

Per modificare una dashboard, fare clic sul pulsante *Modifica dashboard* nell'angolo superiore destro. In alternativa, fare clic sul pulsante *Modifica* di un widget. La dashboard si apre in modalità di modifica.

Da qui è possibile gestire le dashboard come segue:

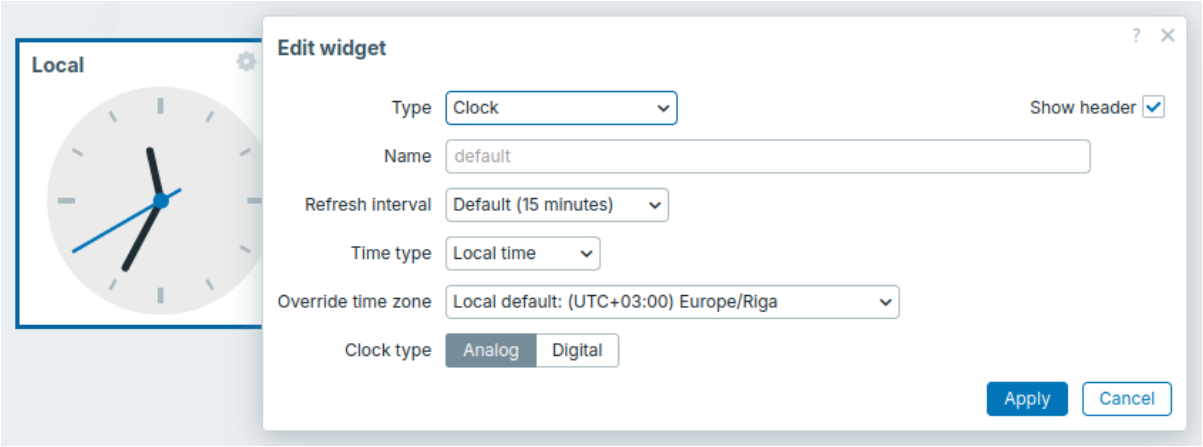
- Modificare le proprietà della dashboard facendo clic sul pulsante a forma di ingranaggio .
- Modificare i widget della dashboard.
- Accedere alle seguenti azioni della dashboard facendo clic sulla freccia del menu a discesa del pulsante *Aggiungi*:
    - *Aggiungi widget* - aggiungere un widget alla dashboard.
    - *Aggiungi pagina* - aggiungere una pagina alla dashboard.
    - *Incolla widget* - incollare un widget copiato. Questa opzione è disabilitata se non è stato copiato alcun widget. È possibile copiare un solo widget alla volta.
    - *Incolla pagina* - incollare una pagina copiata. Questa opzione è disabilitata se non è stata copiata alcuna pagina. È possibile copiare una sola pagina alla volta.

Dopo aver modificato la dashboard, fare clic sul pulsante *Salva modifiche* per salvare la dashboard oppure su *Annulla* per tornare alla schermata precedente.

Modifica dei widget

Un widget può essere modificato nella modalità di modifica del dashboard facendo clic sul pulsante *Modifica* . In alternativa, quando si visualizza un dashboard, è possibile passare il cursore su un widget e fare clic sul pulsante *Modifica*.

Durante la modifica di un widget, questo viene evidenziato e le modifiche vengono visualizzate in anteprima in tempo reale.



È possibile riposizionare il widget e il relativo modulo di configurazione trascinandone le barre del titolo e ridimensionare il widget trascinandone i punti di ridimensionamento.



Durante la modifica di un widget, dal menu *Azioni* del widget ... sono disponibili le seguenti azioni:

- *Copia* - copiare il widget.
- *Incolla* - incollare un widget copiato sopra questo widget. Questa opzione è disabilitata se non è stato copiato alcun widget.
- *Elimina* - eliminare il widget.

Se i parametri obbligatori del widget non sono configurati, il widget entra nello stato *Il widget non è completamente configurato*.

Dopo la modifica, fare clic su *Applica* nel modulo di configurazione del widget, quindi su *Salva modifiche* nell'angolo superiore destro del dashboard per applicare le modifiche.

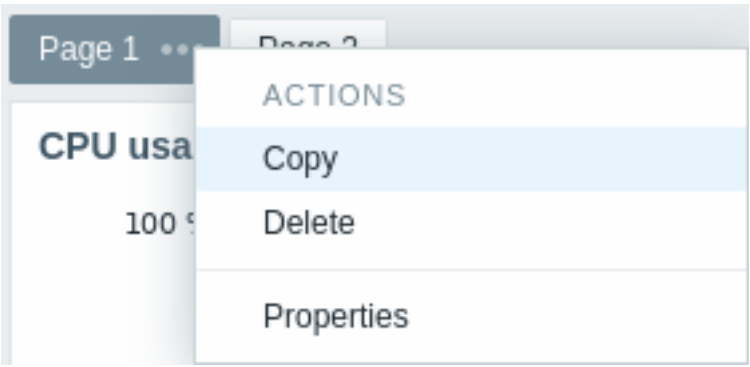
Modifica delle pagine

Le pagine possono essere modificate nella modalità di modifica del dashboard.

Le pagine possono essere riordinate trascinando e rilasciando le schede delle pagine. Il riordinamento non rinomina le pagine, ad eccezione delle pagine con nomi assegnati automaticamente (*Pagina N*), che vengono rinumerate.

Durante la modifica di una pagina, dal menu *Azioni* della pagina sono disponibili le seguenti azioni:

- *Copia* - copiare la pagina.
- *Elimina* - eliminare la pagina.
- *Proprietà* - personalizzare le proprietà della pagina.



Autorizzazioni per i dashboard

Le autorizzazioni per i dashboard degli utenti normali e degli utenti di tipo *Admin* sono limitate come descritto di seguito, a condizione che l'accesso ai dashboard e l'opzione *Crea e modifica dashboard* siano abilitati per il ruolo utente assegnato:

- Possono visualizzare e clonare un dashboard pubblico anche quando nessun utente o gruppo di utenti è stato aggiunto alle preferenze di condivisione del dashboard.
- Possono visualizzare e clonare un dashboard privato se dispongono almeno dei diritti di *Lettura* su di esso, come specificato nelle preferenze di condivisione.
- Possono modificare ed eliminare un dashboard solo se dispongono dei diritti di *Lettura-scrittura* su di esso, come specificato nelle preferenze di condivisione.

- Non possono modificare il proprietario del dashboard.

Widget della dashboard

Panoramica

Zabbix offre una varietà di widget della dashboard che visualizzano e riassumono i dati di monitoraggio, fornendoti una chiara panoramica delle prestazioni e della disponibilità del sistema.

Di seguito è riportata una panoramica di tutti i widget, raggruppati in base al loro scopo principale.

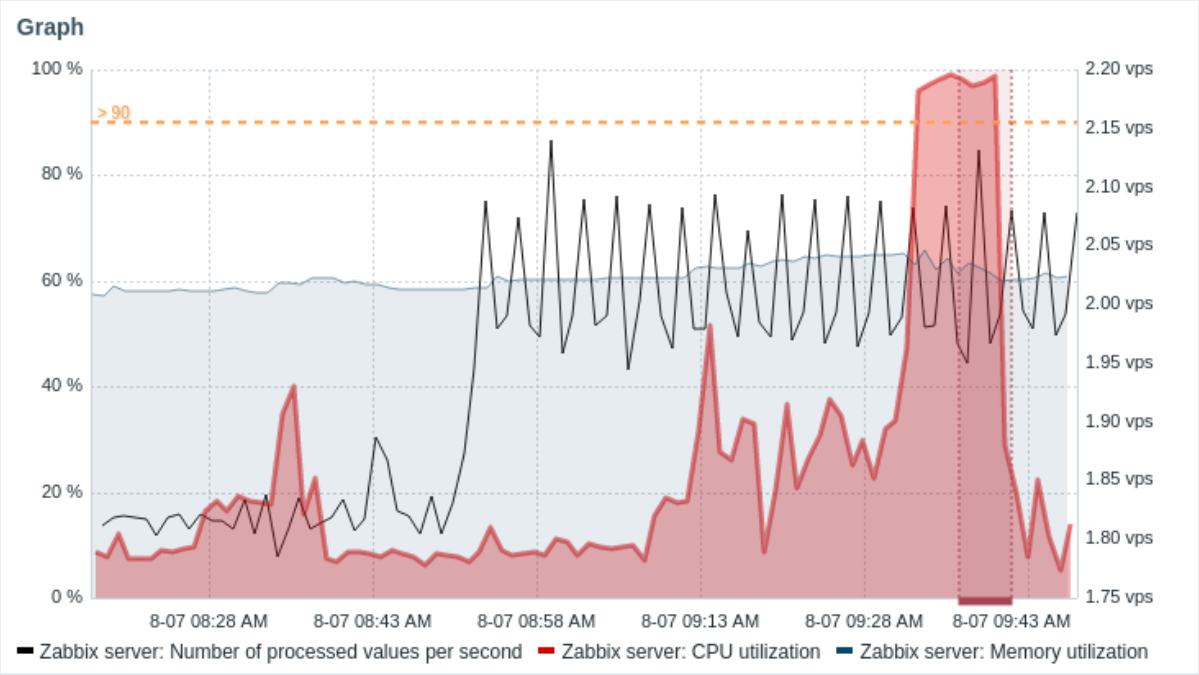
Per i dettagli sulla configurazione dei widget, vedere Dashboard widget parameters.

Visualizzazione dei dati

Questi widget presentano i dati in vari formati visivi, aiutandoti a monitorare i dati e i problemi in modi diversi.

Grafico

Il widget Graph visualizza i dati numerici degli item come un grafico vettoriale. Può aiutarti a monitorare le metriche, individuare i problemi e confrontare i valori nel tempo e tra host.



Vedi anche:

- Graph (classic) - il widget Graph legacy che visualizza i dati numerici degli item come un grafico basato su immagini.
- Graph prototype - visualizza una griglia di grafici personalizzati creati automaticamente in base ai prototipi di grafico o ai prototipi di item delle regole di low-level discovery.
- Favorite graphs - visualizza un elenco alfabetico di scorciatoie ai grafici contrassegnati come preferiti dall'utente corrente.

Grafico a dispersione

Il widget Scatter plot mostra la relazione tra due metriche tracciando singoli punti dati lungo un asse X e un asse Y. Questo aiuta a individuare schemi, gruppi, correlazioni e valori anomali nel set di dati.

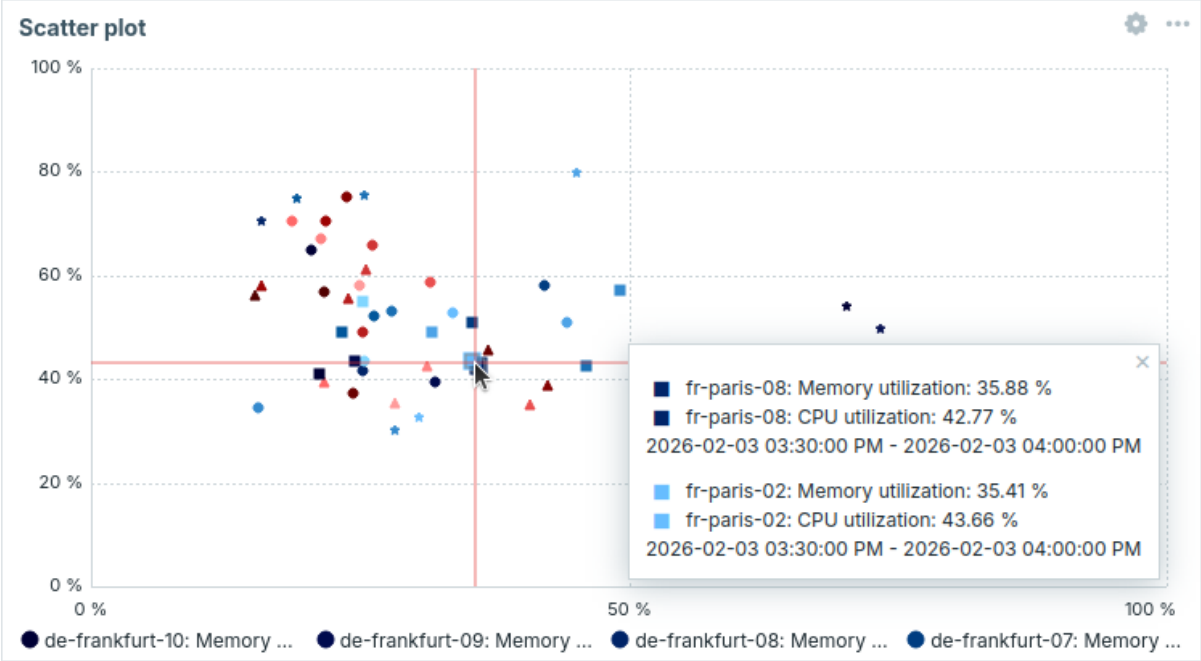
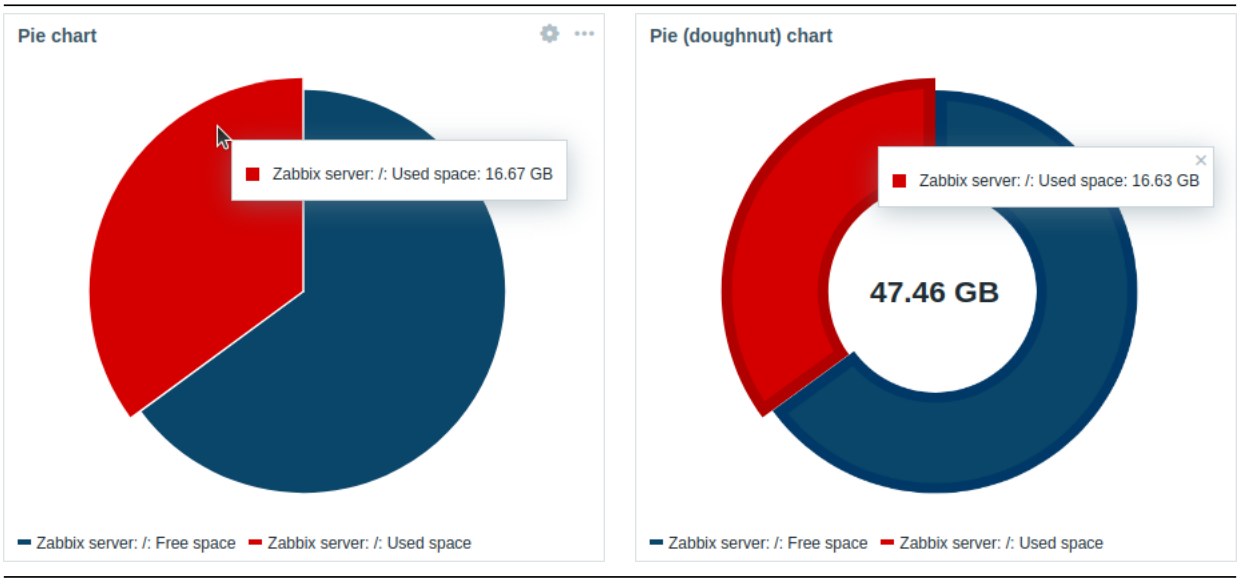


Grafico a torta

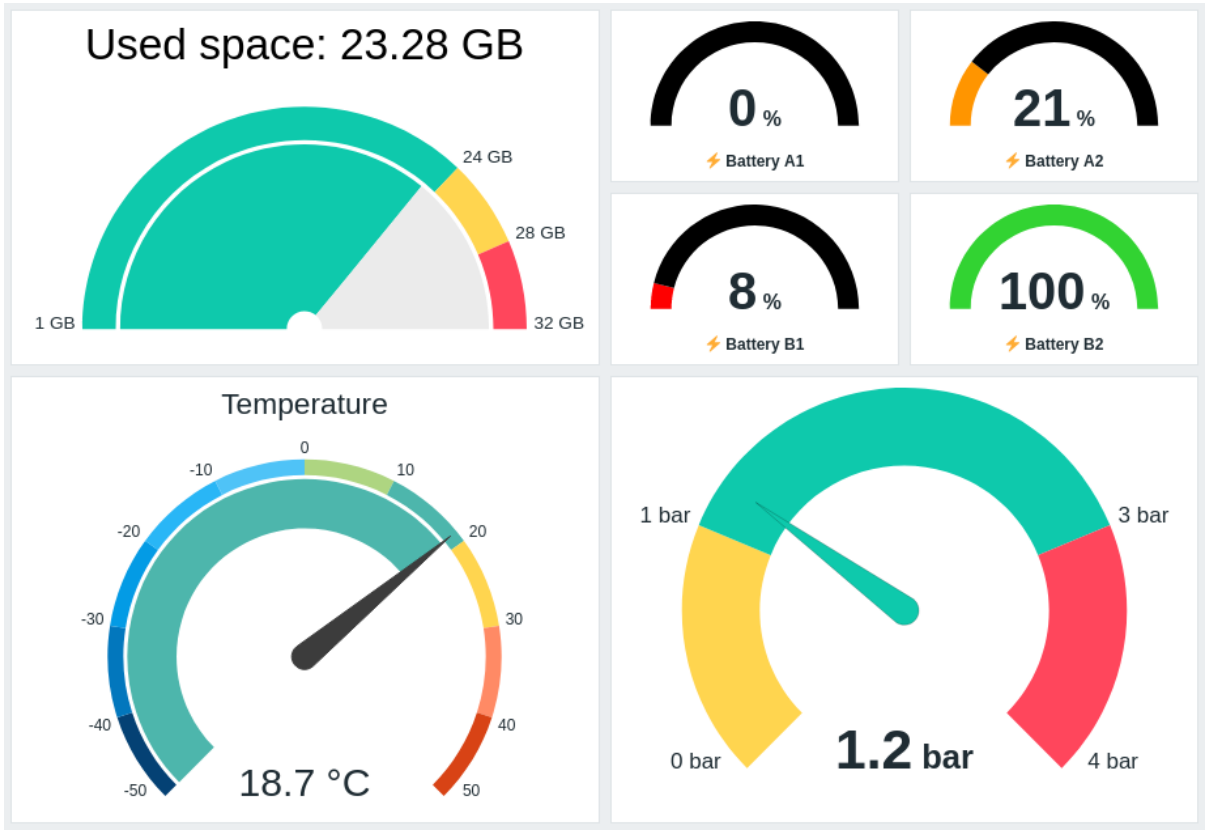
Il widget Pie chart visualizza i dati numerici degli item come un grafico a torta o ad anello vettoriale. Questo widget può aiutarti a visualizzare come gli item o gli host contribuiscono al dataset complessivo.



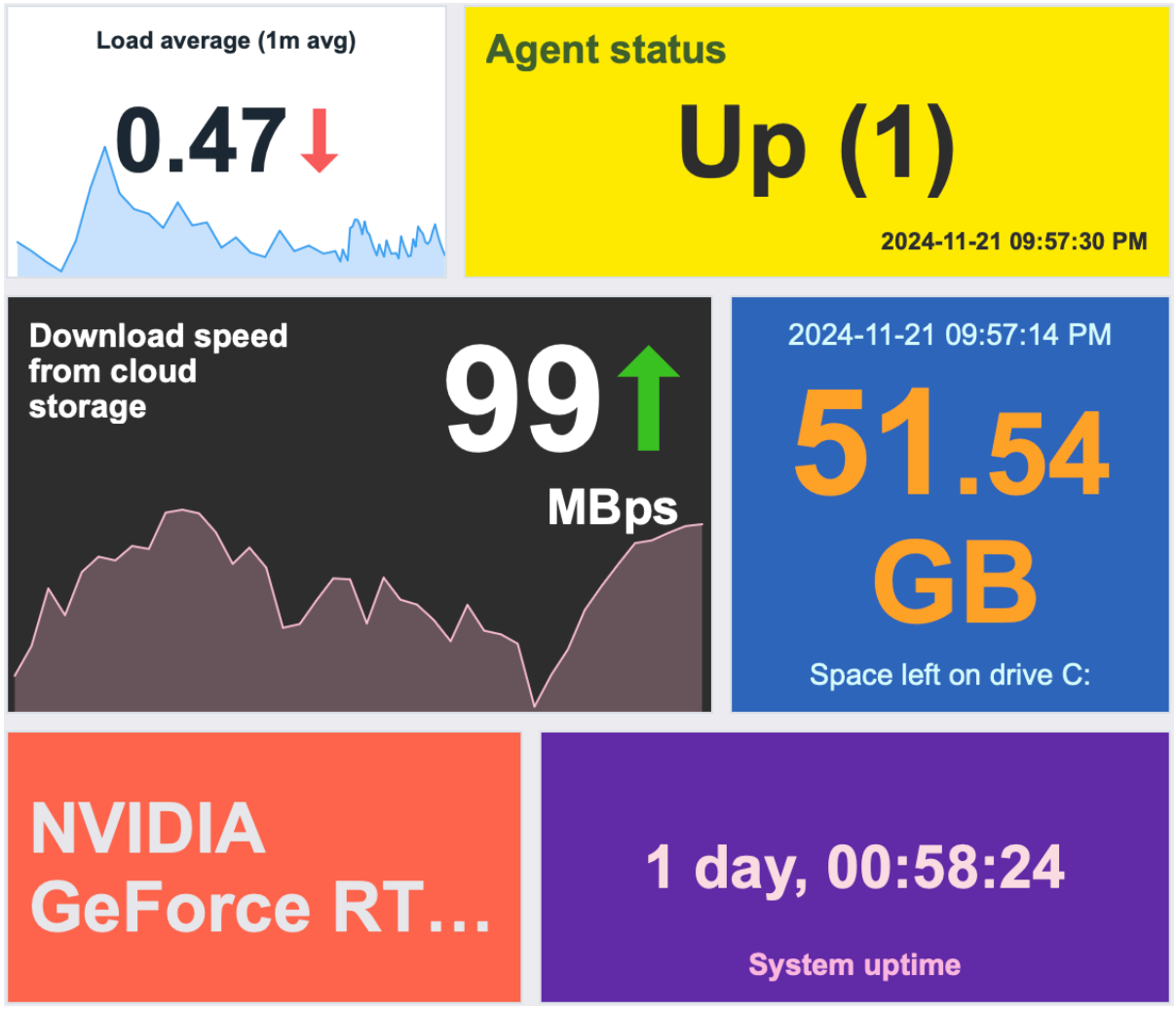
Valore indicatore/Valore item

I widget Indicatore/Valore item visualizzano il valore di un singolo item, utili per tenere sotto controllo metriche importanti, visualizzare le soglie e rilevare cambiamenti improvvisi nei dati.

Widget Indicatore:

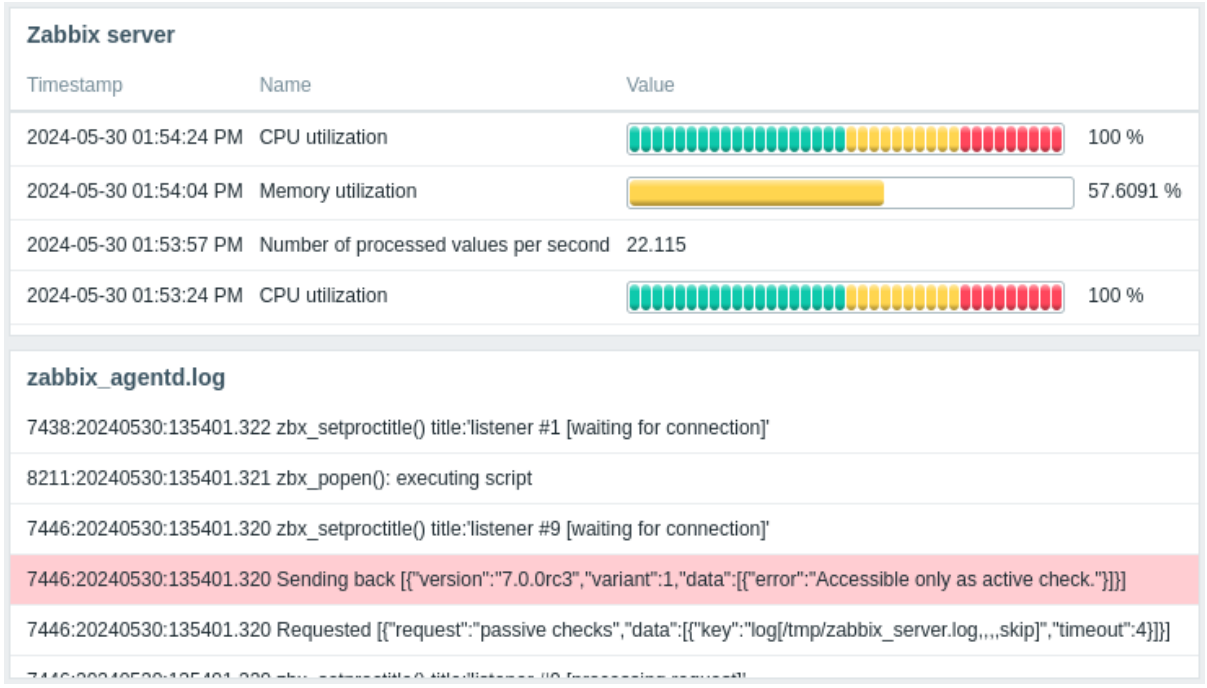


Widget Valore item:



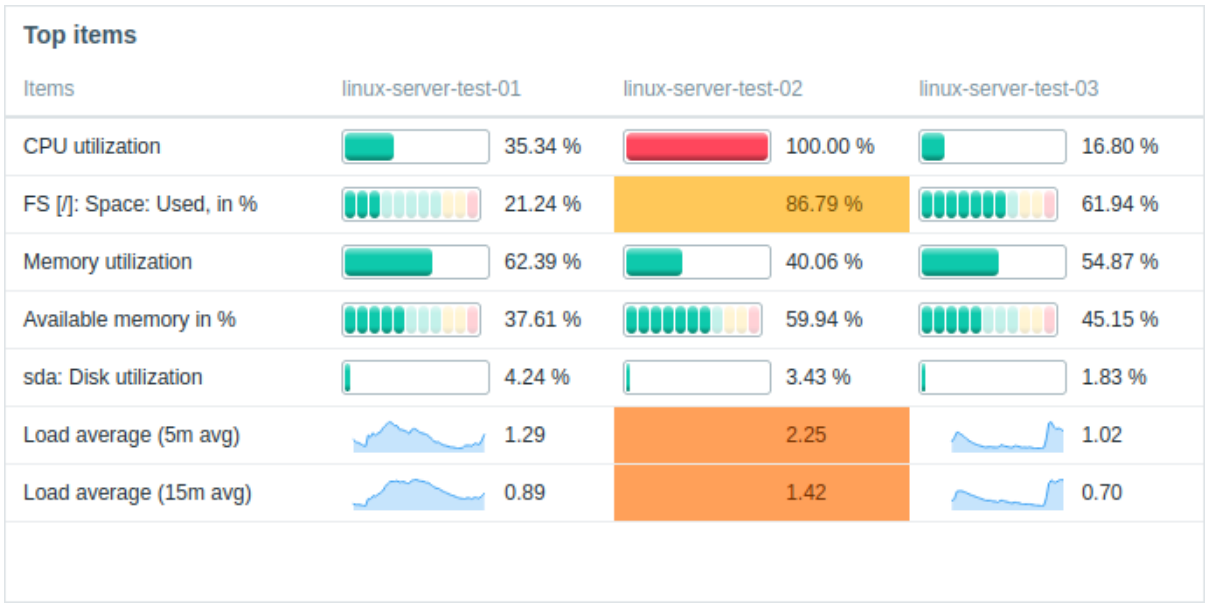
Cronologia degli item

Il widget Cronologia degli item visualizza i dati più recenti per tutti i tipi di item (numerici, testuali, ecc.) in formato tabellare. Può mostrare barre di avanzamento e immagini (utili per gli item del browser) e può evidenziare i valori (utile per il monitoraggio dei file di log).



Item principali

Il widget Top items visualizza i valori massimi/minimi per gli item selezionati, fornendo una rapida panoramica delle loro prestazioni.

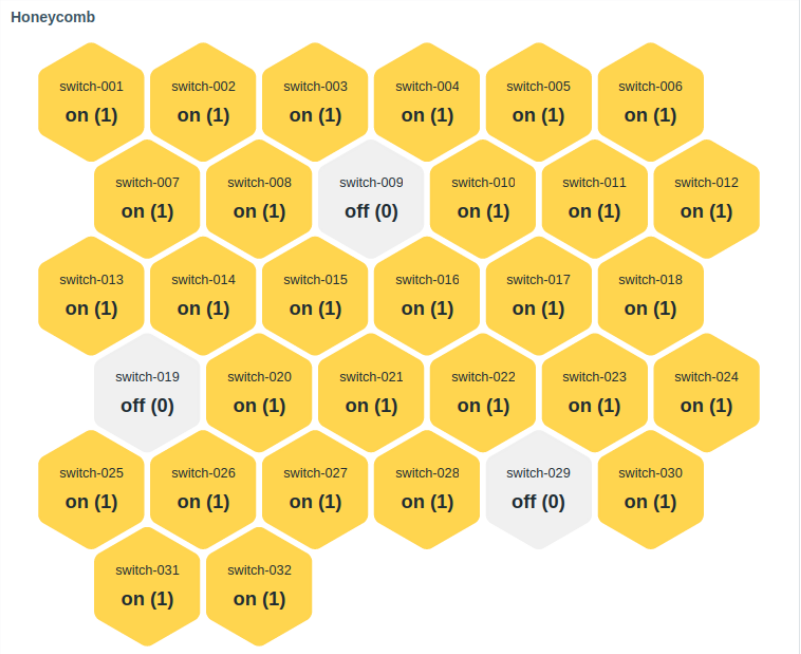


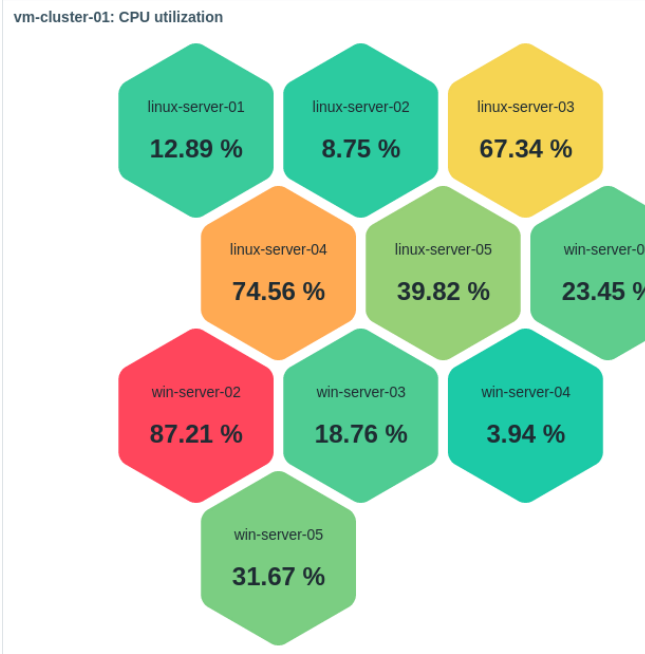
Vedi anche:

- Top hosts - visualizza i valori massimi/minimi per gli item selezionati da più host.
- Top triggers - visualizza i trigger con il maggior numero di problemi.

Nido d'ape

Il widget Honeycomb visualizza host o item come celle esagonali in una griglia. Questa disposizione consente di individuare facilmente host/item con problemi, riconoscere gruppi di problemi, confrontare gruppi a colpo d'occhio e altro ancora.



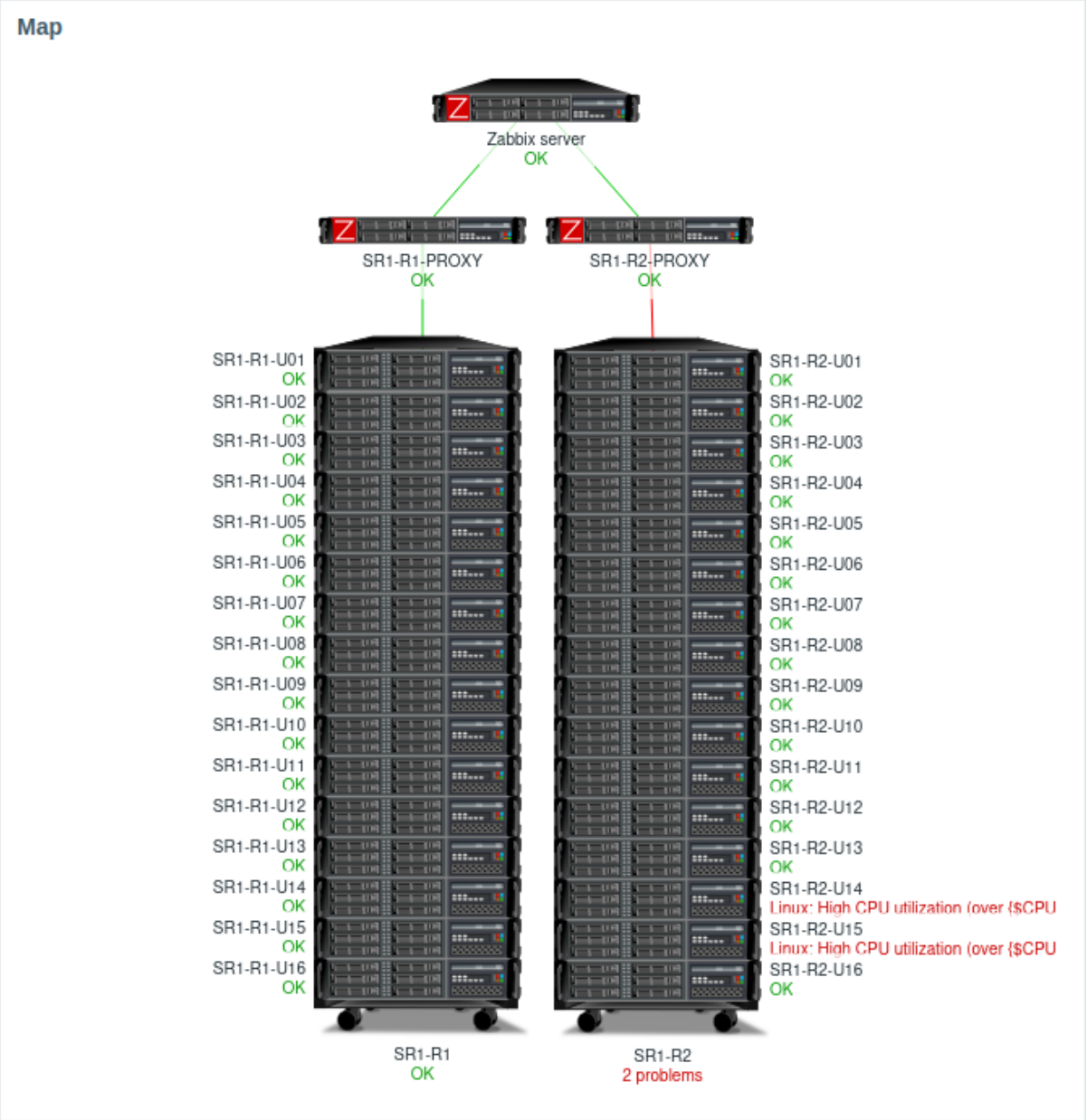


Mappe

I widget mappa visualizzano la topologia di rete o le posizioni degli host, aiutando a valutare lo stato generale e quello specifico per posizione.

Mappa

Il widget Mappa può visualizzare una mappa di rete o una visualizzazione simile, fornendoti una panoramica dinamica della tua rete.

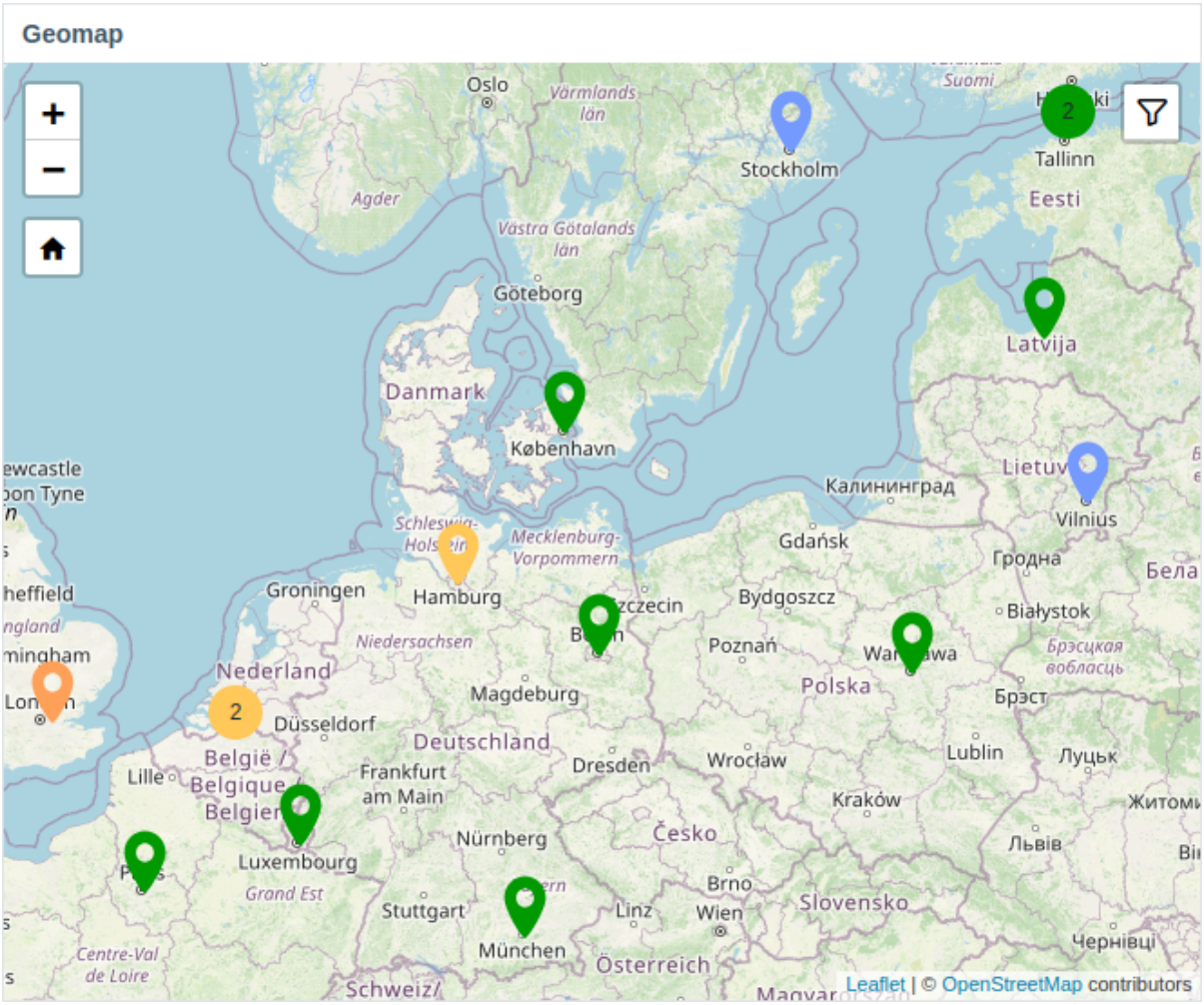


Vedi anche:

- Albero di navigazione delle mappe - visualizza una gerarchia delle mappe esistenti e il numero di problemi per ciascuna mappa e gruppo di mappe.
- Mappe preferite - visualizza un elenco di collegamenti rapidi alle mappe contrassegnate come preferite dall'utente corrente, ordinate alfabeticamente.

Geomap

Il widget Geomap visualizza gli host come marcatori su una mappa interattiva basata su Leaflet, con i colori dei marcatori che indicano il problema più grave di ciascun host.

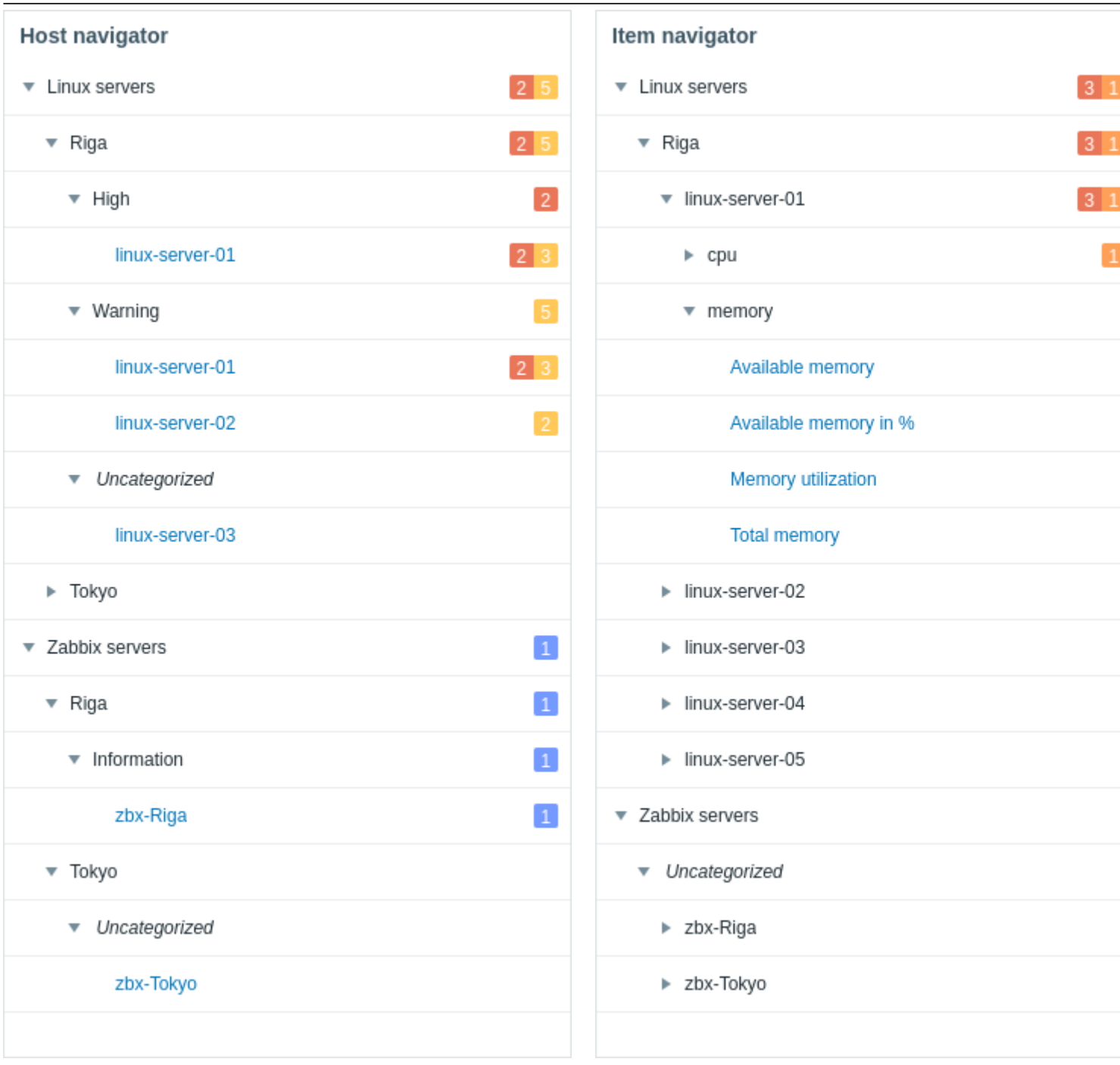


Dettagli di host/item

Questi widget forniscono elenchi e viste dettagliate di host o item.

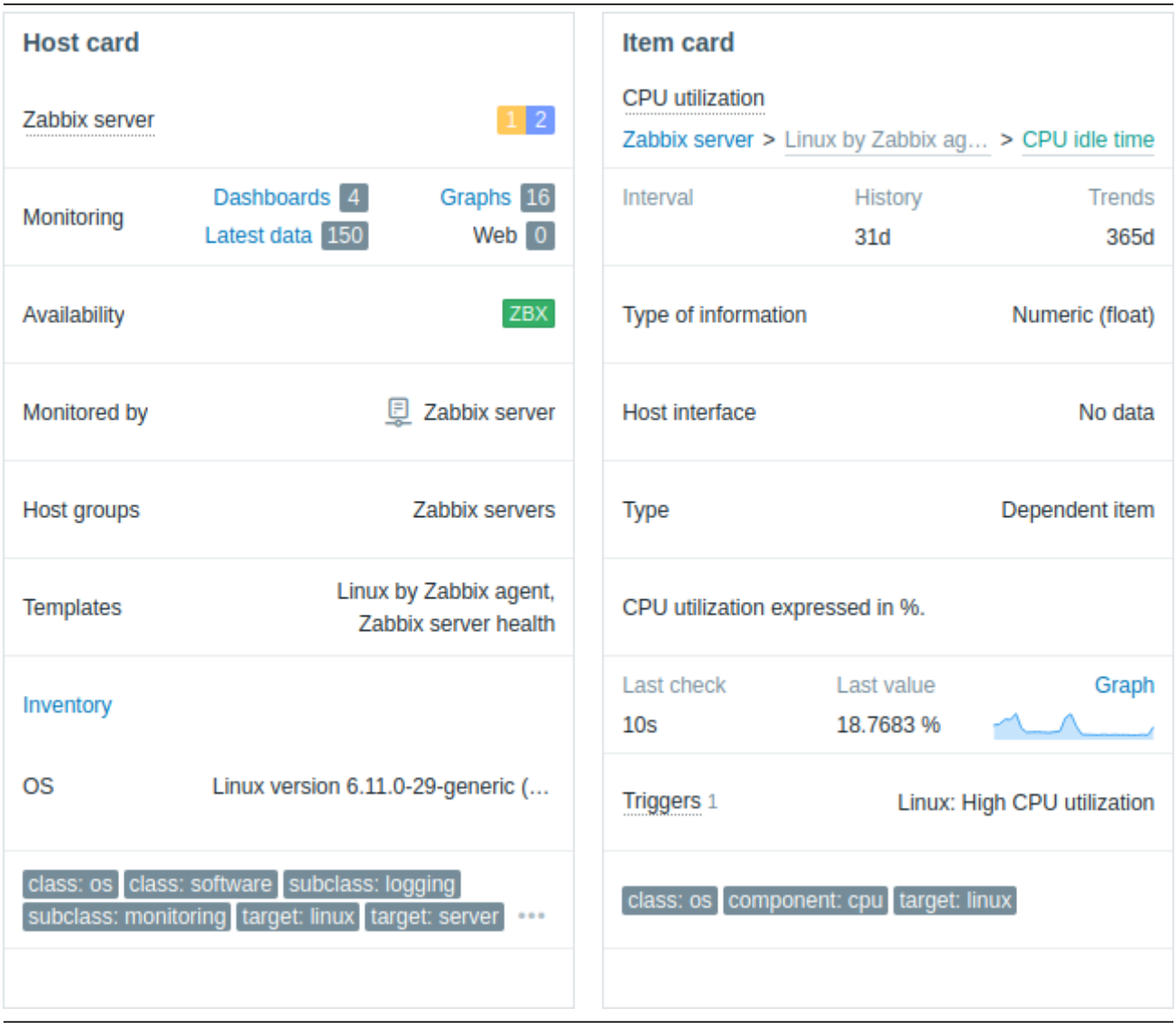
Navigatore host/item

I widget Navigatore host/Navigatore item visualizzano un elenco di host/item in base a varie opzioni di filtro e raggruppamento. Questi widget sono particolarmente utili per controllare ciò che viene visualizzato da altri widget in base all'host/item selezionato.



Scheda host/item

I widget Scheda host/Scheda item visualizzano i dettagli di un singolo host/item, consentendo di valutare rapidamente lo stato e la configurazione di un host o di un item (disponibilità dell'host, problemi, inventario, dati più recenti dell'item, errori, descrizione, ecc.).

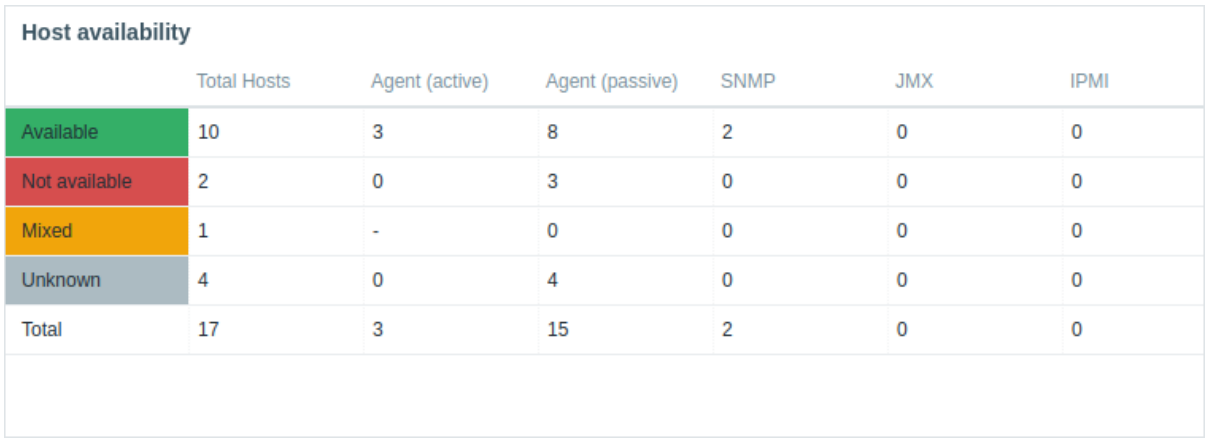


Stato e report

Questi widget visualizzano dati di stato (ad esempio disponibilità, trigger, problemi) nonché report.

Disponibilità dei host

Il widget Disponibilità dei host visualizza la disponibilità dei host nei gruppi di host selezionati, consentendo di monitorare il numero di host attivi o non attivi.

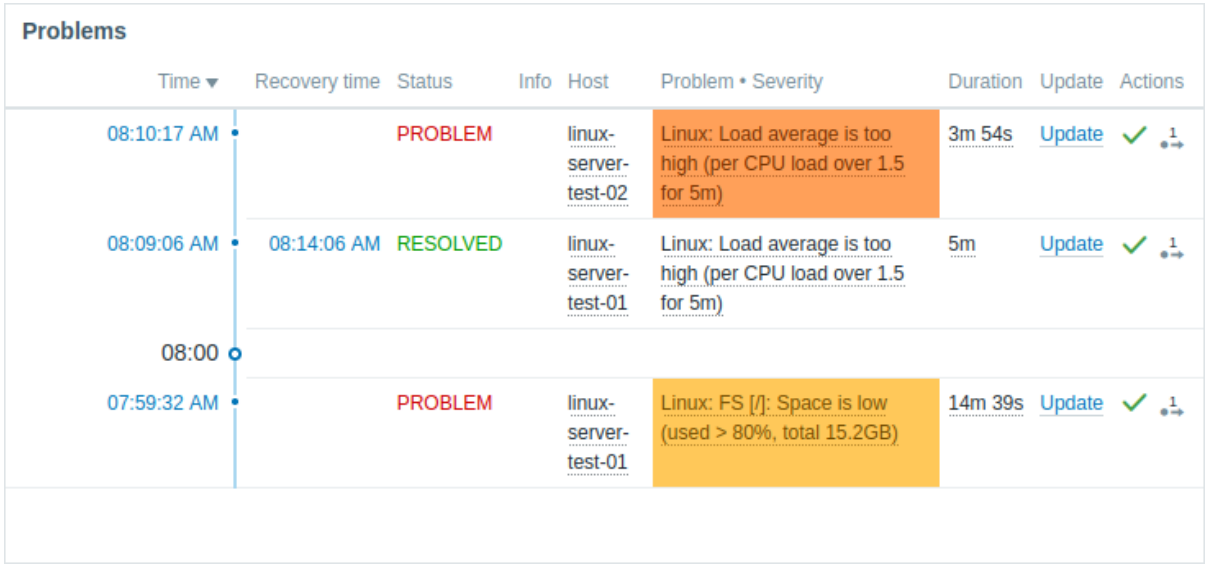


Host availability

	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

Problemi

Il widget Problems visualizza i problemi correnti filtrati in base a diversi parametri (gruppi di host, host, nomi dei problemi, ecc.), offrendo una vista chiara di ciò che richiede attenzione.



Vedi anche:

- Problems by severity - visualizza il numero di problemi causa raggruppati per gravità.
- Problem hosts - visualizza il numero di problemi causa per gruppo di host e mostra la gravità del problema più alta in ciascun gruppo.

Panoramica trigger

Il widget Trigger overview visualizza gli stati correnti dei trigger dei host come una tabella di blocchi colorati. Fornisce una rapida panoramica visiva dello stato di salute e dell'attività dei trigger su host diversi.



Log delle azioni

Il widget Log delle azioni visualizza i dettagli delle operazioni (notifiche, comandi remoti) eseguite all'interno di un'azione. Il log dettagliato aiuta nelle attività di auditing, risoluzione dei problemi e monitoraggio dell'esecuzione di ciascuna azione.

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

Report SLA

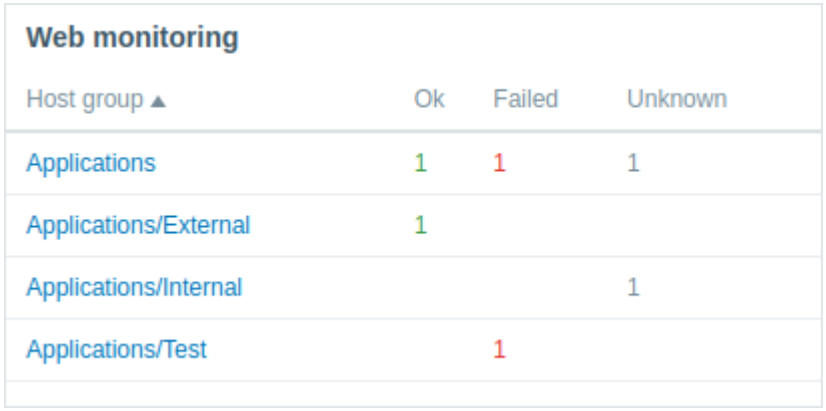
Il widget SLA report visualizza report SLA, aiutandoti a monitorare le prestazioni del servizio rispetto agli obiettivi definiti.

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

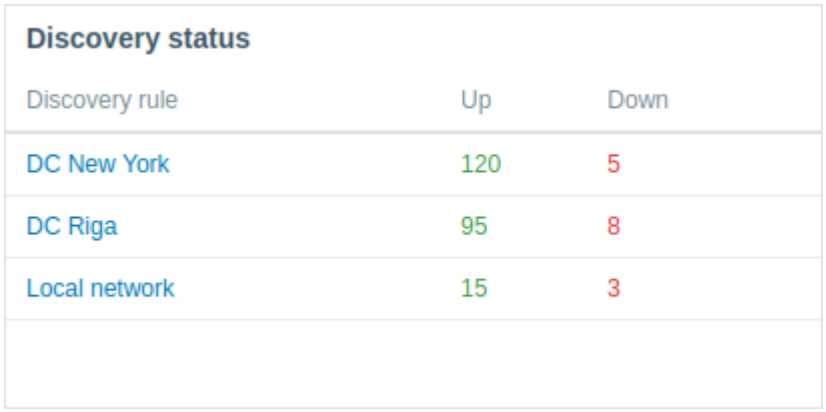
Monitoraggio web

Il widget Web monitoring visualizza lo stato degli scenari web attivi.



Stato del rilevamento

Il widget Discovery status visualizza lo stato dei dispositivi rilevati dalle regole di rilevamento di rete abilitate. Ti aiuta a monitorare se il rilevamento di rete funziona correttamente e il numero di dispositivi attivi o non attivi.



Utilità della dashboard e informazioni di sistema

Questi widget forniscono informazioni supplementari e strumenti per migliorare le dashboard.

Informazioni di sistema

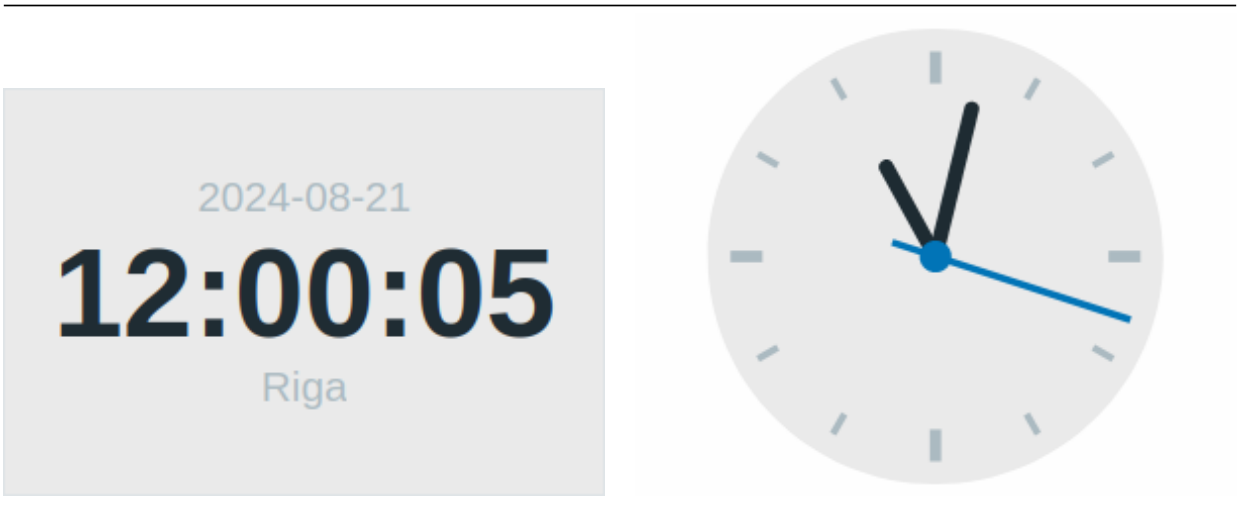
Il widget Informazioni di sistema visualizza un riepilogo dei principali dati del server Zabbix e del sistema oppure i dettagli dei nodi ad alta disponibilità.

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	

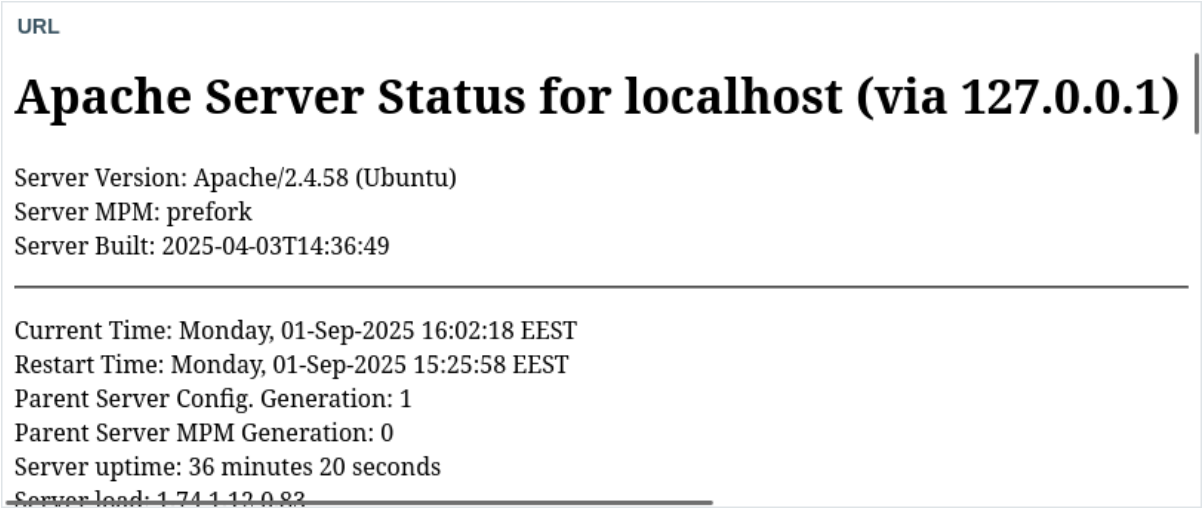
Orologio

Il widget Clock visualizza l'ora locale, del server o dell'host in formato analogico o digitale.



URL

Il widget URL visualizza il contenuto recuperato da un URL.



Parametri del widget della dashboard

Panoramica

Questa pagina descrive i parametri comuni a tutti i widget della dashboard, nonché i parametri dinamici che consentono a più widget di scambiarsi dati di configurazione tra loro o con la dashboard.

Per visualizzare i parametri specifici di ciascun widget, utilizzare la barra laterale per passare alla pagina del widget corrispondente.

Parametri comuni

I seguenti parametri sono comuni a ogni widget:

Nome	Inserire il nome del widget.
Intervallo di aggiornamento	Configurare l'intervallo di aggiornamento predefinito. Gli intervalli di aggiornamento predefiniti per i widget vanno da *Nessun aggiornamento* a *15 minuti*, a seconda del tipo di widget. Ad esempio: - *Nessun aggiornamento* per il widget *URL*; - *1 minuto* per il widget *Registro delle azioni*; - *15 minuti* per il widget *Orologio*. È possibile impostare un valore predefinito per gli intervalli di aggiornamento per tutti gli utenti. Passare la dashboard alla modalità di modifica, fare clic sul pulsante modifica un widget e selezionare l'intervallo di aggiornamento desiderato dall'elenco a discesa. Ogni utente può anche impostare il proprio intervallo di aggiornamento del widget. Nella modalità di visualizzazione della dashboard, fare clic sul pulsante con i tre puntini su un widget e selezionare l'intervallo di aggiornamento desiderato dall'elenco a discesa. L'intervallo di aggiornamento personalizzato dell'utente ha la precedenza sull'impostazione del widget e viene mantenuto anche quando l'impostazione del widget viene modificata.
Mostra intestazione	Selezionare la casella di controllo per visualizzare permanentemente l'intestazione del widget. Se non è selezionata, l'intestazione viene nascosta per risparmiare spazio e diventa visibile solo quando si passa con il mouse sul widget (sia in modalità di visualizzazione che di modifica). L'intestazione è inoltre parzialmente visibile quando si trascina un widget in una nuova posizione.

Parametri dinamici

Diversi parametri dei widget (ad esempio *Hosts*, *Override host*, *Time period*) supportano origini dati dinamiche. Invece di configurare valori statici, è possibile collegare questi parametri ad altri widget o alla dashboard stessa. I widget vengono quindi aggiornati automaticamente in base alle selezioni o alle trasmissioni provenienti da altre parti della dashboard.

Compatibilità dei widget

Alcuni widget possono trasmettere dati di configurazione ad altri widget, alcuni possono ricevere dati e alcuni possono fare entrambe le cose. Per esempio:

- Il widget *Action log* può recuperare solo i dati del periodo di tempo dai widget *Graph*, *Graph (classic)* e *Graph prototype*.

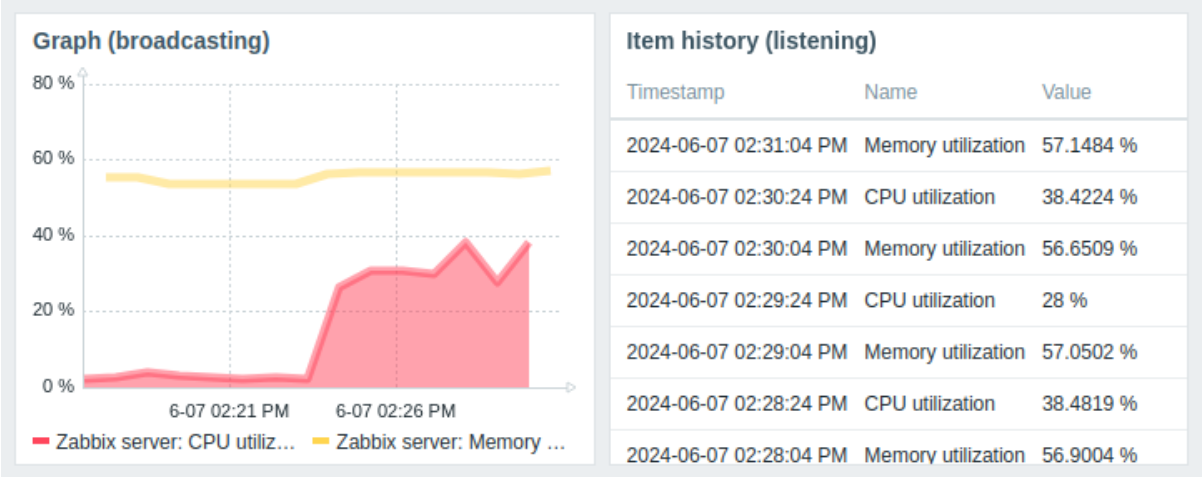
- Il widget *Geomap* può trasmettere dati dei host ai widget che li ricevono (*Honeycomb*, *Top items*, ecc.) e può anche ricevere dati di gruppi di host e host dai widget che li trasmettono (*Honeycomb*, *Problem hosts*, ecc.).
- Il widget *Clock* non può né trasmettere né ricevere alcun dato.

La tabella seguente riassume le capacità di trasmissione e ricezione di ciascun widget.

Widget	Trasmette	Riceve
Action log	-	Periodo di tempo
Clock	-	-
Discovery status	-	-
Favorite graphs	-	-
Favorite maps	-	-
Gauge	-	Host, Item
Geomap	Host	Gruppi di host, Host
Graph	Periodo di tempo, Item	Host, Item, Periodo di tempo
Graph (classic)	Periodo di tempo	Host, Item, Grafici, Periodo di tempo
Graph prototype	Periodo di tempo	Host, Periodo di tempo
Honeycomb	Host, Item	Gruppi di host, Host
Host availability	-	Gruppi di host
Host card	-	Host
Host navigator	Host	Gruppi di host
Item card	-	Host, Item, Periodo di tempo
Item history	Item	Host, Periodo di tempo
Item navigator	Item	Gruppi di host, Host
Item value	-	Host, Item, Periodo di tempo
Map	Gruppi di host, Host	Mappe
Map navigation tree	Mappe	-
Pie chart	-	Item, Periodo di tempo
Problem hosts	Gruppi di host	Gruppi di host, Host
Problems	Eventi	Gruppi di host, Host
Problems by severity	Gruppi di host	Gruppi di host, Host
Scatter plot	Periodo di tempo, Item	Host, Item, Periodo di tempo
SLA report	-	-
System information	-	-
Top hosts	Host	Gruppi di host, Host
Top items	-	Gruppi di host, Host
Top triggers	-	Periodo di tempo
Trigger overview	-	Gruppi di host, Host
URL	-	Host
Web monitoring	Gruppi di host	Gruppi di host, Host

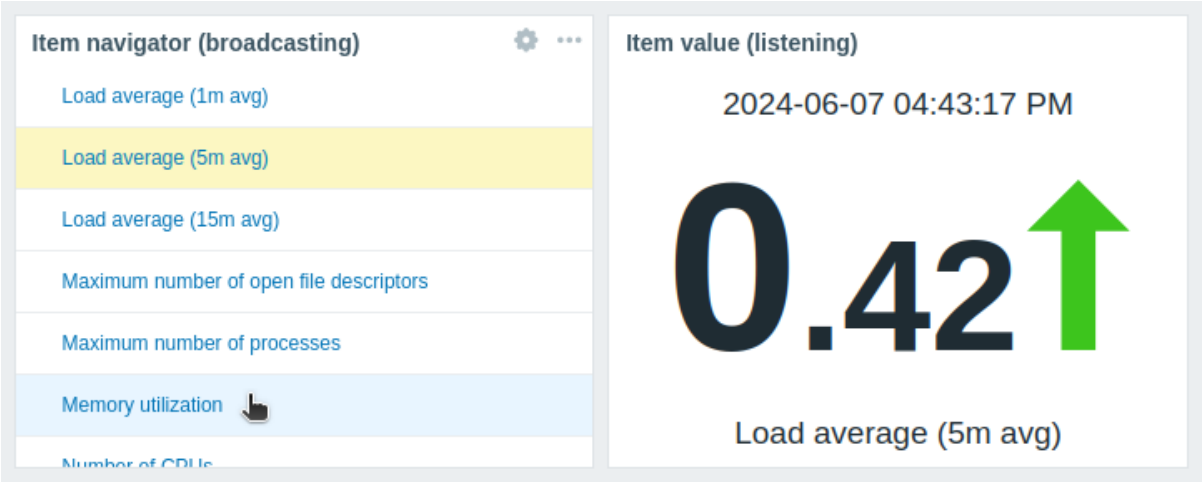
I widget differiscono nel modo in cui **trasmettono** dati ad altri widget.

I widget che supportano la trasmissione iniziano a farlo automaticamente quando vengono creati. Ad esempio, il widget *Graph* trasmette immediatamente i dati dell'intervallo di tempo ai widget in ascolto.

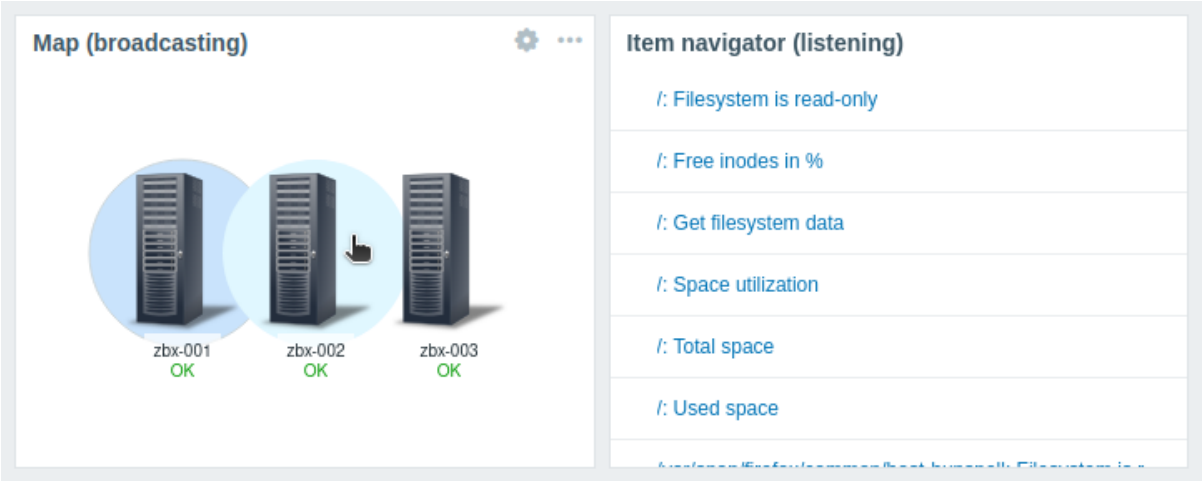


I widget che supportano la selezione di entità trasmettono automaticamente i dati per la prima entità disponibile. Ad esempio, il widget *Item navigator* trasmette i dati per il primo item nel proprio elenco di item. Quando viene selezionato un item diverso,

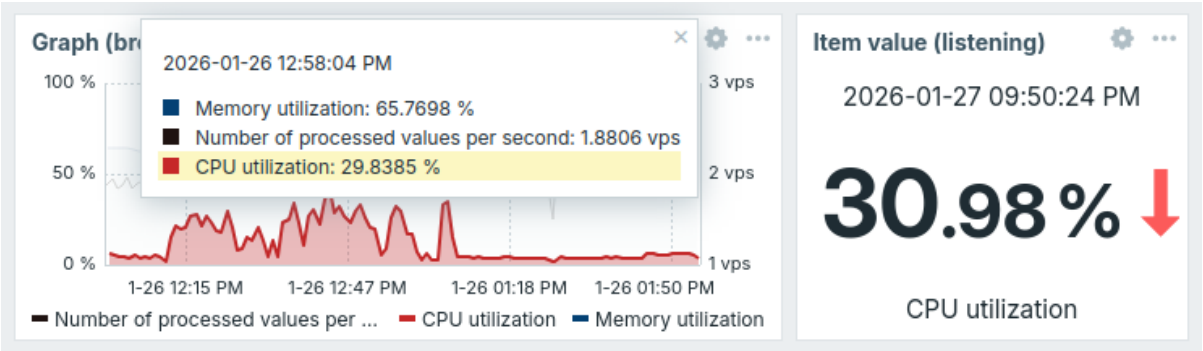
trasmette i dati di quell'item. Al passaggio del mouse, l'item viene evidenziato in azzurro chiaro; quando viene selezionato, viene evidenziato in giallo.



Il widget *Map* si comporta in modo simile, trasmettendo i dati per il primo elemento più vicino all'angolo superiore sinistro dell'area visibile del widget (per il widget *Geomap*, quello più vicino al centro). Quando viene selezionato un elemento diverso, trasmette i dati di quell'elemento. Al passaggio del mouse, l'elemento viene evidenziato in azzurro chiaro; quando viene selezionato, viene evidenziato in blu scuro.

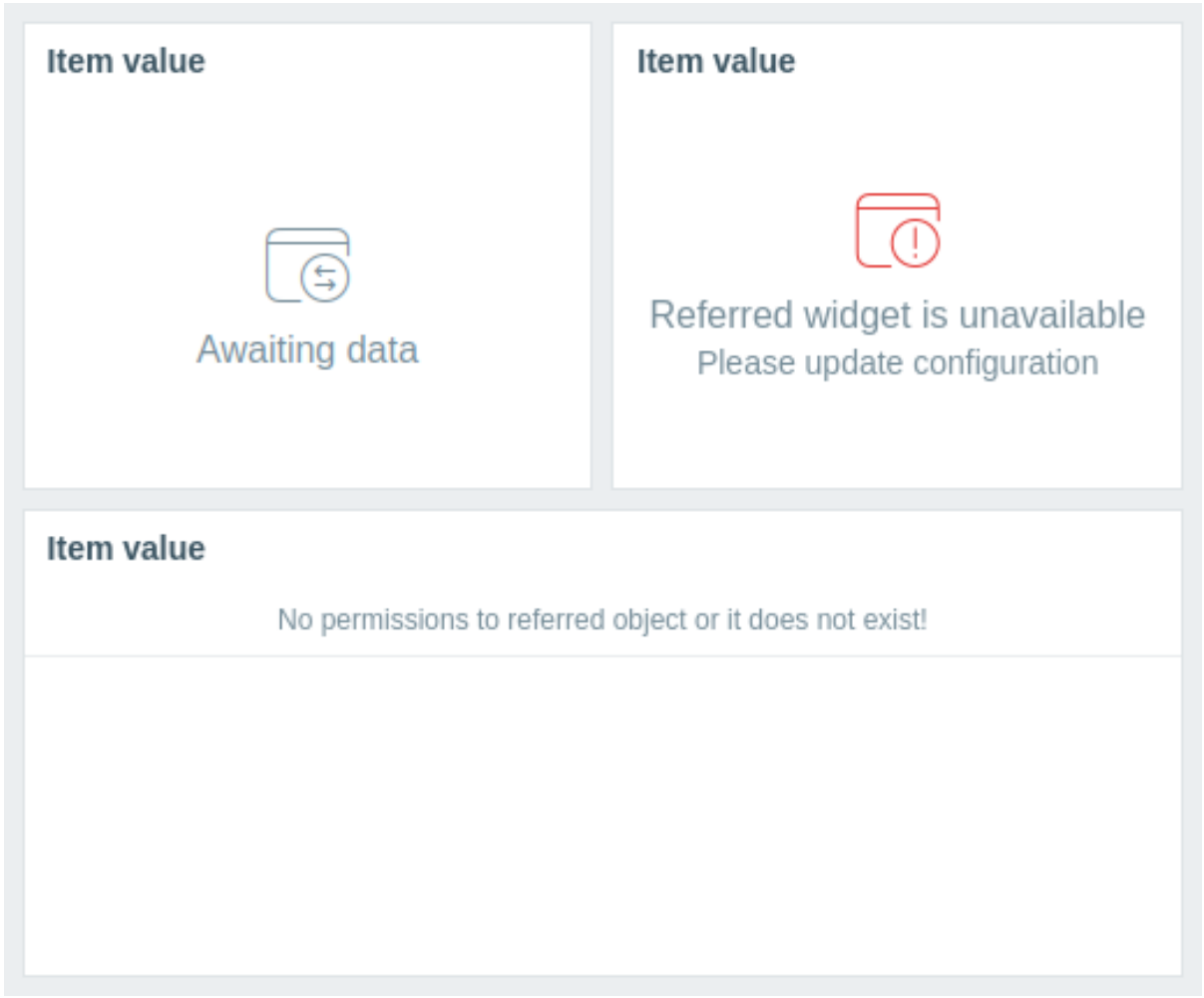


Widget come *Graph* e *Scatter plot* possono trasmettere dati degli item quando un item viene selezionato nei relativi tooltip:



I widget hanno anche comportamenti specifici quando **ricevono** dati da altri widget:

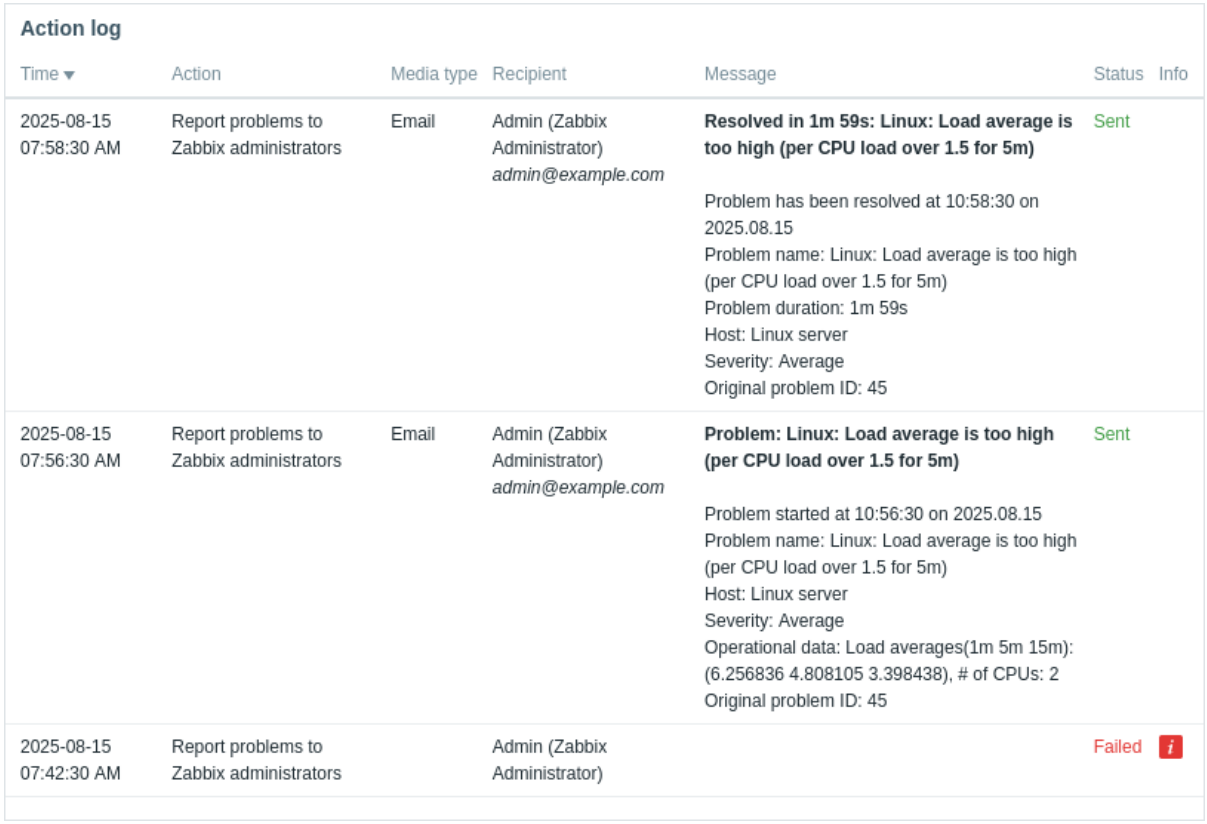
- Se il widget origine dati non sta trasmettendo dati, il widget in ascolto entra nello stato *In attesa di dati*.
- Se il widget origine dati è stato eliminato, sostituito con un widget incompatibile oppure spostato in un'altra pagina della dashboard, il widget in ascolto entra nello stato *Il widget di riferimento non è disponibile*.
- Se l'host specificato nell'origine dati (widget o dashboard) non dispone dell'entità configurata nel widget in ascolto (item, graph, map, ecc.) oppure se l'utente non dispone delle autorizzazioni per accedere all'host, il widget in ascolto visualizza il seguente messaggio: *"Nessun permesso per l'oggetto di riferimento oppure non esiste!"*



Registro azioni

Panoramica

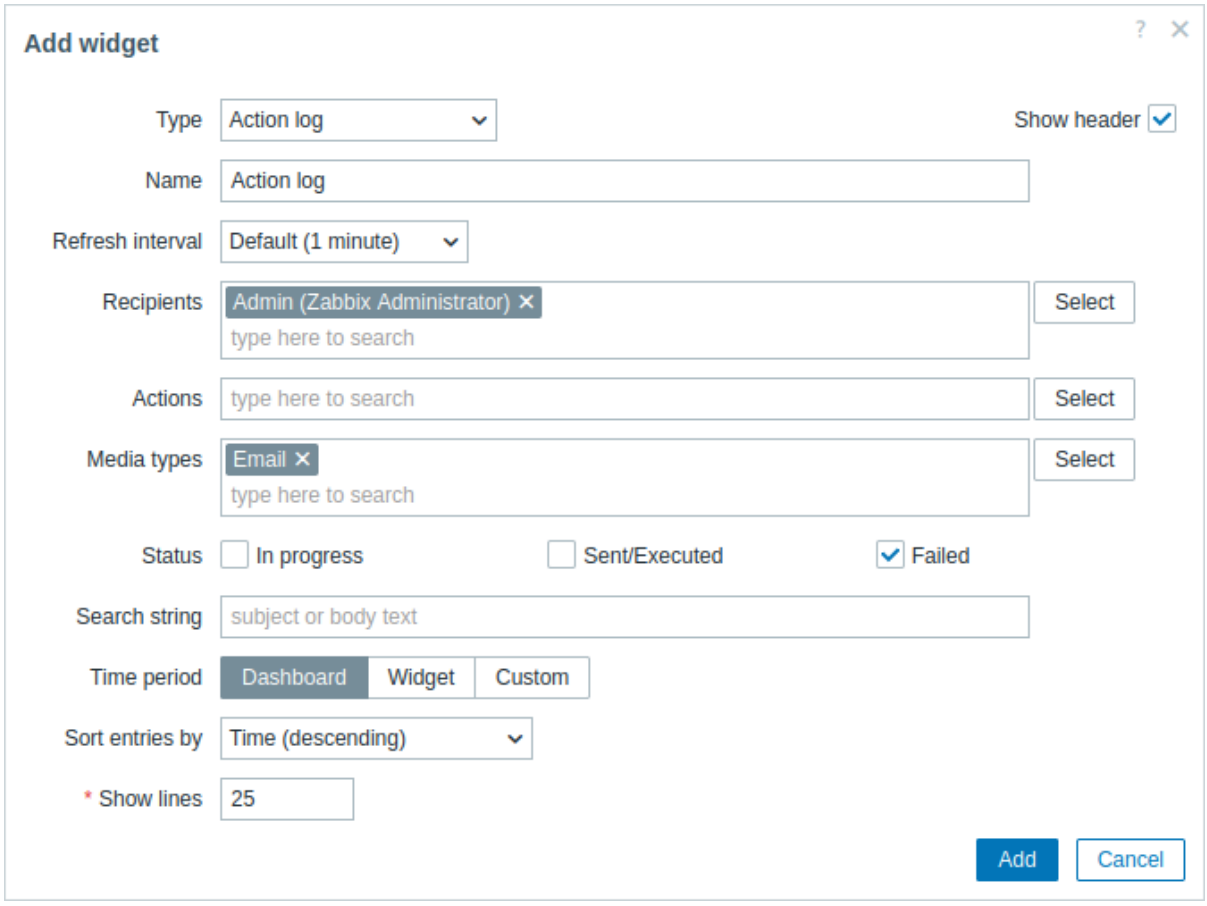
Il widget *Log azioni* visualizza i dettagli delle operazioni (notifiche, comandi remoti) eseguite all'interno di un'azione. Il log dettagliato aiuta nelle attività di audit, risoluzione dei problemi e monitoraggio dell'esecuzione di ciascuna azione.



Visualizza gli stessi dati di *Report* > *Log azioni* e può mostrare fino a 1000 record.

Configurazione

Per configurare il widget, selezionare *Registro azioni* come tipo:



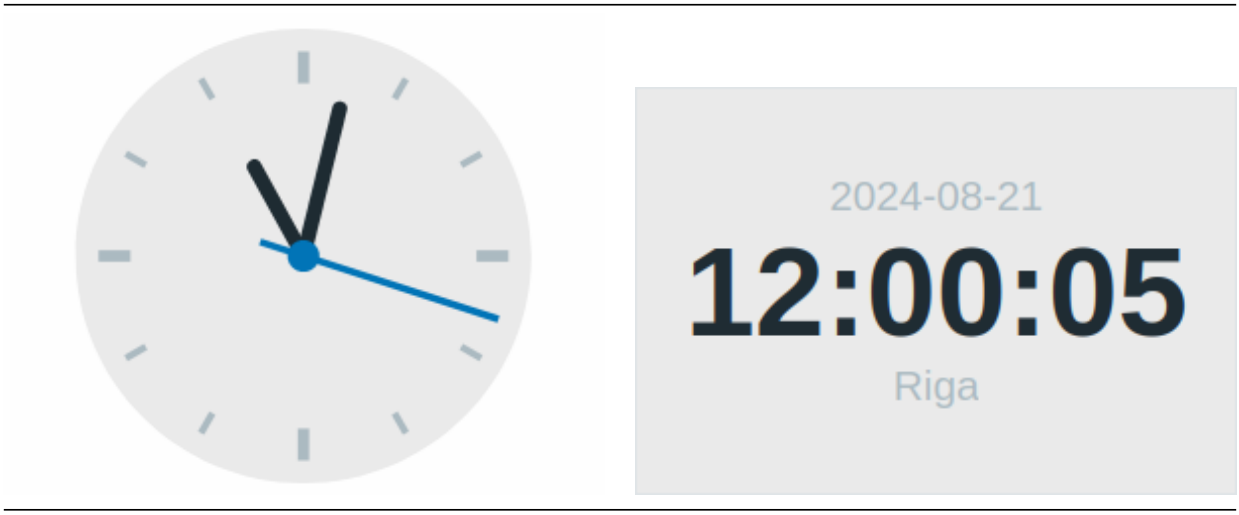
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Destinatari	Filtra le voci per destinatari. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un destinatario verrà visualizzato un menu a discesa con i destinatari corrispondenti. Se non viene selezionato alcun destinatario, verranno visualizzati i dettagli delle operazioni delle azioni per tutti i destinatari.
Azioni	Filtra le voci per azioni. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un'azione verrà visualizzato un menu a discesa con le azioni corrispondenti. Se non viene selezionata alcuna azione, verranno visualizzati i dettagli delle operazioni delle azioni per tutte le azioni.
Tipi di supporto	Filtra le voci per tipi di supporto. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un tipo di supporto verrà visualizzato un menu a discesa con i tipi di supporto corrispondenti. Se non viene selezionato alcun tipo di supporto, verranno visualizzati i dettagli delle operazioni delle azioni per tutti i tipi di supporto.
Stato	Selezionare la casella di controllo per filtrare le voci in base al rispettivo stato: **In corso** - vengono visualizzate le operazioni delle azioni in corso; **Inviato/Eseguito** - vengono visualizzate le operazioni delle azioni che hanno inviato una notifica o sono state eseguite; **Non riuscito** - vengono visualizzate le operazioni delle azioni non riuscite.
Stringa di ricerca	Filtra le voci in base al contenuto del messaggio/comando remoto. Se si inserisce qui una stringa, verranno visualizzate solo le operazioni delle azioni il cui messaggio/comando remoto contiene la stringa inserita. Le macro non vengono risolte.
Periodo di tempo	Filtra le voci per periodo di tempo. Selezionare l'origine dati per il periodo di tempo: **Dashboard** - utilizza il selettore del periodo di tempo della dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizza un periodo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che indica l'ora impostata al passaggio del mouse. Si noti che, indipendentemente dalla configurazione di *Periodo di tempo* del widget, i widget compatibili possono comunque usarlo come origine dati per il periodo di tempo.
Widget	Immettere o selezionare un widget compatibile come origine dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Immettere o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
A	Immettere o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
Ordina voci per	Ordina le voci per: **Ora** (decrescente o crescente); **Tipo** (decrescente o crescente); **Stato** (decrescente o crescente); **Destinatario** (decrescente o crescente).
Mostra righe	Imposta quante righe del registro azioni verranno visualizzate nel widget.

Orologio

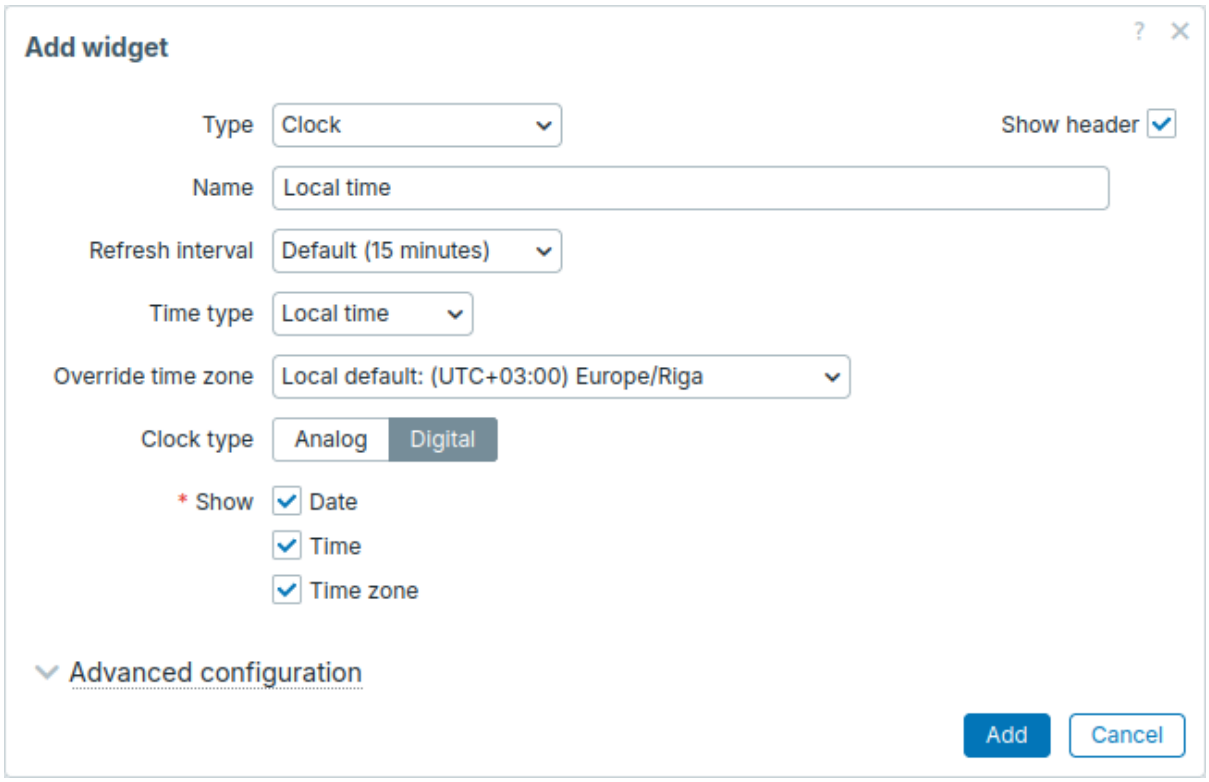
Panoramica

Il widget *Clock* visualizza l'ora locale, del server o dell'host in formato analogico o digitale.



Configurazione

Per configurare, selezionare *Orologio* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

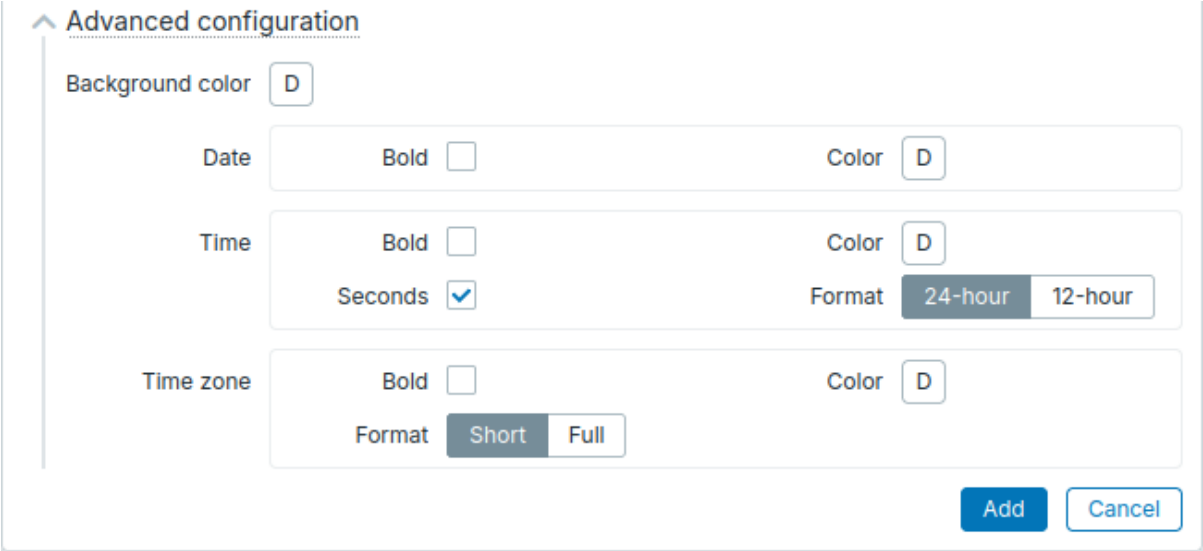
Tipo di ora	Selezionare la sorgente dell'ora: **Ora locale** - ora di sistema del dispositivo su cui è in esecuzione il browser; **Ora del server** - fuso orario impostato globalmente o per l'utente Zabbix; **Ora dell'host** - ora recuperata da un item dell'host.
Item	Selezionare l'item con la chiave `system.localtime[local]` per visualizzare l'ora dell'host. Questo parametro è disponibile se *Tipo di ora* è impostato su "Ora dell'host".
Tipo di orologio	Selezionare il tipo di orologio (analogico o digitale).
Sovrascrivi fuso orario	Selezionare il fuso orario. Questo parametro è disponibile per i tipi di ora *Ora locale* e *Ora del server*.
Mostra	Selezionare una o più unità informative (data, ora, fuso orario) da visualizzare nell'orologio digitale. La dimensione dell'unità si adatta automaticamente alle unità visualizzate, alle dimensioni del widget e al ridimensionamento dello schermo. Questo parametro è disponibile se *Tipo di orologio* è impostato su "Digitale".

Configurazione avanzata	Fare clic sull'intestazione *Configurazione avanzata* per visualizzare le opzioni di configurazione avanzata dell'orologio digitale. Questa sezione è disponibile se *Tipo di orologio* è impostato su "Digitale".

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata* e solo per gli elementi selezionati nel campo *Mostra* (vedere sopra).

Inoltre, la configurazione avanzata consente di modificare il colore di sfondo dell'intero widget.

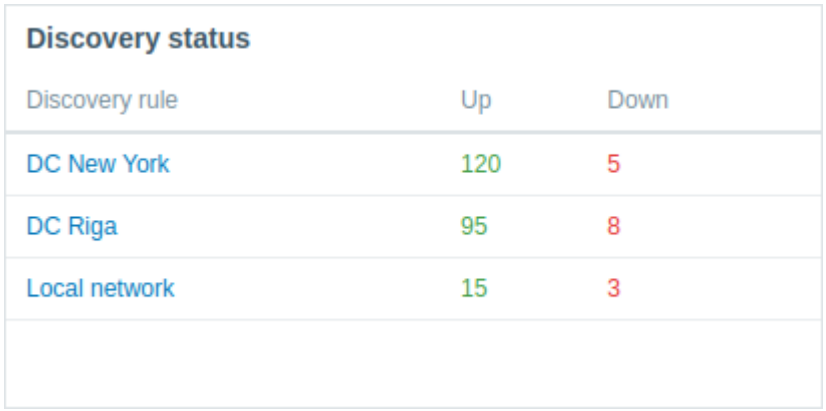


Colore di sfondo		Selezionare il colore di sfondo dal selettore dei colori. D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore dei colori.
Data		Questa sezione è disponibile quando la casella di controllo *Data* è selezionata per il parametro *Mostra*.
	Grassetto	Selezionare la casella di controllo per visualizzare la data in grassetto.
	Colore	Selezionare il colore della data dal selettore dei colori. D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore dei colori.
Ora		Questa sezione è disponibile quando la casella di controllo *Ora* è selezionata per il parametro *Mostra*.
	Grassetto	Selezionare la casella di controllo per visualizzare l'ora in grassetto.
	Colore	Selezionare il colore dell'ora dal selettore dei colori. D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore dei colori.
	Secondi	Selezionare la casella di controllo per visualizzare i secondi. In caso contrario, verranno visualizzate solo le ore e i minuti.
	Formato	Selezionare se visualizzare l'ora nel formato a 24 o a 12 ore.
Fuso orario		Questa sezione è disponibile quando la casella di controllo *Fuso orario* è selezionata per il parametro *Mostra*.
	Grassetto	Selezionare la casella di controllo per visualizzare il fuso orario in grassetto.
	Colore	Selezionare il colore del fuso orario dal selettore dei colori. D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore dei colori.
	Formato	Selezionare se visualizzare il fuso orario in formato breve (ad esempio, `New York`) o completo (ad esempio, `(UTC-04:00) America/New York`).

Stato di discovery

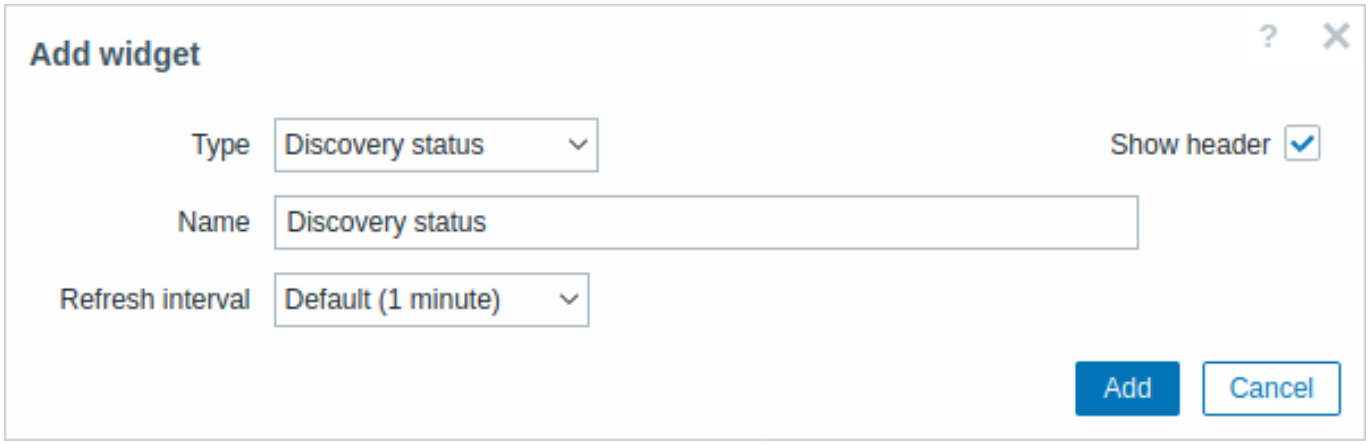
Panoramica

Il widget *Stato del discovery* visualizza lo stato dei dispositivi rilevati dalle regole di network discovery abilitate. Aiuta a monitorare se il network discovery funziona correttamente e il numero di dispositivi attivi o non attivi.



Configurazione

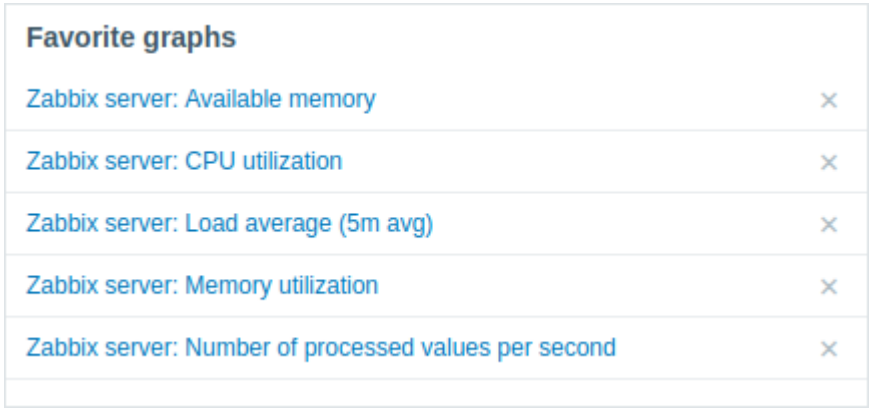
Tutti i parametri di configurazione per questo widget sono comuni a tutti i widget.



Grafici preferiti

Panoramica

Il widget *Grafici preferiti* visualizza un elenco alfabetico di collegamenti ai grafici contrassegnati come preferiti dall'utente corrente.



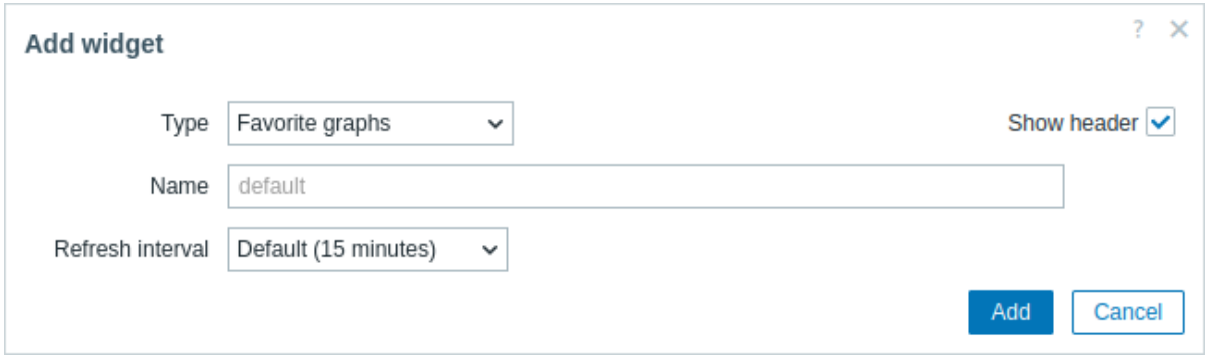
Facendo clic su x, il grafico viene rimosso dall'elenco e dai preferiti.

I grafici vengono contrassegnati come preferiti facendo clic sul pulsante  *Aggiungi ai preferiti* durante la visualizzazione dei grafici degli item in *Monitoring* > *Ultimi dati*.



Configurazione

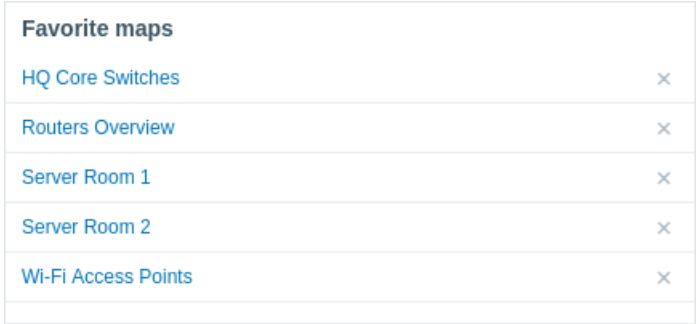
Tutti i parametri di configurazione per questo widget sono comuni a tutti i widget.



Mappe preferite

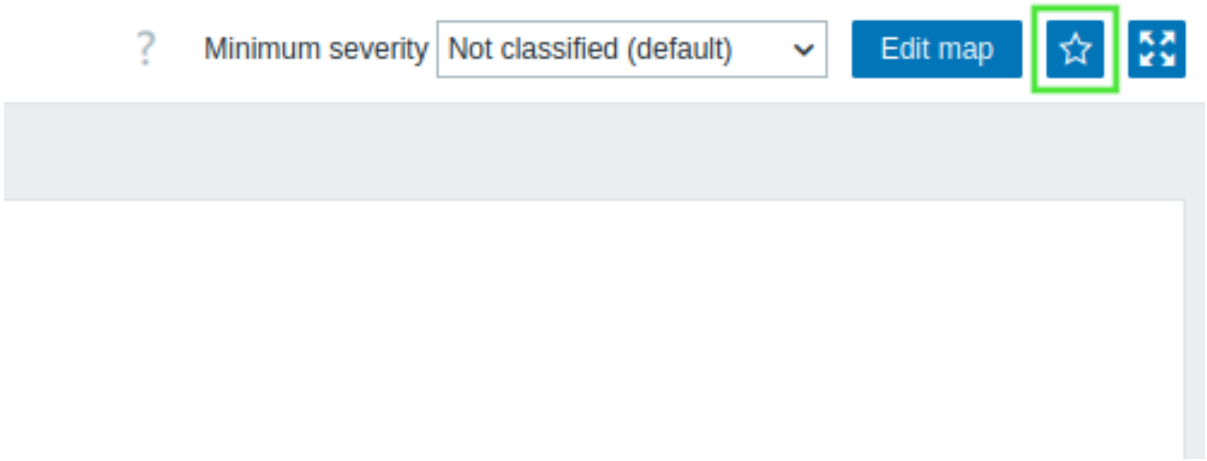
Panoramica

Il widget *Mappe preferite* visualizza un elenco di scorciatoie alle mappe contrassegnate come preferite dall'utente corrente, ordinate alfabeticamente.



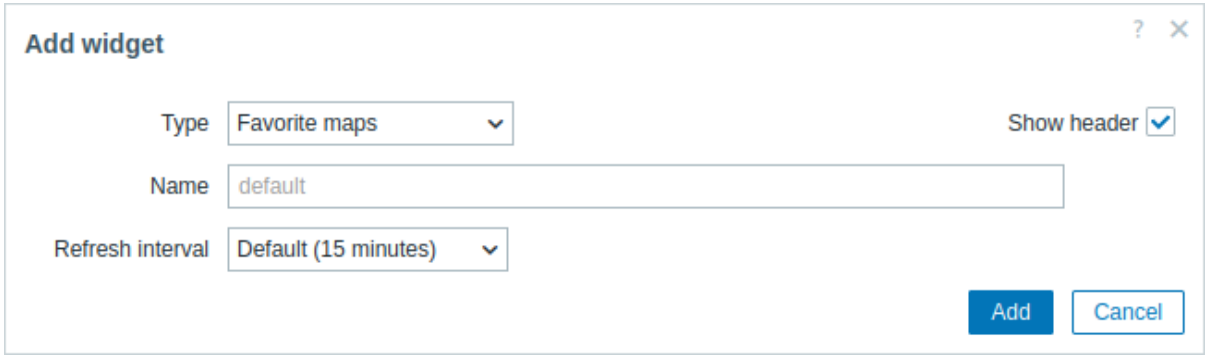
Facendo clic su x la mappa viene rimossa dai preferiti.

Le mappe vengono contrassegnate come preferite facendo clic sul pulsante  *Aggiungi ai preferiti* durante la visualizzazione delle mappe in *Monitoring* > *Maps*.



Configurazione

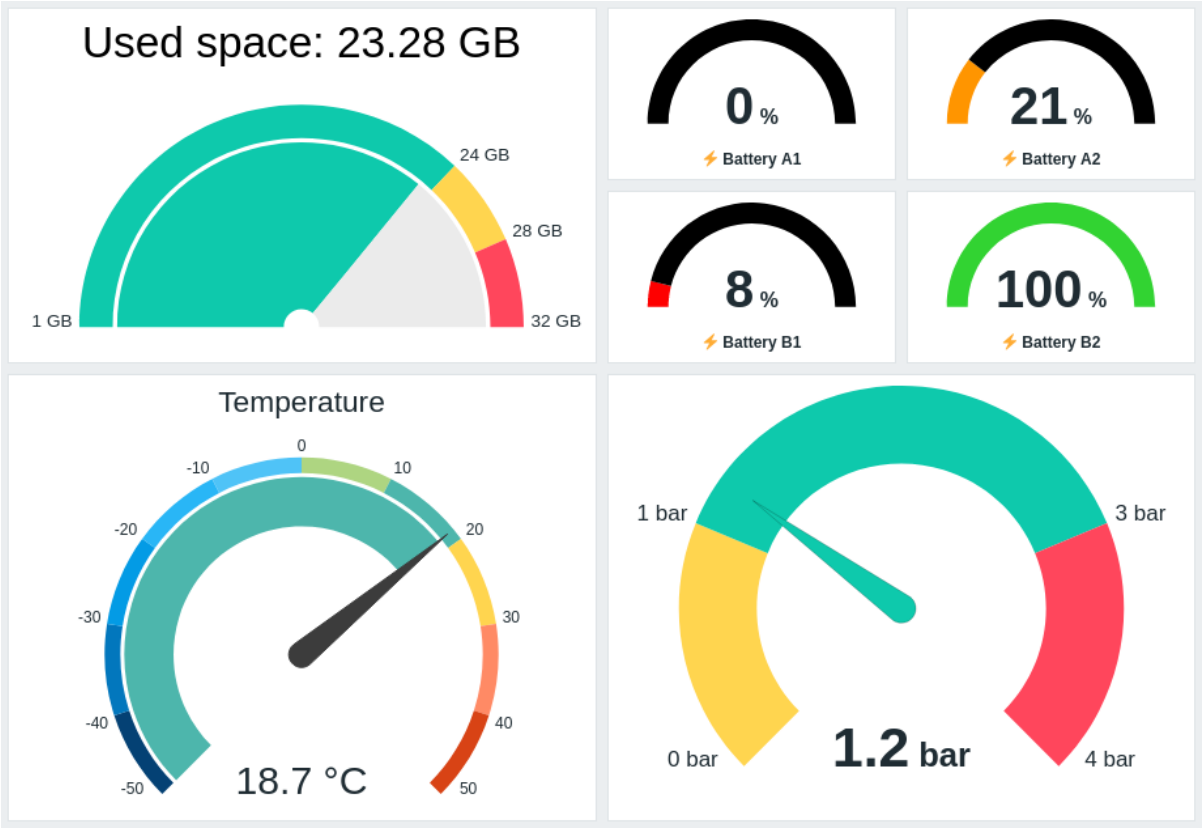
Tutti i parametri di configurazione per questo widget sono comuni a tutti i widget.



Indicatore

Panoramica

Il widget *Gauge* visualizza il valore di un singolo item numerico come indicatore. È utile per tenere sotto controllo metriche importanti, visualizzare le soglie e rilevare cambiamenti improvvisi.



È possibile configurare il widget per visualizzare quanto segue:

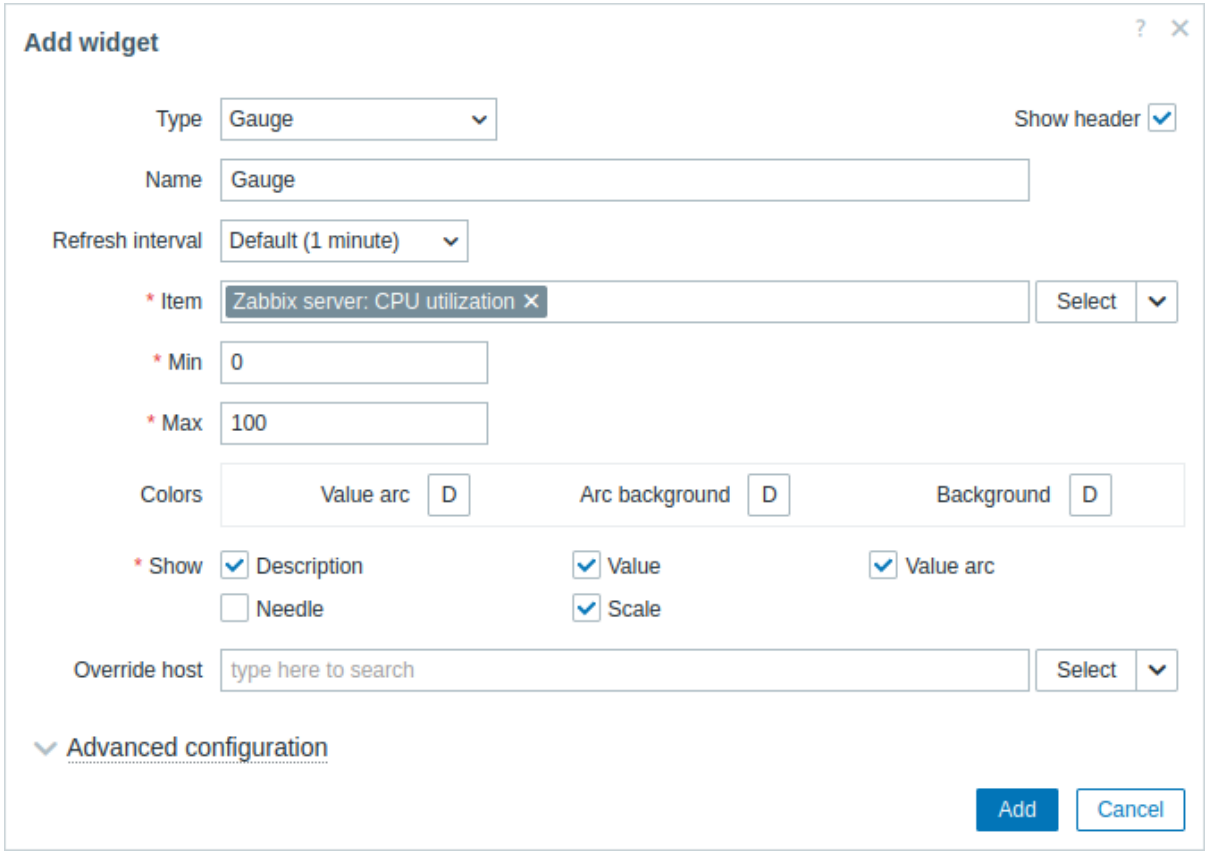
- Descrizione dell'item (Spazio utilizzato: {ITEM.LASTVALUE}, Temperatura)
- Valore e unità dell'item (21 %, 18.7 °C, 1.2 bar)
- Scala dell'indicatore (1GB/24GB/28GB/32GB, -50/-40/-30/ecc.)
- Arco dell'indicatore (arco del valore dell'indicatore e arco delle soglie dell'indicatore)
- Lancetta dell'indicatore

Facendo clic sul widget si apre un grafico semplice per l'item.

Le informazioni visualizzate nel widget *Gauge* possono essere scaricate come immagine PNG selezionando l'opzione *Scarica immagine* nel menu *Azioni* del widget ⋮, visualizzato quando si passa il mouse sul widget. A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella `Downloads` oppure si apre una finestra di dialogo *Salva* che consente di scegliere la posizione.

Configurazione

Per configurare, selezionare *Gauge* come tipo:



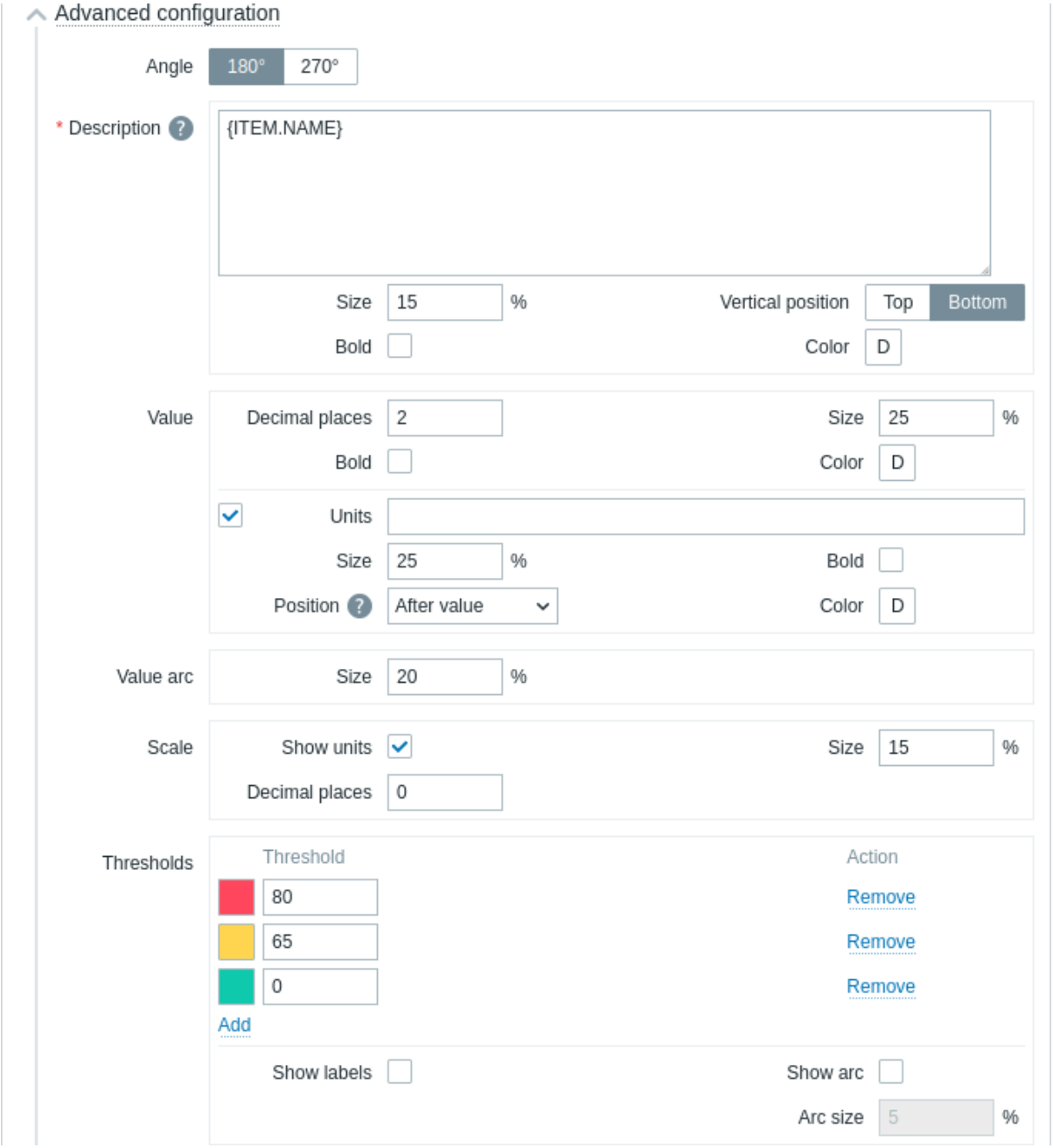
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Item	Selezionare l'item. In alternativa, selezionare un widget compatibile come origine dati per gli item. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un item verrà mostrato un menu a discesa con gli item corrispondenti. Si noti che è possibile selezionare solo item che restituiscono dati numerici (eccetto i dati binari).
Min	Inserire il valore minimo del gauge. I suffissi (ad esempio, "1d", "2w", "4K", "8G") sono supportati. Le mappature dei valori sono supportate.
Max	Inserire il valore massimo del gauge. I suffissi (ad esempio, "1d", "2w", "4K", "8G") sono supportati. Le mappature dei valori sono supportate.
Colors	Selezionare il colore dal selettore colori: **Arco del valore** - selezionare il colore dell'arco del valore del gauge; **Sfondo dell'arco** - selezionare il colore di sfondo dell'arco del valore del gauge e dell'arco delle soglie del gauge; **Sfondo** - selezionare il colore di sfondo del widget. "D" indica il colore predefinito, che dipende dal tema del frontend. Se sono impostate le *Thresholds*, il colore predefinito di **Arco del valore** dipende dal colore della soglia. Per tornare al colore predefinito, fare clic sul pulsante *Use default* nel selettore colori.
Show	Selezionare la casella di controllo per visualizzare il rispettivo elemento del gauge: descrizione, valore, arco del valore, ago, scala (il valore minimo e massimo del gauge all'inizio e alla fine dell'arco del gauge). Deselezionarla per nasconderlo. Deve essere selezionato almeno un elemento. Si noti che l'ago e la scala del gauge possono essere visualizzati se è visualizzato l'arco del valore del gauge o l'arco delle soglie del gauge (vedere le opzioni di configurazione avanzata). Si noti inoltre che, se l'ago del gauge è visualizzato, il valore viene posizionato sotto l'ago; se l'ago è nascosto, il valore viene allineato con la parte inferiore dell'arco del gauge.
Override host	Selezionare un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.

Advanced configuration	Fare clic sull'intestazione *Advanced configuration* per visualizzare le opzioni di configurazione avanzata. Qui sarà inoltre possibile regolare gli elementi del gauge selezionati nel campo *Show*.

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata*:



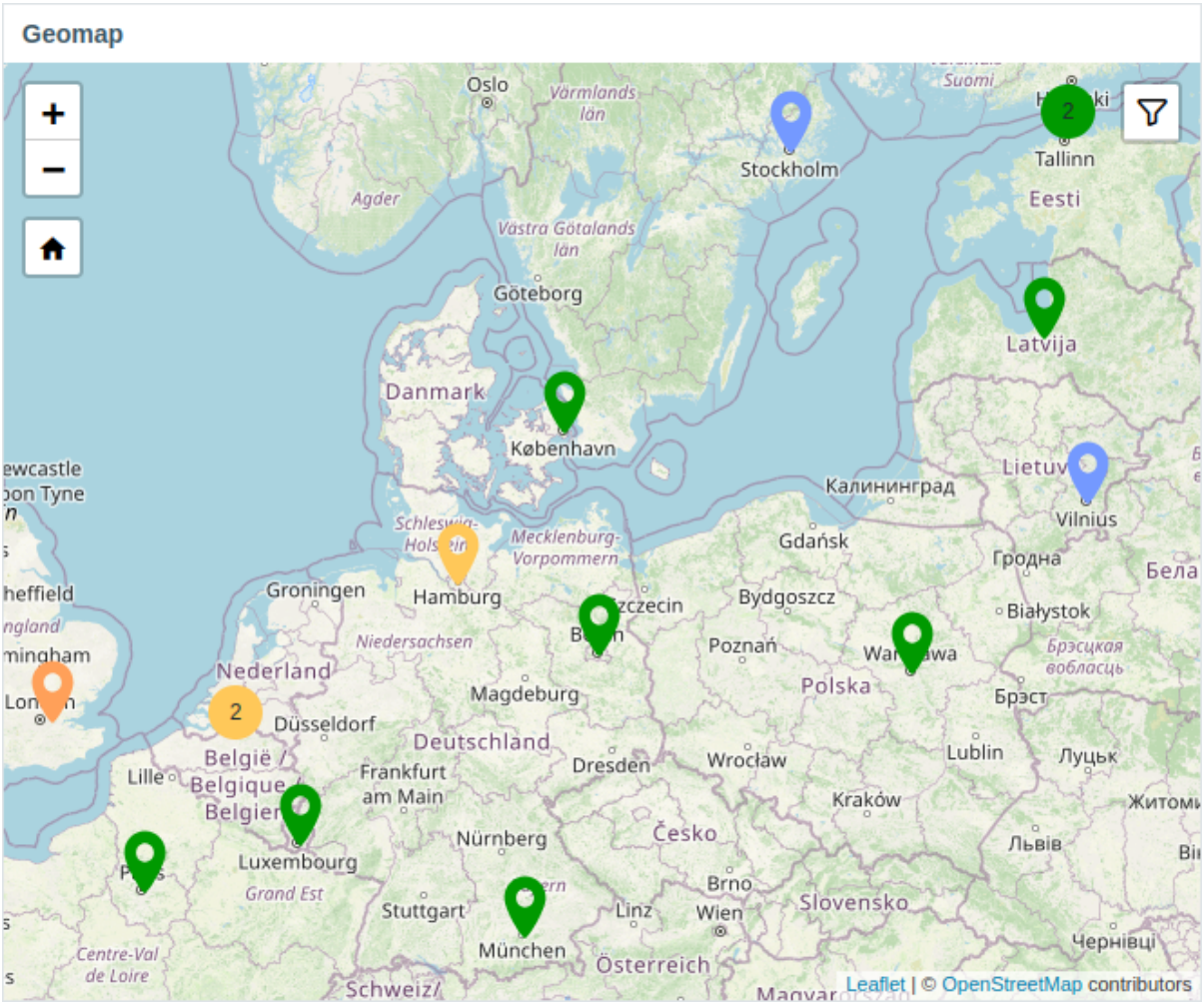
Angolo	Selezionare l'angolo dell'indicatore (180° o 270°).
Descrizione	
Descrizione	Inserire la descrizione dell'item. Questa descrizione può sovrascrivere il nome predefinito dell'item. Sono supportate descrizioni su più righe. È possibile una combinazione di testo e macro supportate. Sono supportate {HOST.*}, {ITEM.*}, {INVENTORY.*} e le macro utente.
Dimensione	Inserire l'altezza della dimensione del carattere per la descrizione dell'item (in percentuale, relativa all'altezza totale del widget).
Posizione verticale	Selezionare la posizione verticale della descrizione dell'item (in alto o in basso, rispetto all'arco dell'indicatore).
Grassetto	Selezionare la casella per visualizzare in grassetto la descrizione dell'item.

Colore	Selezionare il colore della descrizione dell'item dal selettore colori. "D" indica il colore predefinito, che dipende dal tema del frontend. Per tornare al colore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Valore	
Posizioni decimali	Inserire il numero di posizioni decimali da visualizzare con il valore. Questa opzione influisce solo sugli item che restituiscono dati numeric (float).
Dimensione	Inserire l'altezza della dimensione del carattere per il valore (in percentuale, relativa all'altezza dell'arco dell'indicatore).
Grassetto	Selezionare la casella per visualizzare il valore in grassetto.
Colore	Selezionare il colore del valore dal selettore colori. "D" indica il colore predefinito, che dipende dal tema del frontend. Per tornare al colore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Unità	
Unità	Selezionare la casella per visualizzare le unità insieme al valore dell'item. Se si inserisce un nome di unità, questo sovrascriverà le unità impostate nella configurazione dell'item.
Dimensione	Inserire l'altezza della dimensione del carattere per le unità dell'item (in percentuale, relativa all'altezza dell'arco dell'indicatore).
Grassetto	Selezionare la casella per visualizzare in grassetto le unità dell'item.
Posizione	Selezionare la posizione delle unità dell'item (sopra, sotto, prima o dopo, rispetto al valore dell'item). Questa opzione viene ignorata per le seguenti unità relative al tempo: unixtime, uptime, s.
Colore	Selezionare il colore delle unità dell'item dal selettore colori. "D" indica il colore predefinito, che dipende dal tema del frontend. Per tornare al colore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Arco del valore	
Dimensione arco	Inserire l'altezza della dimensione dell'arco del valore dell'indicatore (in percentuale, relativa al raggio dell'arco dell'indicatore).
Lancetta	
Colore	Selezionare il colore della lancetta dell'indicatore dal selettore colori. "D" indica il colore predefinito, che dipende dal tema del frontend. Se sono impostate le *Soglie*, il colore predefinito della lancetta dipende dal colore della soglia. Per tornare al colore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Scala	
Mostra unità	Selezionare la casella per visualizzare le unità insieme al valore minimo e massimo dell'indicatore.
Dimensione	Inserire l'altezza della dimensione del carattere per il valore minimo e massimo dell'indicatore (in percentuale, relativa all'altezza dell'arco dell'indicatore).
Posizioni decimali	Inserire il numero di posizioni decimali da visualizzare con il valore minimo e massimo dell'indicatore. Questa opzione influisce solo sugli item che restituiscono dati numeric (float).
Soglie	
Soglie	Fare clic su *Aggiungi* per aggiungere una soglia, selezionare un colore per la soglia dal selettore colori e specificare un valore numerico. L'elenco delle soglie verrà ordinato in ordine crescente al momento del salvataggio. Si noti che i colori configurati come soglie verranno visualizzati correttamente solo per gli item numerici. Sono supportati i suffissi (ad esempio, "1d", "2w", "4K", "8G"). Sono supportate le mappature dei valori.
Mostra etichette	Selezionare la casella per visualizzare i valori di soglia come etichette sulla scala dell'indicatore.
Mostra arco	Selezionare la casella per visualizzare l'arco delle soglie dell'indicatore.
Dimensione arco	Inserire l'altezza della dimensione dell'arco delle soglie dell'indicatore (in percentuale, relativa al raggio dell'arco dell'indicatore).

Geomap

Panoramica

Il widget *Geomap* visualizza gli host come marcatori su una mappa interattiva basata su Leaflet, con i colori dei marcatori che indicano il problema più grave di ciascun host.

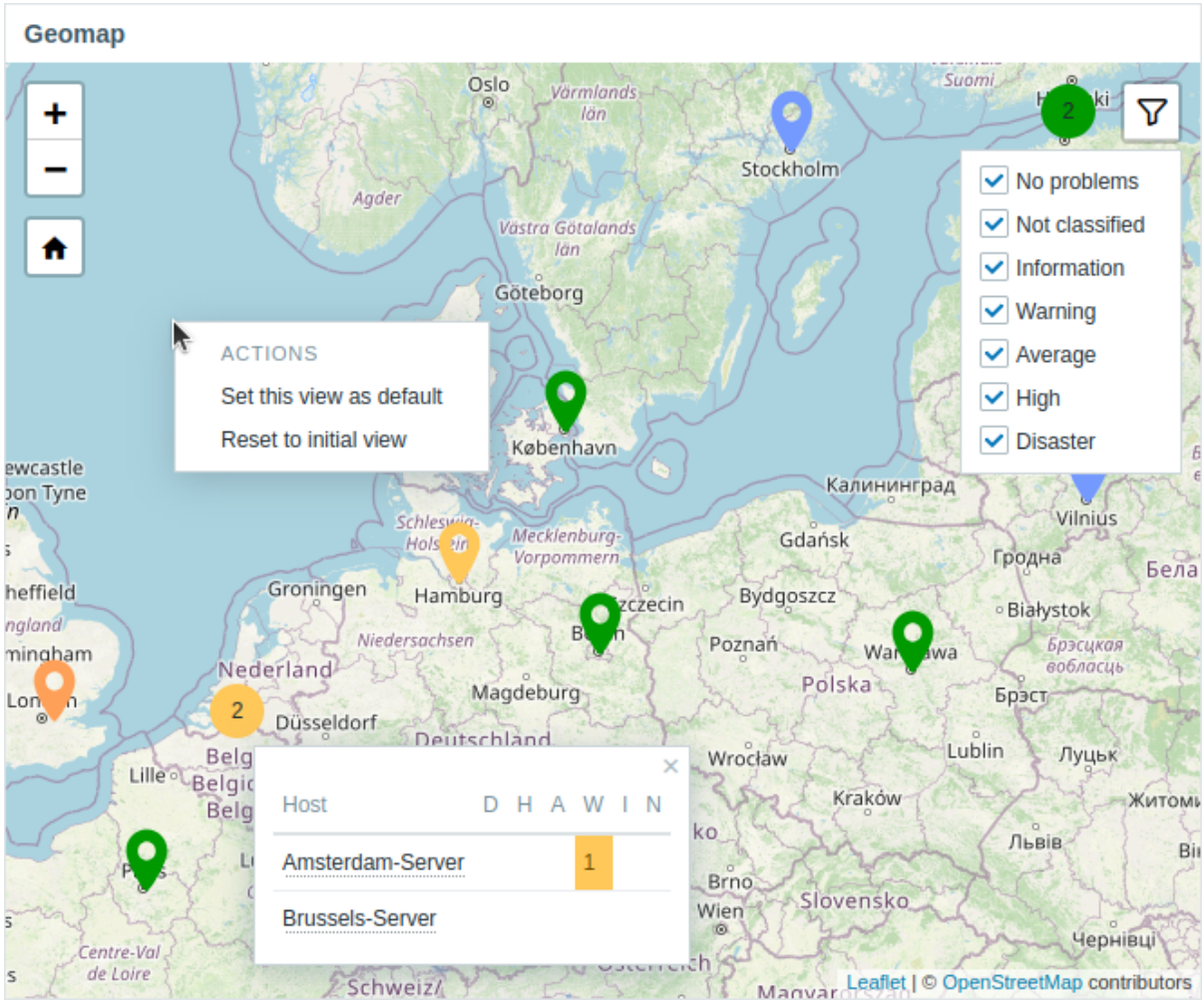


Note:
Zabbix offre diversi provider di tile della mappa predefiniti e consente di aggiungerne di personalizzati in *Administration* > *General* > *Geographical maps*.

Per impostazione predefinita, il widget visualizza tutti gli host abilitati che hanno coordinate valide (latitudine da -90 a 90, longitudine da -180 a 180) nel loro inventory.

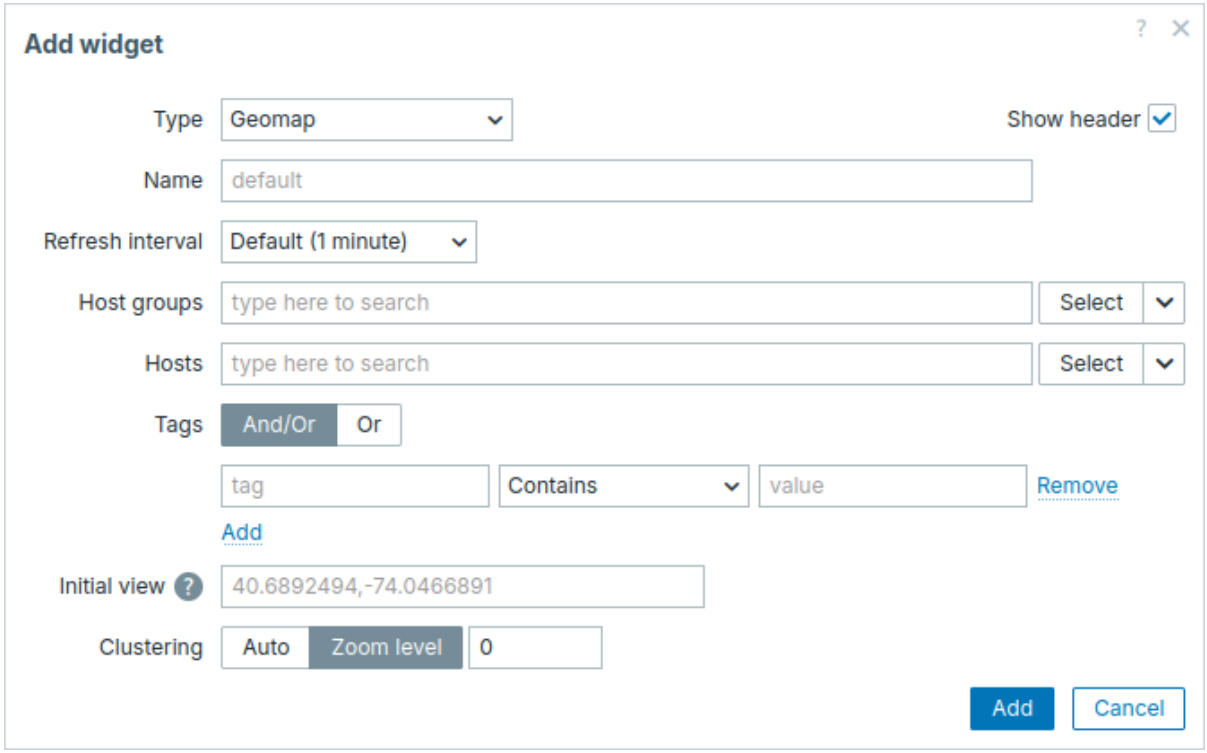
Utilizzare le seguenti opzioni per interagire con il widget:

- Facendo clic su un marcatore vengono mostrati il nome dell'host e i problemi non risolti; facendo clic sul nome dell'host si apre il menu host.
- Filtrare gli host per gravità del problema utilizzando l'icona del filtro nell'angolo superiore destro.
- Fare clic con il pulsante destro del mouse sulla mappa per impostare o reimpostare la vista predefinita; utilizzare l'icona Home per tornare alla vista predefinita.



Configurazione

Per aggiungere il widget, selezionare *Geomap* come tipo.



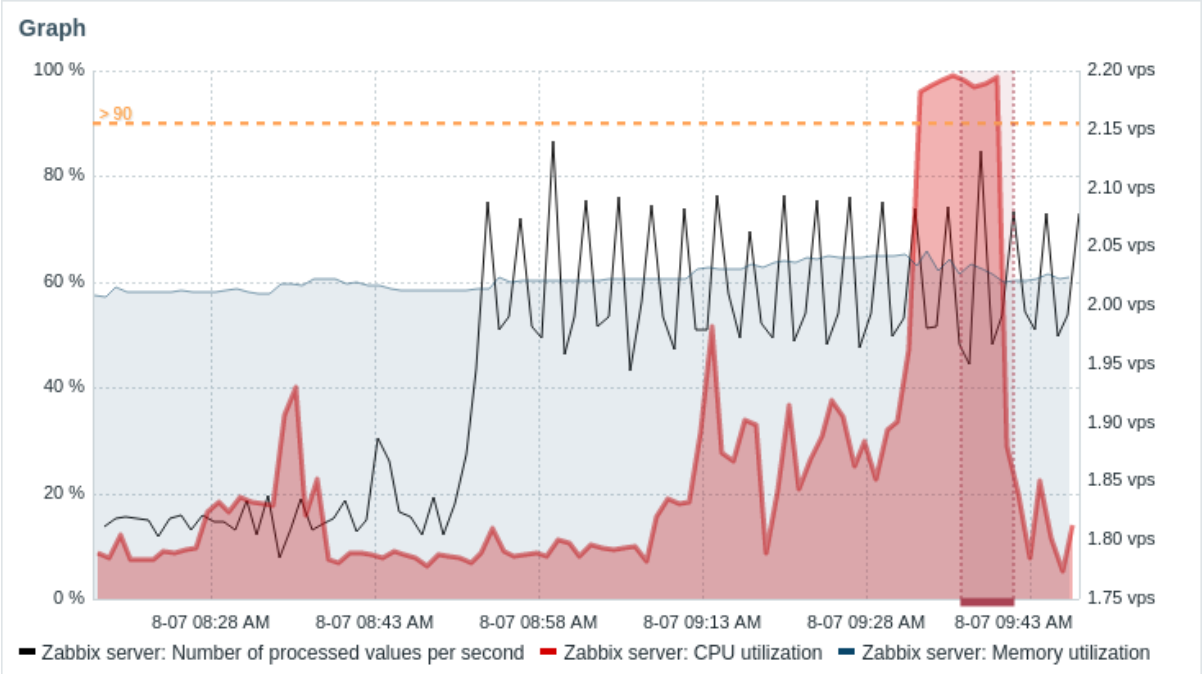
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Host groups	Selezionare i gruppi di host da visualizzare sulla mappa. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un elenco a discesa con i gruppi corrispondenti. Se non viene selezionato nulla nei campi *Host groups* e *Hosts*, verranno visualizzati tutti gli host con coordinate valide. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Hosts	Selezionare gli host da visualizzare sulla mappa. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà mostrato un elenco a discesa con gli host corrispondenti. Se non viene selezionato nulla nei campi *Host groups* e *Hosts*, verranno visualizzati tutti gli host con coordinate valide. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Tags	Specificare i tag per limitare il numero di host visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ogni condizione: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag specificati i cui valori contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag specificati i cui valori contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Sono disponibili due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una sola condizione. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Initial view	Coordinate del centro separate da virgole e un livello di zoom facoltativo da visualizzare al caricamento iniziale del widget, nel formato `<latitude>,<longitude>,<zoom>` Se viene specificato lo zoom iniziale, il widget Geomap viene caricato al livello di zoom indicato. In caso contrario, lo zoom iniziale viene calcolato come metà del max zoom per il provider di tile specifico. La vista iniziale viene ignorata se è impostata la vista predefinita (vedere sotto). Esempi: 40.6892494,-74.0466891 40.6892494,-122.0466891
Clustering	Specificare come i marcatori degli host vicini vengono combinati in un unico marcatore con un conteggio: **Auto** - i marcatori vengono combinati automaticamente; **Zoom level** - i marcatori vengono combinati in base al livello di zoom specificato (0 - zoom minimo, 30 - zoom massimo). Ad esempio, se il parametro è impostato su 0, i marcatori non vengono combinati a nessun livello di zoom; se è impostato su 15, i marcatori vengono combinati a partire dallo zoom minimo fino al livello di zoom 15; se è impostato su 30, i marcatori vengono combinati a tutti i livelli di zoom. Si noti che il livello massimo di zoom è determinato dal provider di tile della mappa.

Grafico

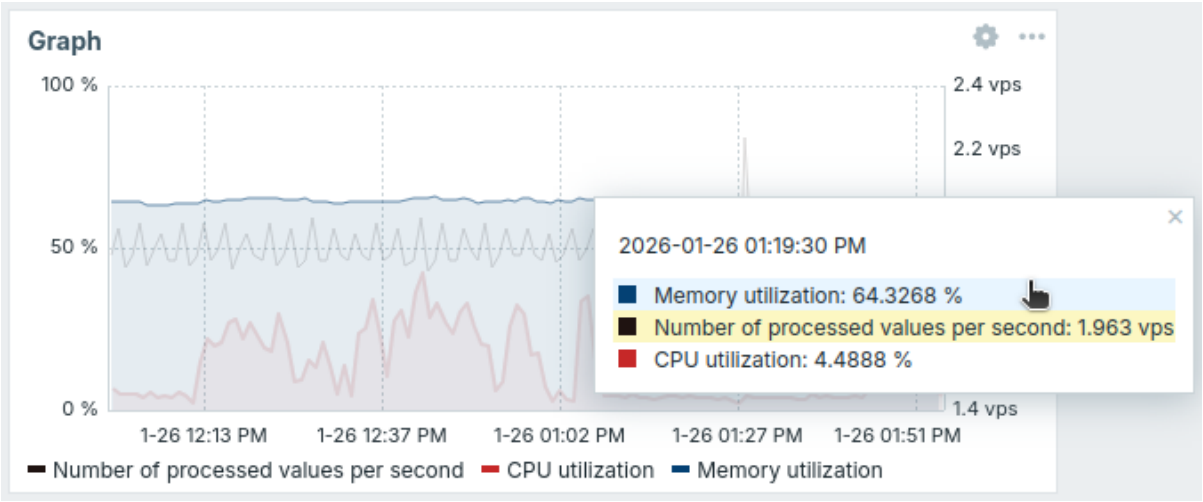
Panoramica

Il widget *Graph* visualizza i dati numerici degli item sotto forma di grafico vettoriale. Può aiutare a monitorare le tendenze, individuare i problemi e confrontare i valori nel tempo e tra gli host.

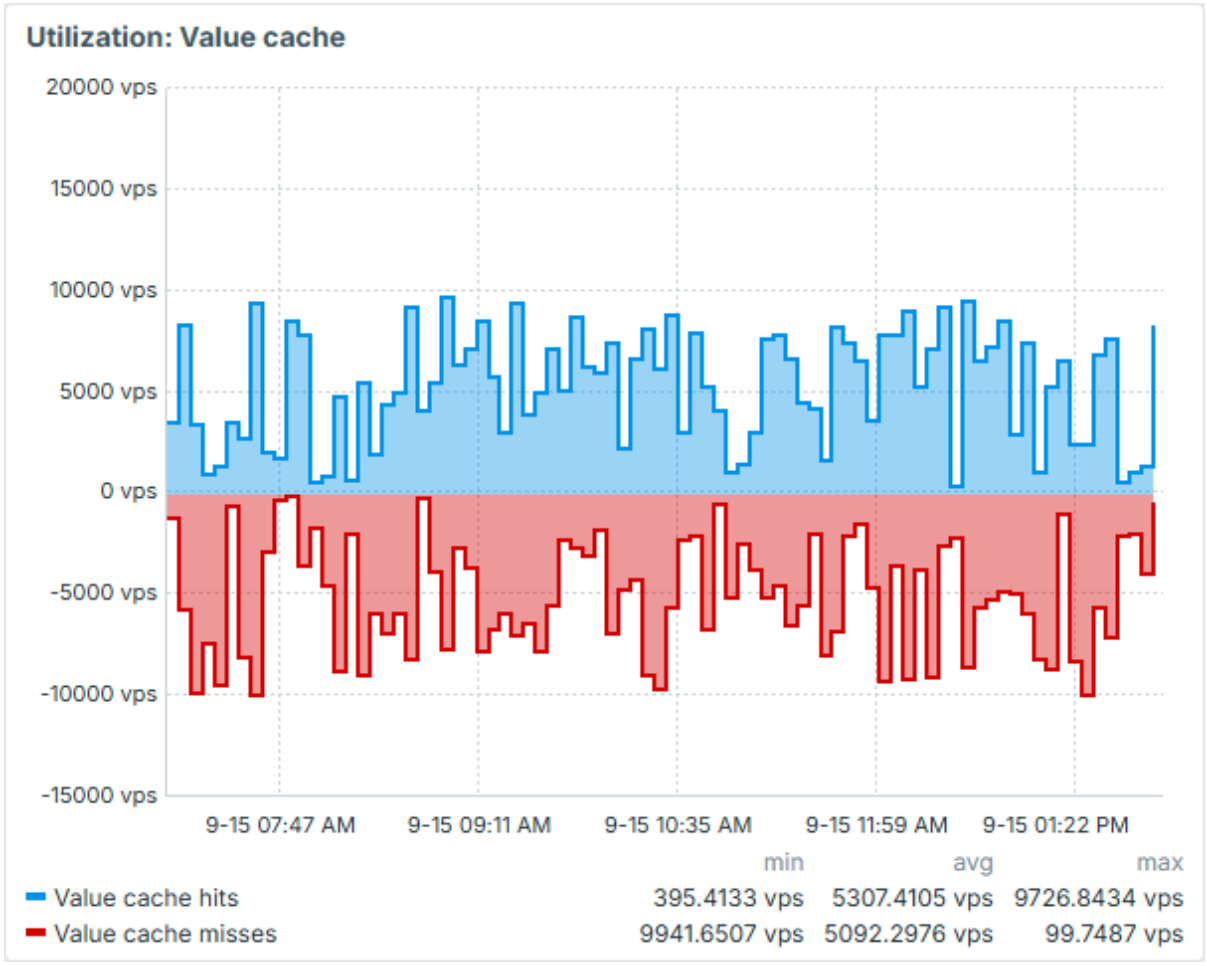


Passando il mouse sul grafico, viene visualizzato un tooltip con un elenco di item ordinati per valore in ordine decrescente.

Passando il mouse su un item nel tooltip, il relativo grafico viene evidenziato (gli altri vengono attenuati); selezionando l'item, i relativi dati vengono trasmessi agli altri widget.



Il grafico può essere visualizzato con i valori dell'asse Y invertiti; ciò influisce solo sulla rappresentazione visiva e non modifica i valori originali.

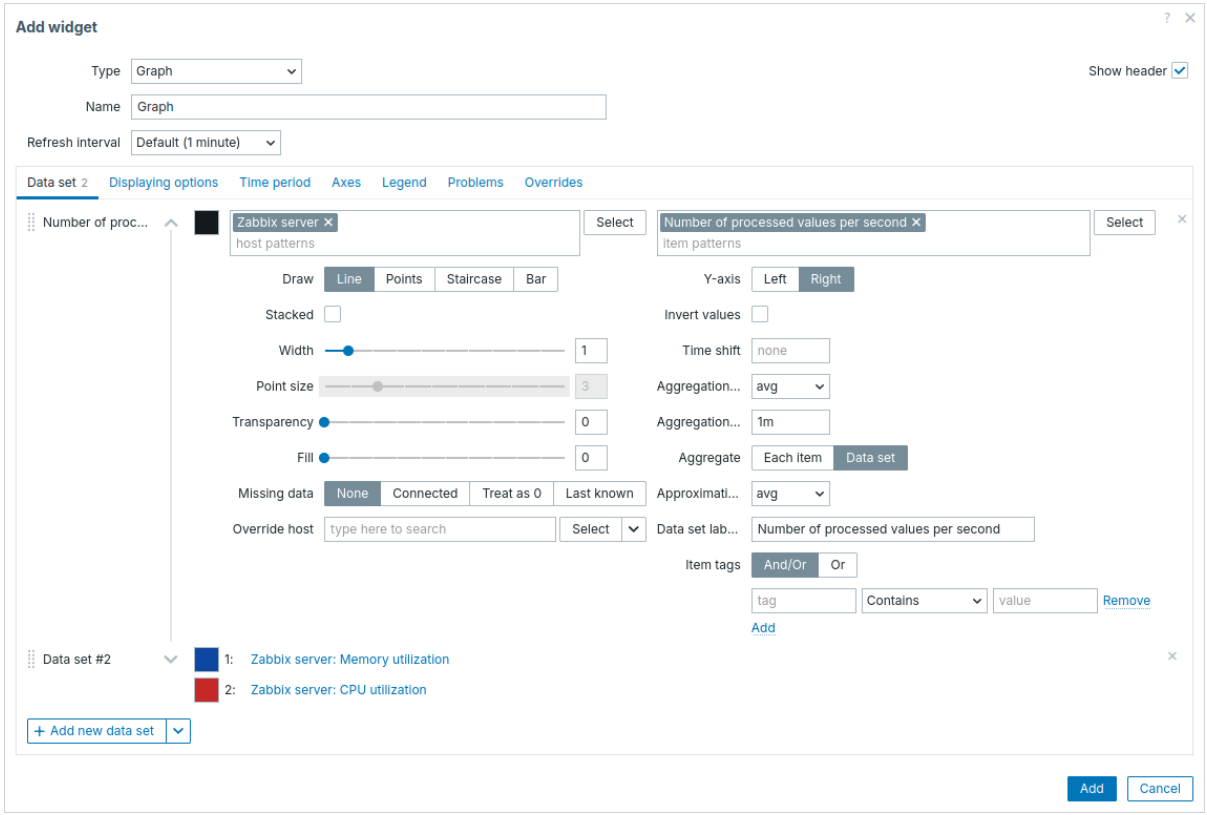


È disponibile anche il widget legacy Graph (classic).

Le informazioni visualizzate nel widget *Graph* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu *Actions* del widget ⋮, visualizzato quando si passa il mouse sul widget. A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella `Downloads` oppure viene aperta una finestra di dialogo *Save*, che consente di scegliere la posizione.

Configurazione

Per configurare, selezionare *Graph* come tipo:



Set di dati

La scheda **Set di dati** consente di selezionare i dati per il grafico aggiungendo set di dati. È possibile aggiungere due tipi di set di dati:

- *Modelli di item* - vengono visualizzati i dati degli item corrispondenti. È possibile scegliere un singolo colore di base oppure selezionare una riga della tavolozza per assegnare colori distinti a ogni item corrispondente.
- *Elenco di item* - vengono visualizzati i dati degli item selezionati. È possibile scegliere individualmente il colore di ogni item dal selettore.

Per impostazione predefinita, viene aggiunto un set di dati *Modelli di item*.

Set di dati	Per il set di dati **Modelli di item**: Selezionare o immettere i modelli di host e item; nel grafico verranno visualizzati i dati degli item che corrispondono a questi modelli; è possibile visualizzare fino a 50 item. Per la selezione è possibile utilizzare modelli con caratteri jolly (ad esempio, ∗ restituirà i risultati che corrispondono a zero o più caratteri). Per specificare un modello con caratteri jolly, immettere manualmente la stringa e premere *Invio*. Il simbolo jolly viene sempre interpretato, pertanto non è possibile aggiungere singolarmente, ad esempio, un item denominato *item** se sono presenti altri item corrispondenti (ad esempio, *item2*, *item3*). La specifica dei modelli di host e item è obbligatoria per i set di dati "Modelli di item". Vedere anche: Dettagli della configurazione del set di dati. Per il set di dati **Elenco di item**: Selezionare gli item per il grafico facendo clic sul pulsante *Aggiungi item*. È inoltre possibile selezionare widget compatibili come origine dati per gli item facendo clic sul pulsante *Aggiungi widget*. La specifica di item o widget è obbligatoria per i set di dati "Elenco di item". Vedere anche: Dettagli della configurazione del set di dati. Tenere presente che sono consentiti solo tipi di item numerici. Quando si configura il widget in una dashboard di template, il parametro per specificare i modelli di host non è disponibile e il parametro per specificare un elenco di item consente solo di selezionare gli item configurati nel template.

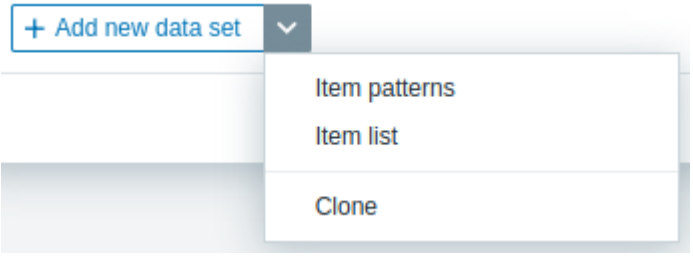
Disegno	Scegliere il tipo di disegno della metrica. Tipi di disegno disponibili: *Linea* (impostato per impostazione predefinita), *Punti*, *Gradini* e *Barre*. Tenere presente che, se nel grafico a linea o a gradini è presente un solo punto dati, questo viene disegnato come un punto indipendentemente dal tipo di disegno. La dimensione del punto viene calcolata in base alla larghezza della linea, ma non può essere inferiore a 3 pixel, anche se la larghezza della linea è minore.
Impilato	Selezionare la casella di controllo per visualizzare i dati impilati (con aree riempite). Questa opzione è disabilitata quando è selezionato il tipo di disegno *Punti*.
Larghezza	Impostare la larghezza della linea. Questa opzione è disponibile quando è selezionato il tipo di disegno *Linea* o *Gradini*.
Dimensione del punto	Impostare la dimensione del punto. Questa opzione è disponibile quando è selezionato il tipo di disegno *Punti*.
Trasparenza	Impostare il livello di trasparenza.
Riempimento	Impostare il livello di riempimento. Questa opzione è disponibile quando è selezionato il tipo di disegno *Linea* o *Gradini*.
Dati mancanti	Selezionare l'opzione per visualizzare i dati mancanti: **Nessuno** - il vuoto viene lasciato tale; **Connessi** - vengono collegati due valori di bordo; **Considera come 0** - i dati mancanti vengono visualizzati come valori 0; **Ultimo valore noto** - i dati mancanti vengono visualizzati con lo stesso valore dell'ultimo valore noto; non applicabile ai tipi di disegno *Punti* e *Barre*.
Sovrascrivi host	Selezionare un widget compatibile o il selettore di host della dashboard come origine dati per gli host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Asse Y	Selezionare il lato del grafico in cui visualizzare l'asse Y.
Inverti valori	Selezionare la casella di controllo per moltiplicare tutti i valori dell'asse Y per -1 prima di visualizzarli nel grafico. Questa opzione influisce solo sulla rappresentazione visiva; i dati originali rimangono invariati. Questa opzione non influisce sulla descrizione comando, sulla legenda o sulla linea del percentile. Se per l'asse è selezionata l'opzione *Trigger semplici*, le soglie dei trigger verranno visualizzate invertite.
Spostamento temporale	Specificare lo spostamento temporale, se necessario. In questo campo è possibile utilizzare i suffissi temporali. Sono consentiti valori negativi.
Funzione di aggregazione	Specificare la funzione di aggregazione da utilizzare per ogni item o per un intero set di dati nell'ambito dell'*Intervallo di aggregazione* selezionato: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **sum** - visualizza la somma dei valori; **count** - visualizza il numero di valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore. Se si seleziona **non utilizzata**, vengono visualizzati tutti i valori (nessuna aggregazione). Per *min*, *max*, *avg* e *sum* è possibile visualizzare solo dati numerici. Per *count*, i dati non numerici vengono convertiti in numerici. Vedere anche: Aggregazione nei grafici.
Intervallo di aggregazione	Specificare l'intervallo per l'aggregazione dei valori. In questo campo è possibile utilizzare i suffissi temporali. Un valore numerico senza suffisso viene considerato espresso in secondi. Tenere presente che, se il widget è configurato per visualizzare dati storici basati sui trend (*Selezione dei dati storici* impostata su *Trend* o *Automatico*), è consigliabile utilizzare un intervallo di aggregazione multiplo di 1 ora (ad esempio, 3600, 60m, 1h, 3h e così via). I trend memorizzano valori aggregati su base oraria, pertanto l'utilizzo di un intervallo di aggregazione non multiplo di 1 ora (ad esempio, 100s, 7min, 15min, 90min e così via) può produrre risultati difficili da interpretare.

Aggrega	Specificare se aggregare: **Ogni item** - ogni item nel set di dati viene aggregato e visualizzato separatamente; **Set di dati** - tutti gli item del set di dati vengono aggregati e visualizzati come un unico valore.
Approssimazione	Specificare quale valore visualizzare quando esiste più di un valore per ogni pixel verticale del grafico: **all** - visualizza i valori minimo, massimo e medio; **min** - visualizza il valore minimo; **max** - visualizza il valore massimo; **avg** - visualizza il valore medio. Questa impostazione è utile quando si visualizza un grafico per un periodo di tempo esteso con un intervallo di aggiornamento frequente, ad esempio un anno di valori raccolti ogni 10 minuti.
Etichetta del set di dati	Specificare l'etichetta del set di dati visualizzata nella configurazione del *Set di dati* del grafico e nella *Legenda* del grafico (per i set di dati aggregati). Tutti i set di dati vengono numerati, inclusi quelli con un'*Etichetta del set di dati* specificata. Se non viene specificata alcuna etichetta, il set di dati viene etichettato automaticamente in base al relativo numero (ad esempio, "Set di dati n. 2", "Set di dati n. 3" e così via). La numerazione dei set di dati viene ricalcolata dopo averli riordinati o trascinati. Le etichette dei set di dati troppo lunghe vengono abbreviate per adattarsi allo spazio disponibile (ad esempio, "Numero di proc....").
Tag degli item	Specificare i tag per filtrare gli item visualizzati nel widget. È possibile includere o escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Esiste** - include i nomi dei tag specificati; **Uguale** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contiene** - include i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Non esiste** - esclude i nomi dei tag specificati; **Non uguale** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Non contiene** - esclude i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - devono essere soddisfatte tutte le condizioni; le condizioni con lo stesso nome di tag vengono raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione.

Dettagli della configurazione del set di dati

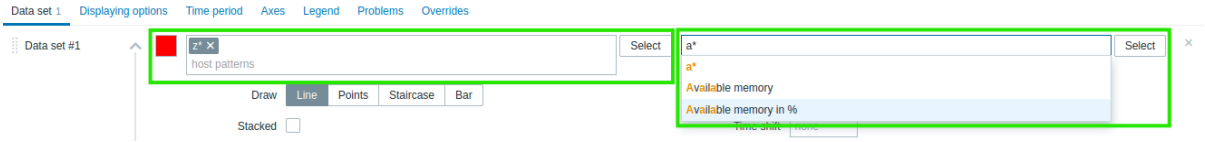
I set di dati esistenti vengono visualizzati in un elenco. È possibile:

- Fare clic sull'icona di spostamento e trascinare un set di dati in una nuova posizione nell'elenco.
- Fare clic sull'icona di espansione per espandere i dettagli del set di dati. Quando è espansa, questa icona diventa un'icona di compressione.
- Fare clic sull'icona del colore per aprire il selettore. È possibile inserire un codice esadecimale, scegliere un campione a tinta unita oppure passare alla scheda *Palette* e selezionare una riga di colori predefiniti. Il colore scelto viene applicato direttamente ai set di dati *Item list* oppure utilizzato come base per generare sfumature nei set di dati *Item patterns*. Utilizzare Tab per spostarsi tra i controlli della finestra di dialogo, i tasti freccia per navigare tra i campioni o le righe della palette, Invio per selezionare ed Esc per annullare.
- Fare clic sul pulsante *Add new data set* per aggiungere un set di dati vuoto che consente di selezionare host e pattern di item. Facendo clic sull'icona rivolta verso il basso accanto al pulsante *Add new data set*, viene visualizzato un menu a discesa che consente di aggiungere un nuovo set di dati *Item patterns* o *Item list*, oppure di *Clone* il set di dati attualmente aperto. Se tutti i set di dati sono compressi, l'opzione *Clone* non è disponibile.

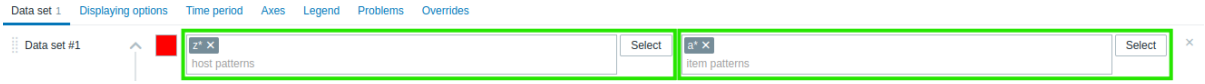


Il set di dati **Pattern degli item** contiene i campi *Pattern degli host* e *Pattern degli item*, entrambi in grado di riconoscere nomi completi o pattern contenenti un simbolo jolly (*). Questa funzionalità consente di selezionare tutti i nomi host e i nomi degli item che contengono il pattern selezionato. Durante la digitazione del nome dell'item o del pattern dell'item nel campo *Pattern degli item*, nell'elenco a discesa vengono visualizzati solo gli item che appartengono ai nomi host selezionati.

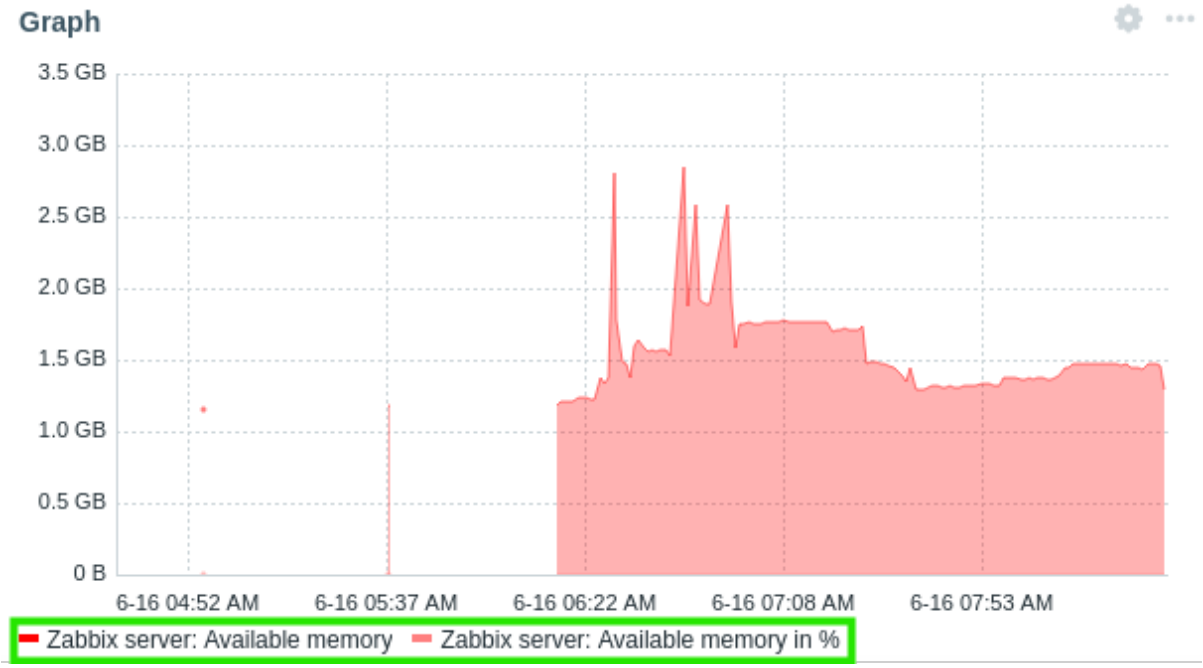
Ad esempio, dopo aver digitato il pattern **z*** nel campo *Pattern degli host*, l'elenco a discesa mostra tutti i nomi host che contengono questo pattern: **z*, Zabbix server, Zabbix proxy**. Dopo aver premuto *Invio*, questo pattern viene accettato e visualizzato come **z***. Allo stesso modo, dopo aver digitato il pattern **a*** nel campo *Pattern degli item*, l'elenco a discesa mostra tutti i nomi degli item che contengono questo pattern: **a*, Available memory, Available memory in %**.



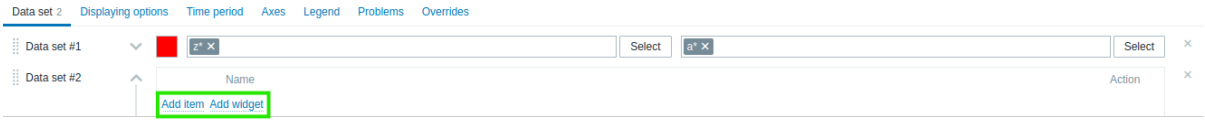
Dopo aver premuto *Invio*, il pattern viene accettato e mostrato come **a***.



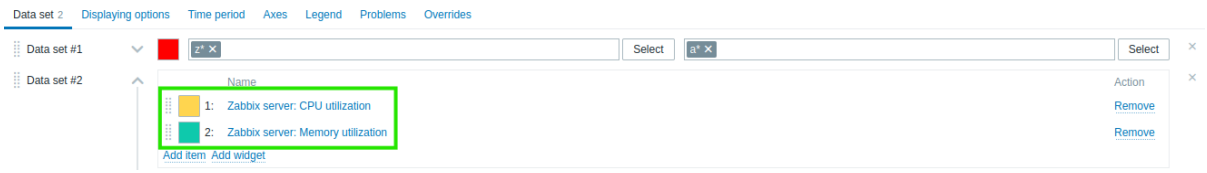
Il grafico mostra quindi tutti gli item che appartengono ai nomi host selezionati.



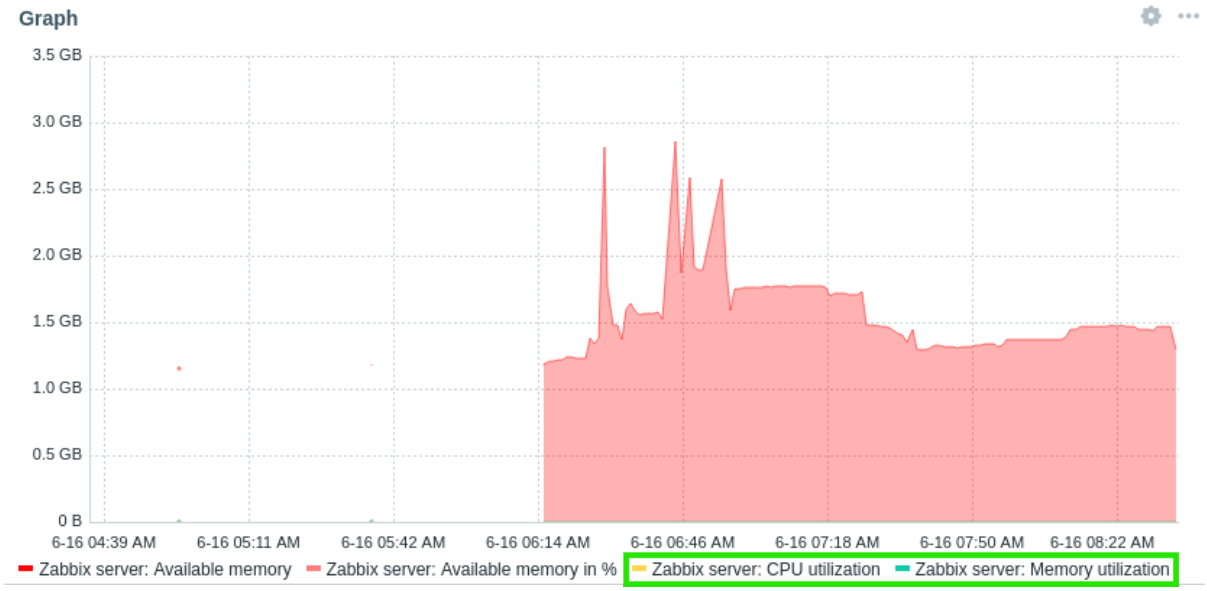
Il set di dati **Elenco item** contiene il pulsante *Aggiungi item*, che consente di aggiungere item da visualizzare nel grafico. È inoltre possibile aggiungere widget compatibili come origine dati per gli item facendo clic sul pulsante *Aggiungi widget*.



Ad esempio, facendo clic sul pulsante *Aggiungi item* si apre una finestra pop-up contenente il parametro *Host*. Dopo aver selezionato un host, tutti i suoi item disponibili per la selezione vengono visualizzati in un elenco.



Dopo aver selezionato uno o più item, questi verranno visualizzati nell'elenco degli item del set di dati e nel grafico.

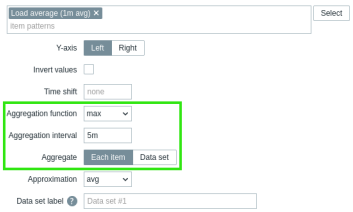


Aggregazione nei grafici

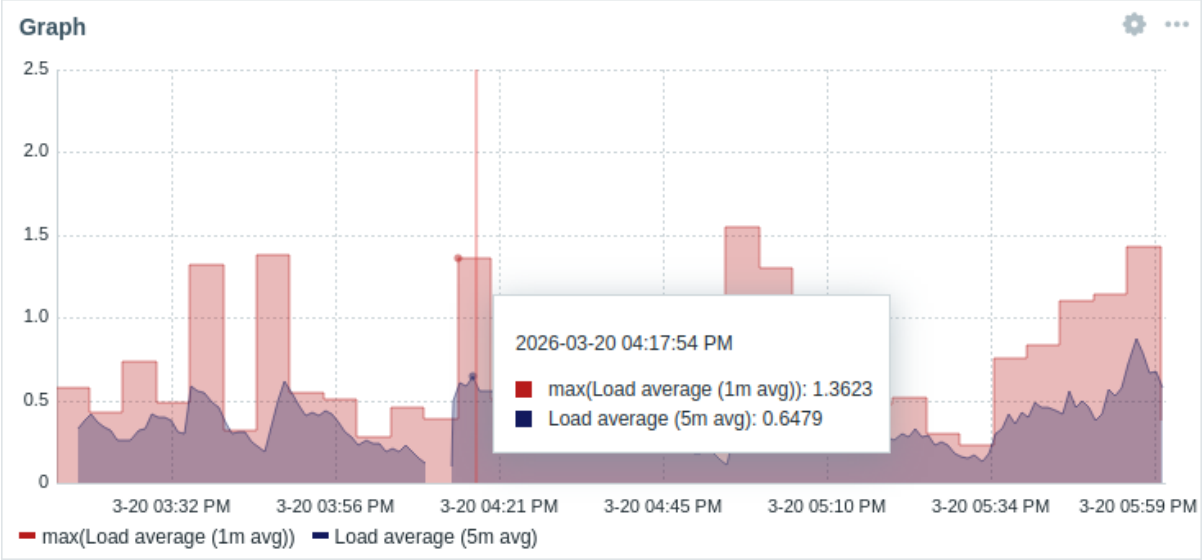
L'aggregazione consente di visualizzare un valore aggregato (minimo, massimo, media, ecc.) per l'intervallo specificato (5 minuti, un'ora, un giorno, ecc.) invece di tutti i singoli valori. All'interno di un set di dati, l'aggregazione può essere applicata a ciascun item separatamente oppure all'intero set di dati.

Alcuni casi d'uso della funzione di aggregazione possono essere i seguenti: - Il numero medio di richieste a un server NGINX al giorno. - Lo spazio su disco minimo tra i cluster alla settimana.

Nell'esempio seguente, la funzione di aggregazione "max" viene applicata all'item con un intervallo di aggregazione di cinque minuti:



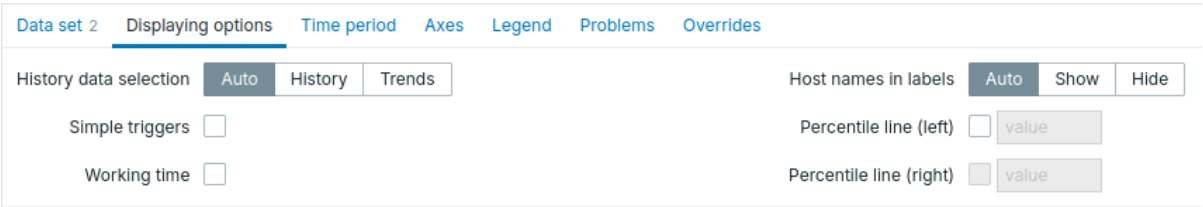
L'esempio mostra il carico massimo della CPU nell'arco di cinque minuti (aggregato a partire da valori medi di un minuto), confrontato con il carico medio della CPU nell'arco di cinque minuti.



Quando *Show aggregation function* è selezionato per la legenda, gli item vengono visualizzati tra parentesi, preceduti dalla funzione di aggregazione nella legenda e nel tooltip. Se per un grafico è specificato *Data set label*, l'etichetta viene visualizzata tra parentesi, preceduta dalla funzione di aggregazione.

Opzioni di visualizzazione

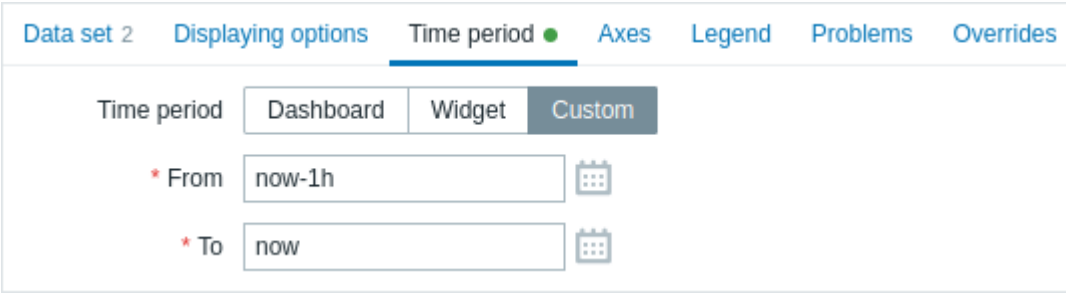
La scheda **Opzioni di visualizzazione** consente di definire la selezione dei dati storici:



Selezione dei dati storici	Imposta l'origine dei dati del grafico: **Automatico** - i dati vengono recuperati in base all'algoritmo del grafico classico (impostazione predefinita); **Cronologia** - dati dalla cronologia; **Trend** - dati dai trend.
Trigger semplici	Selezionare la casella di controllo per mostrare le soglie dei trigger semplici. Le soglie verranno tracciate come linee tratteggiate utilizzando il colore della gravità del trigger. Un trigger semplice è un trigger con una funzione (solo `last`, `max`, `min`, `avg`) per un item nell'espressione. È possibile tracciare un massimo di tre trigger. Il trigger deve rientrare nell'intervallo visualizzato per essere visibile.
Orario di lavoro	Selezionare la casella di controllo per mostrare l'orario di lavoro nel grafico. L'orario di lavoro (giorni lavorativi) viene visualizzato nei grafici con uno sfondo bianco, mentre l'orario non lavorativo viene visualizzato in grigio (con il tema frontend predefinito *Original blue*).
Nomi degli host nelle etichette	Scegliere se visualizzare i nomi degli host nella legenda del grafico e nel tooltip che appare passando il mouse sui dati del grafico: **Automatico** - i nomi degli host vengono visualizzati solo se è presente più di un host tra i set di dati (impostazione predefinita). **Mostra** - i nomi degli host vengono visualizzati. **Nascondi** - i nomi degli host vengono nascosti. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Linea del percentile (sinistra)	Selezionare la casella di controllo e inserire il valore del percentile per mostrare il percentile specificato come linea sull'asse Y sinistro del grafico. Ad esempio, se viene impostato il percentile del 95%, la linea del percentile si troverà al livello al di sotto del quale ricade il 95% dei valori.
Linea del percentile (destra)	Selezionare la casella di controllo e inserire il valore del percentile per mostrare il percentile specificato come linea sull'asse Y destro del grafico. Ad esempio, se viene impostato il percentile del 95%, la linea del percentile si troverà al livello al di sotto del quale ricade il 95% dei valori.

Intervallo di tempo

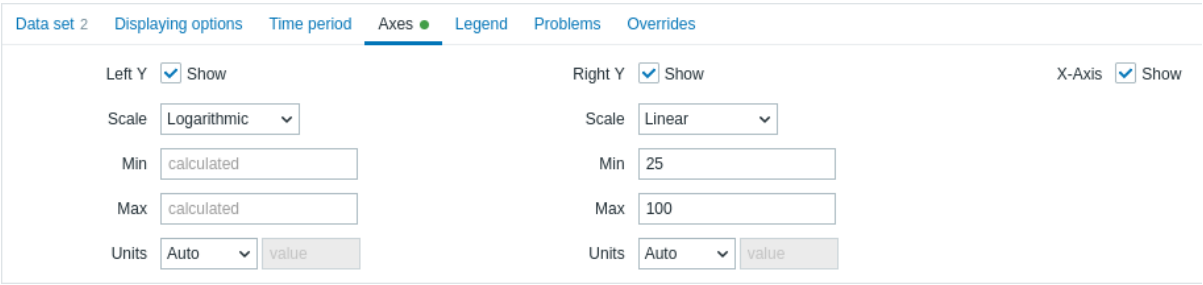
La scheda **Intervallo di tempo** consente di impostare un intervallo di tempo per il quale visualizzare i dati nel grafico:



Intervallo di tempo	Selezionare la sorgente dati per l'intervallo di tempo: **Dashboard** - utilizzare il selettore dell'intervallo di tempo della dashboard; **Widget** - utilizzare un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizzare un intervallo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio, che al passaggio del mouse indica l'ora impostata. Si noti che, indipendentemente dalla configurazione *Intervallo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per l'intervallo di tempo.
Widget	Inserire o selezionare un widget compatibile come sorgente dati per l'intervallo di tempo. Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Widget".
Da	Inserire o selezionare l'inizio dell'intervallo di tempo. È supportata la sintassi dell'ora relativa (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Personalizzato".
A	Inserire o selezionare la fine dell'intervallo di tempo. È supportata la sintassi dell'ora relativa (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Personalizzato".

Assi

La scheda **Assi** consente di personalizzare la visualizzazione degli assi:

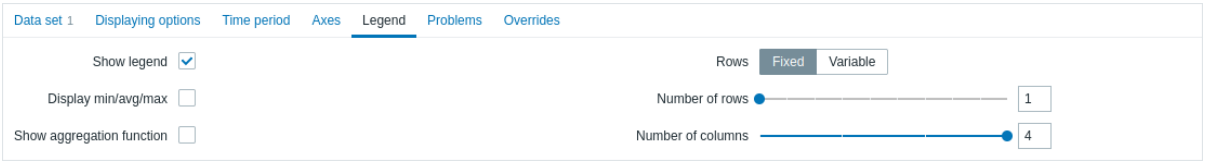


Y sinistro	Selezionare questa casella per rendere visibile l'asse Y sinistro. La casella potrebbe essere disabilitata se l'opzione non è selezionata nella scheda *Data set* o *Overrides*.
Y destro	Selezionare questa casella per rendere visibile l'asse Y destro. La casella potrebbe essere disabilitata se l'opzione non è selezionata nella scheda *Data set* o *Overrides*.
Asse X	Deselezionare questa casella per nascondere l'asse X (selezionata per impostazione predefinita).
Scala	Scegliere la scala per i valori degli assi del grafico dall'elenco a discesa: **Lineare** - i valori dell'asse aumentano di una quantità fissa (ad esempio, 10, 20, 30), adatta ai dati che cambiano in modo costante o coprono un intervallo da piccolo a moderato; **Logaritmica** - i valori dell'asse aumentano in modo esponenziale (ad esempio, 10, 100, 1000), adatta ai dati che cambiano rapidamente o coprono un intervallo ampio.
Min	Impostare il valore minimo dell'asse corrispondente. Viene specificato il valore minimo dell'intervallo visibile dell'asse Y.
Max	Impostare il valore massimo dell'asse corrispondente. Viene specificato il valore massimo dell'intervallo visibile dell'asse Y.

Unità	Scegliere l'unità per i valori degli assi del grafico dall'elenco a discesa: **Automatico** - i valori degli assi vengono visualizzati utilizzando l'unità del primo item nel set di dati; **Statico** - i valori degli assi vengono visualizzati utilizzando l'unità specificata nel campo di input *value*; se il campo viene lasciato vuoto, vengono visualizzati solo valori numerici.

Legenda

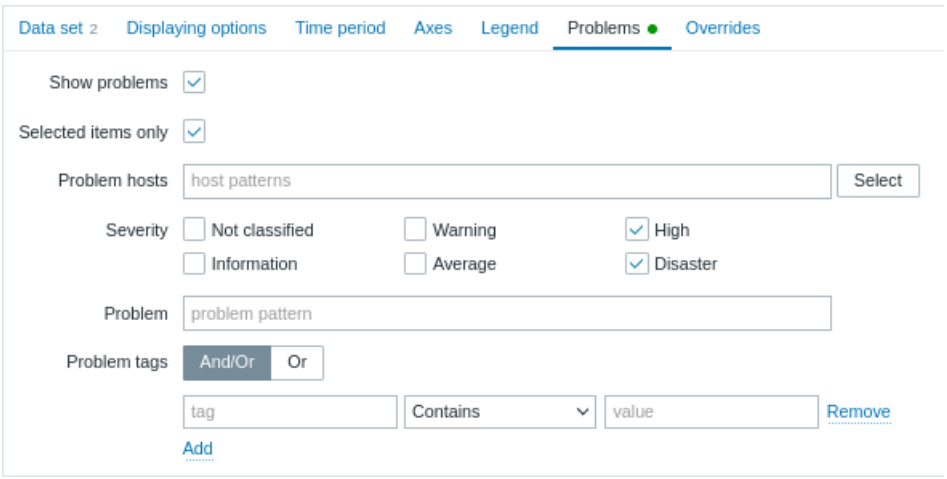
La scheda **Legenda** consente di personalizzare la legenda del grafico:



Mostra legenda	Deselezionare questa casella di controllo per nascondere la legenda nel grafico (selezionata per impostazione predefinita).
Visualizza min/media/max	Selezionare questa casella di controllo per visualizzare nella legenda i valori minimo, medio e massimo dell'item.
Mostra funzione di aggregazione	Selezionare questa casella di controllo per visualizzare la funzione di aggregazione nella legenda.
Righe	Selezionare la modalità di visualizzazione delle righe della legenda: **Fisso** - il numero di righe visualizzate è determinato dal valore del parametro *Numero di righe*; **Variabile** - il numero di righe visualizzate è determinato dalla quantità di item configurati, senza superare il valore del parametro *Numero massimo di righe*.
Numero di righe/ Numero massimo di righe	Se *Righe* è impostato su "Fisso", impostare il numero di righe della legenda da visualizzare (1-10). Se *Righe* è impostato su "Variabile", impostare il numero massimo di righe della legenda da visualizzare (1-10).
Numero di colonne	Impostare il numero di colonne della legenda da visualizzare (1-4). Questo parametro è disponibile se *Visualizza min/media/max* non è selezionato.

Problemi

La scheda **Problemi** consente di personalizzare la visualizzazione dei problemi:

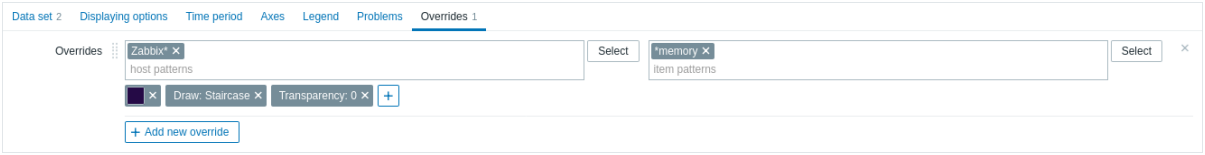


Mostra problemi	Selezionare questa casella di controllo per abilitare la visualizzazione dei problemi nel grafico (non selezionata, ovvero disabilitata per impostazione predefinita).
Solo item selezionati	Selezionare questa casella di controllo per visualizzare nel grafico solo i problemi relativi agli item selezionati.

Host con problemi	Selezionare gli host con problemi da visualizzare nel grafico. È possibile utilizzare modelli con caratteri jolly (ad esempio, * restituirà i risultati che corrispondono a zero o più caratteri). Per specificare un modello con caratteri jolly, inserire manualmente la stringa e premere *Invio*. Durante la digitazione, osservare come tutti gli host corrispondenti vengono visualizzati nell'elenco a discesa. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Gravità	Selezionare le gravità dei problemi per filtrare i problemi da visualizzare nel grafico. Se non viene selezionata alcuna gravità, verranno visualizzati tutti i problemi.
Problema	Specificare il nome del problema da visualizzare nel grafico.
Tag dei problemi	Specificare i tag dei problemi per limitare il numero di problemi visualizzati nel widget. È possibile includere o escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Esiste** - include i nomi dei tag specificati; **Uguale** - include i nomi e i valori dei tag specificati (distingue tra maiuscole e minuscole); **Contiene** - include i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Non esiste** - esclude i nomi dei tag specificati; **Non uguale** - esclude i nomi e i valori dei tag specificati (distingue tra maiuscole e minuscole); **Non contiene** - esclude i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Sono disponibili due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate tramite la condizione *Or*; **Or** - è sufficiente che venga soddisfatta una condizione.

Override

La scheda **Override** consente di aggiungere override personalizzati per i set di dati:



Gli override sono utili quando vengono selezionati diversi item per un set di dati utilizzando il carattere jolly * e si desidera modificare il modo in cui gli item vengono visualizzati per impostazione predefinita (ad esempio, il colore di base predefinito o qualsiasi altra proprietà).

Gli override esistenti, se presenti, vengono visualizzati in un elenco. Per aggiungere un nuovo override:

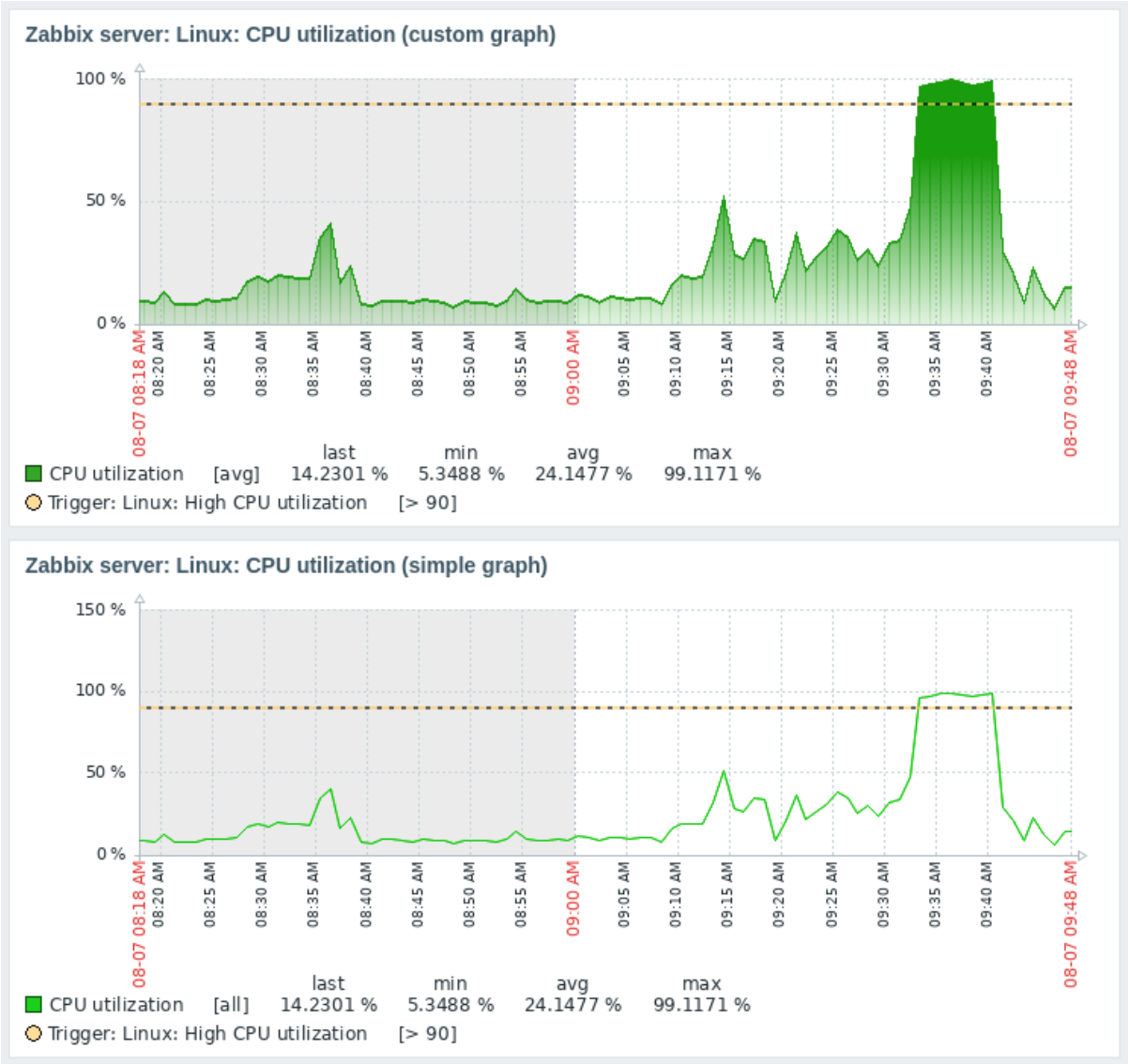
- Fare clic sul pulsante 

- Selezionare gli host e gli item per l'override. In alternativa, è possibile inserire i pattern degli host e degli item. È possibile utilizzare pattern con caratteri jolly (ad esempio, * restituirà i risultati che corrispondono a zero o più caratteri). Per specificare un pattern con caratteri jolly, inserire manualmente la stringa e premere *Invio*. Durante la digitazione, osservare come tutti gli host corrispondenti vengono visualizzati nell'elenco a discesa. Il simbolo jolly viene sempre interpretato; pertanto, non è possibile aggiungere singolarmente, ad esempio, un item denominato "item*" se sono presenti altri item corrispondenti (ad esempio, item2, item3). I parametri relativi al pattern dell'host e al pattern dell'item sono obbligatori. Il parametro per specificare i pattern degli host non è disponibile quando si configura il widget in una dashboard del template. Il parametro per specificare un elenco di item consente di selezionare solo gli item configurati nel template quando si configura il widget in una dashboard del template.
- Fare clic su  per selezionare i parametri dell'override. È necessario selezionare almeno un parametro dell'override. Per le descrizioni dei parametri, vedere la scheda *Set di dati* sopra.

Grafico (classico)

Panoramica

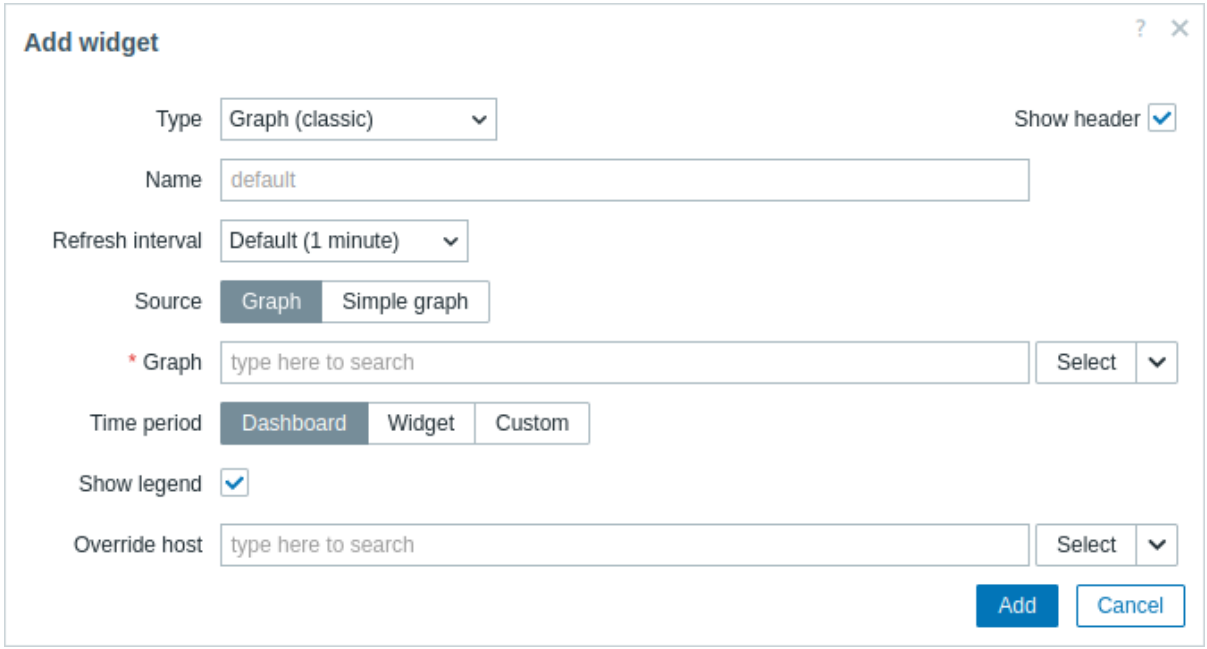
Il widget *Graph (classic)* visualizza i dati numerici degli item come un grafico personalizzato o un grafico semplice basato su immagini. Supporta la selezione del periodo di tempo, le linee dei trigger e la visualizzazione facoltativa della legenda.



Le informazioni visualizzate nel widget *Graph (classic)* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu *Actions* del widget ⋯, visualizzato quando si passa il cursore sul widget. A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella `Downloads` oppure viene aperta una finestra di dialogo *Save*, che consente di scegliere la posizione.

Configurazione

Per configurare, selezionare *Graph (classic)* come tipo:



Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Source	Selezionare il tipo di grafico: **Graph** - grafico personalizzato; **Simple graph** - grafico semplice.
Graph	Selezionare il grafico personalizzato da visualizzare. In alternativa, selezionare un widget compatibile come origine dati per i grafici. Questo parametro è disponibile se *Source* è impostato su "Graph".
Item	Selezionare l'item da visualizzare in un grafico semplice. In alternativa, selezionare un widget compatibile come origine dati per gli item. Questo parametro è disponibile se *Source* è impostato su "Simple graph".
Time period	Impostare un periodo di tempo per il quale visualizzare i dati nel grafico. Selezionare l'origine dati per il periodo di tempo: **Dashboard** - utilizzare il selettore del periodo di tempo della dashboard; **Widget** - utilizzare un widget compatibile (impostato nel parametro *Widget*); **Custom** - utilizzare un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che, al passaggio del mouse, indicherà l'ora impostata. Si noti che, indipendentemente dalla configurazione di *Time period* del widget, i widget compatibili possono comunque usarlo come origine dati per il periodo di tempo.
Widget	Immettere o selezionare un widget compatibile come origine dati per il periodo di tempo. Questo parametro è disponibile se *Time period* è impostato su "Widget".
From	Immettere o selezionare l'inizio del periodo di tempo. La sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.) è supportata. Questo parametro è disponibile se *Time period* è impostato su "Custom".
To	Immettere o selezionare la fine del periodo di tempo. La sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.) è supportata. Questo parametro è disponibile se *Time period* è impostato su "Custom".
Show legend	Deselezionare questa casella di controllo per nascondere la legenda nel grafico (selezionata per impostazione predefinita).
Override host	Selezionare un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.

La legenda del grafico è composta da tre sezioni:

- Item e relativi valori aggregati
- Percentili (se configurati)
- Trigger (se associati agli item visualizzati)

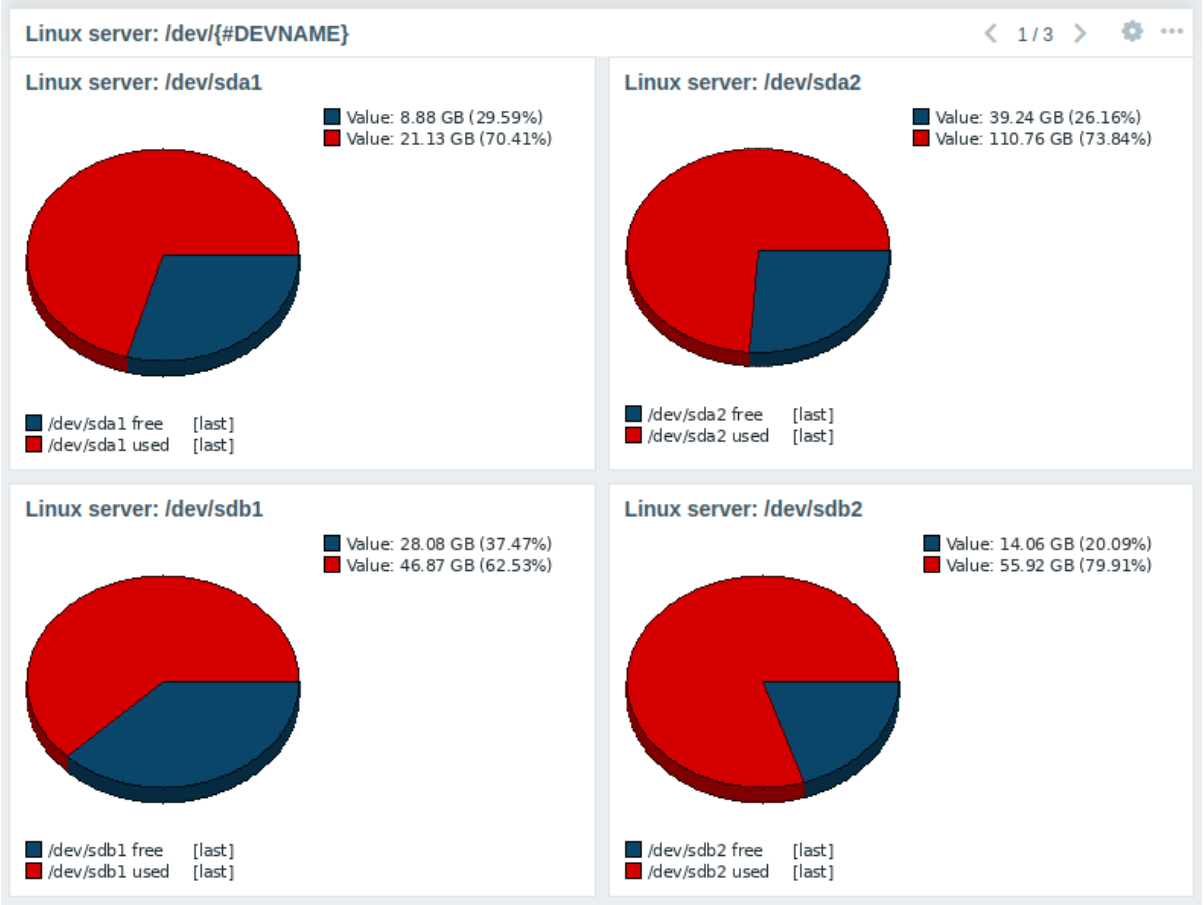
Se l'altezza del grafico all'interno del widget è insufficiente, la legenda potrebbe non essere visualizzata oppure essere visualizzata solo parzialmente. I trigger e i percentili vengono nascosti per primi, seguiti dalla legenda degli item. Per visualizzare la legenda completa, aumentare la dimensione verticale del widget.

Note:
Non possono essere visualizzate più di 3 linee di trigger. Se sono presenti più trigger, per la visualizzazione vengono prioritizzati i trigger con gravità inferiore.

Prototipo di grafico

Panoramica

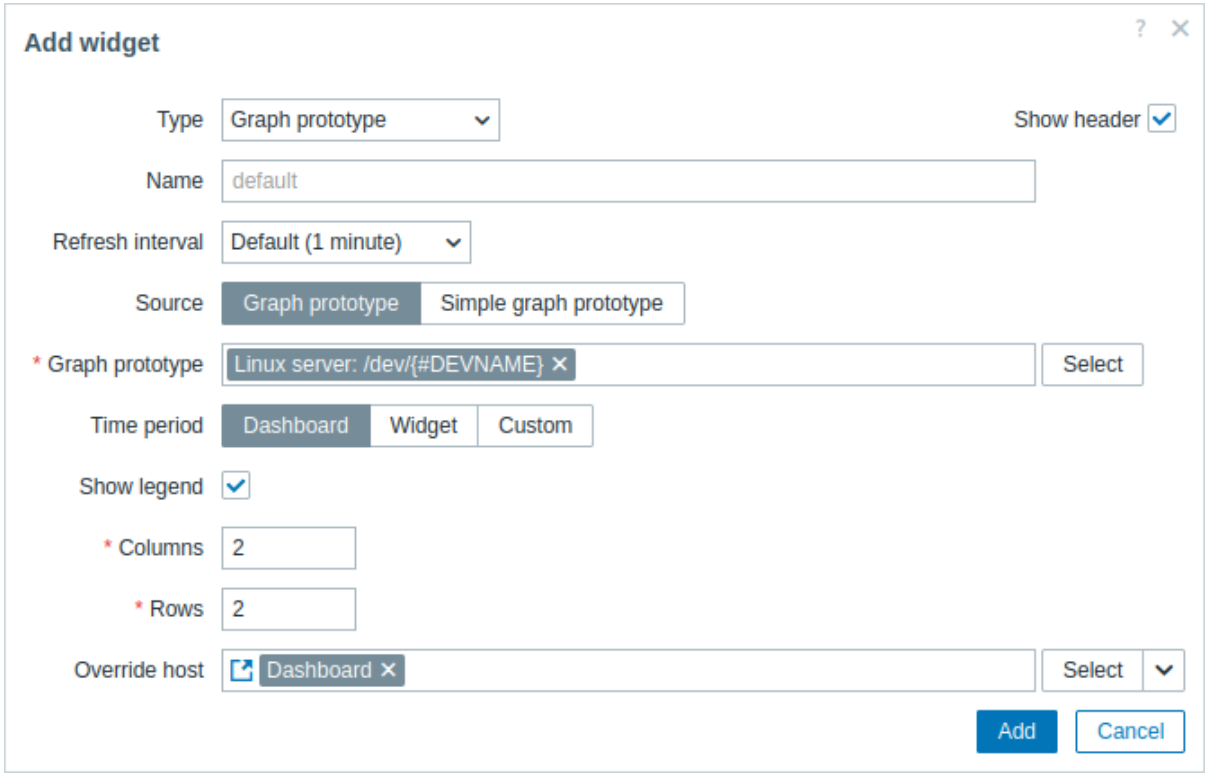
Il widget *Prototipo di grafico* visualizza una griglia di grafici personalizzati creati automaticamente in base ai prototipi di grafico o ai prototipi di item della regola di low-level discovery.



I grafici sono disposti in una griglia configurabile; se il numero di grafici supera il layout, nel widget viene abilitata la paginazione.

Configurazione

Per configurare il widget, selezionare *Graph prototype* come tipo di widget:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Source	Selezionare l'origine dei grafici: **Graph prototype** o **Simple graph prototype**.
Graph prototype	Selezionare un prototipo di grafico per visualizzare i grafici rilevati dal prototipo di grafico. Questo parametro è disponibile se *Source* è impostato su "Graph prototype".
Item prototype	Selezionare un prototipo di item per visualizzare grafici semplici per gli item rilevati dal prototipo di item. Questo parametro è disponibile se *Source* è impostato su "Simple graph prototype".
Time period	Impostare il periodo di tempo per il quale visualizzare i dati nei grafici. Selezionare la sorgente dati per il periodo di tempo: **Dashboard** - utilizzare il selettore del periodo di tempo del dashboard; **Widget** - utilizzare un widget compatibile (impostato nel parametro *Widget*); **Custom** - utilizzare un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio, che al passaggio del mouse indica l'ora impostata. Si noti che, indipendentemente dalla configurazione di *Time period* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per il periodo di tempo.
Widget	Inserire o selezionare un widget compatibile come sorgente dati per il periodo di tempo. Questo parametro è disponibile se *Time period* è impostato su "Widget".
From	Inserire o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
To	Inserire o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
Show legend	Deselezionare questa casella di controllo per nascondere la legenda nei grafici (selezionata per impostazione predefinita).
Override host	Selezionare un widget compatibile o il selettore host del dashboard come sorgente dati per gli host. Questo parametro non è disponibile quando si configura il widget in un dashboard template.
Columns	Inserire il numero di colonne di grafici da visualizzare all'interno di un widget con prototipo di grafico.
Rows	Inserire il numero di righe di grafici da visualizzare all'interno di un widget con prototipo di grafico.

La legenda del grafico è composta da tre sezioni:

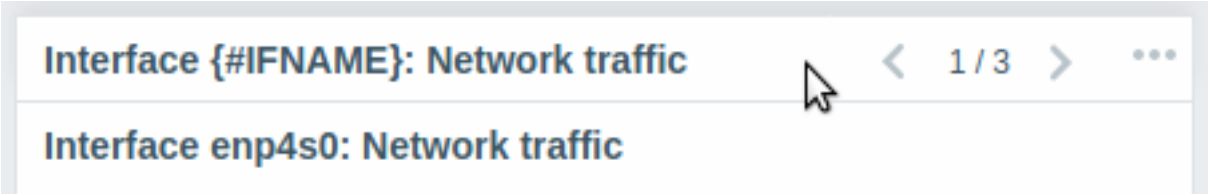
- Item e relativi valori aggregati

- Percentili (se configurati)
- Trigger (se associati agli item visualizzati)

Se l'altezza del grafico all'interno del widget è insufficiente, la legenda potrebbe non essere visualizzata o potrebbe essere visualizzata solo parzialmente. I trigger e i percentili vengono nascosti per primi, seguiti dalla legenda degli item. Per visualizzare la legenda completa, aumentare l'altezza del widget.

Note:
È possibile visualizzare non più di 3 righe di trigger. Se sono presenti più trigger, per la visualizzazione viene data priorità ai trigger con gravità inferiore.

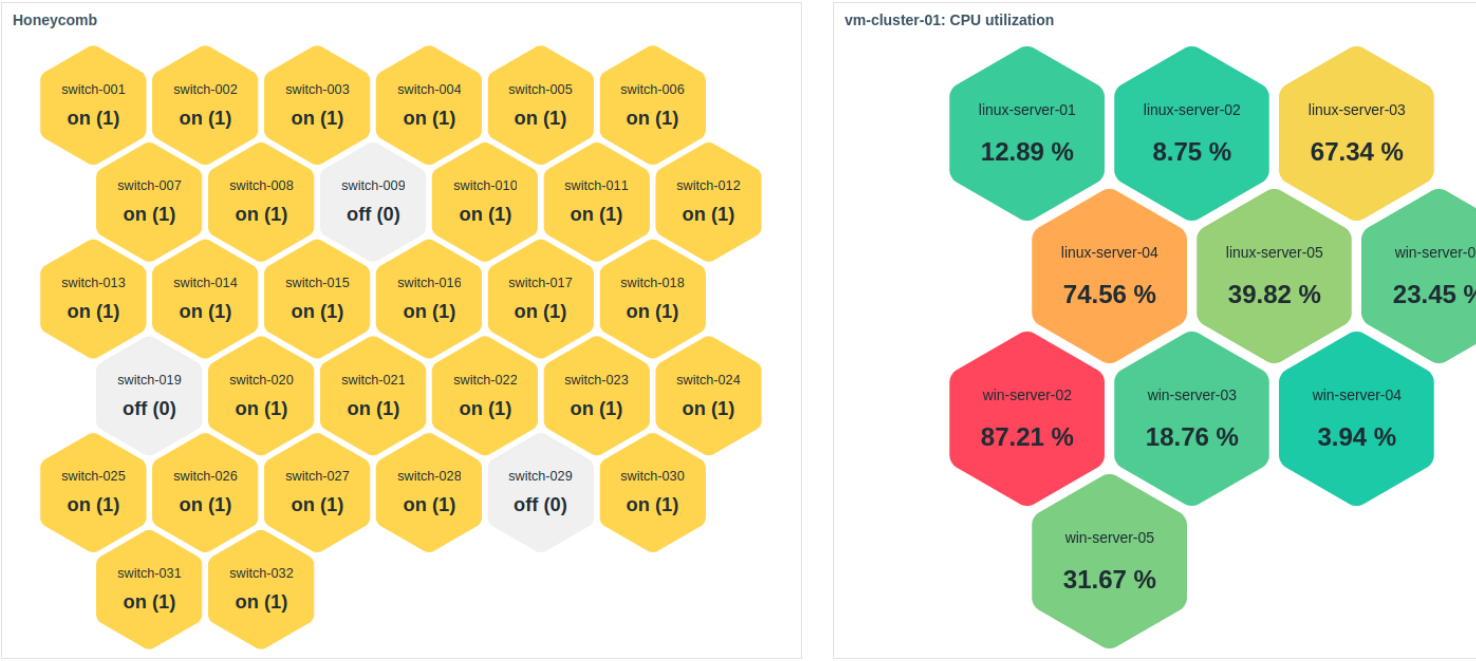
Sebbene i parametri *Columns* e *Rows* consentano di inserire più di un grafico nel widget, potrebbero comunque esserci più grafici rilevati rispetto al numero di colonne/righe disponibili nel widget. In questo caso, nel widget sarà disponibile la paginazione e un'intestazione a comparsa consentirà di passare da una pagina all'altra utilizzando le frecce sinistra e destra:



Honeycomb

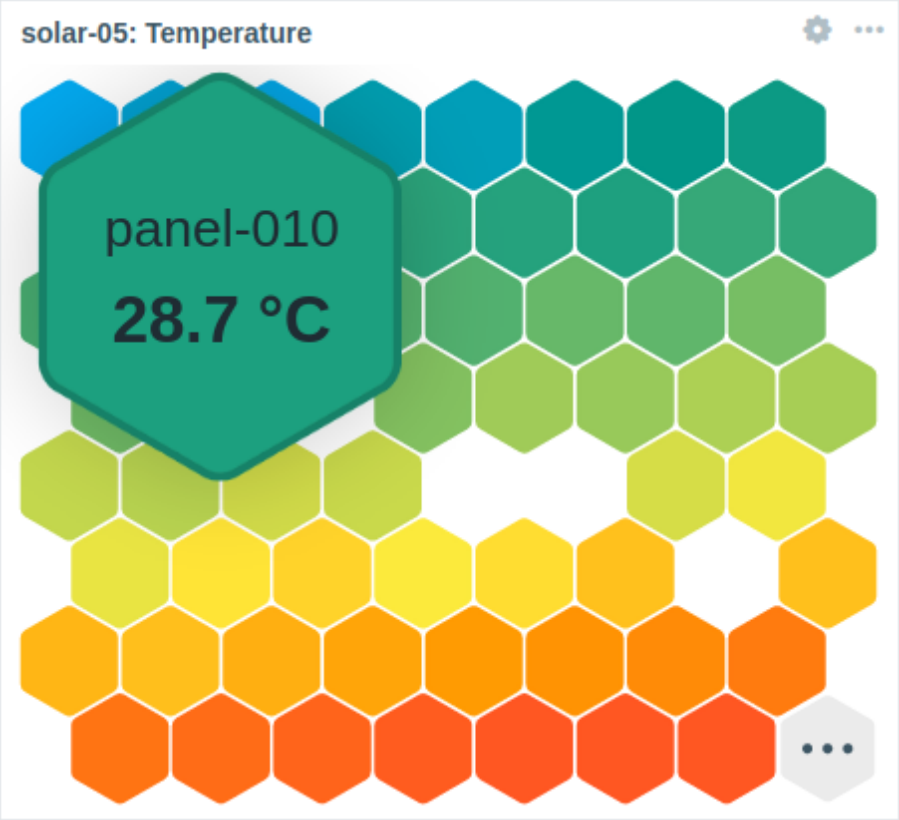
Panoramica

Il widget *Honeycomb* visualizza host o item come celle esagonali in una griglia. Questo layout facilita l'individuazione degli host/item con problemi, l'identificazione di gruppi di problemi, il confronto immediato tra gruppi e altro ancora.



Il numero di celle visualizzate è limitato dalle dimensioni del widget e dalla dimensione minima delle celle (32px). Se non tutte le celle possono essere visualizzate, l'ultima cella mostra un'ellissi. È possibile ridimensionare il widget; il numero, le dimensioni e il posizionamento delle celle vengono adattati dinamicamente.

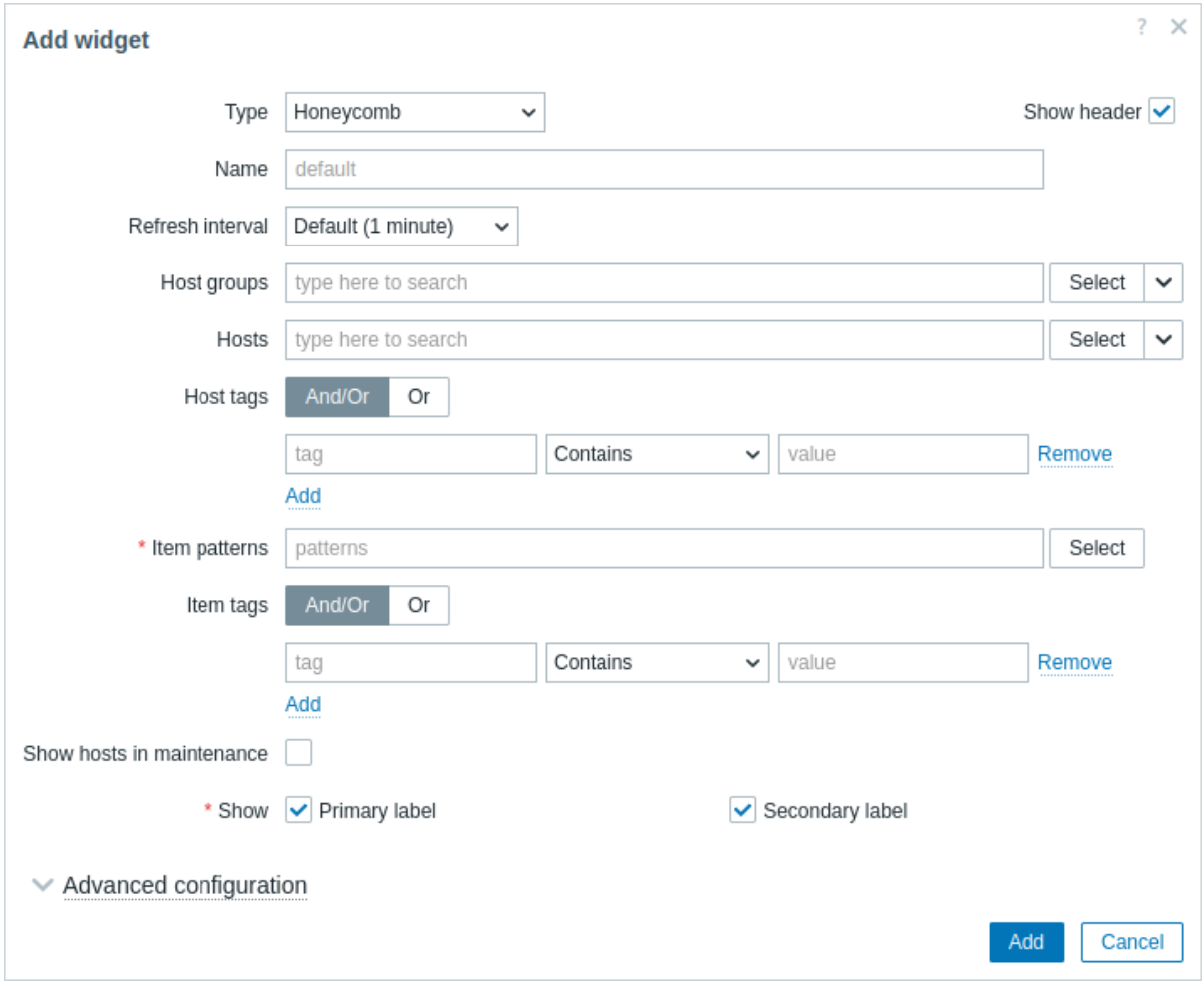
Quando si passa il mouse sopra una cella, quella attiva si ingrandisce per una migliore visibilità. Facendo clic su una cella, il relativo bordo viene evidenziato finché non viene selezionata un'altra cella.



Le informazioni visualizzate nel widget *Honeycomb* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu *Actions* del widget ⋯, visualizzato quando si passa il mouse sopra il widget. A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella `Downloads` oppure viene aperta una finestra di dialogo *Save*, che consente di scegliere la posizione.

Configurazione

Per configurare, selezionare *Honeycomb* come tipo:



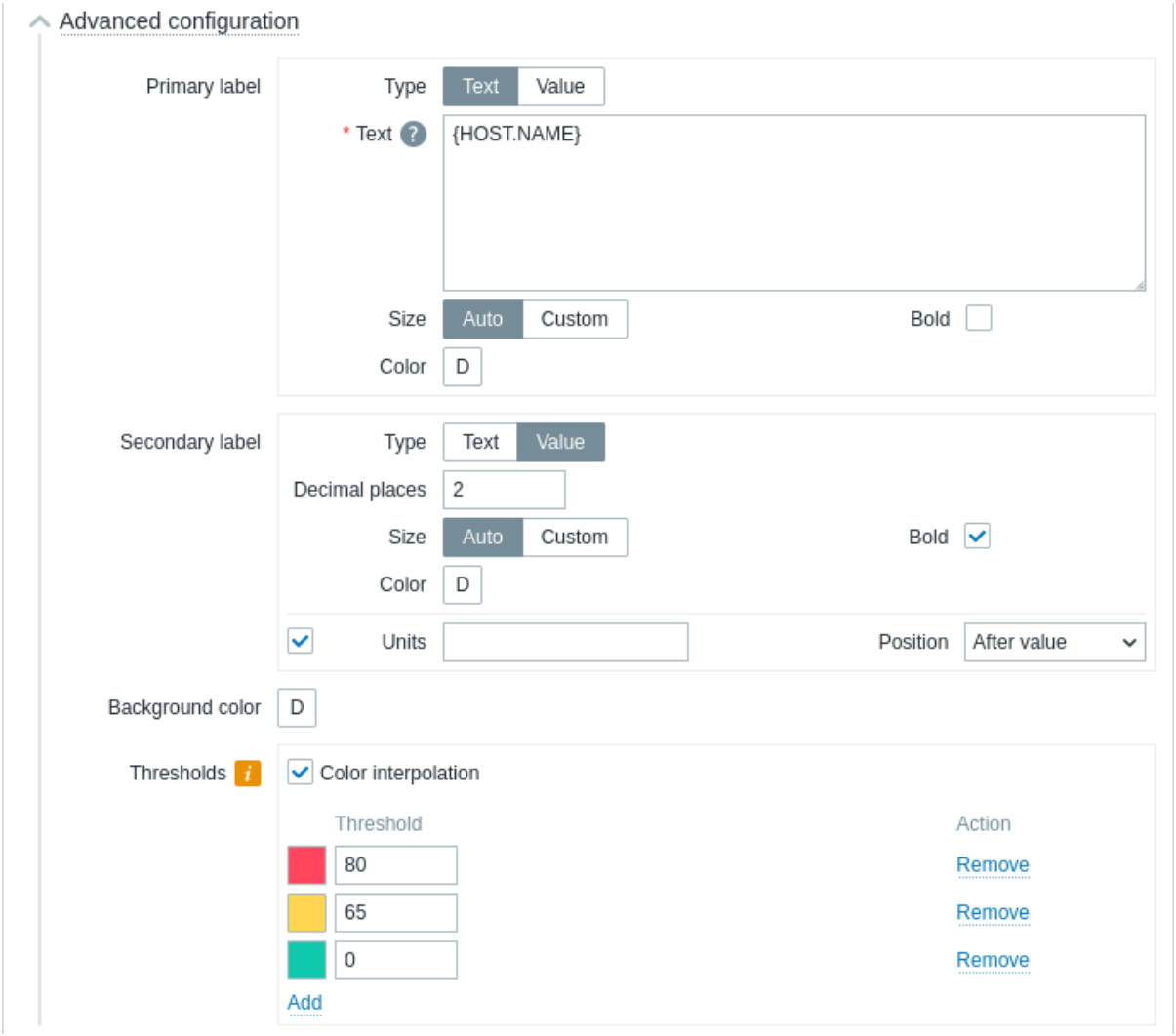
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Host groups	Selezionare i gruppi di host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà visualizzato un menu a discesa con i gruppi corrispondenti. La selezione di un gruppo di host padre seleziona implicitamente tutti i gruppi di host nidificati; se non viene selezionato alcun gruppo di host, il widget visualizzerà tutti i gruppi di host che contengono host con item corrispondenti al *Pattern item* selezionato (vedere sotto). Questo parametro non è disponibile quando si configura il widget in una dashboard template.
Hosts	Selezionare gli host. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà visualizzato un menu a discesa con gli host corrispondenti. Se non viene selezionato alcun host, il widget visualizzerà tutti gli host con item corrispondenti al *Pattern item* selezionato (vedere sotto). Questo parametro non è disponibile quando si configura il widget in una dashboard template.

Host tags	Specificare i tag per limitare il numero di host visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or; **Or** - è sufficiente che sia soddisfatta una condizione.
Item patterns	Questo parametro non è disponibile quando si configura il widget in una dashboard template. Immettere pattern di item oppure selezionare item esistenti come pattern di item. I dati degli item che corrispondono ai pattern immessi o selezionati verranno visualizzati nell'honeycomb. Il parametro *Item patterns* è obbligatorio. Per la selezione possono essere utilizzati pattern con caratteri jolly (ad esempio, `*` restituirà gli item che corrispondono a zero o più caratteri; `Zabbix*` restituirà gli item che iniziano con "Zabbix"). Per specificare un pattern con caratteri jolly, immettere manualmente la stringa e premere *Enter*. Quando si inizia a digitare, un elenco a discesa mostrerà gli item corrispondenti limitati a quelli appartenenti agli *Hosts* selezionati o agli host all'interno dei *Host groups* selezionati, se presenti. Il simbolo jolly viene sempre interpretato, pertanto non è possibile aggiungere singolarmente, ad esempio, un item denominato *item** se esistono altri item corrispondenti (ad esempio, item2, item3). Quando si configura il widget in una dashboard template, questo parametro consente di selezionare solo gli item configurati nel template.
Item tags	Specificare i tag per limitare il numero di item visualizzati nel widget. Per ulteriori informazioni, vedere *Host tags* sopra.
Show hosts in maintenance	Selezionare questa casella di controllo per visualizzare gli host in manutenzione (in questo caso, l'icona di manutenzione verrà mostrata accanto al nome dell'host). Questo parametro è etichettato come *Show data in maintenance* quando si configura il widget in una dashboard template.
Show	Selezionare questa casella di controllo per visualizzare il rispettivo elemento della cella honeycomb: etichetta primaria, etichetta secondaria. Deve essere selezionato almeno un elemento.
Advanced configuration	Fare clic sull'intestazione *Advanced configuration* per visualizzare le opzioni di configurazione avanzata.

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata* e solo per gli elementi selezionati nel campo *Mostra* (vedi sopra), nonché per il colore di sfondo o le soglie delle celle del nido d'ape.



Etichetta primaria/secondaria

Tipo	Selezionare il tipo di etichetta: **Testo** - l'etichetta visualizzerà il testo specificato per il parametro *Testo*; **Valore** - l'etichetta visualizzerà il valore dell'item con il numero di posizioni decimali specificato per il parametro *Posizioni decimali*.
Testo	Inserire il testo dell'etichetta. Questo testo può sostituire il nome predefinito dell'item. È supportato il testo su più righe. È possibile una combinazione di testo e macro supportate. Sono supportati `{HOST.\*}`, `{ITEM.\*}`, `{INVENTORY.\*}` e le macro utente. Le celle del nido d'ape sono ordinate alfabeticamente per nome host e, all'interno di ciascun host, per nome item. Questo parametro è disponibile se *Tipo* è impostato su *Testo*.
Posizioni decimali	Inserire il numero di posizioni decimali da visualizzare con il valore. Questo parametro è disponibile se *Tipo* è impostato su *Valore* e influisce solo sugli item che restituiscono dati numerici (float).
Dimensione	Selezionare la dimensione dell'etichetta: **Auto** - usa una dimensione dell'etichetta regolata automaticamente; **Personalizzata** - inserire una dimensione personalizzata dell'etichetta (in percentuale, relativa alla dimensione della cella del nido d'ape). Si noti che le etichette che non rientrano nella dimensione della cella del nido d'ape vengono troncate.
Grassetto	Selezionare la casella di controllo per visualizzare le etichette in grassetto.
Colore	Selezionare il colore dell'etichetta dal selettore colori. "D" indica il colore predefinito, che dipende dal tema del frontend. Per tornare al colore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.

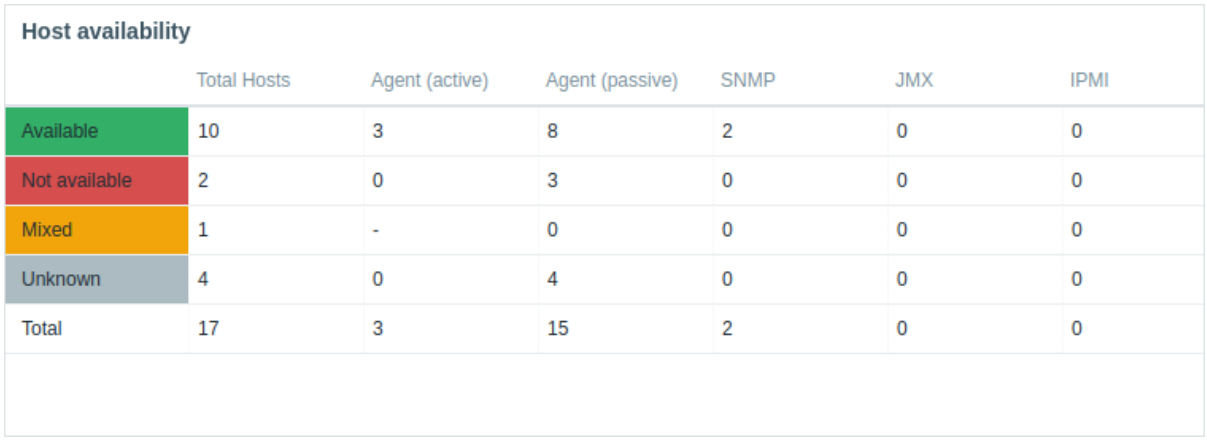
Unità

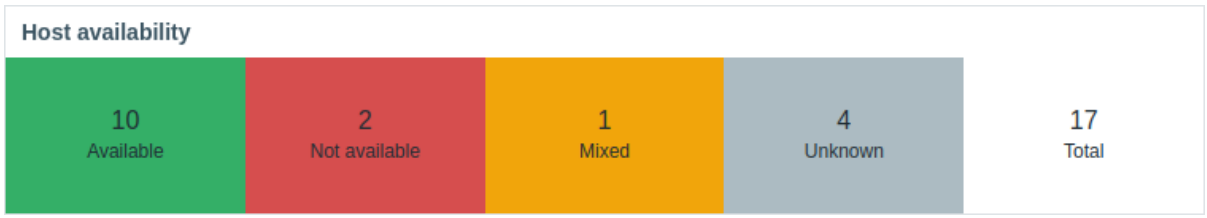
Unità	Selezionare la casella di controllo per visualizzare le unità con il valore dell'item. Se si inserisce un nome di unità, questo sostituirà le unità impostate nella configurazione dell'item. Questo parametro è disponibile se *Tipo* è impostato su *Valore*.
Posizione	Selezionare la posizione delle unità dell'item (prima o dopo il valore dell'item). Questo parametro viene ignorato per le seguenti unità relative al tempo: unixtime, uptime, s. Questo parametro è disponibile se *Tipo* è impostato su *Valore* ed è selezionata l'opzione *Unità*.
Colore di sfondo	
Colore di sfondo	Selezionare il colore di sfondo delle celle del nido d'ape dal selettore colori. "D" indica il colore predefinito, che dipende dal tema del frontend. Per tornare al colore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Soglie	
Interpolazione del colore	Selezionare la casella di controllo per abilitare una transizione uniforme tra i colori delle soglie per le celle del nido d'ape. Questo parametro è disponibile se sono impostate due o più soglie.
Soglia	Fare clic su *Aggiungi* per aggiungere una soglia, selezionare un colore della soglia dal selettore colori e specificare un valore numerico. L'elenco delle soglie verrà ordinato in ordine crescente al momento del salvataggio. Si noti che i colori configurati come soglie verranno visualizzati correttamente solo per gli item numerici. Sono supportati i suffissi (ad esempio, "1d", "2w", "4K", "8G"). Sono supportate le mappature dei valori.

Disponibilità dell'host

Panoramica

Il widget *Disponibilità host* visualizza la disponibilità degli host nei gruppi host selezionati, consentendoti di monitorare il numero di host attivi o non attivi.



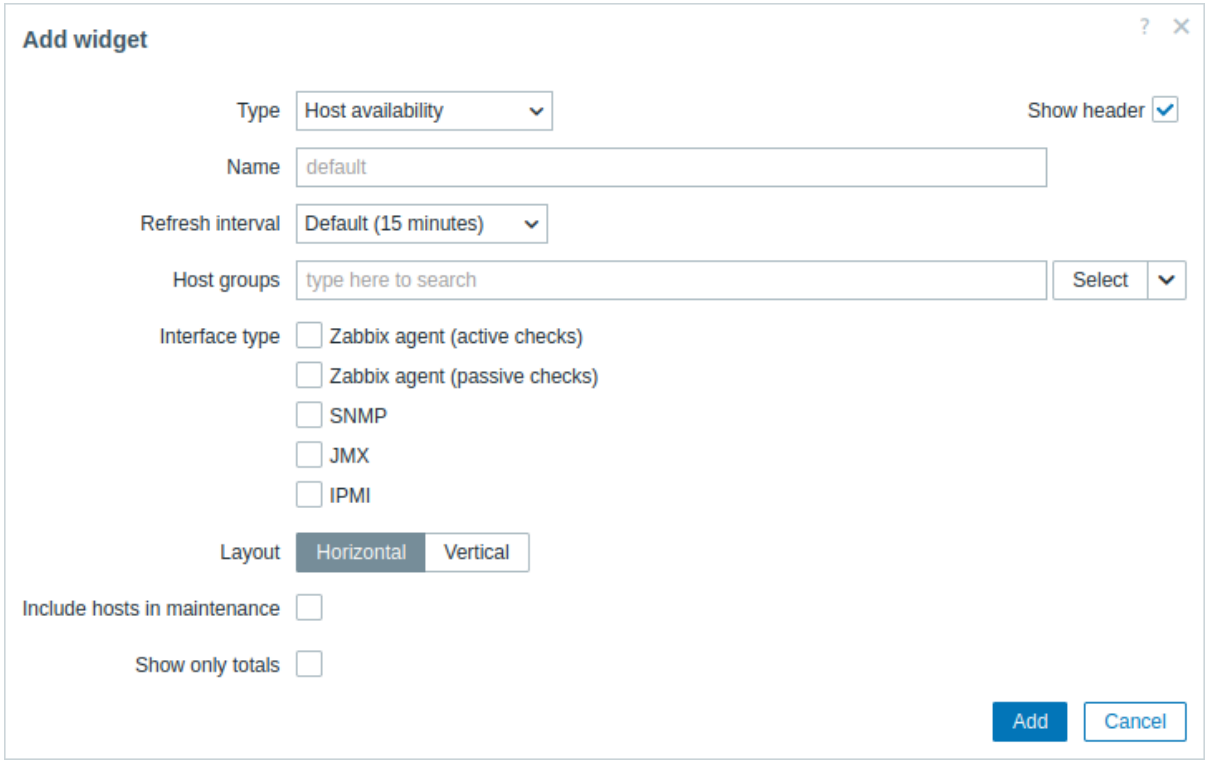


La disponibilità degli host viene conteggiata come segue:

- *Disponibile* - host con tutte le interfacce disponibili
- *Non disponibile* - host con tutte le interfacce non disponibili
- *Mista* - host con almeno un'interfaccia non disponibile e almeno una disponibile o sconosciuta
- *Sconosciuta* - host con almeno un'interfaccia sconosciuta, ma nessuna non disponibile
- *Totale* - totale di tutti gli host

Configurazione

Per configurare il widget, selezionare *Disponibilità host* come tipo:



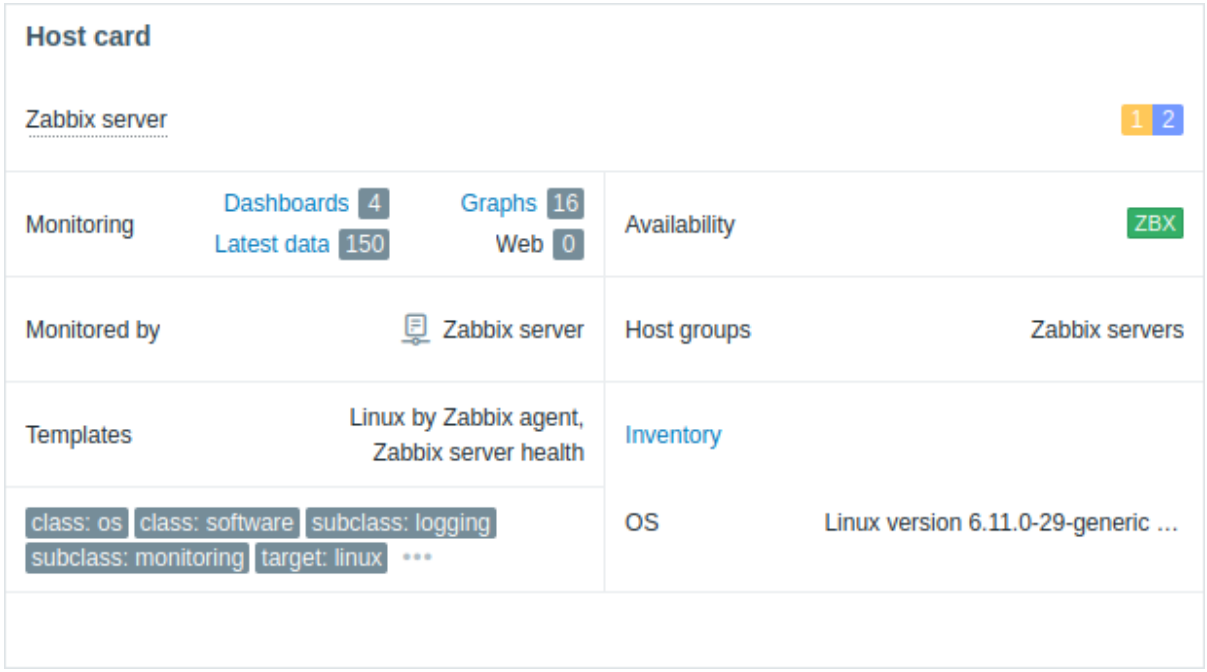
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Gruppi host	Selezionare i gruppi host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà visualizzato un menu a discesa con i gruppi corrispondenti. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Tipo di interfaccia	Selezionare per quali interfacce host si desidera visualizzare i dati di disponibilità: *Zabbix agent (controlli attivi)*, *Zabbix agent (controlli passivi)*, *SNMP*, *JMX*, *IPMI*. Per *Zabbix agent (controlli attivi)*, la cella *Misto* sarà sempre vuota, poiché questo tipo di item non può avere più interfacce. Per impostazione predefinita, se non viene selezionato nulla viene visualizzata la disponibilità di tutte le interfacce.
Layout	Selezionare la visualizzazione orizzontale (colonne) o verticale (righe).
Includi host in manutenzione	Includere nelle statistiche gli host in manutenzione. Questo parametro è etichettato come *Mostra dati in manutenzione* quando si configura il widget in una dashboard di template.
Mostra solo i totali	Se selezionato, viene visualizzato il totale degli host, senza suddivisione per interfacce. Questa opzione è disabilitata se è selezionata una sola interfaccia.

Scheda host

Panoramica

Il widget *Scheda host* visualizza i dettagli di un singolo host, consentendo di valutare rapidamente lo stato e la configurazione di un host (disponibilità, problemi, inventario, ecc.).

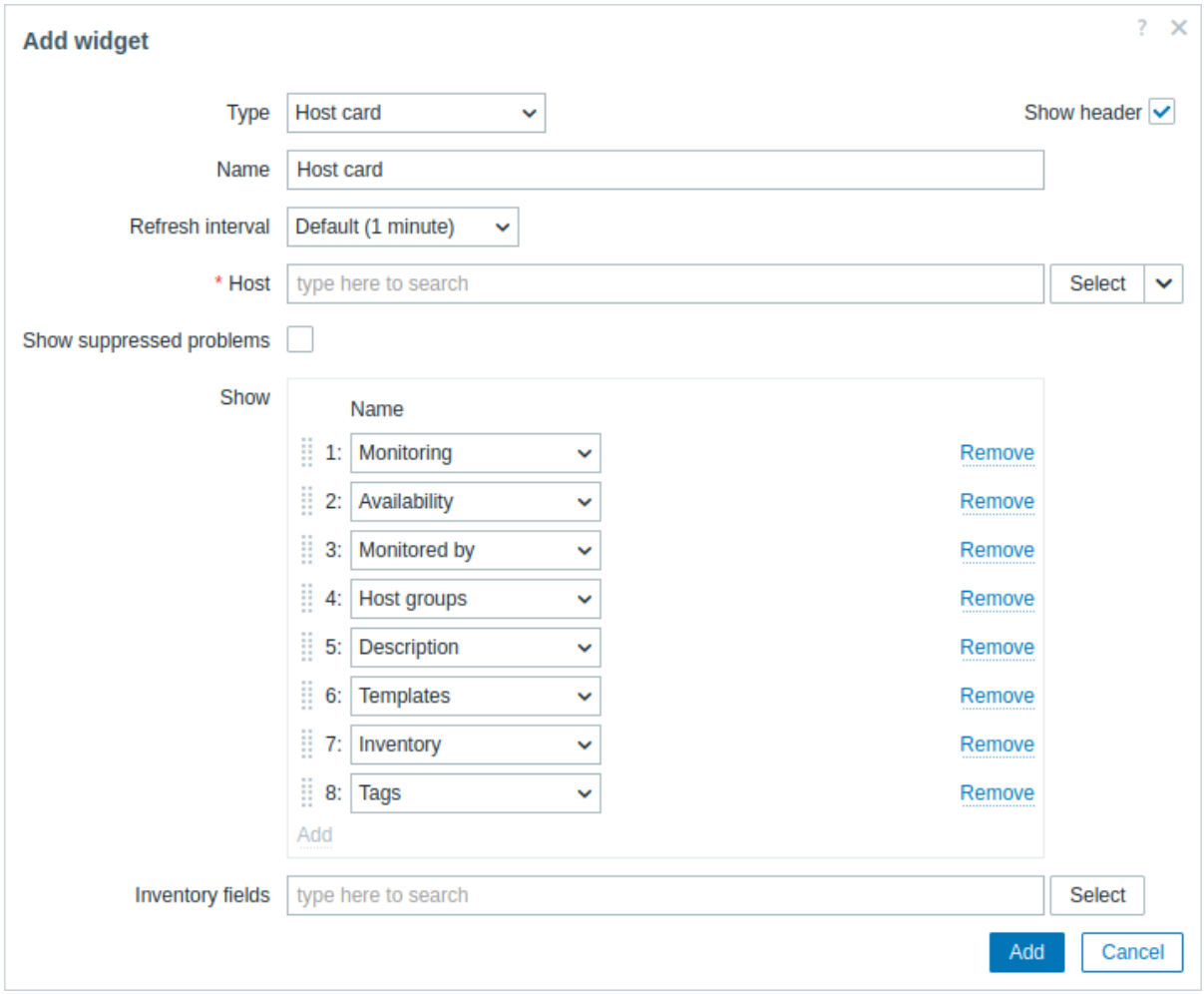


L'host può essere impostato direttamente nella configurazione del widget oppure selezionato da un widget compatibile o dal selettore host nella dashboard.

Il widget può visualizzare più sezioni, ciascuna delle quali mostra informazioni diverse. La disposizione delle sezioni viene regolata automaticamente in base alla larghezza del widget; quando viene espanso orizzontalmente, le sezioni vengono riorganizzate automaticamente in più colonne.

Configurazione

Per configurare, selezionare *Host card* come tipo:



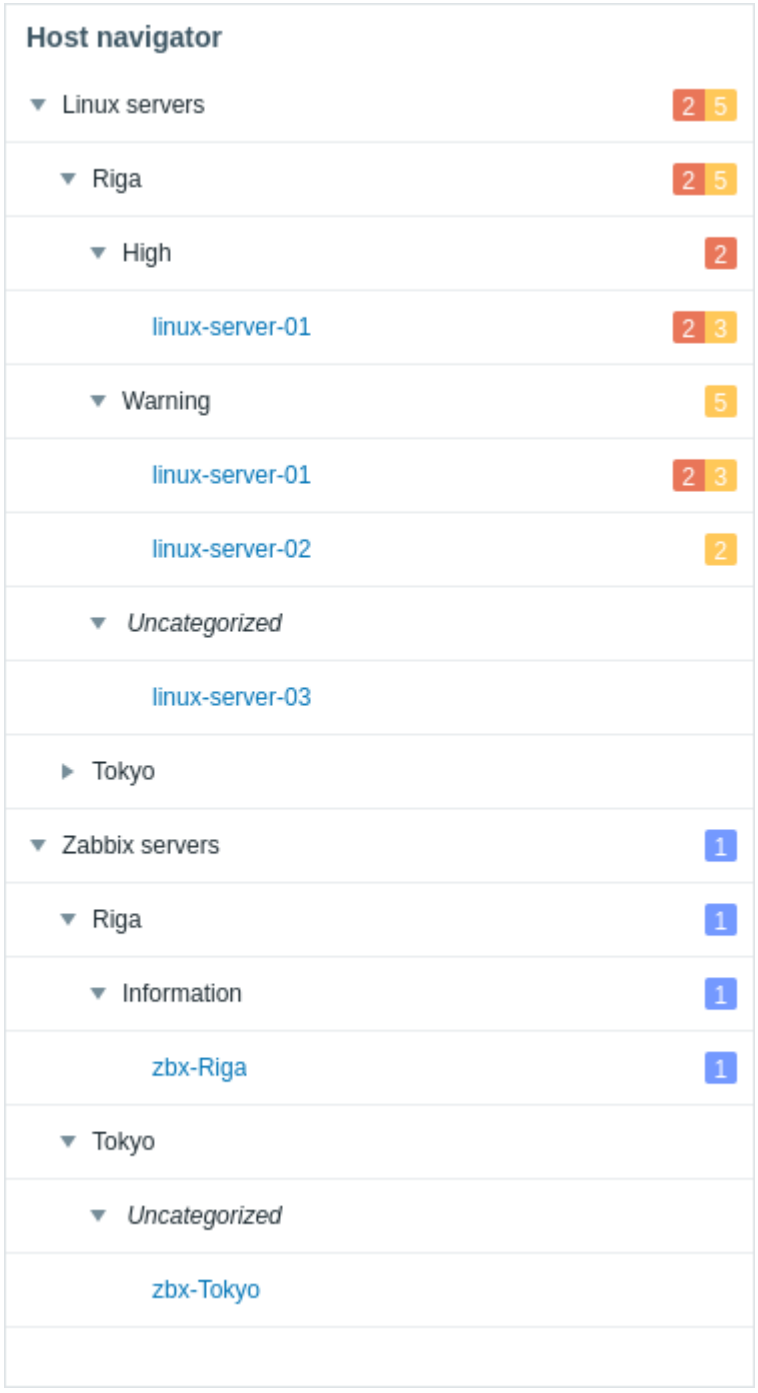
Oltre ai parametri comuni a tutti i widget, è possibile impostare i seguenti parametri specifici:

Host	Selezionare l'host. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. I suggerimenti corrispondenti vengono visualizzati durante la digitazione. La sezione superiore del widget mostra sempre il **nome host** (facendo clic su di esso si apre il menu host): - Se l'host è disabilitato, accanto al nome host viene visualizzata un'etichetta rossa *Disabled*. - Se l'host è in manutenzione, accanto al nome host viene visualizzata l'icona di manutenzione. La sezione superiore del widget mostra anche il numero di **problemi dell'host** raggruppati per gravità (facendo clic su di essi si aprono i problemi dell'host). Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Show suppressed problems	Selezionare questa casella per includere i problemi soppressi nel calcolo dei problemi.
Show	Aggiungere le sezioni da visualizzare nel widget. Le sezioni possono essere riordinate tramite trascinamento.
Host groups	Visualizza i gruppi host a cui appartiene l'host.
Description	Visualizza la descrizione dell'host.
Monitoring	Visualizza collegamenti rapidi per passare a Dashboard, Ultimi dati, Grafici e Scenari web dell'host. Il numero accanto a ciascun collegamento indica il numero delle rispettive entità.
Availability	Visualizza la disponibilità dell'host per interfaccia.
Monitored by	Visualizza se l'host è monitorato da Zabbix server o da uno specifico proxy o gruppo di proxy.
Templates	Visualizza i template collegati all'host.
Inventory	Visualizza i campi dell'inventario host.
Tags	Visualizza i tag dell'host.
Inventory fields	Selezionare i campi dell'inventario da visualizzare. I suggerimenti corrispondenti vengono visualizzati durante la digitazione. Se non viene specificato alcun campo dell'inventario, verranno visualizzati tutti i campi dell'inventario compilati. Questo parametro è disponibile se *Show* contiene *Inventory*.

Navigatore host

Panoramica

Il widget *Navigatore host* visualizza un elenco di host in base a varie opzioni di filtro e raggruppamento.



I gruppi in base ai quali gli host sono organizzati possono essere espansi o compressi.

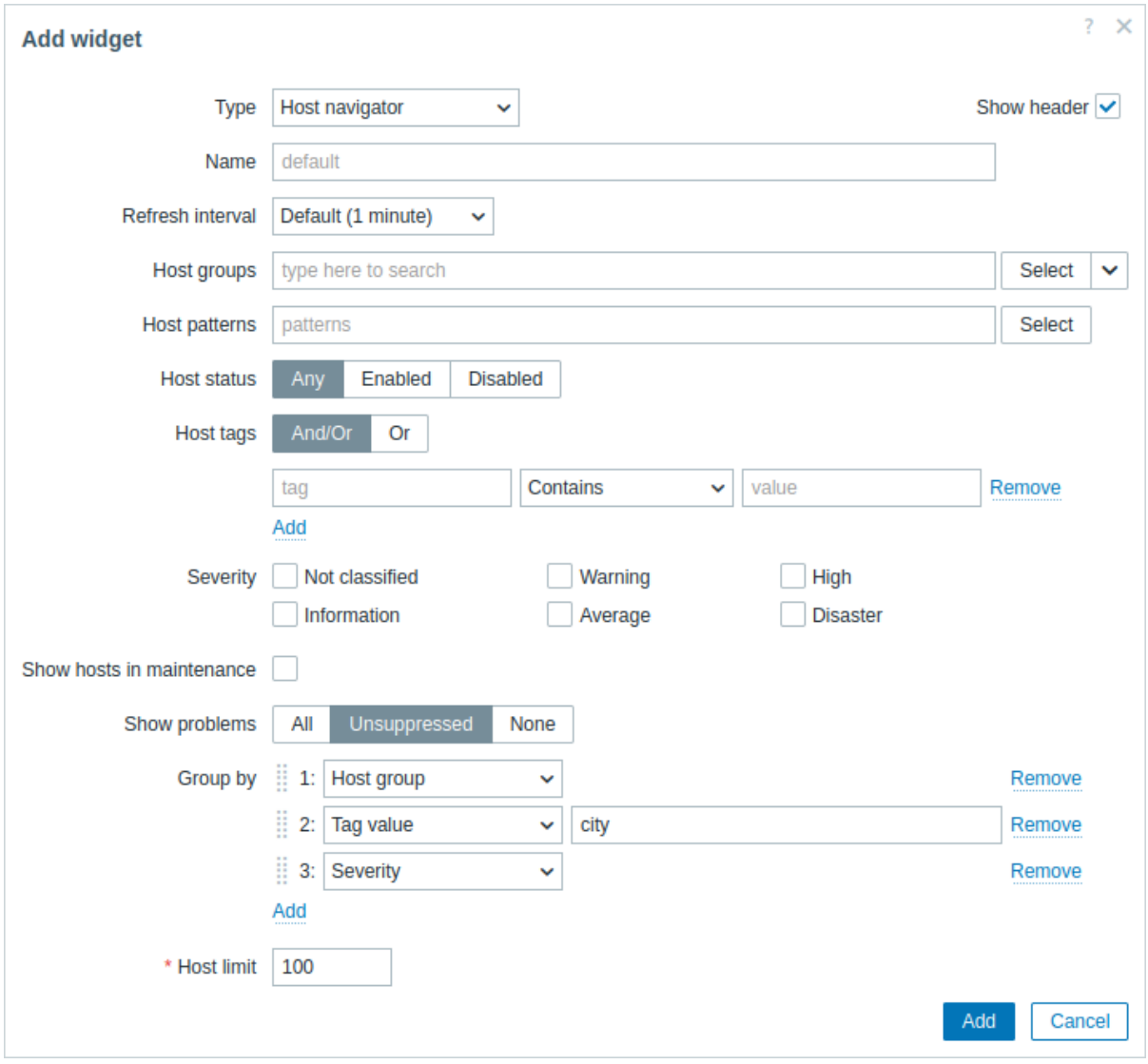
Per i gruppi, i problemi e gli host in manutenzione, ulteriori dettagli sono accessibili tramite suggerimenti al passaggio del mouse.

Il widget è particolarmente utile per controllare ciò che altri widget visualizzano in base all'host selezionato.



Configurazione

Per configurare, selezionare *Host navigator* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

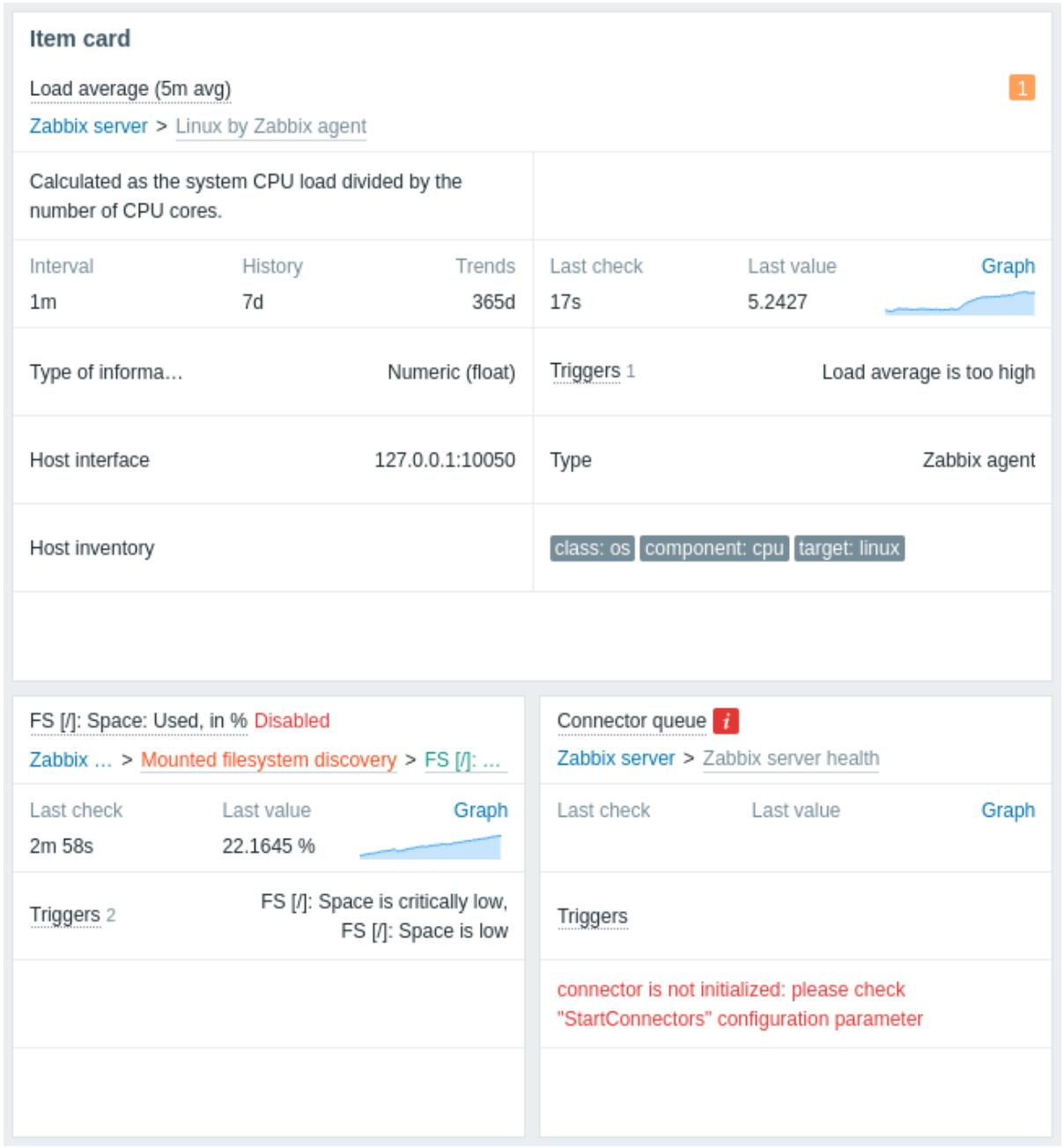
Host groups	Selezionare i gruppi di host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico: iniziando a digitare il nome di un gruppo, verrà visualizzato un elenco a discesa con i gruppi corrispondenti. La selezione di un gruppo di host padre seleziona implicitamente tutti i gruppi di host annidati; se non viene selezionato alcun gruppo di host, il widget visualizzerà tutti gli host di tutti i gruppi di host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Host patterns	Inserire i pattern degli host oppure selezionare host esistenti come pattern degli host. Gli host che corrispondono ai pattern specificati verranno visualizzati nell'host navigator. Questo campo supporta il completamento automatico: iniziando a digitare il nome di un host, verrà visualizzato un elenco a discesa con gli host corrispondenti. Se non viene selezionato alcun host, il widget visualizzerà tutti gli host. È possibile utilizzare pattern con caratteri jolly (ad esempio, `*` restituirà gli host che corrispondono a zero o più caratteri; `Zabbix*` restituirà gli host il cui nome inizia con "Zabbix"). Per specificare un pattern con caratteri jolly, inserire manualmente la stringa e premere *Enter*. Quando si inizia a digitare, un elenco a discesa mostrerà gli host corrispondenti, limitati a quelli appartenenti agli host inclusi nei *Host groups* selezionati, se presenti. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Host status	Filtrare gli host da visualizzare in base al loro stato (qualsiasi, abilitato, disabilitato). Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Host tags	Specificare i tag per filtrare gli host visualizzati nel widget. È possibile includere o escludere tag e valori dei tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi dei tag i cui valori contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag i cui valori contengono la stringa inserita (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - devono essere soddisfatte tutte le condizioni; le condizioni con lo stesso nome di tag vengono raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Severity	Selezionare le gravità dei problemi per filtrare gli host con problemi da visualizzare nel widget. Se non viene selezionata alcuna gravità, verranno visualizzati tutti gli host con tutti i problemi.
Show hosts in maintenance	Selezionare questa casella di controllo per visualizzare gli host in manutenzione (in questo caso, accanto al nome dell'host verrà mostrata l'icona di manutenzione). Questo parametro è denominato *Show data in maintenance* quando si configura il widget in una dashboard di template.
Show problems	Filtrare i problemi da visualizzare in base al loro stato (tutti, non soppressi, nessuno). I problemi sono indicati da blocchi colorati alla fine della riga dell'host. I colori si basano sulla gravità del problema, che può essere personalizzata in *Administration* > *General* > *Trigger displaying options*. Le gravità dei problemi possono essere modificate durante l'aggiornamento dei problemi.

Group by	Aggiungere un attributo di raggruppamento in base al quale raggruppare gli host selezionati: **Host group** - raggruppa gli host in base al relativo gruppo di host; **Tag value** - inserire un nome di tag per raggruppare gli host in base ai valori di questo tag (ad esempio, inserire ”city” per raggruppare gli host in base ai valori ”Riga”, ”Tokyo” ecc.); **Severity** - raggruppa gli host in base alla gravità dei relativi problemi. Se *Show problems* è configurato per visualizzare i problemi, questi vengono visualizzati come segue: - per ogni gruppo di gravità viene visualizzato solo il conteggio dei problemi corrispondente; - per ogni host vengono visualizzati tutti i conteggi dei relativi problemi. Gli host verranno raggruppati solo in base alle gravità selezionate nel parametro *Severity*; se non viene selezionata alcuna gravità, tutti gli host verranno raggruppati in base a tutte le gravità. Gli attributi di raggruppamento possono essere riordinati trascinandoli verso l’alto o verso il basso tramite l’elemento di trascinamento posto prima del nome del gruppo. L’ordine degli attributi di raggruppamento determina l’ordine di annidamento dei gruppi. Ad esempio, specificando più nomi di tag (1: color, 2: city), gli host verranno raggruppati prima per colore (rosso, blu ecc.) e poi per città (Riga, Tokyo ecc.). Un host può essere visualizzato in più gruppi in base agli attributi di raggruppamento configurati (ad esempio, quando si raggruppa per gruppo di host e l’host appartiene a più gruppi di host). Facendo clic su tali host, questi vengono selezionati ed evidenziati in tutti i gruppi. Gli host che non corrispondono agli attributi di raggruppamento configurati vengono visualizzati nel gruppo *Uncategorized*. È possibile specificare fino a 10 attributi di raggruppamento, che devono essere tutti univoci. Se non viene specificato alcun attributo di raggruppamento, gli host non verranno raggruppati.
Host limit	Inserire il numero massimo di host da visualizzare. I valori possibili vanno da 1 a 9999. Quando sono disponibili per la visualizzazione più host rispetto al limite impostato, sotto gli host visualizzati viene mostrato un messaggio corrispondente (ad esempio, ”100 of 100+ hosts are shown”). Il limite di host configurato influisce anche sulla visualizzazione dei gruppi configurati; ad esempio, se il limite di host è impostato su 100 e gli host sono raggruppati per valori dei tag (più di 200), nel widget verranno visualizzati solo i primi 100 valori dei tag con i relativi host. Questo parametro non è influenzato dal parametro *Limit for search and filter results* in *Administration > General > GUI*. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.

Scheda item

Panoramica

Il widget *Scheda item* visualizza i dettagli di un singolo item, consentendoti di valutare rapidamente lo stato e la configurazione di un item (ultimi dati, errori, descrizione, ecc.).

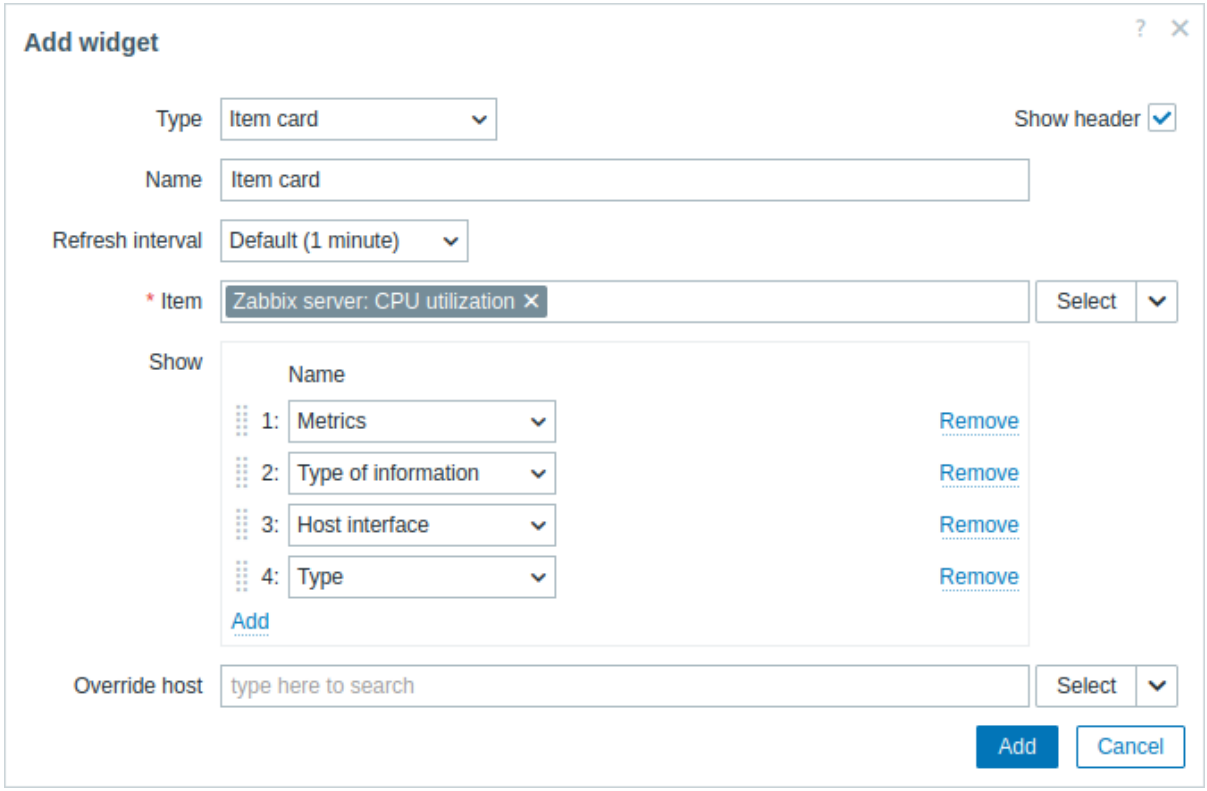


L'item può essere impostato direttamente nella configurazione del widget oppure selezionato da un widget compatibile nella dashboard.

Il widget può visualizzare più sezioni, ciascuna delle quali mostra informazioni diverse. La disposizione delle sezioni viene regolata automaticamente in base alla larghezza del widget; quando viene espanso orizzontalmente, le sezioni vengono automaticamente riorganizzate in più colonne.

Configurazione

Per configurare, selezionare *Item card* come tipo:



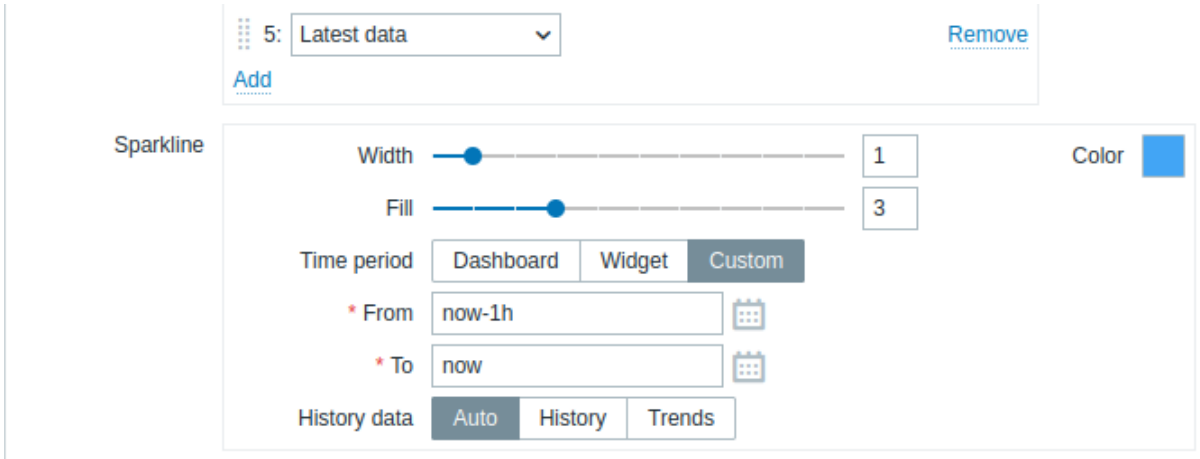
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Item	Selezionare l'item. In alternativa, selezionare un widget compatibile come origine dati per gli item. I suggerimenti corrispondenti vengono visualizzati durante la digitazione. La sezione superiore del widget mostra sempre il **nome dell'item** (facendo clic su di esso si apre il menu item): - Se l'item è disabilitato, accanto al nome dell'item viene visualizzata un'etichetta rossa *Disabled*. - In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". In caso di problemi, viene visualizzata un'icona quadrata con il numero di problemi per ciascuna severità del problema. Passando il cursore sopra un'icona si apre un pop-up con i dettagli. - In una dashboard di template, viene visualizzato il nome tecnico dell'item (invece del nome visibile). La sezione superiore del widget mostra anche il **percorso dell'item**: il suo host e, se applicabile, template, regola di low-level discovery (LLD) o item master (facendo clic su un'entità si apre la relativa configurazione). Il percorso dell'item viene visualizzato solo agli utenti con autorizzazioni per le rispettive entità.
Show	Aggiungere le sezioni da visualizzare nel widget. Le sezioni possono essere riordinate tramite trascinamento.
Description	Visualizza la descrizione dell'item. Le macro vengono risolte.
Error text	Visualizza l'errore dell'item (se presente).
Metrics	Visualizza le seguenti informazioni: - Intervallo di aggiornamento dell'item (eccetto per i trapper item); - Configurazione della history dell'item (se presente); - Configurazione dei trends dell'item (se presente). Le macro vengono risolte. Se l'item utilizza intervalli personalizzati, accanto all'intervallo di aggiornamento dell'item viene visualizzata un'icona di aiuto. Passando il cursore sopra l'icona si apre un pop-up con i dettagli degli intervalli personalizzati.

Latest data	Visualizza le seguenti informazioni: - Tempo trascorso dall'ultimo controllo dell'item; - Ultimo valore dell'item; - Collegamento al grafico semplice/history dei valori dell'item (se history o trends sono memorizzati); - Grafico sparkline configurabile (se l'item è numerico). Per gli item di tipo binario, al posto dell'ultimo valore viene visualizzata una miniatura o l'opzione *Show*. Passando il cursore sopra la miniatura si apre un pop-up con l'immagine. Passando il cursore sopra *Show* si apre un pop-up con il valore dell'item (stringa Base64). In una dashboard di template, al posto dell'ultimo valore dell'item viene visualizzato *No data*.
Type of information	Visualizza il tipo di informazione dell'item.
Triggers	Visualizza le seguenti informazioni: - Etichetta *Triggers* con il numero di trigger (passando il cursore sopra *Triggers* si apre un pop-up con i dettagli del trigger); - Elenco dei nomi dei trigger (le macro vengono risolte).
Host interface	Visualizza l'interfaccia host utilizzata dall'item, oppure *No data* se l'item non dispone di un'interfaccia host (ad esempio, item dipendenti, simple checks). In una dashboard di template, viene visualizzato *No data*.
Type	Visualizza il tipo di item.
Host inventory	Visualizza il campo dell'inventario host popolato dal valore dell'item.
Tags	Visualizza i tag dell'item.
Override host	Selezionare un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.

Sparkline

Le opzioni di configurazione del grafico sparkline sono disponibili solo se la sezione *Ultimi dati* è aggiunta al widget.



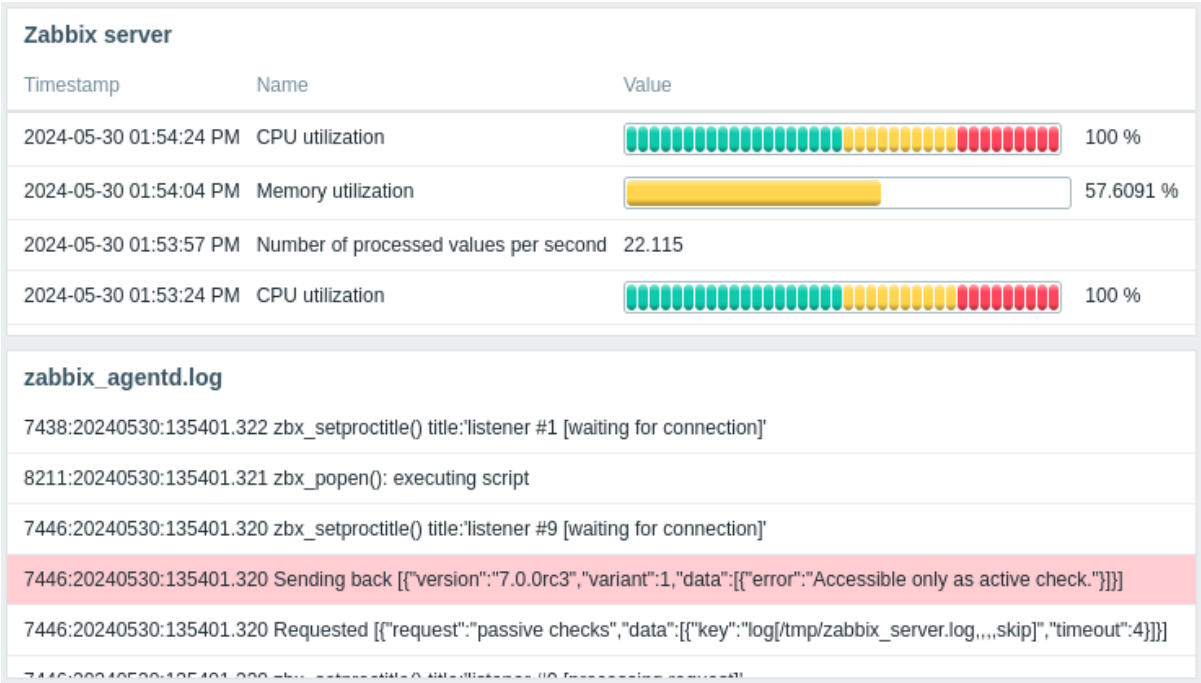
Larghezza	Imposta lo spessore della linea del grafico utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Colore	Seleziona il colore della linea e del riempimento.
Riempimento	Imposta il livello di trasparenza del colore di riempimento utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Periodo di tempo	Seleziona l'origine dati per il periodo di tempo: **Dashboard** - usa il selettore del periodo di tempo della dashboard; **Widget** - usa un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - usa un periodo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che, al passaggio del mouse, indicherà l'ora impostata. Si noti che, indipendentemente dalla configurazione di *Periodo di tempo* del widget, i widget compatibili possono comunque usarlo come origine dati per il periodo di tempo.

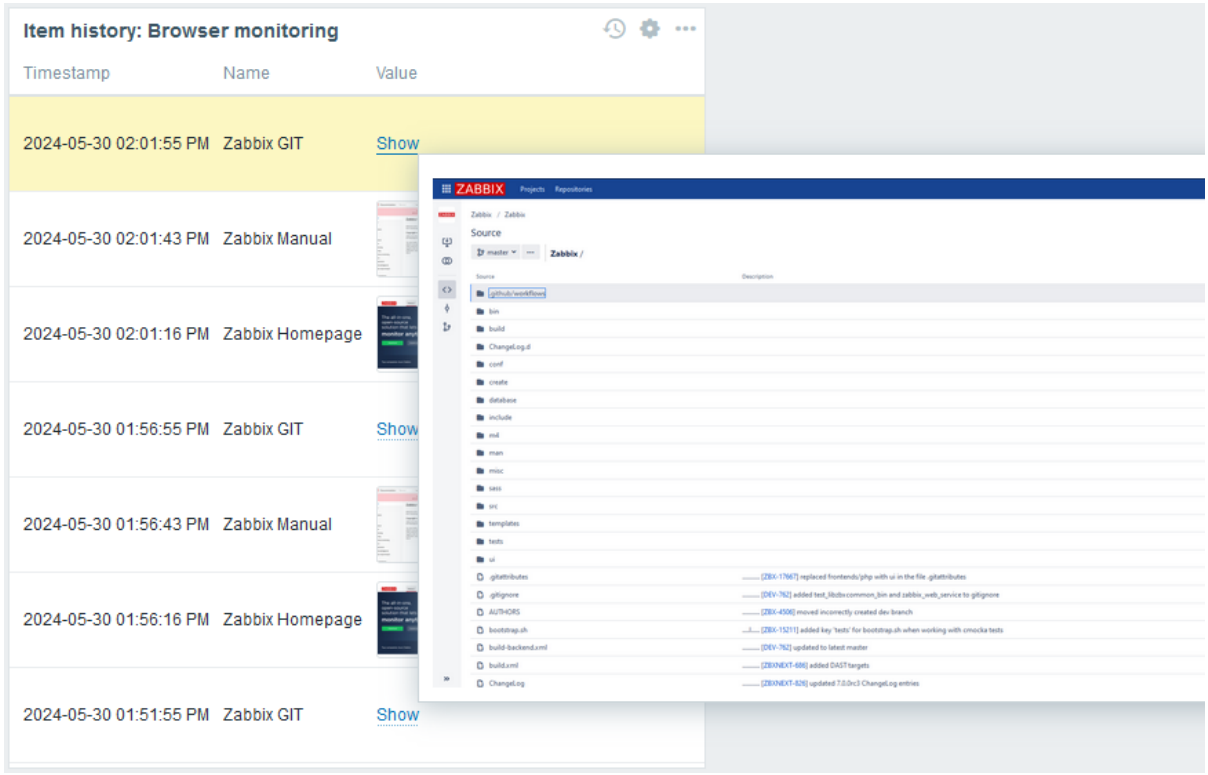
Widget	Inserisci o seleziona un widget compatibile come origine dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Inserisci o seleziona l'inizio del periodo di tempo. La sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.) è supportata. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
A	Inserisci o seleziona la fine del periodo di tempo. La sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.) è supportata. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
Dati storici	Preleva i dati dalla cronologia o dai trend: **Auto** - selezione automatica; **Cronologia** - preleva i dati dalla cronologia; **Trend** - preleva i dati dai trend. Questo parametro si applica solo ai dati numerici. I dati non numerici verranno sempre prelevati dalla cronologia.

Cronologia dell'item

Panoramica

Il widget *Item history* visualizza i dati più recenti per tutti i tipi di item (numerici, testuali, ecc.) in formato tabellare. Può mostrare barre di avanzamento e immagini (utili per browser items) ed evidenziare i valori (utile per il monitoraggio dei file di log).

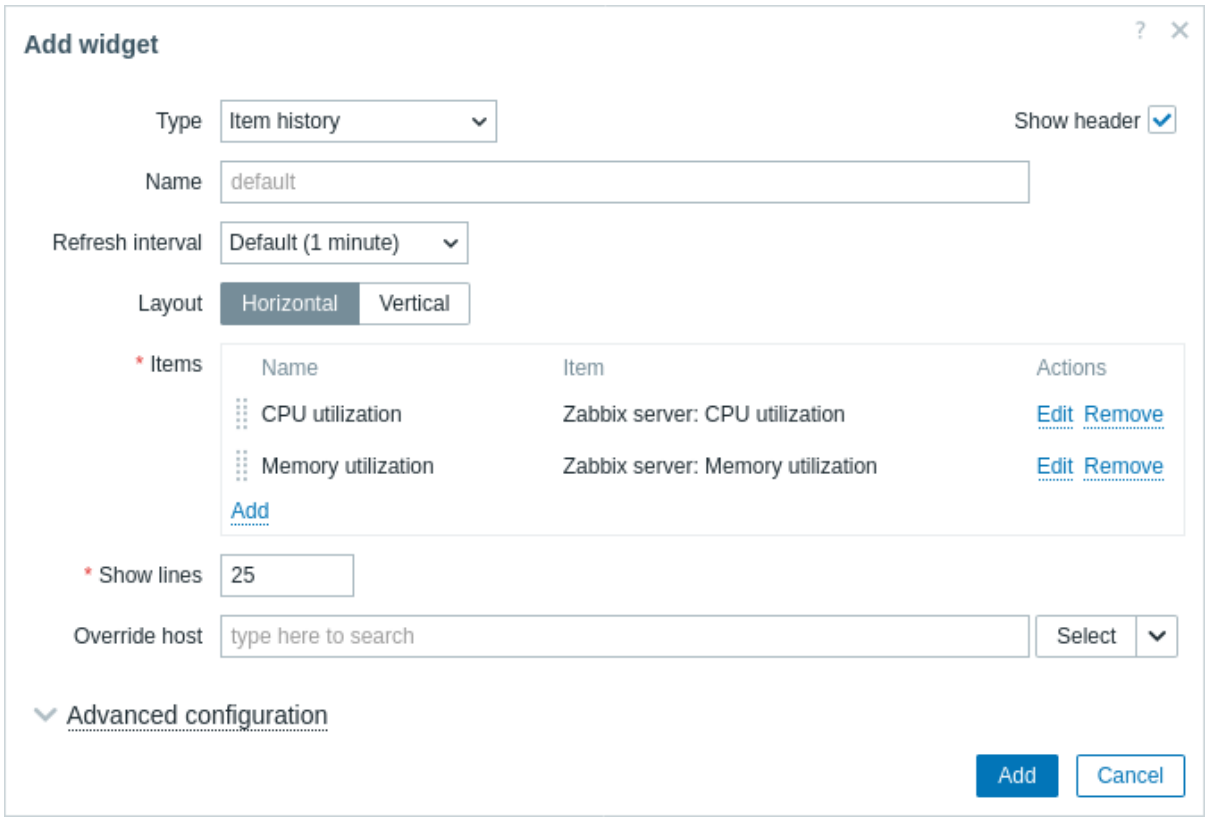




È possibile visualizzare fino a 1000 record.

Configurazione

Per configurare il widget, selezionare *Cronologia item* come tipo:



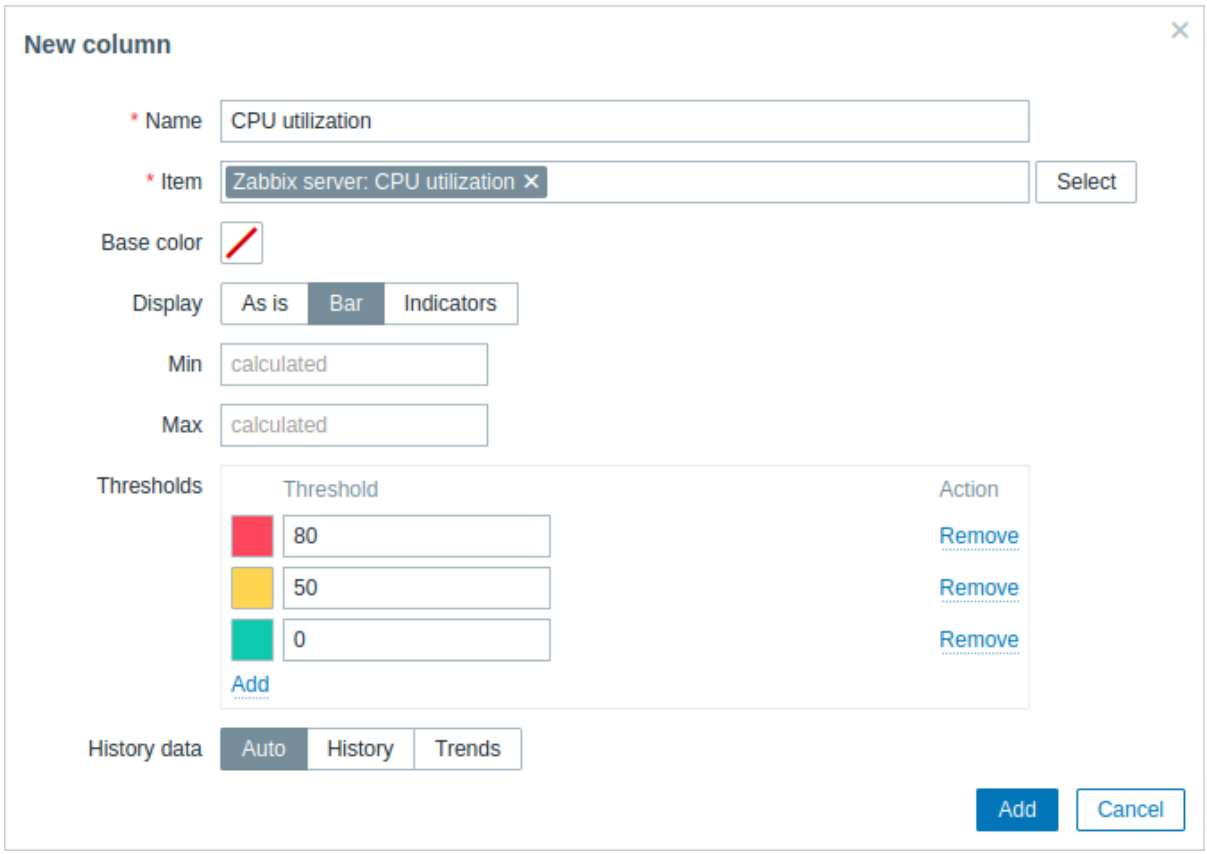
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Layout	Selezionare l'opzione di layout per le colonne degli item: **Orizzontale** - gli item verranno visualizzati orizzontalmente, i valori verticalmente; **Verticale** - gli item verranno visualizzati verticalmente, i valori orizzontalmente.

Items	Aggiungere le colonne degli item da visualizzare. L'ordine degli item determina il loro ordine di visualizzazione. Gli item possono essere riordinati trascinandoli verso l'alto o verso il basso tramite la maniglia posta prima del nome dell'item.
Show lines	Specificare il numero di righe dei valori degli item da visualizzare.
Override host	Selezionare un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Advanced configuration	Fare clic sull'etichetta *Configurazione avanzata* per visualizzare le opzioni di configurazione avanzata.

Configurazione delle colonne

Per configurare le colonne degli item, fare clic su *Aggiungi* nel parametro *Item*:



Parametri comuni delle colonne:

Nome	Inserire il nome della colonna. Se lasciato vuoto, viene utilizzato il nome dell'item dal parametro *Item*.
Item	Selezionare l'item. Si noti che i parametri di configurazione della colonna variano in base al tipo di informazione dell'item selezionato; per ulteriori informazioni, vedere i singoli parametri di seguito. Quando si configura il widget in una dashboard del template, è possibile selezionare solo gli item configurati nel template.
Colore di base	Selezionare il colore di sfondo della colonna o il colore di riempimento se *Display* è impostato su "Bar" o "Indicators". Si noti che il colore di base può essere sostituito dai colori di soglia o di evidenziazione.

Parametri di colonna specifici per item di tipo numerico:

Visualizzazione	Selezionare come deve essere visualizzato il valore dell'item: **Così com'è** - come testo normale; **Barra** - come barra piena colorata; **Indicatori** - come barra segmentata colorata.

Min	Inserire il valore minimo per barra/indicatori. Se lasciato vuoto, il widget utilizzerà il valore minimo dell'item. Questo parametro è disponibile solo quando *Visualizzazione* è impostato su "Barra" o "Indicatori".
Max	Inserire il valore massimo per barra/indicatori. Se lasciato vuoto, il widget utilizzerà il valore massimo dell'item. Questo parametro è disponibile solo quando *Visualizzazione* è impostato su "Barra" o "Indicatori".
Soglie	Fare clic su *Aggiungi* per aggiungere una soglia, selezionare un colore per la soglia dal selettore colori e specificare un valore numerico. L'elenco delle soglie verrà ordinato in ordine crescente al momento del salvataggio. I suffissi (ad esempio, "1d", "2w", "4K", "8G") sono supportati. Le mappature dei valori sono supportate.
Dati cronologici	Selezionare se prendere i dati dalla cronologia o dai trend: **Auto** - selezione automatica; **Cronologia** - usa i dati della cronologia; **Trend** - usa i dati dei trend.

Parametri di colonna specifici per item di tipo carattere, testo, log e JSON:

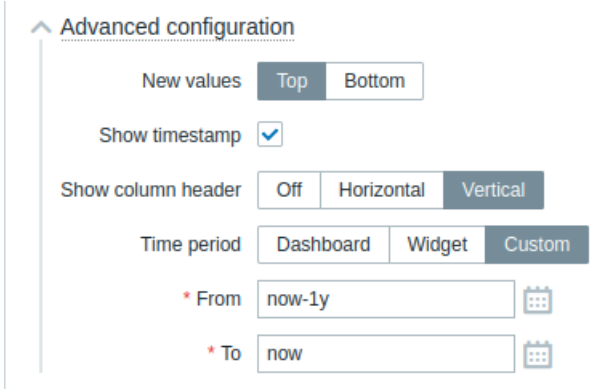
Highlights	Fare clic su *Add* per aggiungere un highlight, selezionare un colore di highlight dal selettore colori e specificare un'espressione regolare. Il colore selezionato verrà usato come colore di sfondo per i valori dell'item in cui l'espressione regolare specificata corrisponde al testo.
Display	Selezionare come deve essere visualizzato il valore dell'item: **As is** - esattamente come ricevuto. Le interruzioni di riga vengono mantenute se il valore contiene ritorni a capo. Il word wrapping viene applicato se *Layout* è impostato su "Vertical" e sono configurate più di una colonna; **HTML** - come testo formattato in HTML; **Single line** - come singola riga, troncata a una lunghezza specificata (1-500 caratteri). Passando il mouse o facendo clic sul valore troncato si apre un pop-up con il valore completo.
Use monospace font	Selezionare questa casella di controllo per visualizzare il valore dell'item con un font monospace (non selezionata per impostazione predefinita).
Display local time	Selezionare questa casella di controllo per visualizzare l'ora locale invece del timestamp nella colonna del timestamp. Si noti che anche la casella di controllo *Show timestamp* nella configurazione avanzata deve essere selezionata. Questo parametro è disponibile solo per item di tipo log.

Parametri della colonna specifici per gli item di tipo binario:

Mostra miniatura	Selezionare questa casella di controllo per visualizzare una miniatura per i file binari immagine oppure l'opzione "Mostra" per i file binari non immagine. Deselezionare questa casella di controllo per visualizzare l'opzione "Mostra" per tutti i valori degli item binari. Passando il cursore sopra o facendo clic sull'opzione "Mostra" si apre una finestra pop-up con il valore dell'item (immagine o stringa Base64). Se il valore dell'item è una stringa vuota, viene visualizzata l'opzione "Mostra"; passandoci sopra con il cursore o facendovi clic si apre un pop-up contenente "Stringa vuota".

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata*:

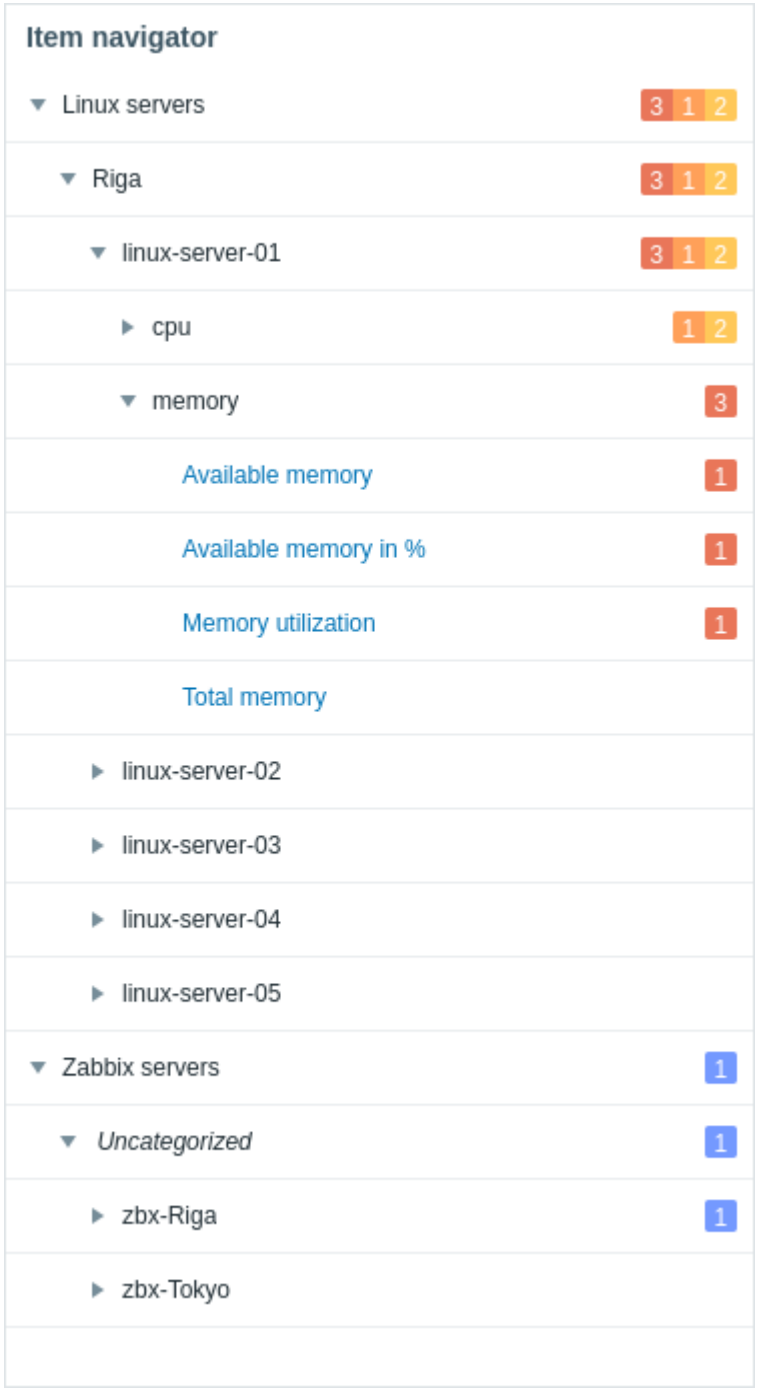


Nuovi valori	Selezionare dove devono essere aggiunti i nuovi valori degli item: **In alto** - nella parte superiore delle colonne; **In basso** - nella parte inferiore delle colonne.
Mostra timestamp	Selezionare questa casella per visualizzare la colonna del timestamp (non selezionata per impostazione predefinita).
Mostra intestazione colonna	Selezionare l'orientamento dell'intestazione della colonna: **Disattivato** - nasconde l'intestazione; **Orizzontale** - visualizza l'intestazione orizzontalmente; **Verticale** - visualizza l'intestazione verticalmente.
Periodo di tempo	Selezionare l'origine dati per il periodo di tempo: **Dashboard** - utilizza il selettore del periodo di tempo della dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizza un periodo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che, al passaggio del mouse, indicherà l'ora impostata. Si noti che, indipendentemente dalla configurazione di *Periodo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come origine dati per il periodo di tempo.
Widget	Immettere o selezionare un widget compatibile (*Graph*, *Graph (classic)*, *Graph prototype*) come origine dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Immettere o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
A	Immettere o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".

Navigatore degli item

Panoramica

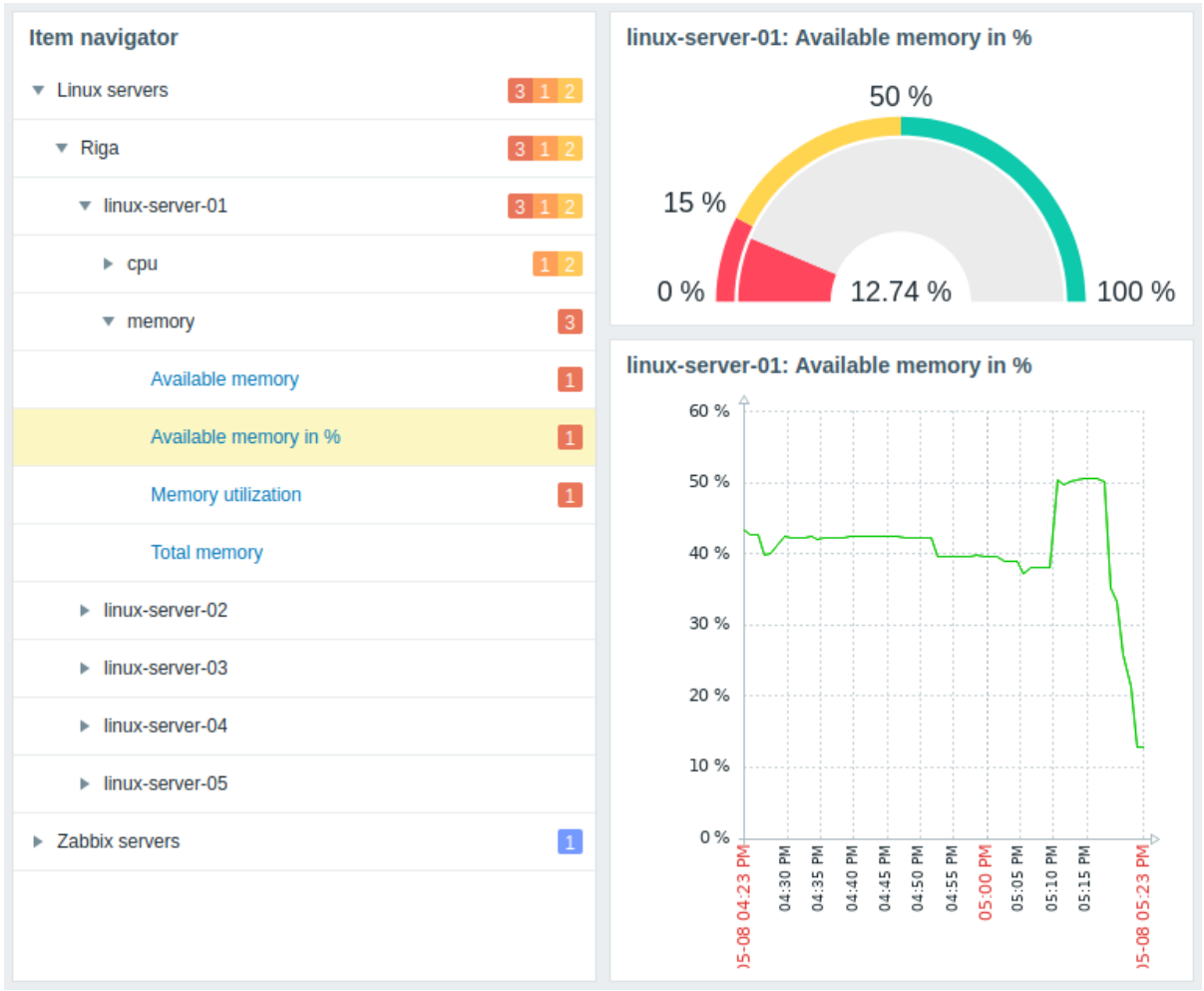
Il widget *Navigatore item* visualizza un elenco di item in base a varie opzioni di filtro e raggruppamento.



I gruppi in base ai quali gli item sono organizzati possono essere espansi o compressi.

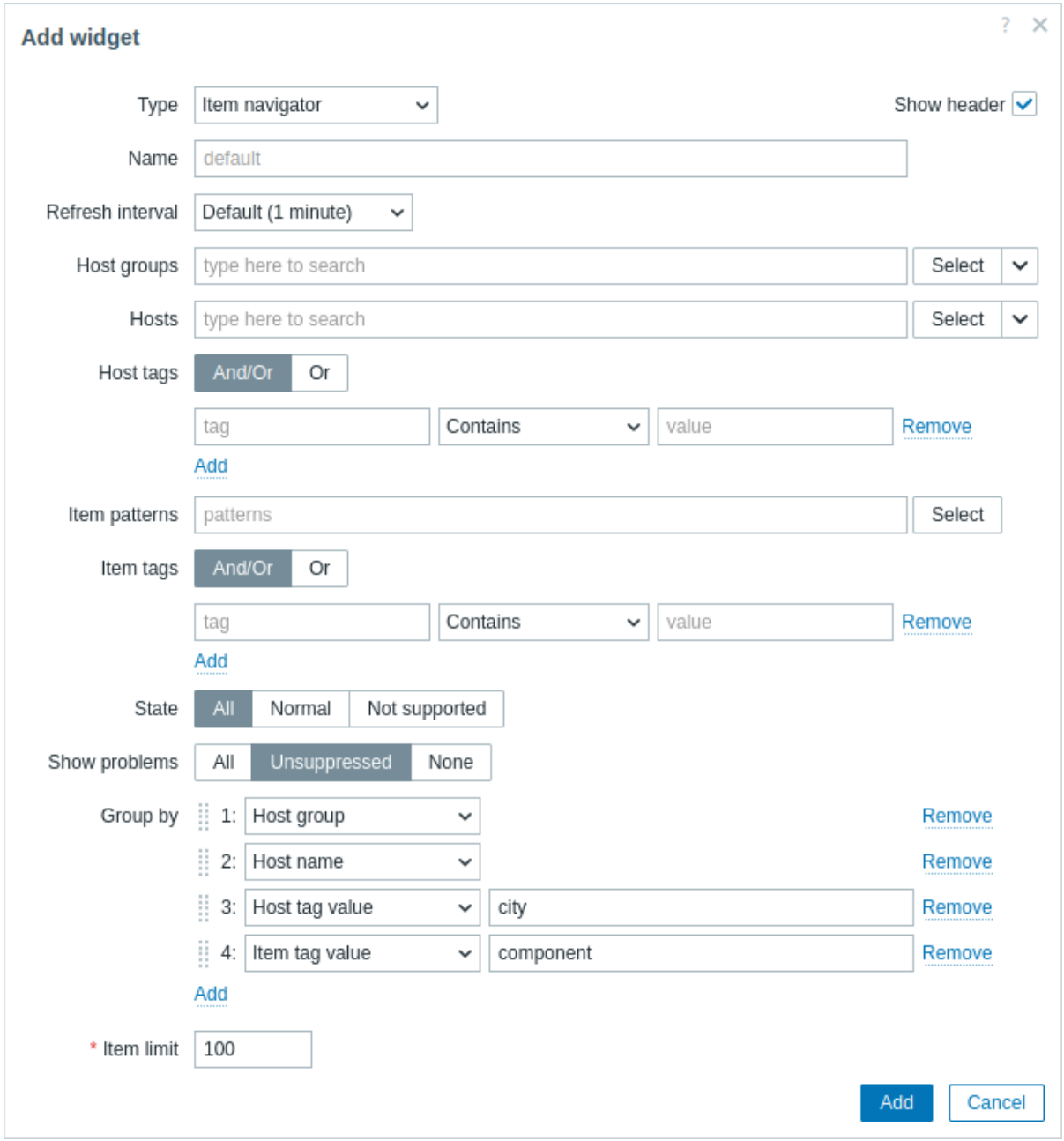
Per i gruppi e i problemi, ulteriori dettagli sono accessibili passando il mouse sui suggerimenti.

Il widget è particolarmente utile per controllare ciò che altri widget visualizzano in base all'item selezionato.



Configurazione

Per configurare, selezionare *Item navigator* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Host groups	Selezionare i gruppi di host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico: iniziando a digitare il nome di un gruppo, verrà visualizzato un elenco a discesa dei gruppi corrispondenti. La selezione di un gruppo di host padre seleziona implicitamente tutti i gruppi di host annidati; se non viene selezionato alcun gruppo di host, il widget visualizzerà gli item appartenenti a tutti gli host di tutti i gruppi di host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Hosts	Selezionare gli host. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico: iniziando a digitare il nome di un host, verrà visualizzato un elenco a discesa degli host corrispondenti. Se non viene selezionato alcun host, il widget visualizzerà gli item appartenenti a tutti gli host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.

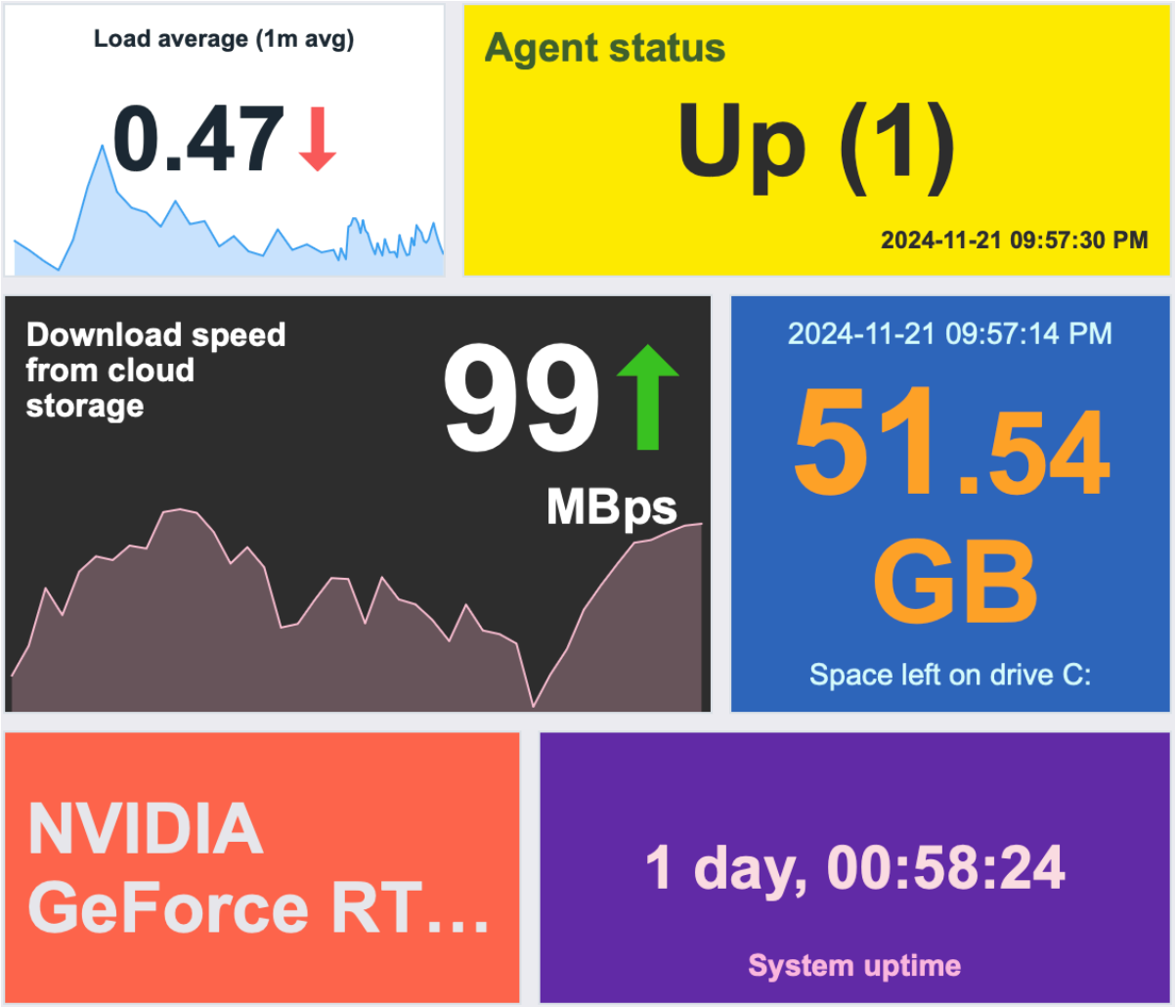
Host tags	Specificare i tag degli host per filtrare gli item visualizzati nel widget. È possibile includere o escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - devono essere soddisfatte tutte le condizioni; le condizioni con lo stesso nome di tag vengono raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione.
Item patterns	Questo parametro non è disponibile quando si configura il widget in una dashboard di template. Immettere i pattern degli item o selezionare item esistenti come pattern degli item. Gli item che corrispondono ai pattern specificati verranno visualizzati nell'item navigator. È possibile utilizzare pattern con caratteri jolly per la selezione (ad esempio, `*` restituirà gli item che corrispondono a zero o più caratteri; `Zabbix*` restituirà gli item che iniziano con "Zabbix"). Per specificare un pattern con caratteri jolly, immettere manualmente la stringa e premere *Enter*. Quando si inizia a digitare, un elenco a discesa mostrerà gli item corrispondenti, limitati a quelli appartenenti agli *Hosts* selezionati o agli host inclusi nei *Host groups* selezionati, se presenti. Il simbolo jolly viene sempre interpretato; pertanto, non è possibile aggiungere singolarmente, ad esempio, un item denominato *item**, se sono presenti altri item corrispondenti (ad esempio, item2, item3). Quando si configura il widget in una dashboard di template, questo parametro consente di selezionare solo gli item configurati nel template.
Item tags	Specificare i tag degli item per filtrare gli item visualizzati nel widget. È possibile includere o escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - devono essere soddisfatte tutte le condizioni; le condizioni con lo stesso nome di tag vengono raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione.
State	Filtrare gli item da visualizzare in base al loro stato (tutti, normale, non supportato).
Show problems	Filtrare i problemi da visualizzare in base al loro stato (tutti, non soppressi, nessuno). I problemi sono indicati da blocchi colorati alla fine della riga dell'item. I colori si basano sulla gravità del problema, che può essere personalizzata in *Administration* > *General* > *Trigger displaying options*. La gravità dei problemi può essere modificata durante l'aggiornamento dei problemi.

Group by	Aggiungere un attributo di raggruppamento in base al quale raggruppare gli item: **Host group** - raggruppa gli item in base ai gruppi di host dei relativi host; **Host name** - raggruppa gli item in base ai relativi host; **Host tag value** - immettere un nome di tag per raggruppare gli item in base ai valori di questo tag dell'host (ad esempio, immettere "city" per raggruppare gli item in base ai valori "Riga", "Tokyo" e così via); **Item tag value** - immettere un nome di tag per raggruppare gli item in base ai valori di questo tag dell'item (ad esempio, immettere "component" per raggruppare gli item in base ai valori "cpu", "memory" e così via). Gli attributi di raggruppamento possono essere riordinati trascinandoli verso l'alto o verso il basso tramite la maniglia prima del nome del gruppo. L'ordine degli attributi di raggruppamento determina l'ordine di annidamento dei gruppi. Ad esempio, specificando più nomi di tag dell'host (1: color, 2: city), gli item verranno raggruppati per color (red, blue e così via) e quindi per city (Riga, Tokyo e così via). Un item può essere visualizzato in più gruppi in base agli attributi di raggruppamento configurati (ad esempio, quando si raggruppa per gruppo di host e l'host dell'item appartiene a più gruppi di host). Facendo clic su tali item, questi vengono selezionati ed evidenziati in tutti i gruppi. Gli item che non corrispondono agli attributi di raggruppamento configurati vengono visualizzati nel gruppo *Uncategorized*. Se *Show problems* è configurato per visualizzare i problemi, questi vengono visualizzati come segue: - per ogni gruppo, viene visualizzato il conteggio di tutti i problemi degli item subordinati; - per ogni item, viene visualizzato solo il relativo conteggio dei problemi. È possibile specificare fino a 10 attributi di raggruppamento, che devono essere tutti univoci. Se non viene specificato alcun attributo di raggruppamento, gli item non verranno raggruppati.
Item limit	Immettere il numero massimo di item da visualizzare. I valori possibili vanno da 1 a 9999. Quando sono disponibili per la visualizzazione più item rispetto al limite impostato, sotto gli item visualizzati viene mostrato un messaggio corrispondente (ad esempio, "100 of 100+ items are shown"). Il limite di item configurato influisce anche sulla visualizzazione dei gruppi configurati; ad esempio, se il limite di item è impostato su 100 e gli item sono raggruppati per host (ciascuno contenente 200 item), nel widget verrà visualizzato solo il primo host con i suoi 100 item. Questo parametro non è influenzato dal parametro *Limit for search and filter results* in *Administration* > *General* > *GUI*.

Valore dell'item

Panoramica

Il widget *Item value* visualizza il valore di un singolo item numerico o di stringa. È utile per tenere sotto controllo metriche importanti, visualizzare soglie e rilevare cambiamenti improvvisi nei dati.



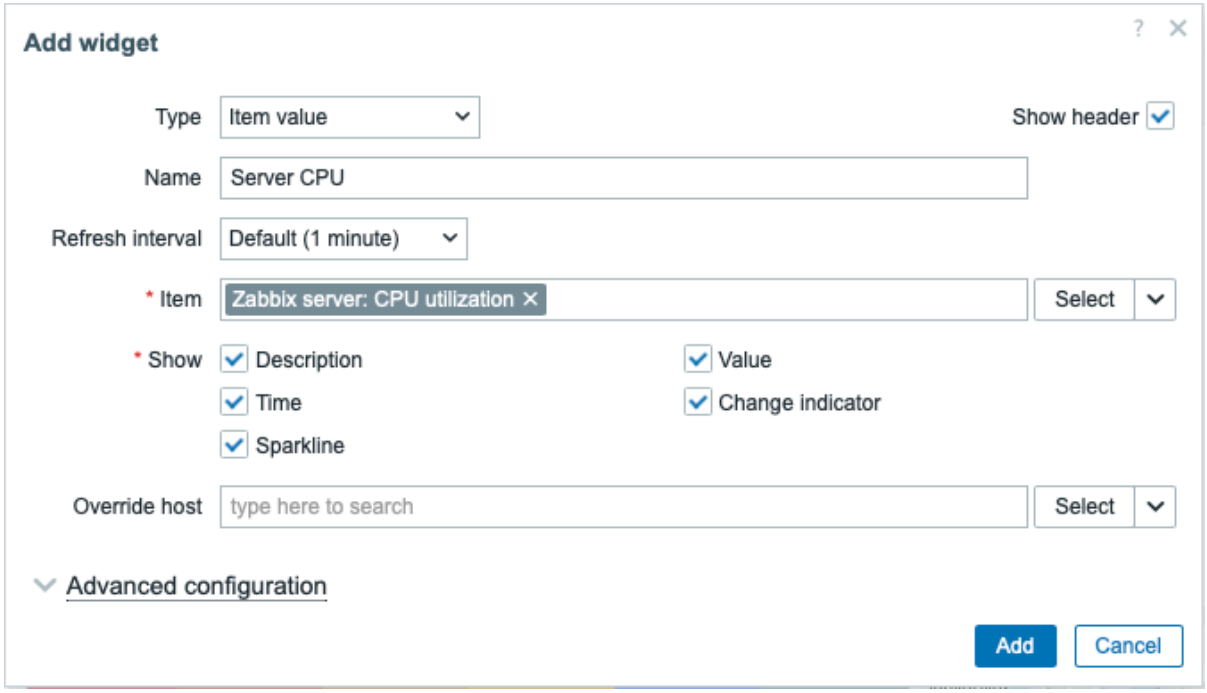
È possibile configurare il widget per visualizzare quanto segue:

- Timestamp della metrica (2024-11-21 09:57:30)
- Descrizione dell'item (Load average (1m), Agent status)
- Valore dell'item, indicatore di variazione e unità (0.47 ↓, 99 ↑ Mbps)
- Colore di sfondo
- Grafico sparkline per i valori del periodo di tempo specificato

Facendo clic sul widget si apre un grafico ad hoc per gli item numerici oppure i dati più recenti per gli item di stringa.

Configurazione

Per configurare, selezionare *Valore item* come tipo di widget:



Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Item	Selezionare l'item. In alternativa, selezionare un widget compatibile come origine dati per l'item. Gli item che restituiscono dati binari non sono supportati.
Mostra	Selezionare la casella di controllo per visualizzare il rispettivo elemento: descrizione, valore, ora, indicatore di variazione, sparkline. Deselezionarla per nasconderlo. Deve essere selezionato almeno un elemento.
Sostituisci host	Selezionare un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard template.
Configurazione avanzata	Fare clic sull'intestazione *Configurazione avanzata* per visualizzare le opzioni di configurazione avanzata.

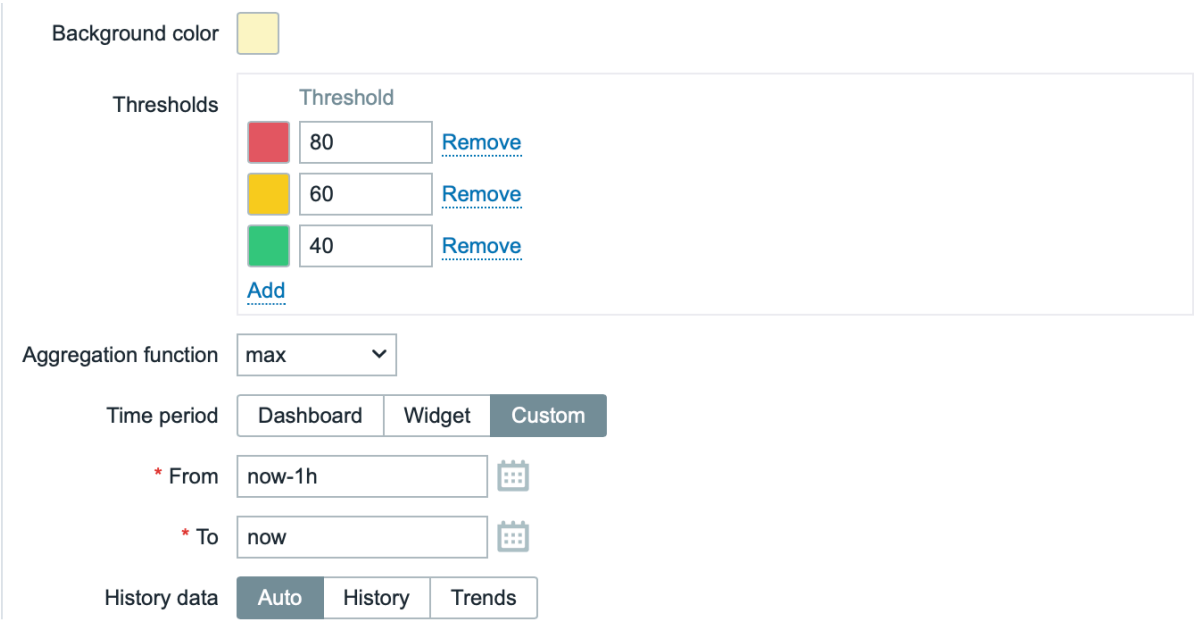
Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata*, e solo per gli elementi selezionati nel campo *Mostra* (vedi sopra).

Si noti che più elementi non possono occupare lo stesso spazio; se vengono posizionati nello stesso spazio, verrà visualizzato un messaggio di errore.

Parametri generici del widget

Questi parametri determinano il colore di sfondo (statico o dinamico) dell'intero widget e una funzione di aggregazione per la visualizzazione dei valori.

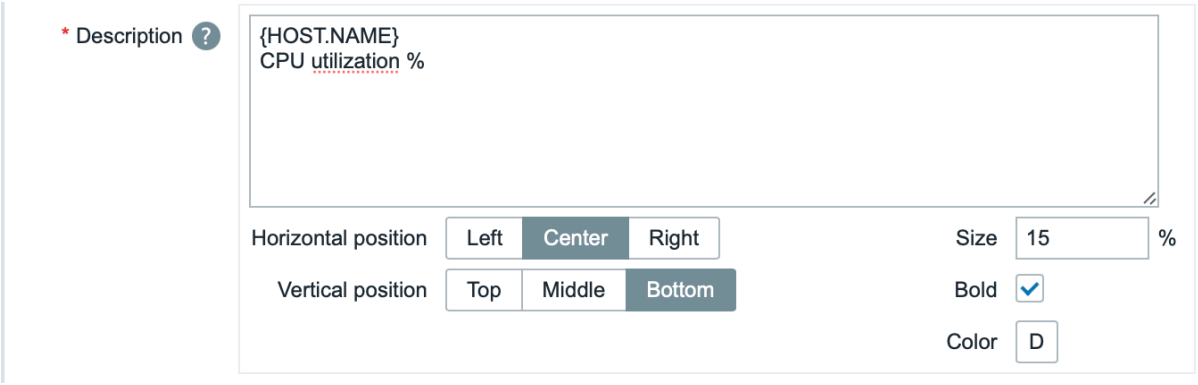


Colore di sfondo	Selezionare il colore di sfondo dell'intero widget dal selettore dei colori. D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore dei colori.
Soglie	Configurare il colore di sfondo dinamico dell'intero widget. Fare clic su *Aggiungi* per aggiungere una soglia, selezionare il colore di sfondo dal selettore dei colori e specificare un valore numerico. Quando il valore dell'item è uguale o superiore al valore della soglia, il colore di sfondo cambierà. L'elenco verrà ordinato in ordine crescente al momento del salvataggio. Il colore di sfondo dinamico viene visualizzato correttamente solo per gli item numerici.
Funzione di aggregazione	Specificare la funzione di aggregazione da utilizzare nell'*Intervallo di tempo* selezionato: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **count** - visualizza il numero di valori; **sum** - visualizza la somma dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore. Se è selezionato **non utilizzata**, viene visualizzato il valore più recente (senza aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo specificato (5 minuti, un'ora, un giorno), invece del valore più recente. Solo i dati numerici possono essere visualizzati per *min*, *max*, *avg* e *sum*. Per *count*, i dati non numerici verranno convertiti in numerici.
Intervallo di tempo	Intervallo di tempo da utilizzare per aggregare i valori. Selezionare la sorgente dati per l'intervallo di tempo: **Dashboard** - utilizza il selettore dell'intervallo di tempo del dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizza un intervallo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio, che al passaggio del mouse indica l'ora impostata. Indipendentemente dalla configurazione dell'*Intervallo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per l'intervallo di tempo. Questo parametro è nascosto quando per *Funzione di aggregazione* è impostato "non utilizzata".
Widget	Inserire o selezionare un widget compatibile come sorgente dati per l'intervallo di tempo. Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Widget".
Da	Inserire o selezionare l'inizio dell'intervallo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Personalizzato".
A	Inserire o selezionare la fine dell'intervallo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Personalizzato".

Dati della cronologia	Prelevare i dati dalla cronologia o dai trend: **Automatico** - selezione automatica; **Cronologia** - preleva i dati della cronologia; **Trend** - preleva i dati dei trend. Questa impostazione si applica solo ai dati numerici. I dati non numerici verranno sempre prelevati dalla cronologia.

Descrizione

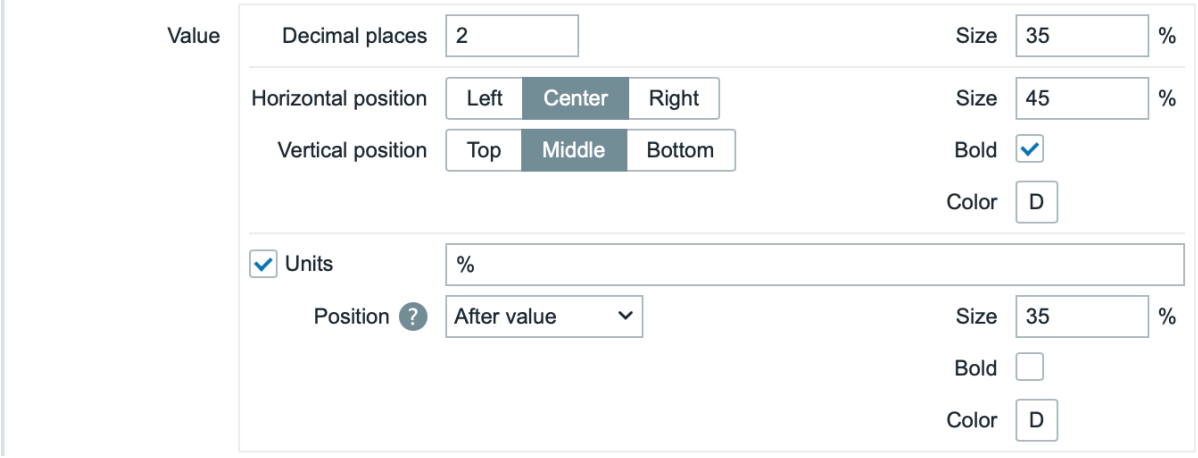
Questi parametri determinano come deve essere visualizzata la descrizione dell'item.



Descrizione	Inserire la descrizione dell'item. Questa descrizione può sostituire il nome predefinito dell'item. Sono supportate descrizioni su più righe. È possibile utilizzare una combinazione di testo e macro supportate. {HOST.*}, {ITEM.*}, {INVENTORY.*} e le macro utente sono supportate.
Posizione orizzontale	Selezionare la posizione orizzontale della descrizione dell'item: a sinistra, a destra o al centro.
Posizione verticale	Selezionare la posizione verticale della descrizione dell'item: in alto, in basso o al centro.
Dimensione	Inserire l'altezza della dimensione del carattere per la descrizione dell'item (in percentuale rispetto all'altezza totale del widget).
Grassetto	Selezionare la casella di controllo per visualizzare la descrizione dell'item in grassetto.
Colore	Selezionare il colore della descrizione dell'item dal selettore colori. D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.

Valore

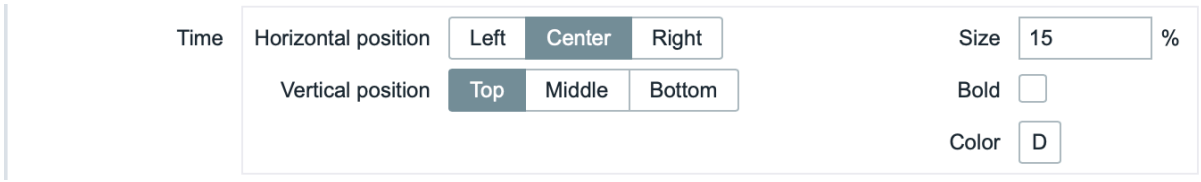
Questi parametri determinano come deve essere visualizzato il valore dell'item.



Posizioni decimali	Selezionare quante posizioni decimali verranno visualizzate con il valore. Questo valore influisce solo sugli item float. Per gli item con unità impostate su ”s”, se questo valore è maggiore di 0 (il valore predefinito è 2), il widget arrotonda l’unità di tempo più significativa e visualizza un valore numerico (ad esempio, ”10.43m”) invece di convertire il valore in una stringa temporale completa. Se impostato su 0, i secondi vengono convertiti in una stringa temporale leggibile (ad esempio, ”4h 56m 30s”).
Dimensione	Inserire l’altezza della dimensione del carattere per le posizioni decimali (in percentuale rispetto all’altezza totale del widget).
Posizione orizzontale	Selezionare la posizione orizzontale del valore dell’item: a sinistra, a destra o al centro.
Posizione verticale	Selezionare la posizione verticale del valore dell’item: in alto, in basso o al centro.
Dimensione	Inserire l’altezza della dimensione del carattere per il valore dell’item (in percentuale rispetto all’altezza totale del widget). Si noti che la dimensione del valore dell’item ha la priorità; gli altri elementi devono cedere spazio al valore. Tuttavia, con l’indicatore di variazione, se il valore è troppo grande, verrà troncato per mostrare l’indicatore di variazione.
Grassetto	Selezionare la casella di controllo per visualizzare il valore dell’item in grassetto.
Colore	Selezionare il colore del valore dell’item dal selettore colori. D indica il colore predefinito (dipende dal tema del frontend). Per tornare al valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Unità	
Unità	Selezionare la casella di controllo per visualizzare le unità con il valore dell’item. Se si inserisce un nome di unità, questo sovrascriverà l’unità della configurazione dell’item.
Posizione	Selezionare la posizione dell’unità dell’item: sopra, sotto, prima o dopo il valore.
Dimensione	Inserire l’altezza della dimensione del carattere per l’unità dell’item (in percentuale rispetto all’altezza totale del widget).
Grassetto	Selezionare la casella di controllo per visualizzare l’unità dell’item in grassetto.
Colore	Selezionare il colore dell’unità dell’item dal selettore colori. D indica il colore predefinito (dipende dal tema del frontend). Per tornare al valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.

Ora

Questi parametri determinano come deve essere visualizzata l’ora (valore del clock dalla cronologia dell’item).



Posizione orizzontale	Selezionare la posizione orizzontale dell’ora: a sinistra, a destra o al centro.
Posizione verticale	Selezionare la posizione verticale dell’ora: in alto, in basso o al centro.
Dimensione	Inserire l’altezza della dimensione del carattere per l’ora (in percentuale rispetto all’altezza totale del widget).
Grassetto	Selezionare la casella di controllo per visualizzare l’ora in grassetto.
Colore	Selezionare il colore dell’ora dal selettore colori. D indica il colore predefinito (dipende dal tema del frontend). Per tornare al valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.

Indicatore di variazione

La sezione consente di selezionare il colore degli indicatori di variazione dal selettore dei colori.



Gli indicatori di variazione sono i seguenti:

- ↑ - il valore dell’item è aumentato (per gli item numerici)
- ↓ - il valore dell’item è diminuito (per gli item numerici)
- ↕ - il valore dell’item è cambiato (per gli item stringa e gli item con mappatura dei valori)

Nota:

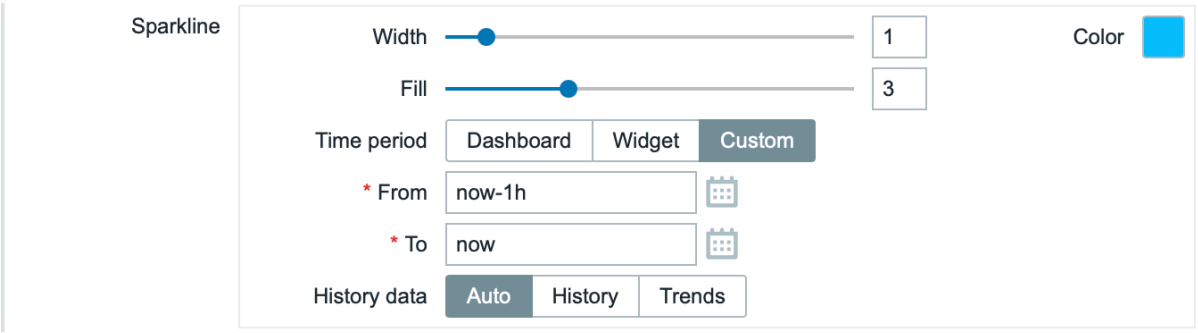
- Gli indicatori di aumento (↑) e diminuzione (↓) non vengono visualizzati se è presente un solo valore.
- I valori vengono sempre confrontati all'interno dello stesso ambito; ad esempio:
  - L'ultimo valore viene confrontato con il valore precedente.
  - Un valore mensile viene confrontato con quello del mese precedente.
  - Per le aggregazioni, il periodo precedente ha la stessa durata del periodo selezionato e termina immediatamente prima dell'inizio del periodo selezionato.

D indica il colore predefinito (dipende dal tema del frontend). Per ripristinare il valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore dei colori.

La dimensione verticale dell'indicatore di variazione è uguale alla dimensione del valore (la parte intera del valore per gli item numerici).

Sparkline

Questi parametri determinano come deve essere visualizzato il grafico sparkline.

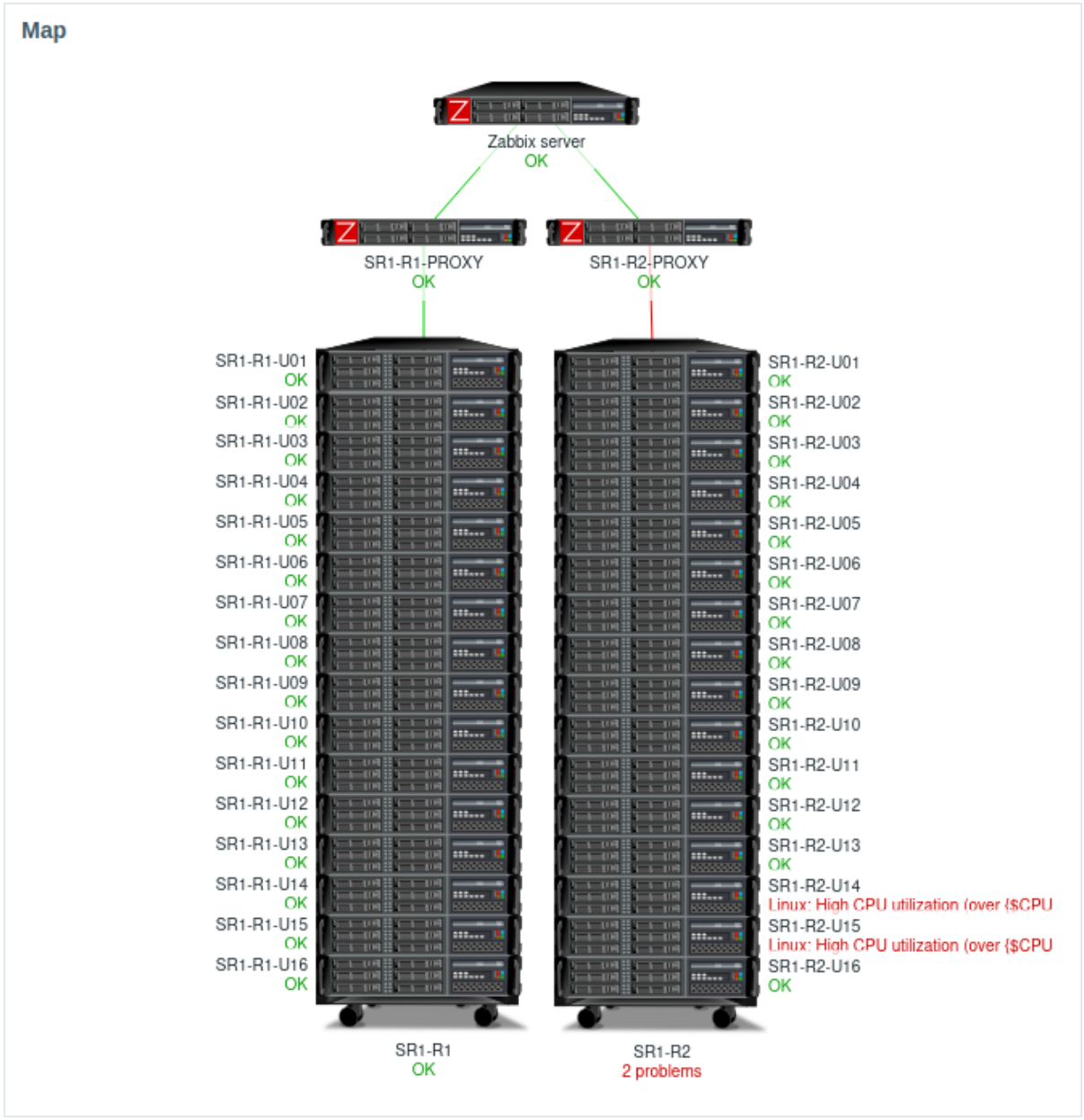


Larghezza	Imposta lo spessore della linea del grafico utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Colore	Seleziona il colore della linea e del riempimento.
Riempimento	Imposta il livello di trasparenza del colore di riempimento utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Periodo di tempo	Specifica il periodo di tempo per i valori da includere nel grafico sparkline. Seleziona l'origine dati per il periodo di tempo: **Dashboard** - utilizza il selettore del periodo di tempo della dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizza un periodo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che, al passaggio del mouse, indicherà l'ora impostata. Si noti che, indipendentemente dalla configurazione di *Periodo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come origine dati per il periodo di tempo. Questo parametro non è disponibile se *Funzione di aggregazione* è impostato su "non utilizzata".
Widget	Inserisci o seleziona un widget compatibile come origine dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Inserisci o seleziona l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
A	Inserisci o seleziona la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
Dati storici	Preleva i dati dalla cronologia o dai trend: **Auto** - selezione automatica; **Cronologia** - utilizza i dati della cronologia; **Trend** - utilizza i dati di trend.

Mappa

Panoramica

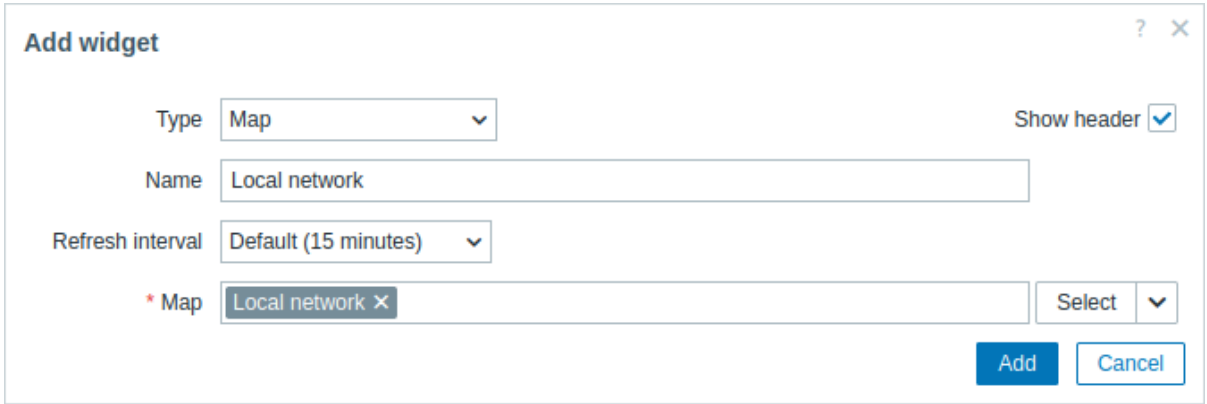
Il widget *Map* può visualizzare una mappa di rete o una visualizzazione simile, fornendoti una panoramica dinamica della tua rete.



Puoi abbinarlo al widget Map navigation tree per visualizzare dinamicamente la mappa selezionata nell'albero.

Configurazione

Per configurare, selezionare *Mappa* come tipo:



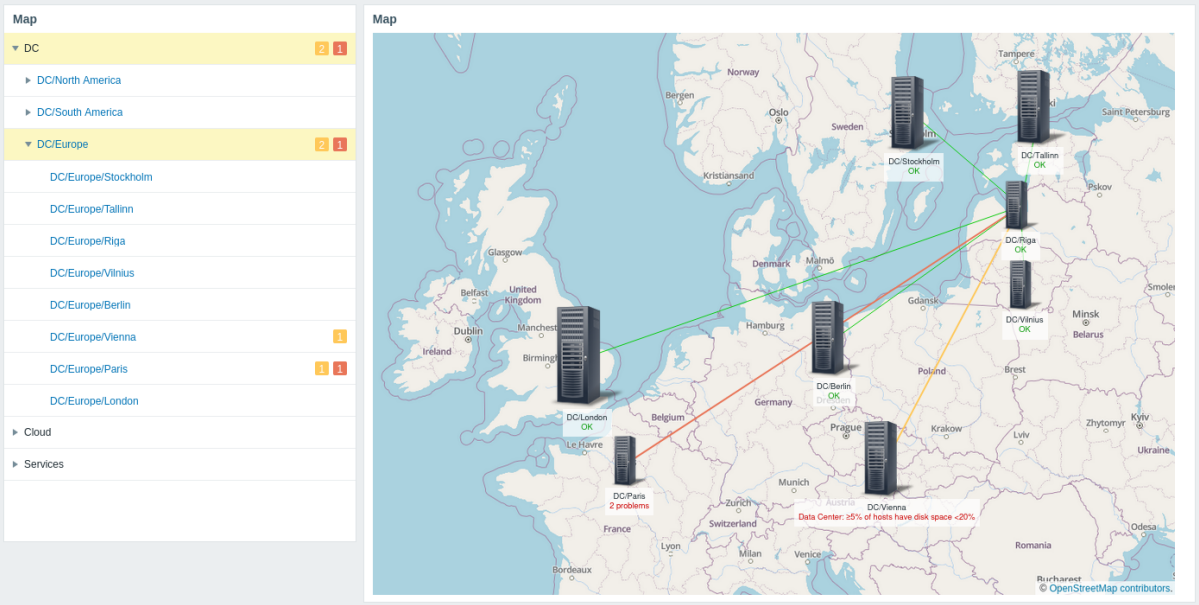
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Mappa	Impostare una mappa da visualizzare. In alternativa, selezionare un widget compatibile come origine dati per la mappa da visualizzare. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome della mappa o del widget verrà visualizzato un menu a discesa con le mappe o i widget corrispondenti.

Albero di navigazione della mappa

Panoramica

Il widget *Albero di navigazione mappe* visualizza una gerarchia delle mappe esistenti e il numero di problemi per ciascuna mappa e gruppo di mappe.

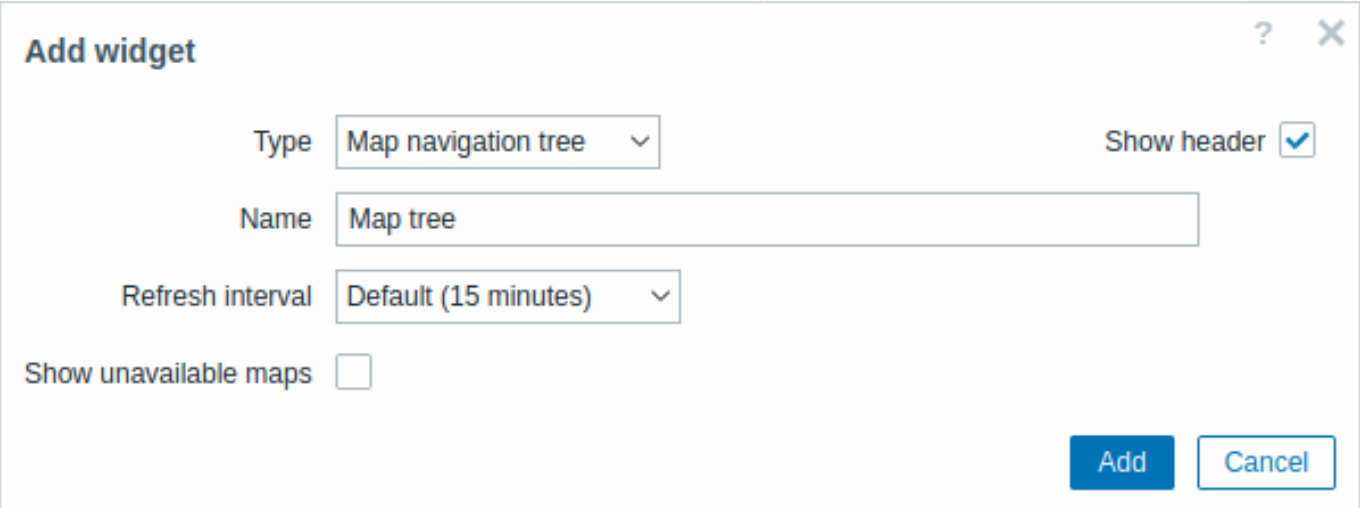


Puoi abbinarlo al widget Map per visualizzare dinamicamente la mappa selezionata nell'albero.

Il numero di problemi per la mappa di primo livello nella gerarchia include tutti i problemi delle sottomappe e i propri.

Configurazione

Per configurare, selezionare *Map navigation tree* come tipo:

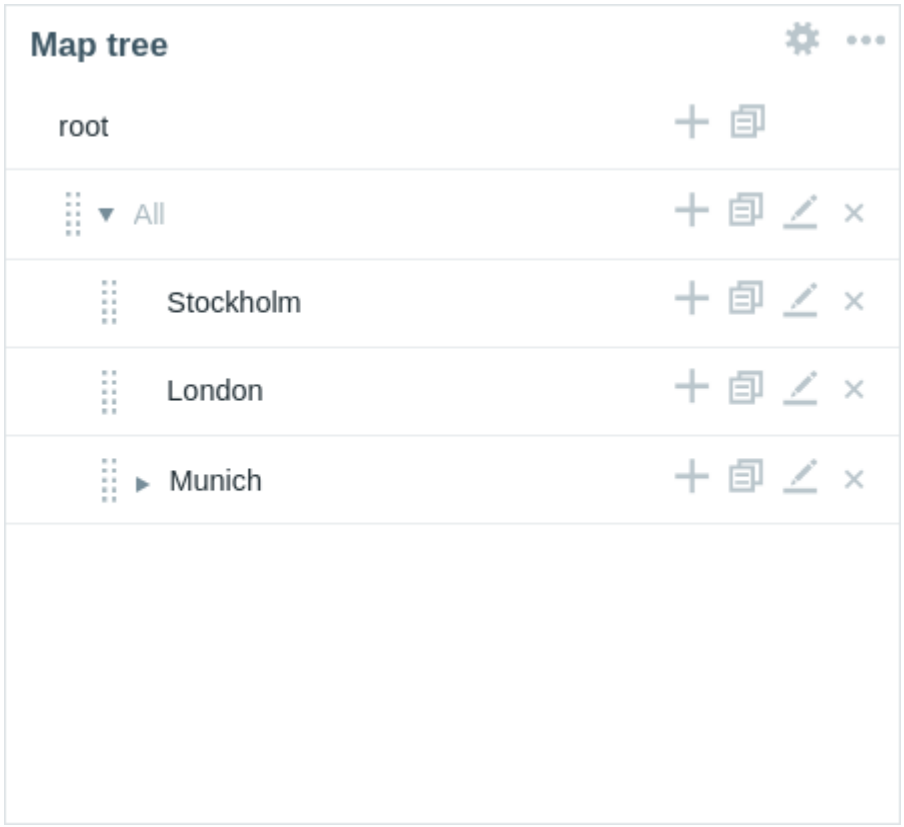


Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Mostra mappe non disponibili	Selezionare questa casella per visualizzare le mappe per cui l'utente non dispone dei permessi di lettura. Le mappe non disponibili nell'albero di navigazione verranno visualizzate con un'icona in grigio. Si noti che, se questa casella è selezionata, le submappe disponibili vengono visualizzate anche se la mappa di livello superiore non è disponibile. Se non è selezionata, le submappe disponibili di una mappa padre non disponibile non verranno visualizzate affatto. Il conteggio dei problemi viene calcolato in base alle mappe disponibili e agli elementi mappa disponibili.

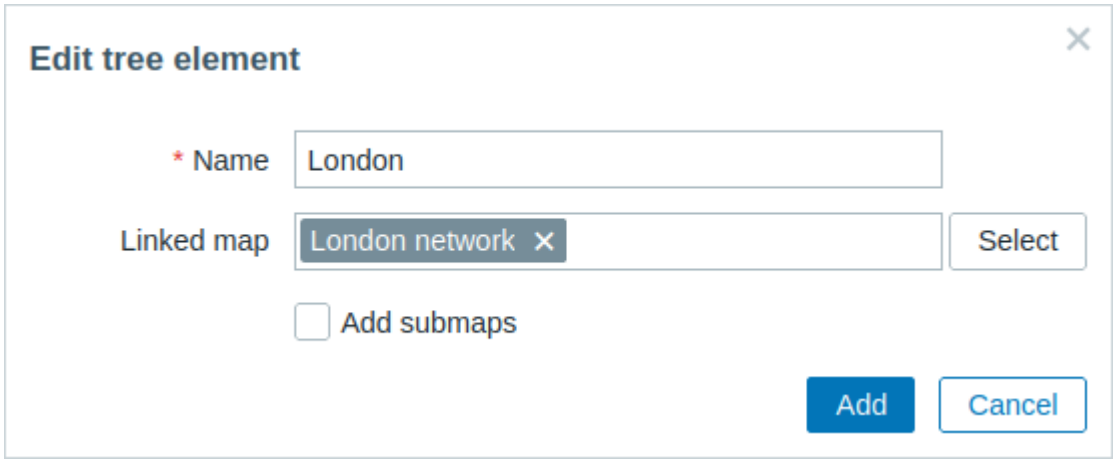
Gli elementi dell'albero di navigazione sono visualizzati in un elenco. È possibile:

- trascinare un elemento (inclusi i suoi elementi figli) in una nuova posizione nell'elenco;
- espandere o comprimere un elemento per visualizzare o nascondere i suoi elementi figli;
- aggiungere un elemento figlio (con o senza una mappa collegata) a un elemento;
- aggiungere più elementi figli (con mappe collegate) a un elemento;
- modificare un elemento;
- rimuovere un elemento (inclusi i suoi elementi figli).



Configurazione dell'elemento

To configure a navigation tree element, either add a new element or edit an existing element.



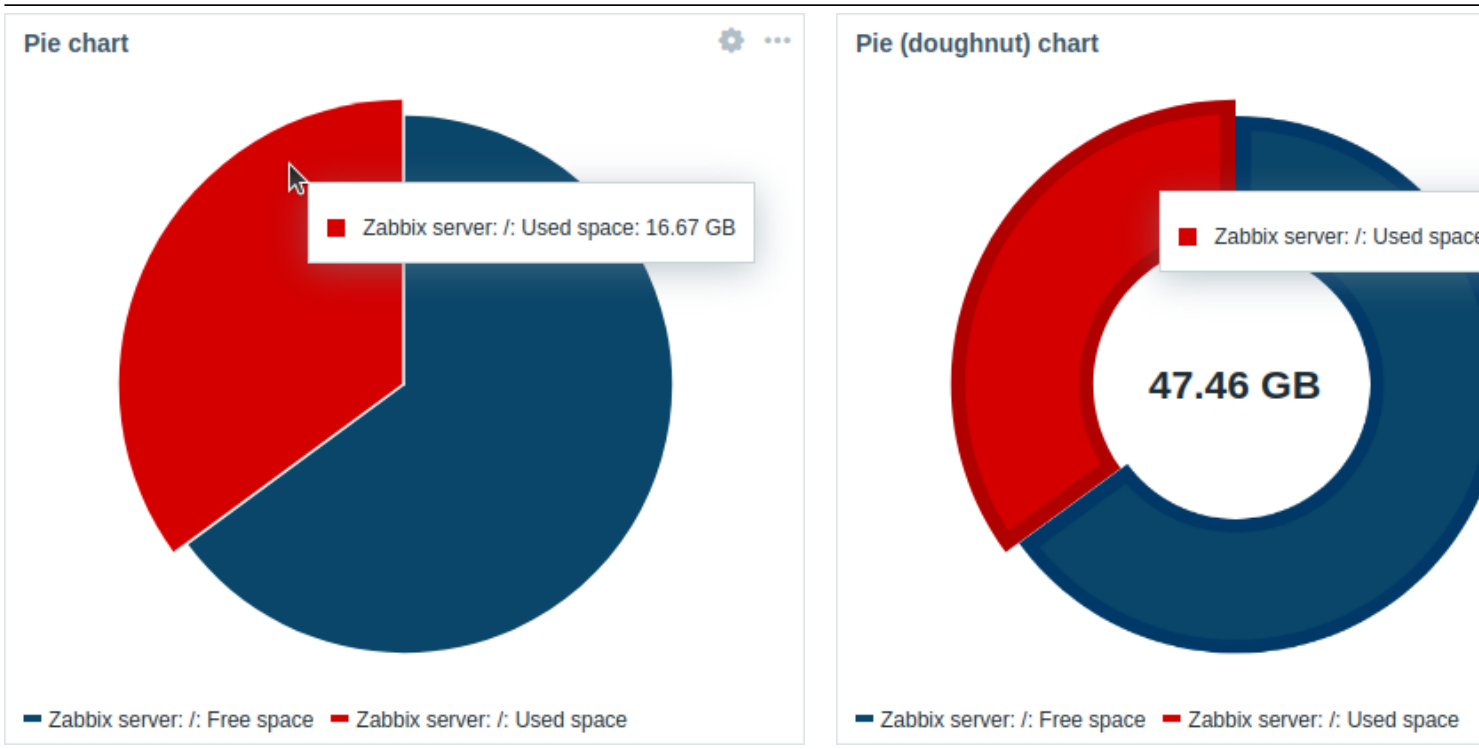
The following navigation tree element configuration parameters are available:

Name	Enter the navigation tree element name.
Linked map	Select the map to link to the navigation tree element. This field is auto-complete so starting to type the name of a map will offer a dropdown of matching maps.
Add submaps	Mark this checkbox to add the submaps of the linked map as child elements to the navigation tree element.

Grafico a torta

Panoramica

Il widget *Grafico a torta* visualizza i dati numerici degli item come grafico a torta o ad anello basato su vettori. Questo widget può aiutare a visualizzare il contributo degli item o degli host al dataset complessivo.

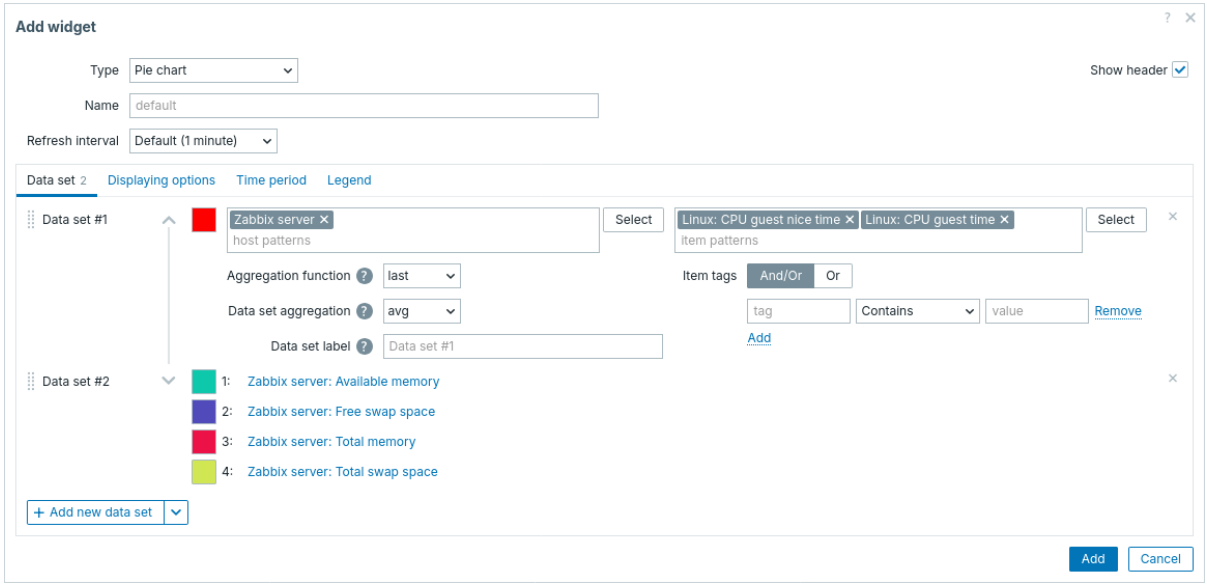


Al passaggio del mouse, il settore interessato si espande e ne mostra la legenda; facendo clic sul settore, questo rimane espanso e la relativa legenda resta visibile fino alla chiusura.

Le informazioni visualizzate nel widget *Grafico a torta* possono essere scaricate come immagine PNG selezionando l'opzione *Scarica immagine* nel menu *Azioni* del widget ⋯, visualizzato quando si passa il mouse sul widget. A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella `Downloads` oppure viene aperta una finestra di dialogo *Salva*, che consente di scegliere la posizione.

Configurazione

Per configurare, selezionare *Pie chart* come tipo:



Set di dati

La scheda **Set di dati** consente di selezionare i dati per il grafico a torta aggiungendo set di dati. È possibile aggiungere due tipi di set di dati:

- *Pattern degli item* - vengono visualizzati i dati degli item corrispondenti. È possibile scegliere un singolo colore di base o selezionare una riga della tavolozza per assegnare colori distinti a ogni item corrispondente.
- *Elenco di item* - vengono visualizzati i dati degli item selezionati. È possibile scegliere individualmente il colore di ogni item dal selettore.

Per impostazione predefinita, viene aggiunto un set di dati di tipo *Pattern degli item*.

Set di dati	Per il set di dati **Pattern degli item**: Selezionare o immettere i pattern di host e item; nel grafico a torta verranno visualizzati i dati degli item che corrispondono ai pattern immessi; è possibile visualizzare fino a 50 item. Per la selezione è possibile utilizzare pattern con caratteri jolly (ad esempio, ∗ restituisce i risultati che corrispondono a zero o più caratteri). Per specificare un pattern con caratteri jolly, immettere manualmente la stringa e premere *Invio*. Il simbolo jolly viene sempre interpretato, pertanto non è possibile aggiungere singolarmente, ad esempio, un item denominato *item** se sono presenti altri item corrispondenti (ad esempio, *item2*, *item3*). La specifica dei pattern di host e item è obbligatoria per i set di dati di tipo "Pattern degli item". Vedere anche: Dettagli sulla configurazione del set di dati. Per il set di dati **Elenco di item**: Selezionare gli item per il grafico a torta facendo clic sul pulsante *Aggiungi item*. Il menu a discesa *Tipo* dopo il nome dell'item consente di selezionare il tipo di visualizzazione per ogni item: **Normale** - il valore dell'item viene rappresentato proporzionalmente nel grafico a torta (impostazione predefinita); **Totale** - il valore dell'item occupa l'intero grafico a torta. È possibile avere un solo item "Totale" per grafico a torta e questo verrà posizionato per primo nella legenda del grafico a torta. Se un item è impostato su "Totale", il parametro *Aggregazione del set di dati* (vedere di seguito) verrà disabilitato e impostato su "non utilizzato". È inoltre possibile selezionare widget compatibili come origine dati per gli item facendo clic sul pulsante *Aggiungi widget*. Per i widget si applicano le stesse opzioni degli item singoli. La specifica di item o widget è obbligatoria per i set di dati di tipo "Elenco di item". Vedere anche: Dettagli sulla configurazione del set di dati. Tenere presente che sono consentiti solo tipi di item numerici. Quando si configura il widget in una dashboard di template, il parametro per specificare i pattern di host non è disponibile e il parametro per specificare un elenco di item consente di selezionare solo gli item configurati nel template.
Funzione di aggregazione	Specificare la funzione di aggregazione da utilizzare per ogni item nel set di dati nell'ambito del *Periodo di tempo* selezionato: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **sum** - visualizza la somma dei valori; **count** - visualizza il numero di valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore (impostazione predefinita). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo specificato (5 minuti, un'ora, un giorno). Per *min*, *max*, *avg* e *sum* è possibile visualizzare solo dati numerici. Per *count*, i dati non numerici verranno convertiti in numerici.
Aggregazione del set di dati	Specificare la funzione di aggregazione da utilizzare per l'intero set di dati: **non utilizzato** - nessuna aggregazione, gli item vengono visualizzati separatamente (impostazione predefinita); **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **sum** - visualizza la somma dei valori; **count** - visualizza il numero di valori. L'aggregazione consente di visualizzare un valore aggregato per l'intervallo specificato (5 minuti, un'ora, un giorno) selezionato nella scheda *Periodo di tempo* o utilizzato per l'intera dashboard.

Etichetta del set di dati	Specificare un'etichetta personalizzata per il set di dati. L'etichetta viene visualizzata nella configurazione del set di dati e nella legenda del grafico a torta (per i set di dati aggregati). Tutti i set di dati vengono numerati, inclusi quelli con un' *Etichetta del set di dati* specificata. Se non viene specificata alcuna etichetta, il set di dati verrà etichettato automaticamente in base al relativo numero (ad esempio, "Set di dati n. 2", "Set di dati n. 3", ecc.). La numerazione dei set di dati viene ricalcolata dopo averli riordinati o trascinati. Le etichette dei set di dati troppo lunghe verranno abbreviate per adattarsi allo spazio disponibile (ad esempio, "Numero di proc..."). Specificare i tag per filtrare gli item visualizzati nel widget. È possibile includere o escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Esiste** - include i nomi dei tag specificati; **È uguale a** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contiene** - include i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Non esiste** - esclude i nomi dei tag specificati; **Non è uguale a** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Non contiene** - esclude i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - devono essere soddisfatte tutte le condizioni; le condizioni con lo stesso nome di tag verranno raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione.
Tag degli item	

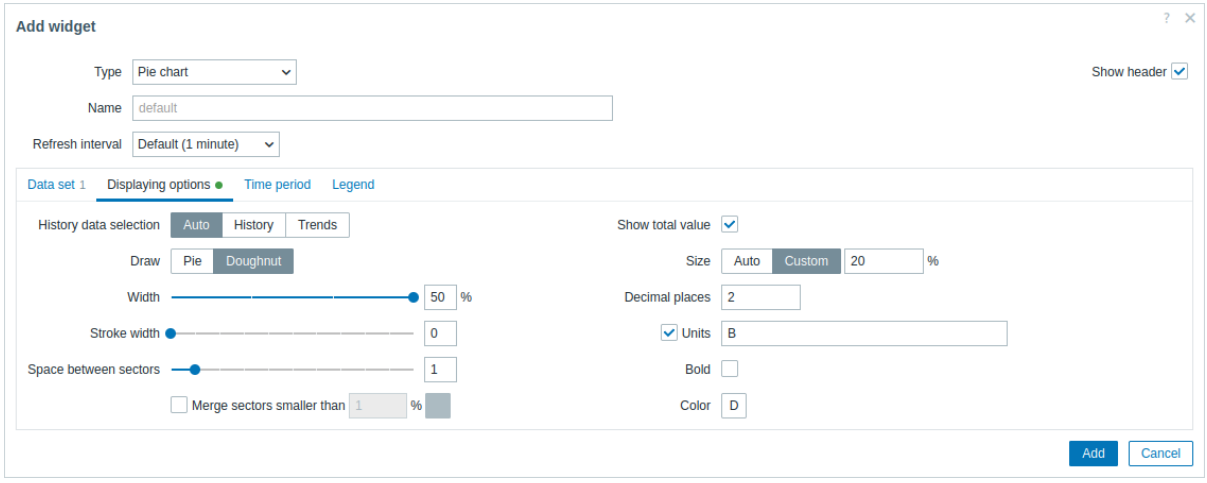
Dettagli di configurazione del set di dati

I set di dati esistenti sono visualizzati in un elenco. Puoi riordinarli, espanderli/comprimerli, cambiarne i colori e clonare questi set di dati.

Per ulteriori informazioni, vedi i dettagli di configurazione del set di dati nel widget *Graph*. Questi dettagli si applicano anche al widget *Pie chart*.

Opzioni di visualizzazione

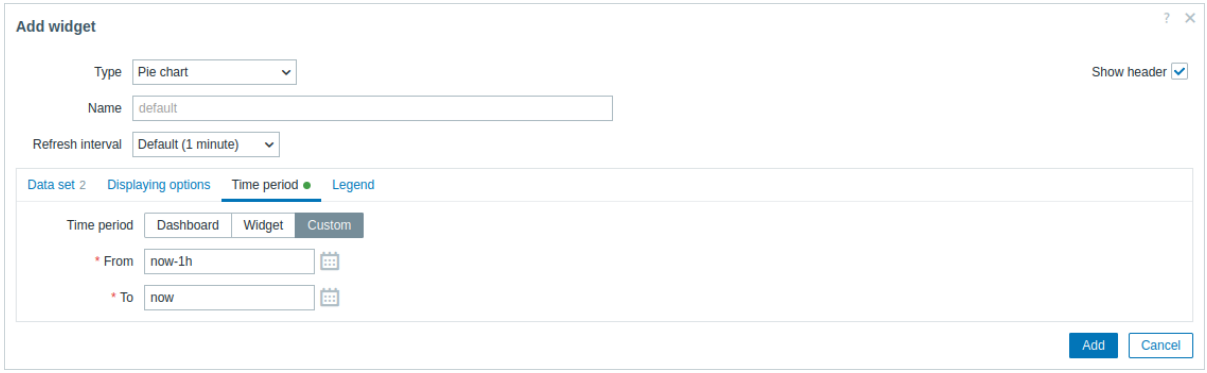
La scheda **Opzioni di visualizzazione** consente di definire la selezione dei dati della cronologia e le opzioni di visualizzazione per il grafico a torta:



Selezione dei dati della cronologia		Selezionare l'origine dei dati: **Auto** - i dati vengono ottenuti in base all'algoritmo classico (impostazione predefinita); **History** - dati dalla cronologia; **Trends** - dati dai trend.
Disegna		Selezionare lo stile di visualizzazione del grafico a torta: **Pie** - una torta completa (i settori occupano il 100% del raggio); **Doughnut** - una torta con un cerchio vuoto al centro (i settori occupano fino al 50% del raggio).
Spazio tra i settori		Selezionare la dimensione dello spazio (in unità da 0 a 10) tra i settori (il valore predefinito è "1").
Unisci i settori inferiori a N%		Selezionare la casella per unire i settori inferiori a N%. Se abilitata, selezionare il colore dei settori uniti e la soglia percentuale (N) per l'unione dei settori piccoli.
Stile di disegno: ***Doughnut***		
	Larghezza	Selezionare la larghezza della ciambella: 20, 30, 40 o 50% (impostazione predefinita) del raggio.
	Larghezza del bordo	Selezionare la larghezza del bordo del settore della ciambella (0-10).
	Mostra valore totale	Selezionare la casella per mostrare il valore totale al centro del grafico a ciambella.
		Dimensione	Selezionare l'opzione relativa alla dimensione del valore totale: **Auto** - la dimensione del testo viene regolata automaticamente per adattarsi in modo leggibile al centro della ciambella; **Custom** - specificare la dimensione del testo come percentuale dell'altezza totale del widget.
		Posizioni decimali	Specificare il numero di posizioni decimali per il valore totale (0-6).
		Unità	Specificare le unità per il valore totale.
		Grassetto	Selezionare la casella per visualizzare il valore totale in grassetto.
		Colore	Selezionare il colore per il valore totale.

Periodo di tempo

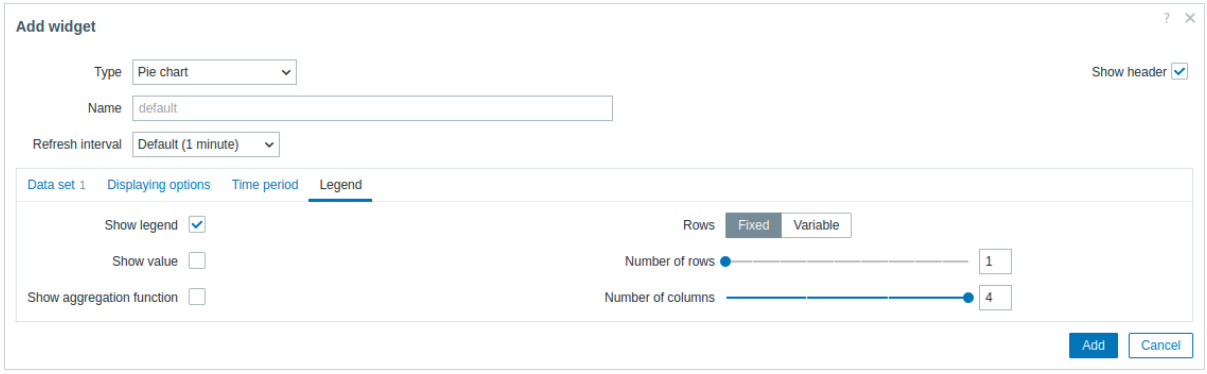
La scheda **Periodo di tempo** consente di impostare un periodo di tempo personalizzato per le impostazioni di aggregazione del grafico a torta:



Periodo di tempo	Selezionare la sorgente dati per il periodo di tempo: **Dashboard** - utilizzare il selettore del periodo di tempo del dashboard; **Widget** - utilizzare un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizzare un periodo di tempo personalizzato specificato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che, al passaggio del mouse, indica l'ora impostata. Si noti che, indipendentemente dalla configurazione *Periodo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per il periodo di tempo.
Widget	Immettere o selezionare un widget compatibile (*Grafico*, *Grafico (classico)*, *Prototipo di grafico*) come sorgente dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Immettere o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
A	Immettere o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".

Legenda

La scheda **Legenda** consente di personalizzare la legenda del grafico a torta:

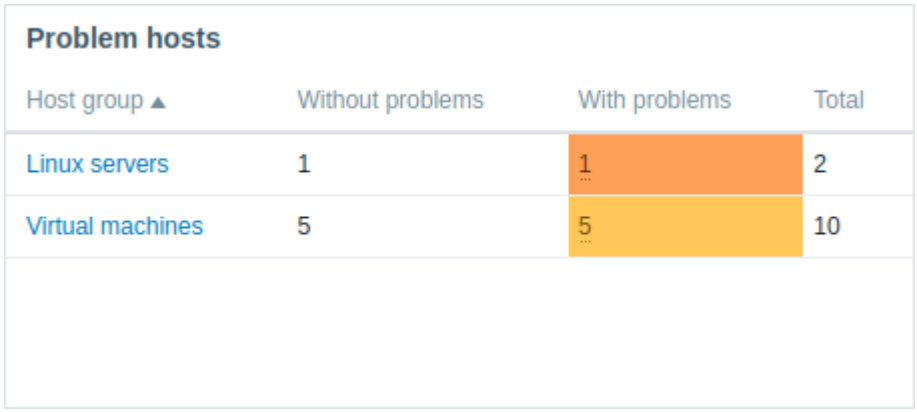


Mostra legenda	Deselezionare questa casella di controllo per nascondere la legenda nel grafico a torta (selezionata per impostazione predefinita).
Mostra valore	Selezionare questa casella di controllo per mostrare il valore dell'item nella legenda.
Mostra funzione di aggregazione	Selezionare questa casella di controllo per mostrare la funzione di aggregazione nella legenda.
Righe	Selezionare la modalità di visualizzazione delle righe della legenda: **Fisso** - il numero di righe visualizzate è determinato dal valore del parametro *Numero di righe*; **Variabile** - il numero di righe visualizzate è determinato dalla quantità di item configurati, senza superare il valore del parametro *Numero massimo di righe*.
Numero di righe/ Numero massimo di righe	Se *Righe* è impostato su "Fisso", impostare il numero di righe della legenda da visualizzare (1-10). Se *Righe* è impostato su "Variabile", impostare il numero massimo di righe della legenda da visualizzare (1-10).
Numero di colonne	Impostare il numero di colonne della legenda da visualizzare (1-4). Questo parametro è disponibile se *Mostra valore* non è selezionato.

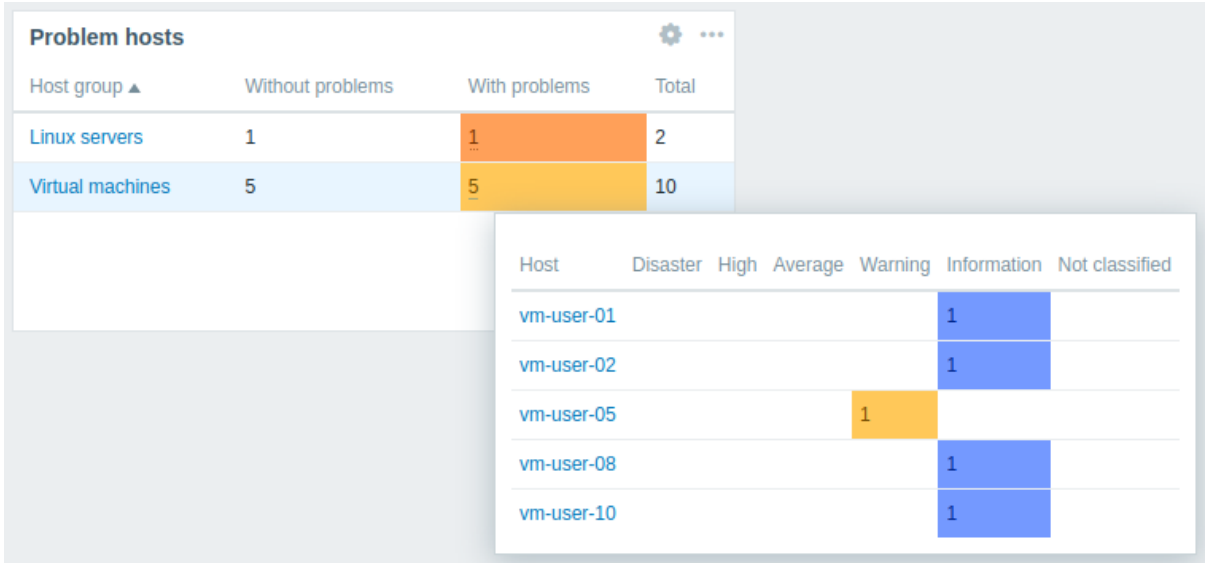
Host con problemi

Panoramica

Il widget *Host con problemi* visualizza il numero di problemi causa per gruppo di host e mostra la gravità più alta del problema in ciascun gruppo. Consente di filtrare i problemi in base a diversi parametri (gruppi di host, host, nomi dei problemi, ecc.).

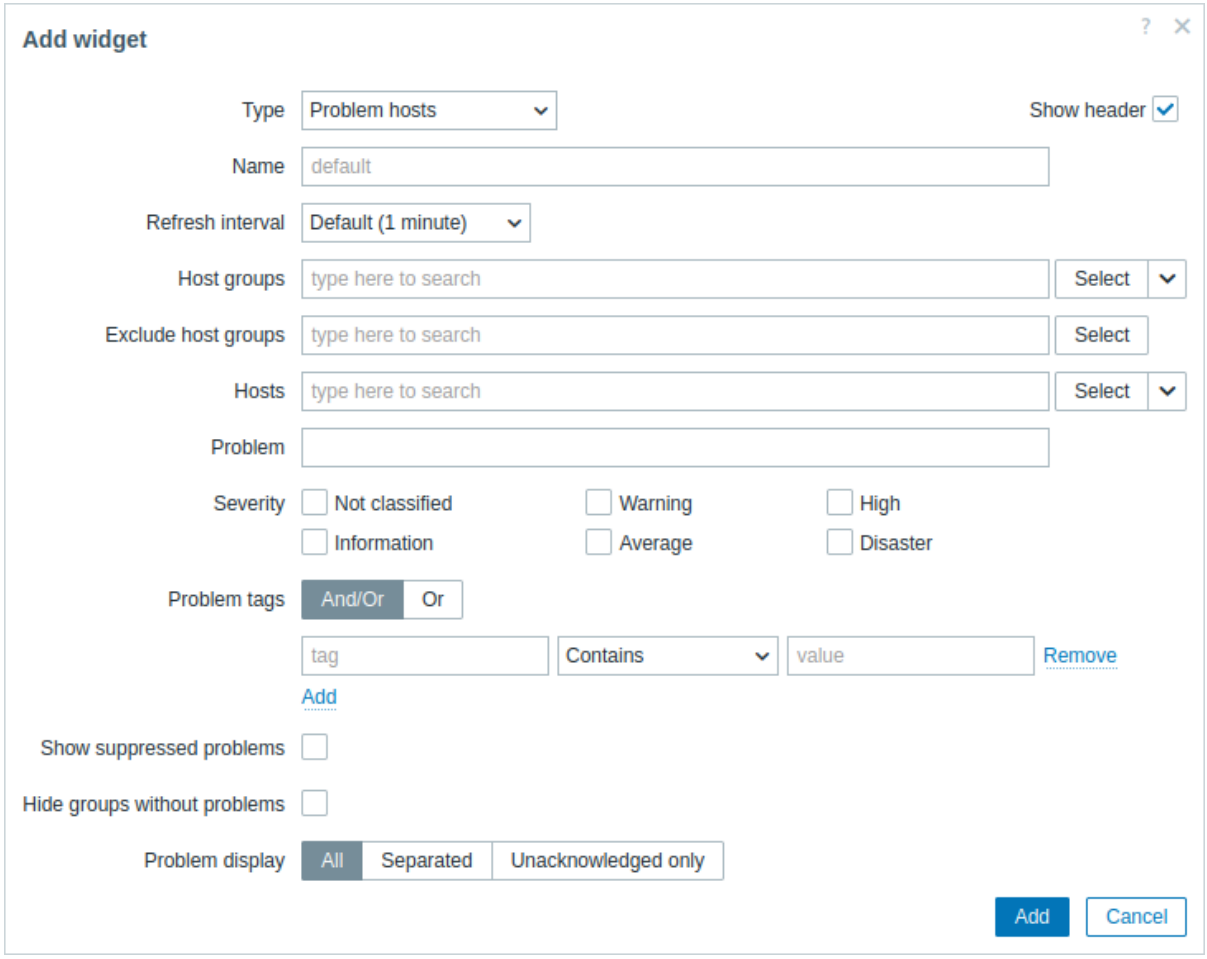


Passando il puntatore sul conteggio dei problemi viene mostrato l'elenco degli host con problemi:



Configurazione

Per configurare il widget, selezionare *Problem hosts* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

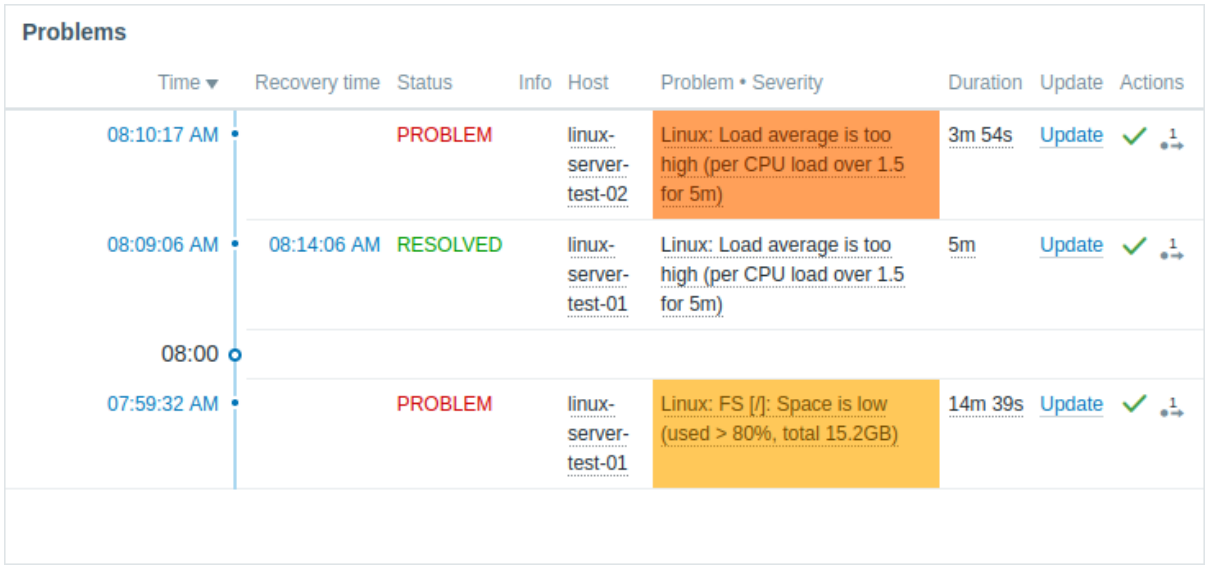
Host groups	Selezionare i gruppi di host da visualizzare nel widget. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Specificando un gruppo di host padre vengono implicitamente selezionati tutti i gruppi di host nidificati. I dati degli host di questi gruppi di host verranno visualizzati nel widget; se non viene inserito alcun gruppo di host, verranno visualizzati tutti i gruppi di host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Exclude host groups	Selezionare i gruppi di host da nascondere nel widget. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Specificando un gruppo di host padre vengono implicitamente selezionati tutti i gruppi di host nidificati. I dati degli host di questi gruppi di host non verranno visualizzati nel widget. Ad esempio, gli host 001, 002, 003 possono trovarsi nel Gruppo A e gli host 002, 003 anche nel Gruppo B. Se selezioniamo contemporaneamente di *mostrare* il Gruppo A e di *escludere* il Gruppo B, nella dashboard verranno visualizzati solo i dati dell'host 001. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Hosts	Selezionare gli host da visualizzare nel widget. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà mostrato un menu a discesa con gli host corrispondenti. Se non viene inserito alcun host, verranno visualizzati tutti gli host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Problem	È possibile limitare il numero di host con problemi visualizzati in base al nome del problema. Se si inserisce qui una stringa, verranno visualizzati solo gli host con problemi il cui nome contiene la stringa immessa. Le macro non vengono espanse.

Severity	Selezionare le severity dei problemi per filtrare i problemi da visualizzare nel widget. Se non viene selezionata alcuna severity, verranno visualizzati tutti i problemi.
Problem tags	Specificare i tag dei problemi per limitare il numero di problemi visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una condizione.
Show suppressed problems	Selezionare la casella per visualizzare i problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione dell'host.
Hide groups without problems	Selezionare l'opzione *Hide groups without problems* per nascondere nel widget i dati dei gruppi di host senza problemi. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Problem display	Visualizzare il conteggio dei problemi come: **All** - verrà visualizzato il conteggio completo dei problemi; **Separated** - il conteggio dei problemi non riconosciuti verrà visualizzato separatamente come numero rispetto al conteggio totale dei problemi; **Unacknowledged only** - verrà visualizzato solo il conteggio dei problemi non riconosciuti.

Problemi

Panoramica

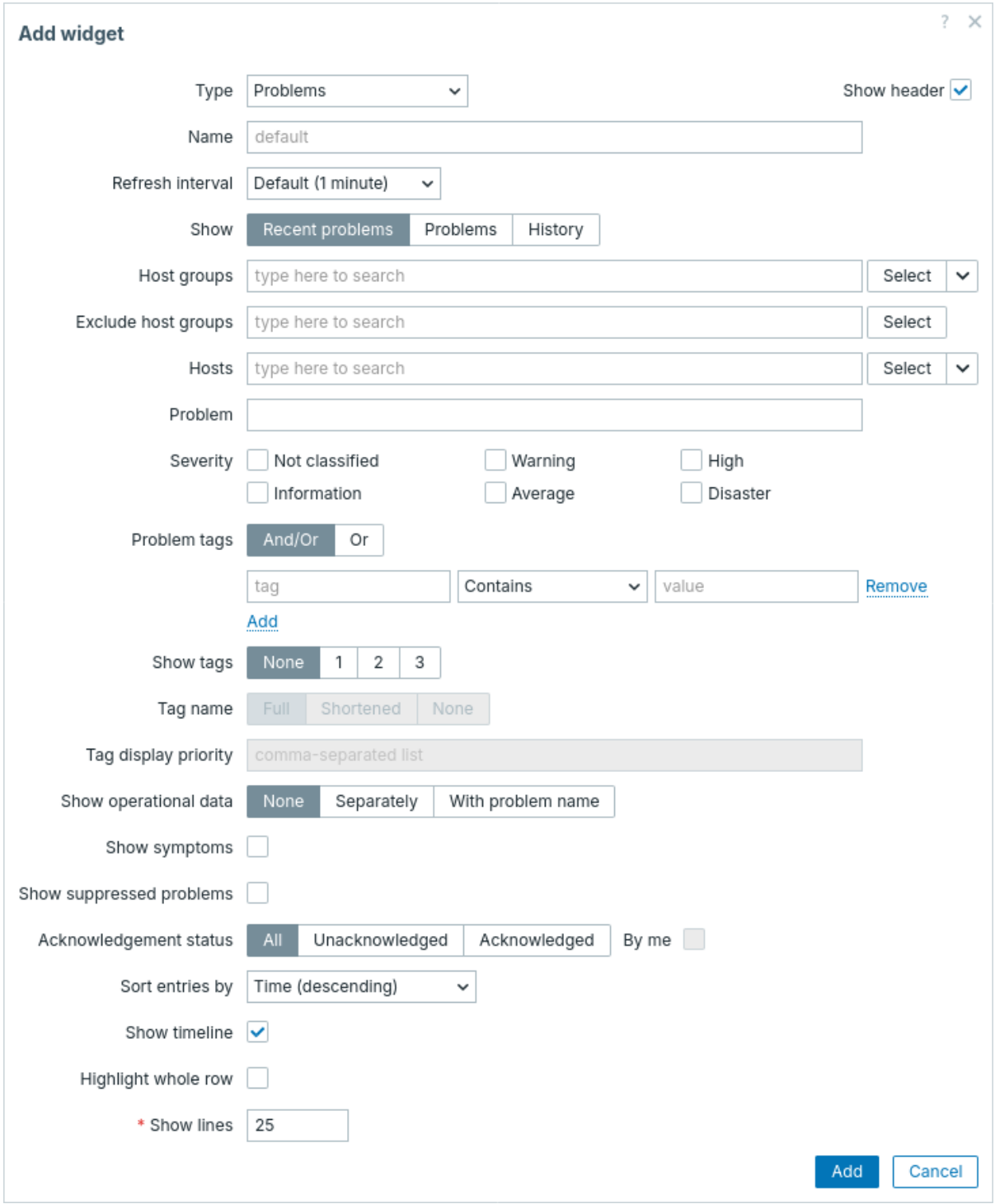
Il widget *Problems* visualizza i problemi correnti filtrati in base a diversi parametri (gruppi di host, host, nomi dei problemi, ecc.), offrendo una vista chiara di ciò che richiede attenzione.



Visualizza gli stessi dati di *Monitoring* > *Problems* e può mostrare fino a 1000 record.

Configurazione

Per configurare, seleziona *Problems* come tipo:



Puoi limitare in vari modi il numero di problemi visualizzati nel widget: in base allo stato del problema, al nome del problema, alla gravità, al gruppo di host, all'host, al tag dell'evento, allo stato di conferma, ecc.

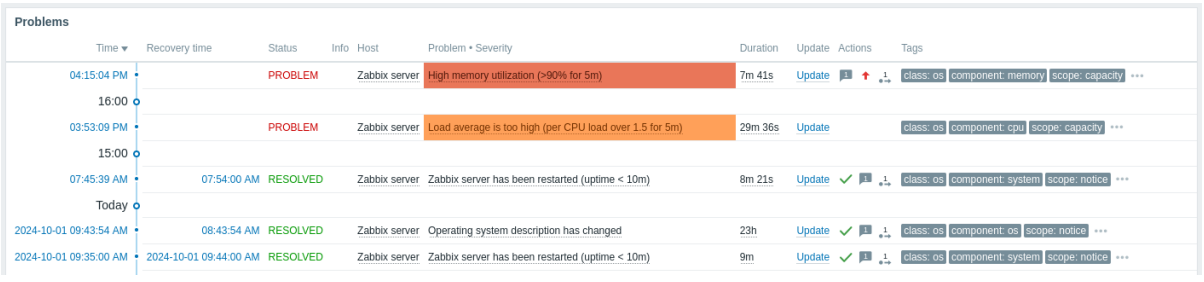
Oltre ai parametri comuni a tutti i widget, puoi impostare le seguenti opzioni specifiche:

Show	Filtra per stato del problema: **Recent problems** - vengono visualizzati i problemi non risolti e quelli risolti di recente (predefinito); **Problems** - vengono visualizzati i problemi non risolti; **History** - viene visualizzata la cronologia di tutti gli eventi.

Host groups	Seleziona i gruppi di host di cui visualizzare i problemi nel widget. In alternativa, seleziona un widget compatibile come origine dati per i gruppi di host. Questo campo supporta l'autocompletamento, quindi iniziando a digitare il nome di un gruppo verrà mostrato un elenco a discesa dei gruppi corrispondenti. Specificando un gruppo di host padre, vengono implicitamente selezionati tutti i gruppi di host annidati. I problemi di questi gruppi di host verranno visualizzati nel widget; se non viene specificato alcun gruppo di host, verranno visualizzati i problemi di tutti i gruppi di host. Questo parametro non è disponibile quando si configura il widget in una template dashboard.
Exclude host groups	Seleziona i gruppi di host di cui nascondere i problemi nel widget. Questo campo supporta l'autocompletamento, quindi iniziando a digitare il nome di un gruppo verrà mostrato un elenco a discesa dei gruppi corrispondenti. Specificando un gruppo di host padre, vengono implicitamente selezionati tutti i gruppi di host annidati. I problemi di questi gruppi di host non verranno visualizzati. Ad esempio, gli host 001, 002, 003 possono trovarsi nel Group A e gli host 002, 003 anche nel Group B. Se scegliamo di *show* Group A ed *exclude* Group B contemporaneamente, nel widget verranno visualizzati solo i problemi dell'host 001. Questo parametro non è disponibile quando si configura il widget in una template dashboard.
Hosts	Seleziona gli host di cui visualizzare i problemi nel widget. In alternativa, seleziona un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta l'autocompletamento, quindi iniziando a digitare il nome di un host verrà mostrato un elenco a discesa degli host corrispondenti. Se non viene specificato alcun host, verranno visualizzati i problemi di tutti gli host. Questo parametro non è disponibile quando si configura il widget in una template dashboard.
Problem	Puoi limitare il numero di problemi visualizzati in base al loro nome. Se inserisci una stringa qui, verranno visualizzati solo i problemi il cui nome contiene la stringa inserita. Le macro non vengono espanse.
Severity	Seleziona le gravità dei problemi per filtrare i problemi da visualizzare nel widget. Se non viene selezionata alcuna gravità, verranno visualizzati tutti i problemi.
Problem tags	Specifica i tag dei problemi per limitare il numero di problemi visualizzati nel widget. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che una condizione sia soddisfatta. Quando vengono filtrati, i tag specificati qui verranno visualizzati per primi con il problema, a meno che non siano sovrascritti dall'elenco *Tag display priority* (vedi sotto).
Show tags	Seleziona il numero di tag visualizzati: **None** - nessuna colonna *Tags*; **1** - la colonna *Tags* contiene un tag; **2** - la colonna *Tags* contiene due tag; **3** - la colonna *Tags* contiene tre tag. Per vedere tutti i tag del problema, passa il mouse sull'icona con i tre puntini.

Tag name	Seleziona la modalità di visualizzazione del nome del tag: **Full** - i nomi e i valori dei tag vengono visualizzati per intero; **Shortened** - i nomi dei tag sono abbreviati a 3 simboli, ma i valori dei tag vengono visualizzati per intero; **None** - vengono visualizzati solo i valori dei tag; nessun nome.
Tag display priority	Inserisci la priorità di visualizzazione dei tag per un problema, come elenco di tag separati da virgole. Devono essere usati solo i nomi dei tag, non i valori. Esempio: `Services,Applications,Application` I tag di questo elenco verranno sempre visualizzati per primi, sostituendo l'ordinamento naturale alfabetico.
Show operational data	Seleziona la modalità di visualizzazione dei dati operativi: **None** - nessun dato operativo viene visualizzato; **Separately** - i dati operativi vengono visualizzati in una colonna separata; **With problem name** - aggiunge i dati operativi al nome del problema, usando le parentesi per i dati operativi. Per i dettagli, consulta la descrizione del campo *Operational data* nella configurazione del trigger.
Show symptoms	Seleziona la casella di controllo per visualizzare su una riga separata i problemi classificati come sintomi.
Show suppressed problems	Seleziona la casella di controllo per visualizzare i problemi che altrimenti sarebbero soppressi (non mostrati) a causa della manutenzione dell'host o della singola soppressione del problema.
Acknowledgement status	Filtra per visualizzare tutti i problemi, solo i problemi non confermati o solo i problemi confermati.
Sort entries by	Seleziona la casella aggiuntiva per filtrare i problemi che sono stati confermati da te in passato. Ordina le voci per: **Time** (decrescente o crescente); **Severity** (decrescente o crescente); **Problem name** (decrescente o crescente); **Host** (decrescente o crescente). L'ordinamento delle voci per **Host** (decrescente o crescente) non è disponibile quando si configura il widget in una template dashboard.
Show timeline	Seleziona la casella di controllo per visualizzare una timeline grafica.
Highlight whole row	Seleziona la casella di controllo per evidenziare l'intera riga per i problemi non risolti. Per l'evidenziazione viene usato il colore della gravità del problema. *Highlight whole row* non è disponibile nei temi ad alto contrasto.
Show lines	Specifica il numero di righe di problemi da visualizzare.

Utilizzo del widget



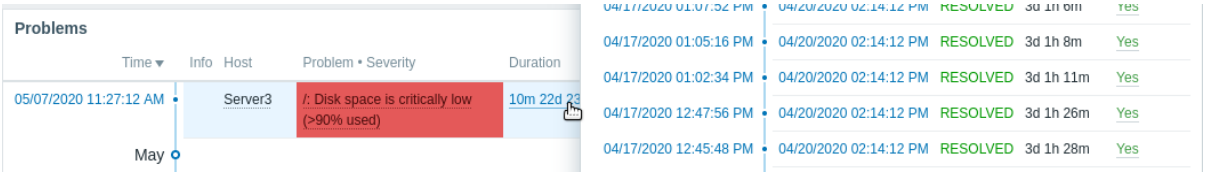
Il widget Problems offre un accesso rapido a informazioni aggiuntive:

- Fare clic sulla data del problema (nella colonna Time) o sulla data di ripristino (nella colonna Recovery Time) per visualizzare i dettagli dell'evento.
- Se il campo Info non è vuoto, passare il mouse sull'icona visualizzata per vedere ulteriori dettagli.
- Fare clic sul nome dell'host per aprire il menu host.
- Fare clic sul nome del problema per aprire il menu evento.
- Passare il mouse sopra o fare clic sulla durata del problema per visualizzare il popup dell'evento del problema.
- Facendo clic su *Update* si apre una finestra Aggiorna problema.
- Passare il mouse sopra o fare clic sull'icona della freccia grigia nella colonna Actions per visualizzare l'elenco delle azioni eseguite.

Popup dell'evento del problema

Il popup dell'evento del problema include l'elenco degli eventi di problema per questo trigger e, se definita, la descrizione del

trigger e un URL cliccabile.



Per visualizzare il popup dell'evento del problema:

- Passare il mouse sopra la durata del problema nella colonna *Duration* del widget *Problems*. Il popup scompare non appena si allontana il mouse dalla durata.
- Fare clic sulla durata nella colonna *Duration* del widget *Problems*. Il popup scompare solo se si fa nuovamente clic sulla durata.

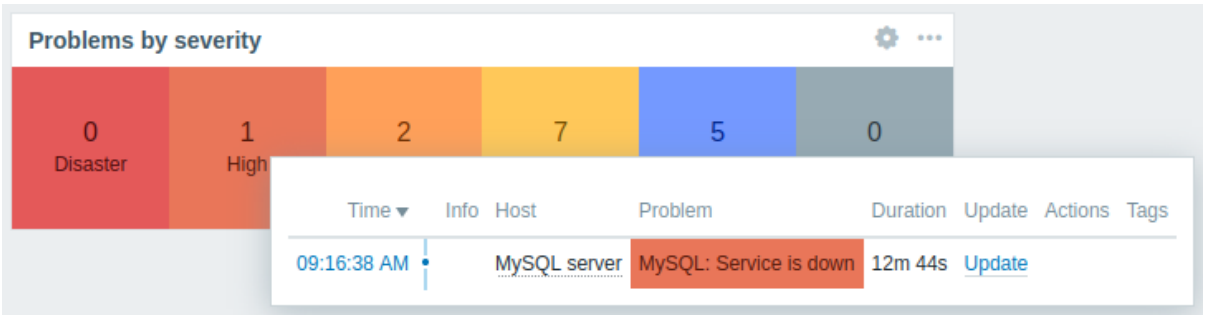
Problemi per gravità

Panoramica

Il widget *Problemi per gravità* visualizza il numero di problemi causa raggruppati per gravità. Consente di filtrare i problemi in base a diversi parametri (gruppi di host, host, nomi dei problemi, ecc.).

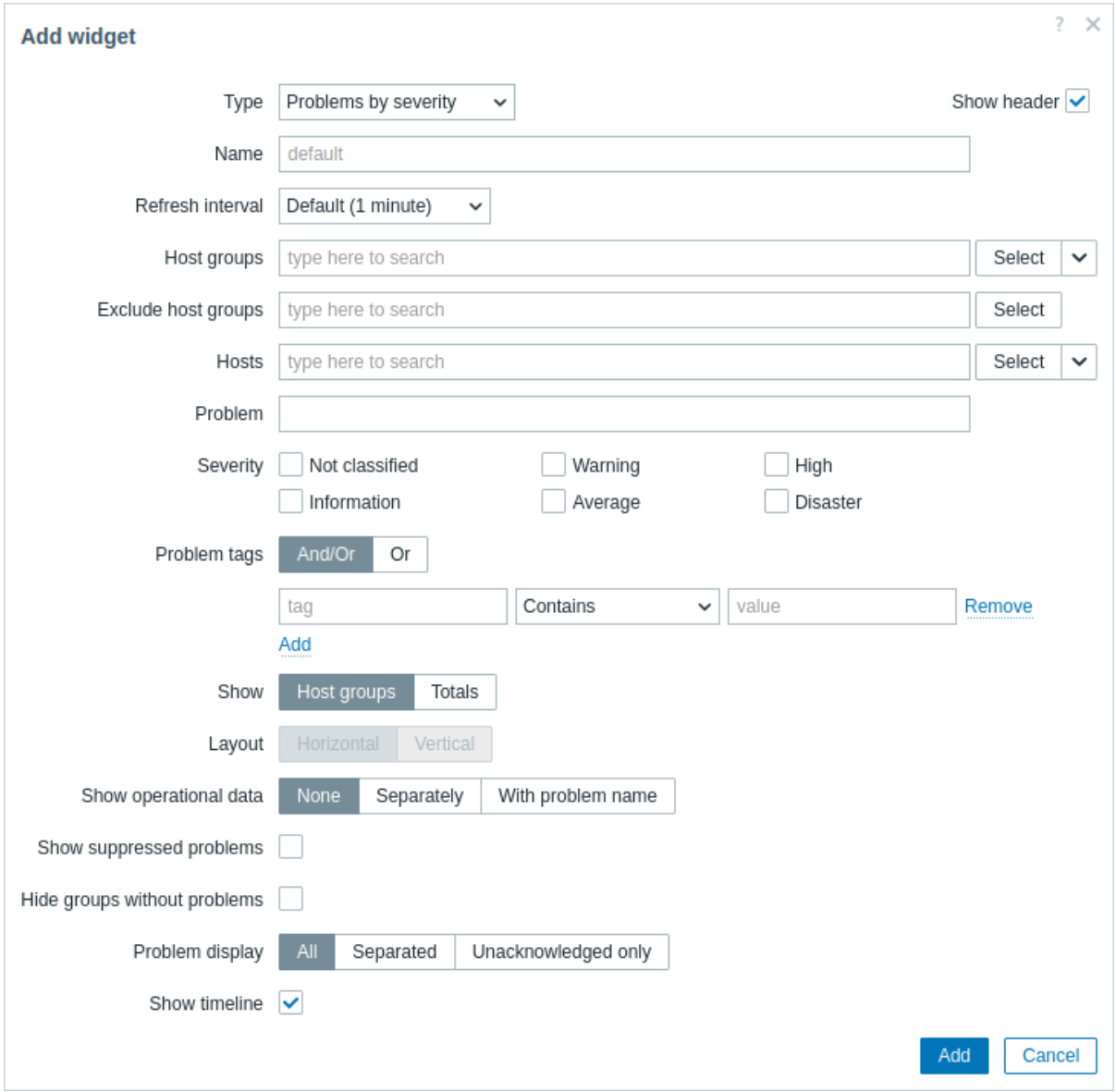


Passando il cursore sul conteggio di un problema viene mostrato l'elenco dei problemi correlati:



Configurazione

Per configurare, selezionare *Problems by severity* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

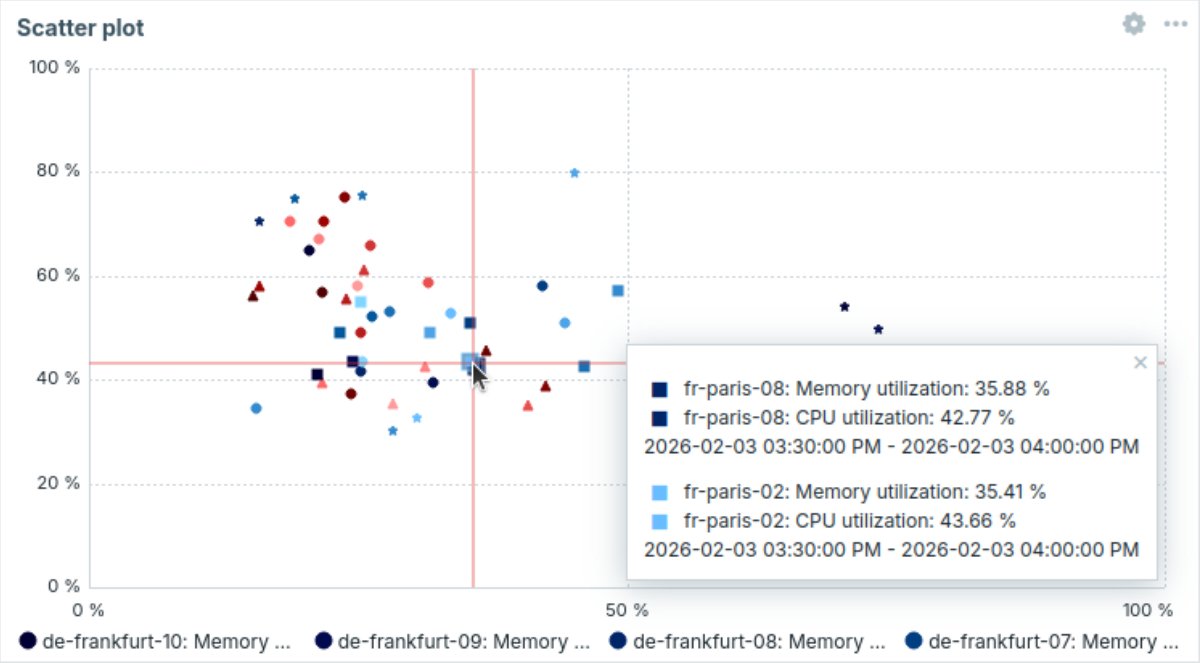
Host groups	Selezionare gli host group da visualizzare nel widget. In alternativa, selezionare un widget compatibile come origine dati per gli host group. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Specificando un host group padre, vengono selezionati implicitamente tutti gli host group annidati. I dati degli host provenienti da questi host group verranno visualizzati nel widget; se non vengono inseriti host group, verranno visualizzati tutti gli host group. Questo parametro non è disponibile quando si configura il widget su un template dashboard.
Exclude host groups	Selezionare gli host group da nascondere nel widget. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Specificando un host group padre, vengono selezionati implicitamente tutti gli host group annidati. I dati degli host provenienti da questi host group non verranno visualizzati nel widget. Ad esempio, gli host 001, 002, 003 possono trovarsi nel Group A e gli host 002, 003 anche nel Group B. Se si sceglie contemporaneamente di *mostrare* il Group A ed *escludere* il Group B, nella dashboard verranno visualizzati solo i dati dell'host 001. Questo parametro non è disponibile quando si configura il widget su un template dashboard.

Hosts	Selezionare gli host da visualizzare nel widget. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà mostrato un menu a discesa con gli host corrispondenti. Se non vengono inseriti host, verranno visualizzati tutti gli host. Questo parametro non è disponibile quando si configura il widget su un template dashboard.
Problem	È possibile limitare il numero di host con problemi visualizzati in base al nome del problema. Se si inserisce una stringa qui, verranno visualizzati solo gli host con problemi il cui nome contiene la stringa inserita. Le macro non vengono espanse.
Severity	Selezionare le severità dei problemi per filtrare i problemi da visualizzare nel widget. Se non viene selezionata alcuna severità, verranno visualizzati tutti i problemi.
Problem tags	Specificare i tag dei problemi per limitare il numero di problemi visualizzati nel widget. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole/minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole); **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole); **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole). Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che una condizione sia soddisfatta.
Show	Selezionare l'opzione di visualizzazione: **Host groups** - visualizza i problemi per host group; **Totals** - visualizza il totale dei problemi per tutti gli host group selezionati in blocchi colorati corrispondenti alla severità del problema. Questo parametro non è disponibile quando si configura il widget su un template dashboard e verrà visualizzato solo un totale dei problemi.
Layout	Selezionare l'opzione di layout: **Horizontal** - i blocchi colorati dei totali verranno visualizzati orizzontalmente; **Vertical** - i blocchi colorati dei totali verranno visualizzati verticalmente. Questo parametro è disponibile se *Show* è impostato su "Totals".
Show operational data	Visualizza i dati operativi come: **None** - i dati operativi non verranno visualizzati **Separately** - i dati operativi verranno visualizzati come una riga separata **With problem name** - i dati operativi verranno visualizzati insieme al nome del problema. Per i dettagli, vedere la descrizione del campo *Operational data* nella configurazione del trigger.
Show suppressed problems	Selezionare la casella di controllo per visualizzare i problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione dell'host.
Hide groups without problems	Selezionare l'opzione *Hide groups without problems* per nascondere nel widget i dati degli host group senza problemi. Questo parametro non è disponibile quando si configura il widget su un template dashboard.
Problem display	Visualizza il conteggio dei problemi come: **All** - verrà visualizzato il conteggio completo dei problemi; **Separated** - il conteggio dei problemi non riconosciuti verrà visualizzato separatamente come numero rispetto al conteggio totale dei problemi; **Unacknowledged only** - verrà visualizzato solo il conteggio dei problemi non riconosciuti.
Show timeline	Selezionare la casella di controllo per visualizzare una timeline visiva.

Grafico a dispersione

Panoramica

Il widget *Grafico a dispersione* mostra la relazione tra due metriche rappresentando singoli punti dati lungo un asse X e un asse Y. Questo aiuta a individuare modelli, raggruppamenti, correlazioni e valori anomali nel set di dati.



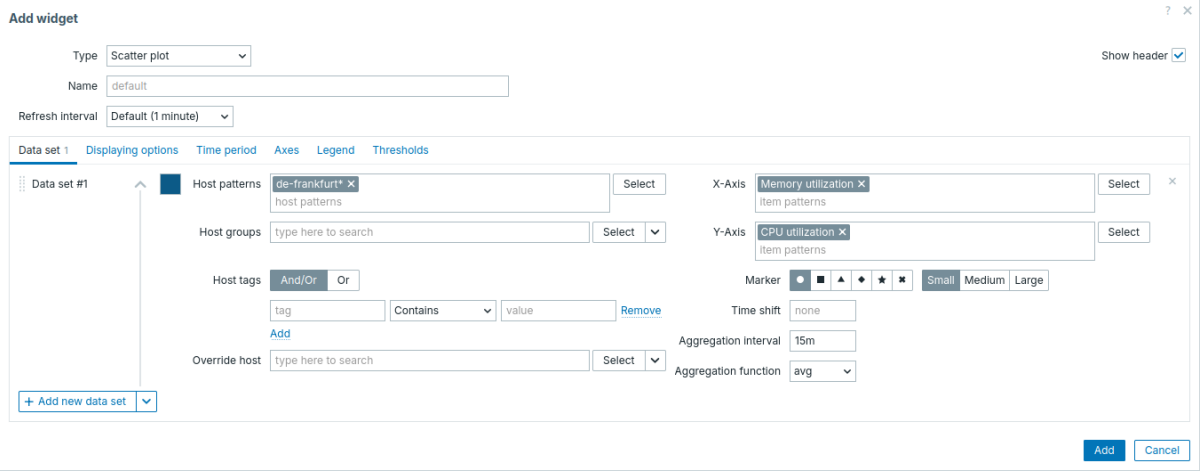
Passando il mouse su un punto dati viene visualizzato un tooltip con un elenco di item appartenenti a quel punto dati e agli altri punti entro un raggio di 6 px; l'elenco è ordinato prima in base ai valori dell'asse X del grafico a dispersione in ordine crescente e poi in base ai valori dell'asse Y in ordine decrescente.

Selezionando un item nel tooltip, i relativi dati vengono trasmessi agli altri widget.

Le informazioni visualizzate nel widget *Grafico a dispersione* possono essere scaricate come immagine PNG selezionando l'opzione *Scarica immagine* nel menu *Azioni* del widget ..., visualizzato quando si passa il mouse sul widget. A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella `Downloads` oppure viene aperta una finestra di dialogo *Salva*, che consente di scegliere la posizione.

Configurazione

Per configurare, selezionare *Scatter plot* come tipo:



Set di dati

La scheda **Set di dati** consente di selezionare i dati per il grafico a dispersione aggiungendo set di dati. È possibile aggiungere due tipi di set di dati:

- *Modelli di item* - gli host con item corrispondenti vengono visualizzati come punti dati. È possibile scegliere un singolo colore di base oppure selezionare una riga della tavolozza per assegnare colori distinti a ciascun host con gli item corrispondenti.
- *Elenco di item* - gli item selezionati vengono visualizzati come punti dati, dove ogni punto rappresenta una coppia di item (X e Y). È possibile scegliere individualmente il colore di ciascun set di dati dal selettore.

I punti dati vengono creati solo per le coppie host/item che hanno almeno un valore su entrambi gli assi.

Per impostazione predefinita, viene aggiunto un set di dati *Modelli di item*.

Set di dati	Per il set di dati **Modelli di item**: Selezionare o immettere i modelli di host e item; i dati degli item che corrispondono a questi modelli verranno visualizzati nel grafico a dispersione. Se il numero di punti dati supera 8k, è necessario aumentare `memory_limit` nel file di configurazione `php.ini`. Il limite di memoria consigliato per 20k punti dati è di 512 MB. Per la selezione è possibile utilizzare modelli con caratteri jolly (ad esempio, * restituirà i risultati che corrispondono a zero o più caratteri). Per specificare un modello con caratteri jolly, immettere manualmente la stringa e premere *Invio*. Il simbolo jolly viene sempre interpretato, pertanto non è possibile aggiungere singolarmente, ad esempio, un item denominato *item** se sono presenti altri item corrispondenti (ad esempio, *item2*, *item3*). La specifica dei modelli di host e item è obbligatoria per i set di dati "Modelli di item". Vedere anche: Dettagli della configurazione del set di dati. Per il set di dati **Elenco di item**: Selezionare gli item per il grafico a dispersione facendo clic sul pulsante *Aggiungi item*. È inoltre possibile selezionare widget compatibili come origine dati per gli item facendo clic sul pulsante *Aggiungi widget*. La specifica di item o widget è obbligatoria per i set di dati "Elenco di item". Vedere anche: Dettagli della configurazione del set di dati. Sono consentiti solo tipi di item numerici. Quando si configura il widget in una dashboard di template, il parametro per specificare i modelli di host non è disponibile e il parametro per specificare un elenco di item consente di selezionare solo gli item configurati nel template.
Gruppi di host	Selezionare i gruppi di host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo dispone del completamento automatico: iniziando a digitare il nome di un gruppo, verrà visualizzato un elenco a discesa dei gruppi corrispondenti. La specifica di un gruppo di host padre seleziona implicitamente tutti i gruppi di host annidati. I dati degli host appartenenti a questi gruppi di host verranno visualizzati nel widget; se non viene immesso alcun gruppo di host, verranno visualizzati tutti i gruppi di host. Questo parametro è disponibile solo per i set di dati *Modelli di item* e non è disponibile quando si configura il widget in una dashboard di template.
Tag host	Specificare i tag per filtrare gli host visualizzati nel widget. È possibile includere o escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza dei nomi dei tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Esiste** - include i nomi dei tag specificati; **È uguale a** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contiene** - include i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole); **Non esiste** - esclude i nomi dei tag specificati; **Non è uguale a** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Non contiene** - esclude i nomi dei tag specificati i cui valori contengono la stringa immessa (corrispondenza di sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate tramite la condizione Or; **Or** - è sufficiente che venga soddisfatta una condizione. Questo parametro è disponibile solo per i set di dati *Modelli di item* e non è disponibile quando si configura il widget in una dashboard di template.

Sovrascrivi host	Selezionare un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Indicatore	Scegliere il tipo e la dimensione dell'indicatore per i punti dati degli host nel set di dati.
Spostamento temporale	Specificare lo spostamento temporale, se necessario. In questo campo è possibile utilizzare i suffissi temporali. Sono consentiti valori negativi.
Funzione di aggregazione	Specificare la funzione di aggregazione da utilizzare per ogni item nel set di dati all'interno dell'*Intervallo di aggregazione* selezionato: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **sum** - visualizza la somma dei valori; **count** - visualizza il numero di valori (se esiste almeno un valore su entrambi gli assi); **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore. L'aggregazione consente di visualizzare un valore aggregato per l'intervallo specificato (5 minuti, un'ora, un giorno), invece di tutti i valori. Per *min*, *max*, *avg* e *sum* è possibile visualizzare solo dati numerici. Per *count*, i dati non numerici verranno convertiti in numerici.
Intervallo di aggregazione	Specificare l'intervallo per l'aggregazione dei valori. In questo campo è possibile utilizzare i suffissi temporali. Un valore numerico senza suffisso verrà considerato espresso in secondi. Se il widget è configurato per visualizzare dati storici basati sui trend (*Selezione dei dati storici* impostata su *Trend* o *Automatico*), si consiglia di utilizzare un intervallo di aggregazione che sia un multiplo di 1 ora (ad esempio, 3600, 60m, 1h, 3h e così via). I trend memorizzano valori aggregati su base oraria, pertanto l'utilizzo di un intervallo di aggregazione che non sia un multiplo di 1 ora (ad esempio, 100s, 7min, 15min, 90min e così via) può produrre risultati difficili da interpretare. La specifica di un intervallo di aggregazione è obbligatoria. L'intervallo di aggregazione determina l'intervallo temporale e le coordinate di un punto dati; ad esempio, se gli item dell'asse X hanno 4 valori e gli item dell'asse Y hanno 1 valore, le coordinate riflettono l'aggregazione di 4 valori sull'asse X e di 1 valore sull'asse Y nell'intervallo specificato.

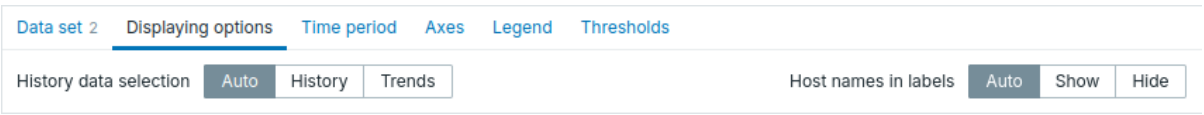
Dettagli della configurazione del set di dati

I set di dati esistenti vengono visualizzati in un elenco. È possibile riordinare, espandere/comprimere, cambiare i colori e clonare questi set di dati.

Per ulteriori informazioni, vedere i dettagli della configurazione del set di dati nel widget *Graph*. Questi dettagli si applicano anche al widget *Scatter plot*.

Opzioni di visualizzazione

La scheda **Opzioni di visualizzazione** consente di definire la selezione dei dati della cronologia:

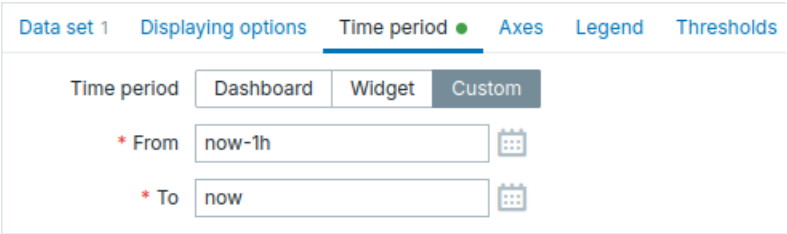


Selezione dei dati della cronologia	Imposta l'origine dei dati del grafico a dispersione: **Auto** - i dati vengono ottenuti in base all'algoritmo del grafico classico (impostazione predefinita). **Cronologia** - dati dalla cronologia. **Trend** - dati dai trend.

Nomi degli host nelle etichette	Scegli se visualizzare i nomi degli host nella legenda del grafico e nel tooltip che appare passando il mouse sui dati del grafico: **Auto** - i nomi degli host vengono visualizzati solo se nei set di dati è presente più di un host (impostazione predefinita). **Mostra** - i nomi degli host vengono visualizzati. **Nascondi** - i nomi degli host vengono nascosti. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.

Periodo di tempo

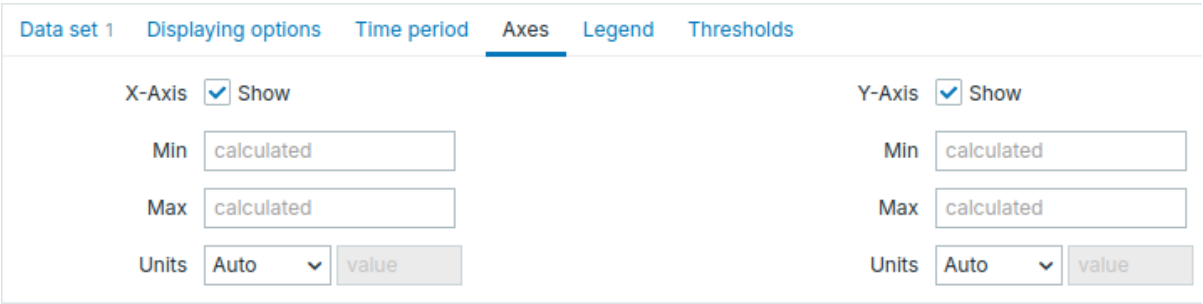
La scheda **Periodo di tempo** consente di impostare un periodo di tempo per il quale visualizzare i dati nel grafico a dispersione:



Periodo di tempo	Selezionare la sorgente dati per il periodo di tempo: **Dashboard** - utilizzare il selettore del periodo di tempo della dashboard; **Widget** - utilizzare un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizzare un periodo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio, che al passaggio del mouse indica l'ora impostata. Tenere presente che, indipendentemente dalla configurazione *Periodo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per il periodo di tempo.
Widget	Inserire o selezionare un widget compatibile come sorgente dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Inserire o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".
A	Inserire o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Periodo di tempo* è impostato su "Personalizzato".

Assi

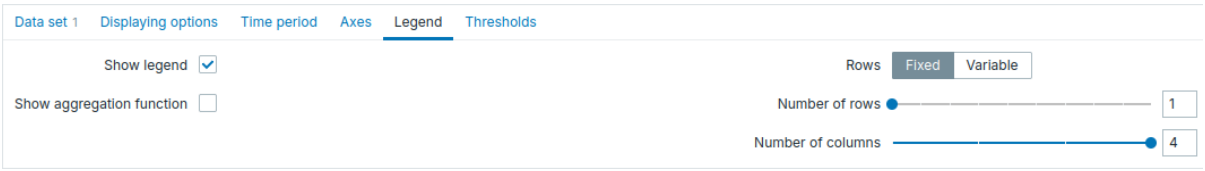
La scheda **Assi** consente di personalizzare la visualizzazione degli assi:



Asse X/Asse Y	Deselezionare questa casella di controllo per nascondere l'asse X/l'asse Y (selezionata per impostazione predefinita).
Min	Impostare il valore minimo dell'asse corrispondente.
Max	Impostare il valore massimo dell'asse corrispondente.
Unità	Scegliere l'unità per i valori degli assi del grafico a dispersione dall'elenco a discesa: **Automatico** - i valori degli assi vengono visualizzati utilizzando l'unità del primo item nel set di dati; **Statico** - i valori degli assi vengono visualizzati utilizzando l'unità specificata nel campo di input *valore*; se il campo viene lasciato vuoto, vengono visualizzati solo valori numerici.

Legenda

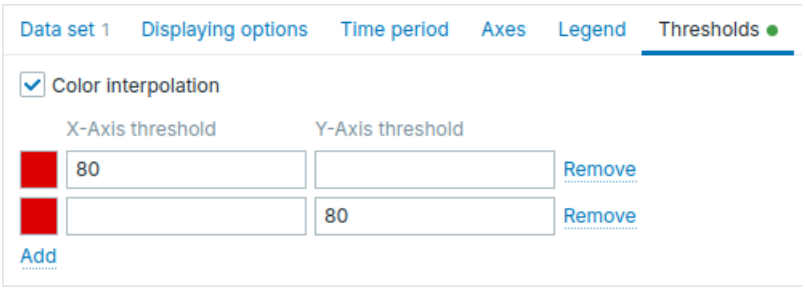
La scheda **Legenda** consente di personalizzare la legenda del grafico a dispersione:



Mostra legenda	Deselezionare questa casella di controllo per nascondere la legenda nel grafico a dispersione (selezionata per impostazione predefinita).
Mostra funzione di aggregazione	Selezionare questa casella di controllo per mostrare la funzione di aggregazione nella legenda.
Righe	Selezionare la modalità di visualizzazione delle righe della legenda: **Fisso** - il numero di righe visualizzate è determinato dal valore del parametro *Numero di righe*; **Variabile** - il numero di righe visualizzate è determinato dalla quantità di item configurati, senza superare il valore del parametro *Numero massimo di righe*.
Numero di righe/ Numero massimo di righe	Se *Righe* è impostato su ”Fisso”, impostare il numero di righe della legenda da visualizzare (1-10). Se *Righe* è impostato su ”Variabile”, impostare il numero massimo di righe della legenda da visualizzare (1-10).
Numero di colonne	Impostare il numero di colonne della legenda da visualizzare (1-4). Questo parametro è disponibile se *Visualizza min/media/max* non è selezionato.

Soglie

La scheda **Soglie** consente di configurare le soglie che determinano i colori degli indicatori degli host:



Interpolazione del colore	Selezionare la casella di controllo per abilitare una transizione uniforme tra i colori delle soglie per gli indicatori degli host. Questo parametro si applica se sono impostate due o più soglie per l’asse X, l’asse Y o entrambi gli assi contemporaneamente.
Soglia asse X/asse Y	Fare clic su *Aggiungi* per aggiungere una soglia, selezionare un colore per la soglia dal selettore dei colori e specificare un valore numerico. Le soglie possono essere applicate a un singolo asse, a entrambi gli assi separatamente o a entrambi gli assi contemporaneamente. Ad esempio, una singola soglia con X=80 e Y=80 colorerà gli indicatori quando entrambe le condizioni sono soddisfatte. L’elenco delle soglie verrà ordinato in ordine crescente al momento del salvataggio. I suffissi (ad esempio, ”1d”, ”2w”, ”4K”, ”8G”) sono supportati. Le mappature dei valori sono supportate.

Report SLA

Panoramica

Il widget *Report SLA* visualizza i report SLA, aiutandoti a monitorare le prestazioni del servizio rispetto agli obiettivi definiti.

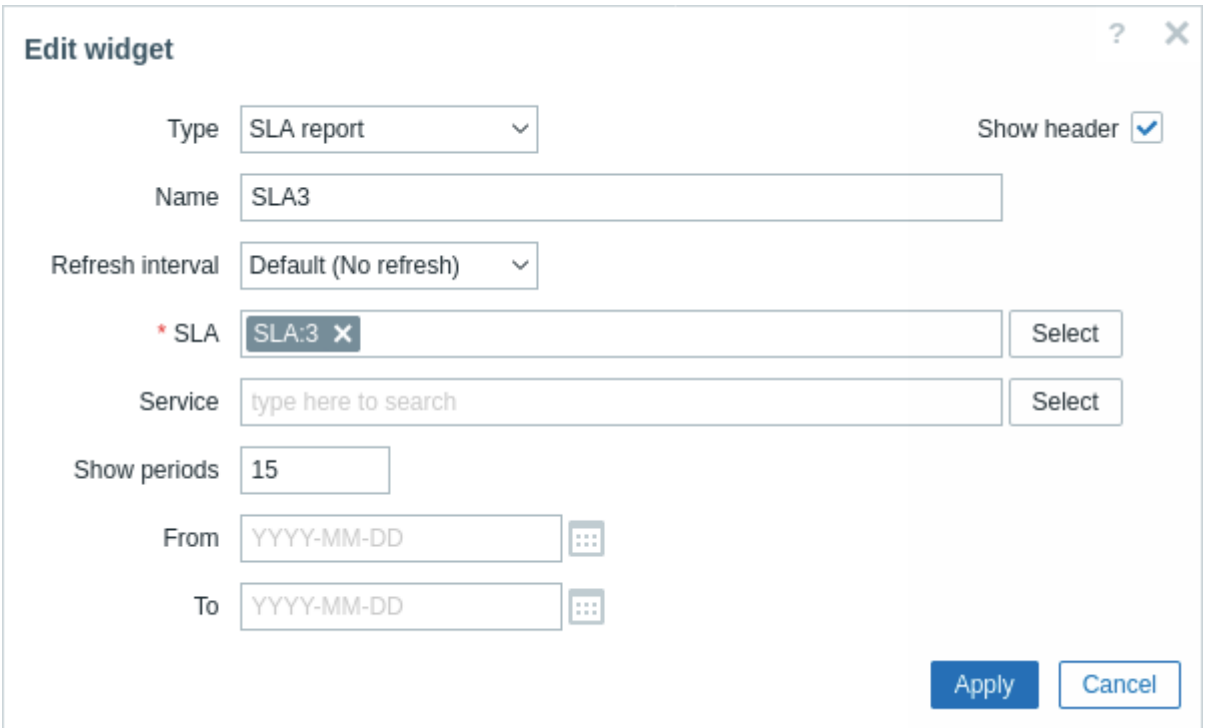
SLA report

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

Visualizza gli stessi dati di *Servizi* > *Report SLA*.

Configurazione

Per configurare il widget, selezionare *Report SLA* come tipo:



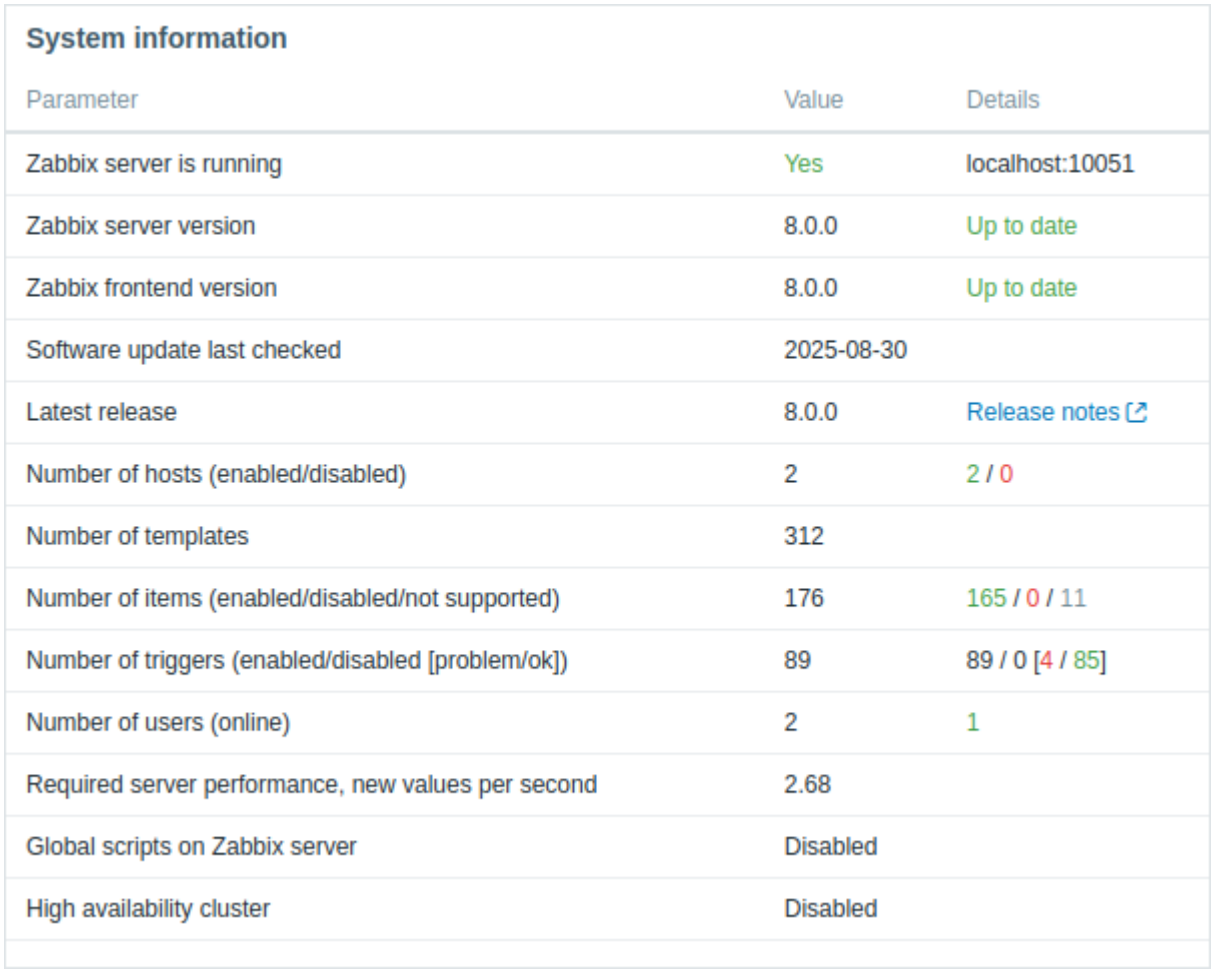
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

SLA	Selezionare l'SLA per il report.
Service	Selezionare il servizio per il report.
Show periods	Impostare quanti periodi verranno visualizzati nel widget (20 per impostazione predefinita, massimo 100).
From	Selezionare la data di inizio del report. Sono supportate le date relative: `now`, `now/d`, `now/w-1w` ecc.; modificatori di data supportati: d, w, M, y.
To	Selezionare la data di fine del report. Sono supportate le date relative: `now`, `now/d`, `now/w-1w` ecc.; modificatori di data supportati: d, w, M, y.

Informazioni di sistema

Panoramica

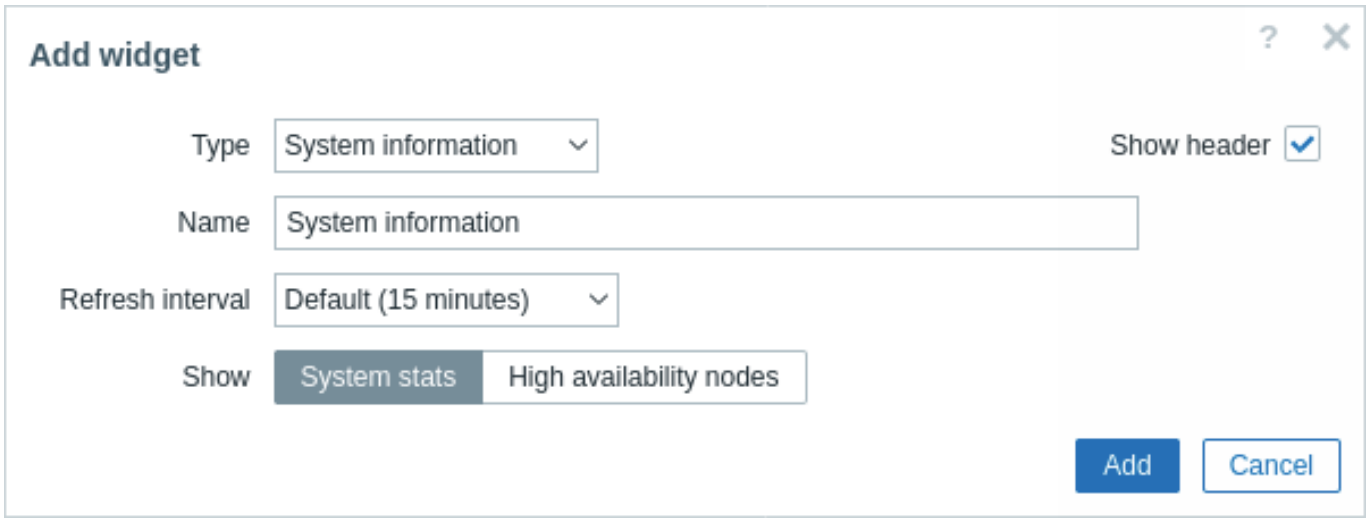
Il widget *System information* visualizza un riepilogo dei dati principali di Zabbix server e del sistema e, quando l'alta disponibilità è abilitata, i dettagli dei nodi ad alta disponibilità.



Visualizza gli stessi dati di *Reports* > *System information*, ad eccezione di *Zabbix server ID*.

Configurazione

Per configurare, selezionare *Informazioni di sistema* come tipo:



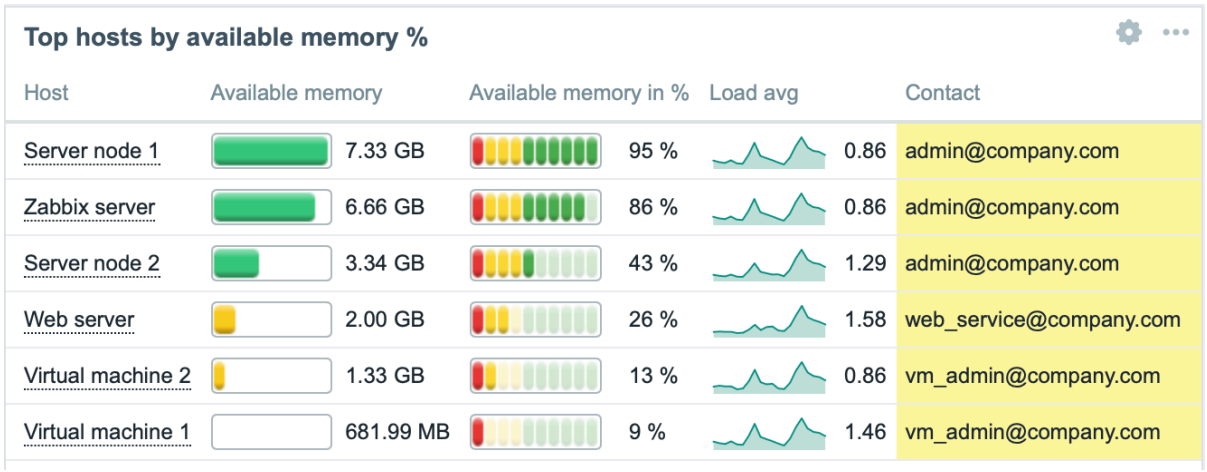
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Mostra	Selezionare cosa visualizzare: **Statistiche di sistema** - visualizza un riepilogo dei dati principali del server Zabbix e del sistema; **Nodi ad alta disponibilità** - visualizza lo stato dei nodi ad alta disponibilità (se il cluster ad alta disponibilità è abilitato).
Mostra i dettagli del controllo degli aggiornamenti software	Selezionare la casella di controllo per visualizzare i dettagli del controllo degli aggiornamenti del software Zabbix. Questa opzione è disponibile solo se il controllo degli aggiornamenti software è abilitato nella configurazione del server Zabbix e se nel campo *Mostra* è selezionato "Statistiche di sistema".

Host principali

Panoramica

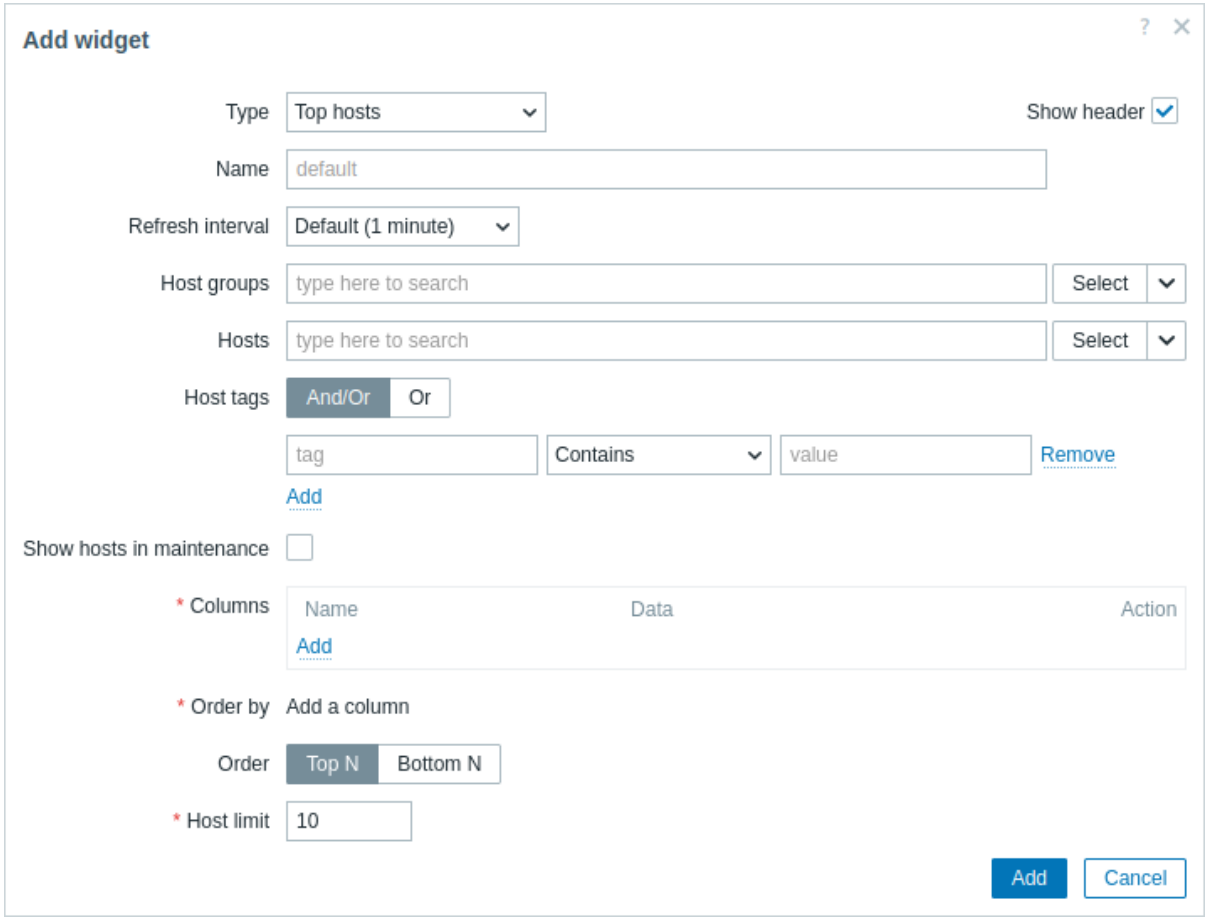
Il widget *Top hosts* visualizza i valori massimi/minimi per gli item selezionati da più host.



Il widget supporta la visualizzazione simultanea di un massimo di 1000 host.

Configurazione

Per configurare, selezionare *Top hosts* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Gruppi host	Selezionare i gruppi host da visualizzare nel widget. In alternativa, selezionare un widget compatibile come origine dati per i gruppi host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Host	Selezionare gli host da visualizzare nel widget. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà mostrato un menu a discesa con gli host corrispondenti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.

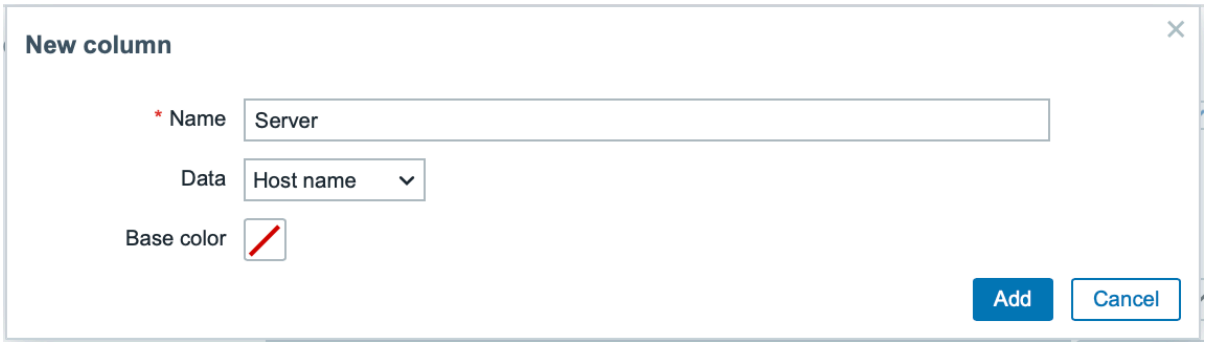
Tag host	Specificare i tag per limitare il numero di host visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Esiste** - include i nomi di tag specificati; **Uguale a** - include i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Contiene** - include i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Non esiste** - esclude i nomi di tag specificati; **Diverso da** - esclude i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Non contiene** - esclude i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Sono disponibili due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or; **Or** - è sufficiente che sia soddisfatta una condizione. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Mostra host in manutenzione	Selezionare questa casella per visualizzare anche gli host in manutenzione (in questo caso, l'icona di manutenzione verrà mostrata accanto al nome dell'host). Per impostazione predefinita non è selezionata.
Colonne	Aggiungere le colonne di dati da visualizzare. L'ordine delle colonne ne determina la visualizzazione da sinistra a destra. Le colonne possono essere riordinate trascinandole verso l'alto o verso il basso tramite la maniglia posta prima del nome della colonna.
Ordina per	Specificare la colonna dall'elenco *Colonne* definito da utilizzare per l'ordinamento *Top N* o *Bottom N*.
Ordine	Specificare l'ordinamento delle righe: **Top N** - in ordine decrescente in base al valore aggregato di *Ordina per*; **Bottom N** - in ordine crescente in base al valore aggregato di *Ordina per*.
Limite host	Numero di righe host da visualizzare (1-1000). Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.

Configurazione delle colonne

È possibile aggiungere colonne di tre tipi di dati: nome host, valore item o testo. L'elenco dei parametri di colonna disponibili dipende dal tipo di dati della colonna e, per il tipo valore item, dal formato del valore.

Colonna Nome host

La colonna Nome host viene utilizzata per visualizzare il nome del host.



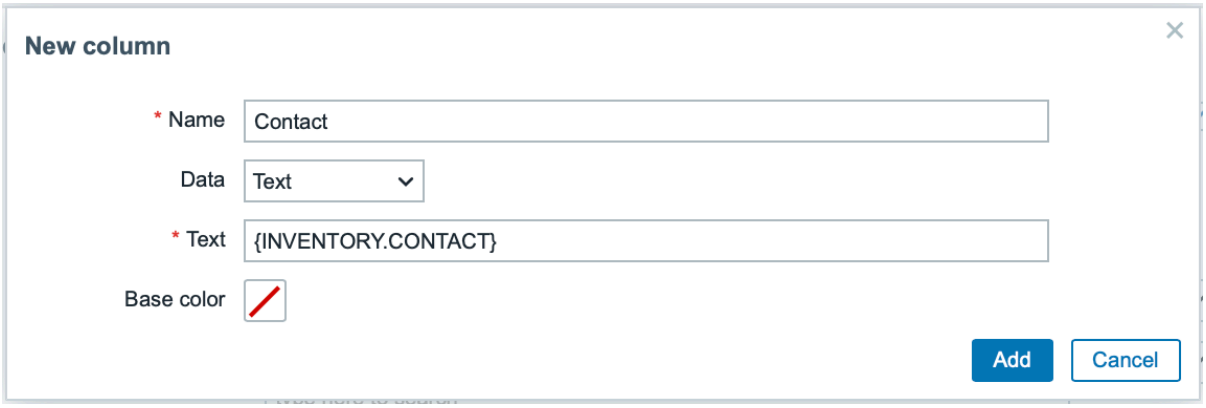
Parametri supportati:

Name	Nome della colonna.
Data	Tipo di dati da visualizzare nella colonna; selezionare **Host name**.

Base color	Colore di sfondo della colonna.

Colonna di testo

La colonna di testo viene utilizzata per visualizzare una qualsiasi stringa di testo specificata.

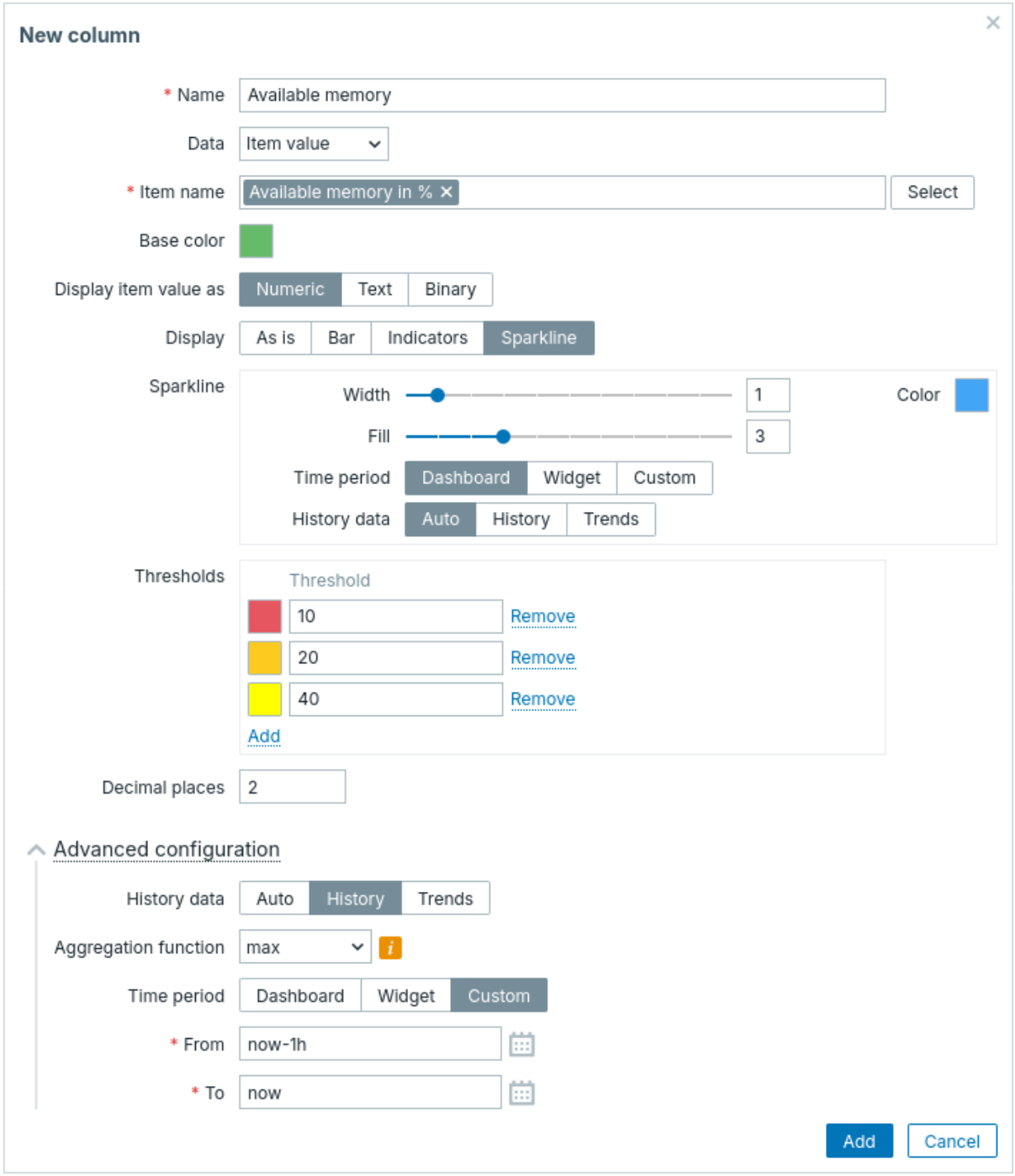


Parametri supportati:

Name	Nome della colonna.
Data	Tipo di dati da visualizzare nella colonna, selezionare **Text**.
Base color	Colore di sfondo della colonna.
Text	Inserire la stringa da visualizzare. Può contenere macro di host e inventario.

Colonna del valore dell'item

La colonna del valore dell'item viene utilizzata per visualizzare il valore dell'item specificato; supporta diversi formati e opzioni di visualizzazione dei valori.



Parametri supportati:

Name	Nome della colonna.
Data	Tipo di dati da visualizzare nella colonna; selezionare **Item value**.
Item name	Selezionare un item; il relativo nome viene utilizzato per trovare e visualizzare gli item con lo stesso nome su tutti gli host selezionati. Se un host dispone di più item di questo tipo, il widget visualizza l'item la cui chiave viene prima in ordine alfabetico (ad esempio, *CPU utilization* con la chiave `proc.cpu.util` invece di `system.cpu.util`). Quando si configura il widget in una dashboard del template, è possibile selezionare solo gli item configurati nel template.
Base color	Colore di sfondo della colonna; colore di riempimento se i dati *Item value* vengono visualizzati come barra/indicatori. Il colore predefinito può essere sostituito da un colore personalizzato se il valore dell'item supera una delle soglie specificate.
Display item value as	Formato di visualizzazione del valore dell'item: **Numeric**, **Text** o **Binary**. L'opzione selezionata determina quali parametri aggiuntivi saranno disponibili. Consultare l'elenco dei parametri per ciascun formato riportato di seguito.
Advanced configuration	

History data	Prelevare i dati dalla cronologia o dai trend: **Auto** - selezione automatica; **History** - preleva i dati dalla cronologia; **Trends** - preleva i dati dai trend. Questo parametro è disponibile solo per i valori numerici degli item. Per i valori testuali e binari, i dati vengono sempre prelevati dalla cronologia.
Aggregation function	Specificare la funzione di aggregazione da utilizzare nell'intervallo *Time period* selezionato: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **count** - visualizza il numero di valori; **sum** - visualizza la somma dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore. Se si seleziona **not used**, viene visualizzato il valore più recente (senza aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo specificato (5 minuti, un'ora, un giorno) invece del valore più recente. Per *min*, *max*, *avg* e *sum* è possibile visualizzare solo dati numerici. Per *count*, i dati non numerici vengono convertiti in numerici.
Time period	Specificare il periodo di tempo da utilizzare per aggregare i valori. Selezionare la sorgente dati per il periodo di tempo: **Dashboard** - utilizza il selettore del periodo di tempo della dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Custom** - utilizza un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, nell'angolo superiore destro del widget viene visualizzata un'icona a forma di orologio che, al passaggio del mouse, indica l'ora impostata. Si noti che, indipendentemente dalla configurazione *Time period* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per il periodo di tempo. Questo parametro non è disponibile se *Aggregation function* è impostato su "not used".
Widget	Inserire o selezionare un widget compatibile come sorgente dati per il periodo di tempo. Questo parametro è disponibile se *Time period* è impostato su "Widget".
From	Inserire o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
To	Inserire o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".

Parametri del **valore numerico dell'item**:

Display	Definire come visualizzare il valore: **As is** - come testo normale; **Bar** - come barra piena con colore di riempimento; **Indicators** - come barra segmentata con colore di riempimento; **Sparkline** - mini grafico a linee.
Thresholds	Specificare i valori di soglia al raggiungimento dei quali deve cambiare il colore di sfondo/riempimento. L'elenco verrà ordinato in ordine crescente al momento del salvataggio. Per gli sparkline, le soglie vengono applicate solo all'ultimo valore dell'item.
Decimal places	Specificare il numero di cifre decimali da visualizzare con il valore.
Bar/Indicators	
Min	Valore minimo.
Max	Valore massimo.
Sparkline	
Width	Impostare lo spessore della linea del grafico utilizzando il cursore oppure inserendo manualmente un valore compreso tra 0 e 10.
Color	Selezionare il colore della linea e del riempimento.
Fill	Impostare il livello di trasparenza del colore di riempimento utilizzando il cursore oppure inserendo manualmente un valore compreso tra 0 e 10.

Time period	Specificare il periodo di tempo dei valori da includere nel grafico sparkline. Selezionare la sorgente dati per il periodo di tempo: **Dashboard** - utilizza il selettore del periodo di tempo della dashboard **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Custom** - utilizza un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, nell'angolo superiore destro del widget viene visualizzata un'icona a forma di orologio che, al passaggio del mouse, indica l'ora impostata. Si noti che, indipendentemente dalla configurazione *Time period* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per il periodo di tempo.
Widget	Inserire o selezionare un widget compatibile come sorgente dati per il periodo di tempo. Questo parametro è disponibile se *Time period* è impostato su "Widget".
From	Inserire o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
To	Inserire o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
History data	Prelevare i dati dalla cronologia o dai trend: **Auto** - selezione automatica; **History** - preleva i dati dalla cronologia; **Trends** - preleva i dati dai trend.

Parametri del **valore testuale dell'item**:

Highlights	Specificare le espressioni regolari la cui corrispondenza deve determinare il cambiamento del colore di sfondo/riempimento.

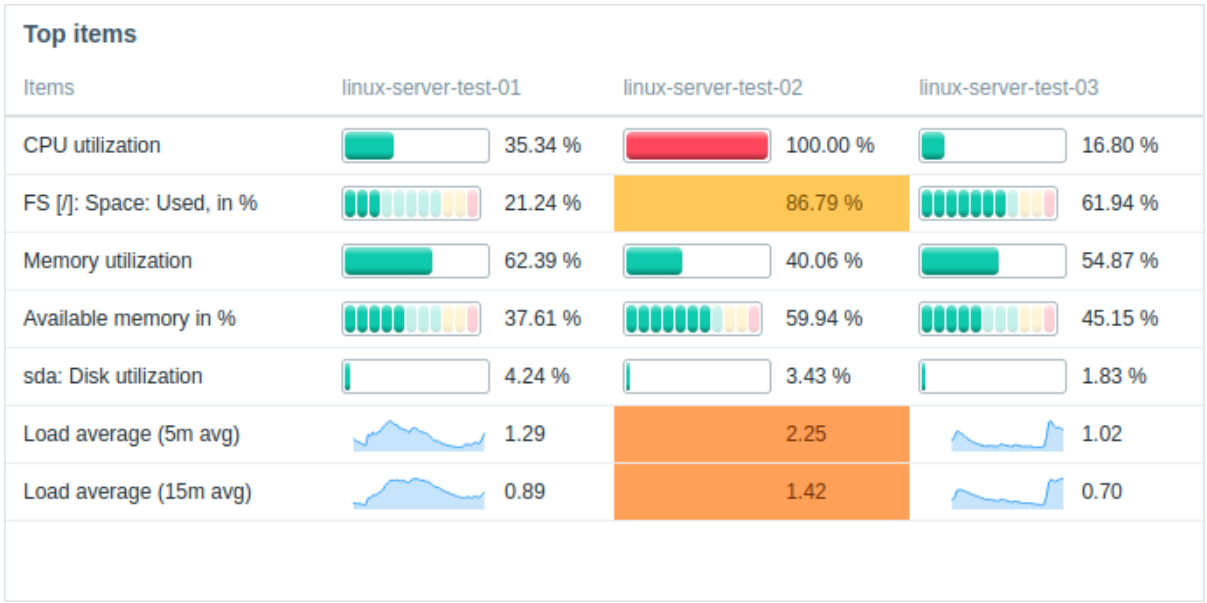
Parametri del **valore binario dell'item**:

Show thumbnail	Specificare se creare e visualizzare una miniatura dell'immagine contenente dati binari oppure visualizzare il collegamento ipertestuale *Show*, che conduce all'immagine a dimensione intera nella colonna del valore.

Elementi principali

Panoramica

Il widget *Top items* visualizza i valori massimi/minimi per gli item selezionati, fornendo una rapida panoramica delle loro prestazioni.

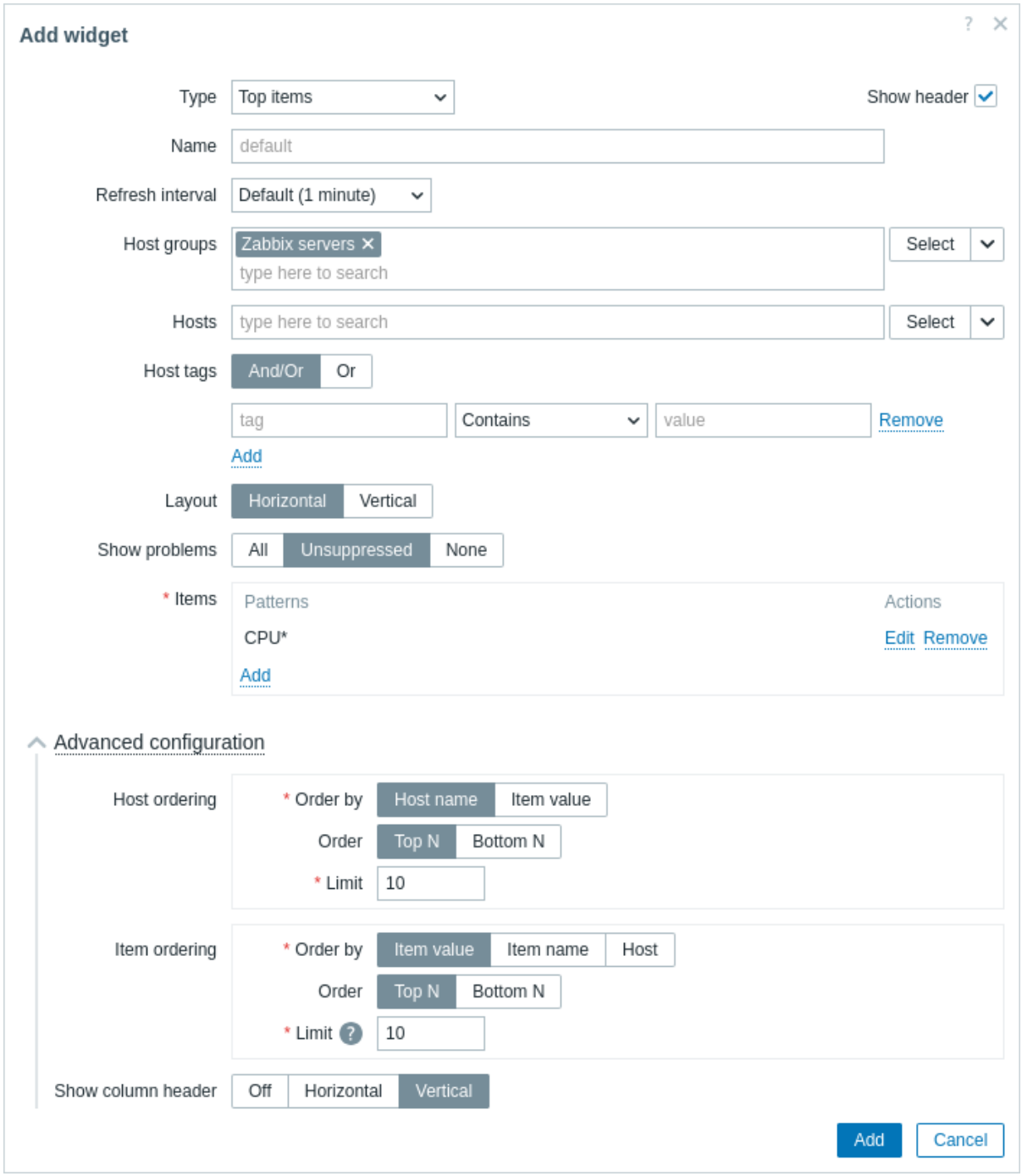


Per impostazione predefinita, il widget visualizza i valori delle ultime 24 ore. Questo periodo può essere modificato in *Administration > General > GUI* (*Max history display period*).

Facendo clic sul valore dell'item si apre il menu dell'item.

Configurazione

Per configurare, seleziona *Top items* come tipo:



Oltre ai parametri comuni a tutti i widget, puoi impostare le seguenti opzioni specifiche:

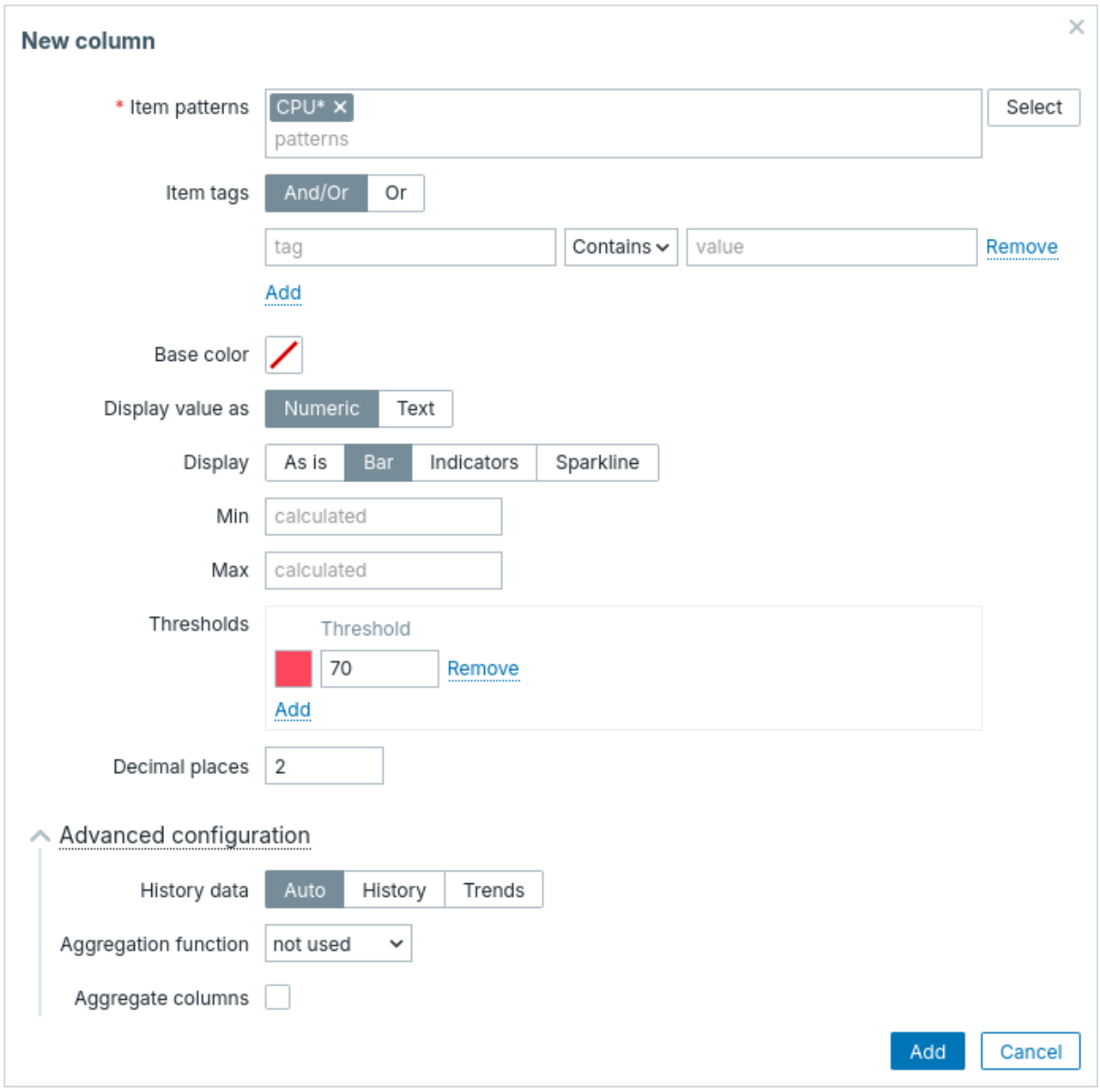
Host groups	Seleziona i gruppi di host. In alternativa, seleziona un widget compatibile come origine dati per i gruppi di host. Questo campo supporta l'autocompletamento, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Questo parametro non è disponibile quando si configura il widget in una template dashboard.

Hosts	Seleziona gli host. In alternativa, seleziona un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta l'autocompletamento, quindi iniziando a digitare il nome di un host verrà mostrato un menu a discesa con gli host corrispondenti. Questo parametro non è disponibile quando si configura il widget in una template dashboard.
Host tags	Specifica i tag per limitare il numero di dati degli host visualizzati nel widget. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Contains** - include i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati i cui valori contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole). Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che una condizione sia soddisfatta.
Layout	Seleziona l'opzione di layout: **Horizontal** - i nomi degli host verranno visualizzati a sinistra; **Vertical** - i nomi degli host verranno visualizzati in alto.
Show problems	Filtra quali problemi visualizzare in base al loro stato (all, unsuppressed, none). I problemi sono indicati con valori colorati degli item, sostituendo le impostazioni di visualizzazione della colonna item. I colori si basano sulla gravità del problema, che può essere personalizzata in *Administration > General > Trigger displaying options*. Le gravità dei problemi possono essere modificate durante l'aggiornamento dei problemi.
Items	Aggiungi pattern di item o item specifici da visualizzare (vedi Configurazione delle colonne).
Advanced configuration	
Host ordering	Seleziona le opzioni di ordinamento per la colonna/riga degli host.
Order by	Ordina gli host in base a: **Host name** - gli host saranno ordinati per nome host; **Item value** - gli host saranno ordinati in base al valore degli item selezionati.
Order	Seleziona se visualizzare i valori più alti o più bassi: **TopN** - i primi N valori; **BottomN** - gli ultimi N valori. Il valore di N viene selezionato nel campo *Limit*.
Limit	Inserisci il limite per gli host visualizzabili (intervallo 1-1000; predefinito 10). Questo valore diventerà il valore di N nel campo *Order*.
Item ordering	Seleziona le opzioni di ordinamento per la colonna/riga degli item.
Order by	Ordina gli item in base a: **Item value** - gli item saranno ordinati per valore dell'item; **Item name** - gli item saranno ordinati per nome dell'item; **Host** - gli item saranno ordinati in base al pattern dell'host selezionato.

Order	Seleziona se visualizzare i valori più alti o più bassi: **TopN** - i primi N valori; **BottomN** - gli ultimi N valori. Il valore di N viene selezionato nel campo *Limit*.
Limit	Inserisci il limite per gli item visualizzabili (intervallo 1-1000; predefinito 10). Questo valore diventerà il valore di N nel campo *Order*.
Show column header	Seleziona le opzioni di visualizzazione dell'intestazione della colonna: **Off** - non visualizzare l'intestazione della colonna; **Horizontal** - visualizzare il testo orizzontalmente nell'intestazione; **Vertical** - visualizzare il testo verticalmente nell'intestazione.

Configurazione delle colonne

Per configurare le colonne degli item, fare clic su *Aggiungi* nel parametro *Items*:



Parametri comuni delle colonne:

Item patterns	Specificare uno o più pattern di item, utilizzando il carattere jolly. In alternativa, selezionare gli item. Quando si configura il widget in una dashboard di template, è possibile selezionare solo gli item configurati nel template.

Item tags	Specificare i tag per limitare il numero di dati degli item visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una condizione.
Base color	Selezionare il colore di sfondo della colonna oppure il colore di riempimento se *Display* è impostato su ”Bar” o ”Indicators”. Si noti che il colore di base può essere sovrascritto dai colori di soglia o di evidenziazione.
Display value as	Formato per la visualizzazione del valore dell’item: **Numeric** o **Text**. L’opzione selezionata determina quali parametri aggiuntivi saranno disponibili. Fare riferimento all’elenco dei parametri per ciascun formato riportato di seguito.

Parametri delle colonne per valori numerici:

Visualizzazione	Definisce come deve essere visualizzato il valore: **Così com’è** - come testo normale; **Barra** - come barra piena con colore; **Indicatori** - come barra segmentata con colore; **Sparkline** - mini grafico a linee.
Min	Valore minimo per la visualizzazione di barre/indicatori.
Max	Valore massimo per la visualizzazione di barre/indicatori.
Larghezza	Imposta lo spessore della linea del grafico utilizzando il cursore oppure inserendo manualmente un valore compreso tra 0 e 10. Questo parametro è disponibile solo per la visualizzazione sparkline.
Riempimento	Imposta il livello di trasparenza del colore di riempimento utilizzando il cursore oppure inserendo manualmente un valore compreso tra 0 e 10. Questo parametro è disponibile solo per la visualizzazione sparkline.
Colore	Seleziona il colore della linea e del riempimento. Questo parametro è disponibile solo per la visualizzazione sparkline.
Per una descrizione dei campi relativi al periodo di tempo e alla selezione dei dati della cronologia, vedere *Configurazione avanzata*.	
Soglie	Specifica i valori di soglia al raggiungimento dei quali deve cambiare il colore dello sfondo/riempimento. L’elenco verrà ordinato in ordine crescente al momento del salvataggio. Per le sparkline, le soglie vengono applicate solo all’ultimo valore dell’item.
Posizioni decimali ***Configurazione avanzata***	Specifica il numero di posizioni decimali da visualizzare con il valore.

Dati della cronologia	Indica se utilizzare i dati della cronologia o dei trend: **Automatico** - selezione automatica; **Cronologia** - utilizza i dati della cronologia; **Trend** - utilizza i dati dei trend. Questo parametro è disponibile solo per i valori numerici degli item. Per i valori testuali, i dati verranno sempre recuperati dalla cronologia.
Funzione di aggregazione	Specifica quale funzione di aggregazione utilizzare nell'*Intervallo di tempo* selezionato: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **count** - visualizza il numero di valori; **sum** - visualizza la somma dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore. Se è selezionato **non utilizzata**, viene visualizzato il valore più recente (senza aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo specificato (5 minuti, un'ora, un giorno), invece del valore più recente. Per *min*, *max*, *avg* e *sum* possono essere visualizzati solo dati numerici. Per *count*, i dati non numerici verranno convertiti in numerici.
Intervallo di tempo	Specifica l'intervallo di tempo da utilizzare per aggregare i valori. Seleziona la sorgente dati per l'intervallo di tempo: **Dashboard** - utilizza il selettore dell'intervallo di tempo del dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Personalizzato** - utilizza un intervallo di tempo personalizzato impostato nei parametri *Da* e *A*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio, che al passaggio del mouse indica l'ora impostata. Nota: indipendentemente dalla configurazione dell'*Intervallo di tempo* del widget, i widget compatibili possono comunque utilizzarlo come sorgente dati per l'intervallo di tempo. Questo parametro non è disponibile se *Funzione di aggregazione* è impostata su "non utilizzata".
Widget	Inserisci o seleziona un widget compatibile come sorgente dati per l'intervallo di tempo. Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Widget".
Da	Inserisci o seleziona l'inizio dell'intervallo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Personalizzato".
A	Inserisci o seleziona la fine dell'intervallo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Intervallo di tempo* è impostato su "Personalizzato".
Aggrega colonne	Se selezionato, raggruppa tutti gli item che corrispondono ai pattern degli item della colonna in un'unica colonna/riga combinata. I valori degli item corrispondenti verranno aggregati utilizzando la *Funzione di aggregazione delle colonne* selezionata e visualizzati sotto il *Nome della colonna combinata*. Predefinito: disattivato (non selezionato).Questo parametro viene visualizzato solo quando *Visualizza valore come* è impostato su **Numerico**.
Funzione di aggregazione delle colonne	Scegli come combinare i valori dei pattern degli item raggruppati. Opzioni disponibili: **min**, **max**, **avg**, **count**, **sum** (predefinita). Questo parametro viene visualizzato solo quando *Aggrega colonne* è selezionato.
Nome della colonna combinata	Inserisci il nome visualizzato per la colonna aggregata. Questo campo è obbligatorio quando *Aggrega colonne* è selezionato e diventa l'intestazione della colonna per i valori combinati.

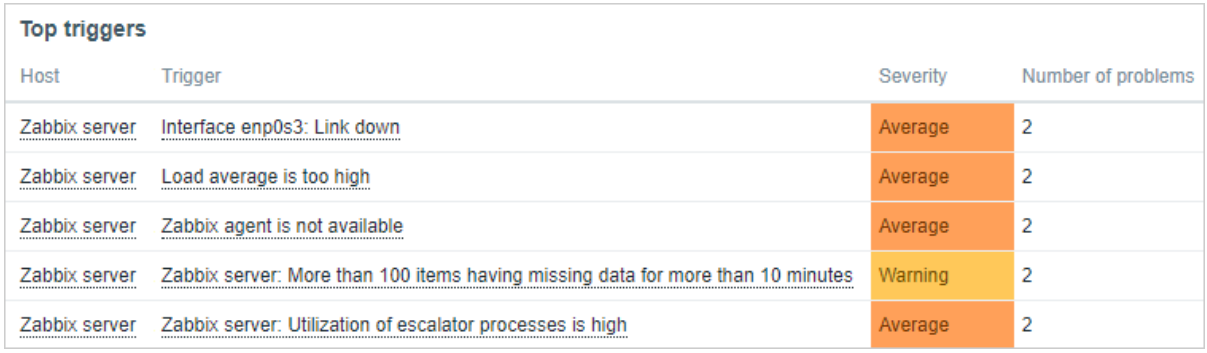
Parametri delle colonne per valori testuali:

Evidenziazioni	Specifica le espressioni regolari in corrispondenza delle quali deve cambiare il colore dello sfondo/riempimento.

Trigger principali

Panoramica

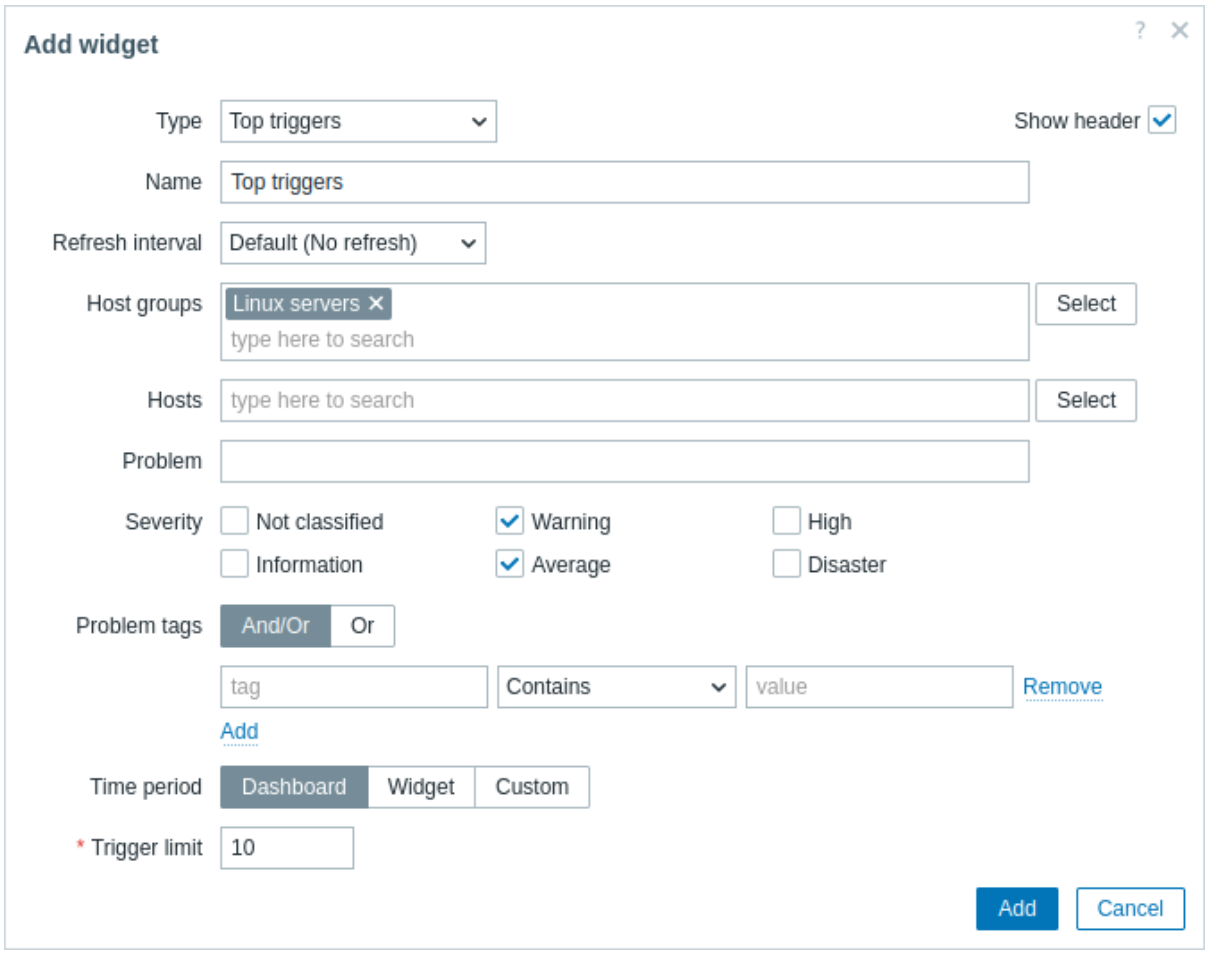
Il widget *Top triggers* visualizza i trigger con il maggior numero di problemi.



Visualizza gli stessi dati di *Reports > Top 100 triggers* e può mostrare fino a 1000 record.

Configurazione

Per configurare il widget, selezionare *Top triggers* come tipo di widget:



Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Host groups	Selezionare i gruppi di host di cui visualizzare i problemi nel widget. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. Specificando un gruppo di host padre vengono selezionati implicitamente anche tutti i gruppi di host nidificati. I problemi di questi gruppi di host verranno visualizzati nel widget; se non viene inserito alcun gruppo di host, verranno visualizzati i problemi di tutti i gruppi di host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Hosts	Selezionare gli host di cui visualizzare i problemi nel widget. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà mostrato un menu a discesa con gli host corrispondenti. Se non viene inserito alcun host, verranno visualizzati i problemi di tutti gli host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.

Problem	È possibile visualizzare i trigger solo per problemi specifici. A tale scopo, inserire la stringa da confrontare nel nome del problema. Le macro non vengono espanse.
Severity	Selezionare le severity dei trigger per filtrare i trigger da visualizzare nel widget. Se non viene selezionata alcuna severity, verranno visualizzati tutti i trigger.
Problem tags	Specificare i tag dei problemi da visualizzare nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ogni condizione: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi di tag e i valori specificati (distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi di tag e i valori specificati (distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una condizione.
Time period	Selezionare l'origine dati per il periodo di tempo: **Dashboard** - utilizza il selettore del periodo di tempo della dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Custom** - utilizza un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, nell'angolo superiore destro del widget verrà visualizzata un'icona a forma di orologio che, al passaggio del mouse, indicherà l'ora impostata. Si noti che, indipendentemente dalla configurazione di *Time period* del widget, i widget compatibili possono comunque usarlo come origine dati per il periodo di tempo.
Widget	Inserire o selezionare un widget compatibile come origine dati per il periodo di tempo. Questo parametro è disponibile se *Time period* è impostato su "Widget".
From	Inserire o selezionare l'inizio del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
To	Inserire o selezionare la fine del periodo di tempo. È supportata la sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.). Questo parametro è disponibile se *Time period* è impostato su "Custom".
Trigger limit	Impostare il numero di trigger da visualizzare. Intervallo di valori possibile: 1-1000.

Panoramica dei trigger

Panoramica

Il widget *Trigger overview* visualizza gli stati correnti dei trigger dell'host come una tabella di blocchi colorati. Fornisce un'istantanea visiva rapida dello stato e dell'attività dei trigger su diversi host.



Il widget utilizza più elementi visivi per mostrare i dettagli dei trigger:

- I colori dei blocchi per i trigger PROBLEM corrispondono alle severità dei problemi.
- I blocchi lampeggianti indicano recenti cambiamenti di stato del trigger (entro gli ultimi 2 minuti).
- Le frecce grigie indicano trigger con dipendenze (passa il mouse sopra di esse per vedere i dettagli).
- Le icone con casella di controllo indicano problemi riconosciuti; tutti i problemi attivi o risolti per il trigger devono essere riconosciuti affinché questa icona venga visualizzata.

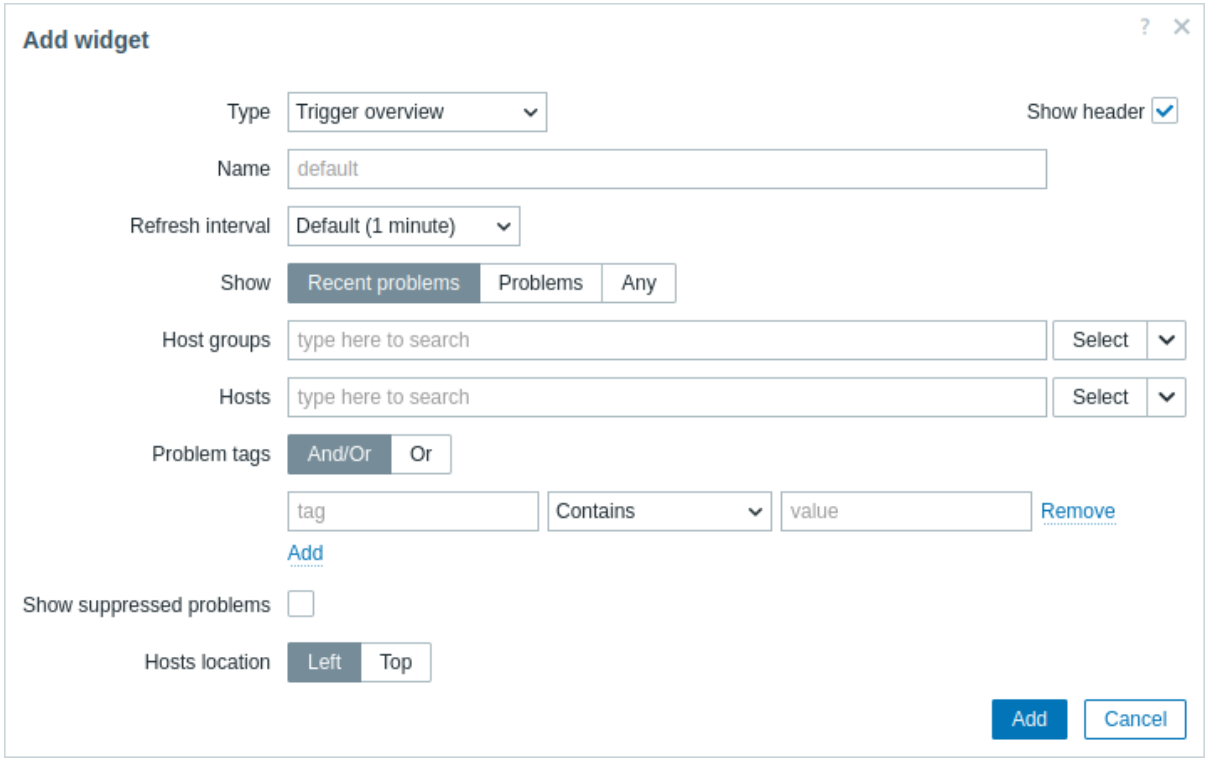
Note:
I colori della severità dei trigger possono essere regolati in *Administration* > *General*. Le severità dei trigger possono essere modificate durante l'aggiornamento dei problemi.

Facendo clic su un blocco si ottengono collegamenti alle informazioni relative al trigger (problemi, configurazione del trigger, ecc.).

Per impostazione predefinita, vengono visualizzati fino a 50 record (configurabile in *Administration* > *General* > *GUI* sotto *Max number of columns and rows in overview tables*). Se esistono più trigger, viene mostrato un messaggio che invita a restringere i criteri di filtro. Questo limite viene applicato prima di qualsiasi filtro aggiuntivo (ad esempio, per tag).

Configurazione

Per configurare il widget, selezionare *Panoramica trigger* come tipo:



Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

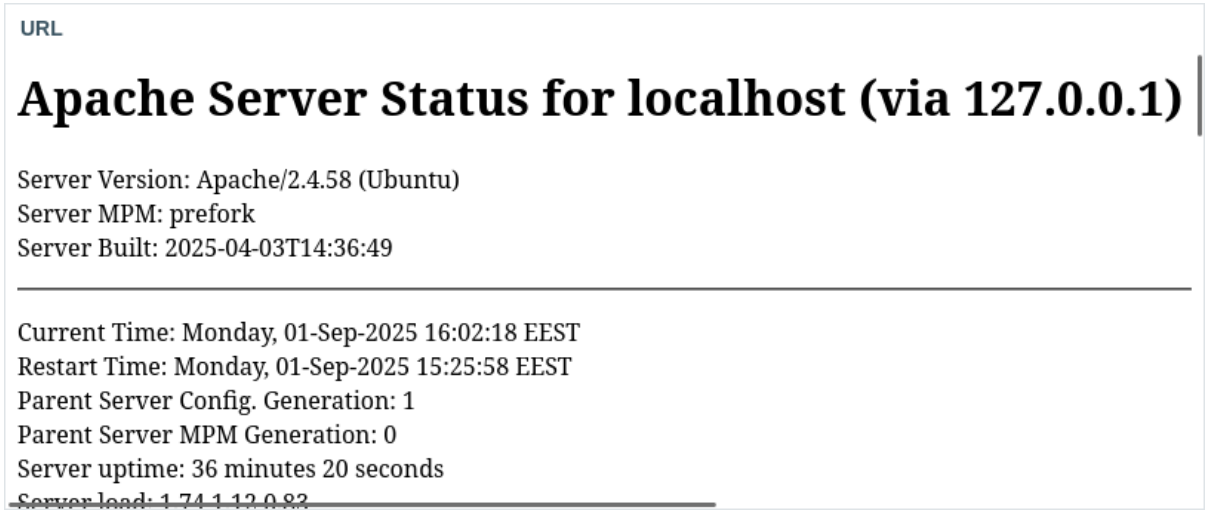
Mostra	Filtra i trigger in base allo stato del trigger: **Problemi recenti** - *(predefinito)* mostra i trigger che recentemente sono stati o sono ancora nello stato PROBLEM (risolti e non risolti); **Problemi** - mostra i trigger che sono nello stato PROBLEM (non risolti); **Qualsiasi** - mostra tutti i trigger.
Gruppi host	Selezionare i gruppi host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà visualizzato un menu a discesa con i gruppi corrispondenti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard template.
Host	Selezionare gli host. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà visualizzato un menu a discesa con gli host corrispondenti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard template.

Tag problema	Specificare i tag per filtrare i trigger visualizzati nel widget. È possibile includere così come escludere tag specifici e valori dei tag. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. **Nota:** Se il parametro *Mostra* è impostato su 'Qualsiasi', verranno visualizzati tutti i trigger anche se sono specificati dei tag. Tuttavia, mentre i cambiamenti recenti dello stato del trigger (visualizzati come blocchi lampeggianti) verranno aggiornati per tutti i trigger, i dettagli dello stato del trigger (il colore della gravità del problema e se il problema è riconosciuto) verranno aggiornati solo per i trigger che corrispondono ai tag specificati. Per ogni condizione sono disponibili diversi operatori: **Esiste** - include i nomi dei tag specificati; **Uguale a** - include i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Contiene** - include i nomi dei tag specificati in cui i valori dei tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Non esiste** - esclude i nomi dei tag specificati; **Diverso da** - esclude i nomi e i valori dei tag specificati (con distinzione tra maiuscole e minuscole); **Non contiene** - esclude i nomi dei tag specificati in cui i valori dei tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Per le condizioni sono disponibili due tipi di calcolo: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una condizione.
Mostra problemi soppressi	Selezionare la casella di controllo per visualizzare i problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione dell'host.
Layout	Selezionare l'opzione di layout: **Orizzontale** - i nomi host verranno visualizzati a sinistra; **Verticale** - i nomi host verranno visualizzati in alto.

URL

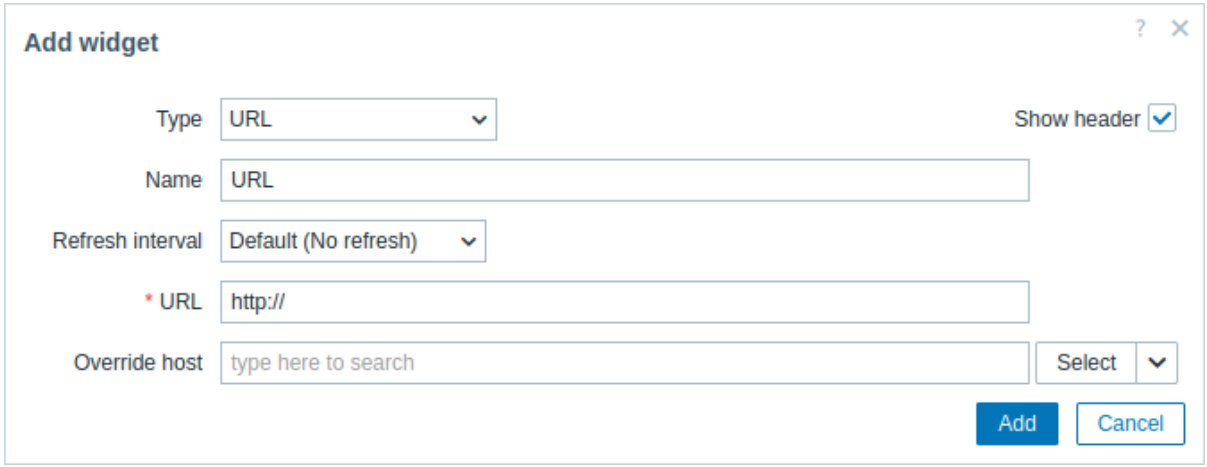
Panoramica

Il widget *URL* visualizza il contenuto recuperato da un URL.



Configurazione

Per configurare, seleziona *URL* come tipo:



Oltre ai parametri comuni a tutti i widget, puoi impostare le seguenti opzioni specifiche:

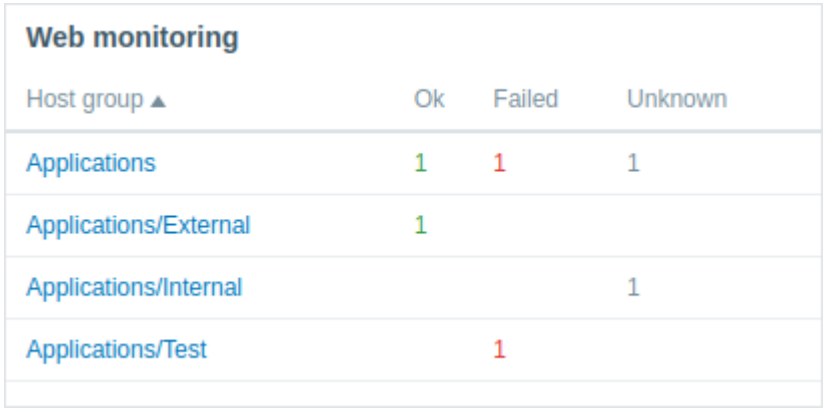
URL	Inserisci l'URL da visualizzare (fino a 2048 caratteri). Gli URL esterni devono iniziare con `http://` o `https://`. Gli URL interni supportano i percorsi relativi (ad esempio, `zabbix.php?action=report.status`). Le macro `{HOST.*}` sono supportate.
Override host	Seleziona un widget compatibile o il selettore host della dashboard come origine dati per gli host. Questo parametro non è disponibile quando si configura il widget su una template dashboard.

Attention:
I browser potrebbero non caricare una pagina HTTP configurata nel widget se il frontend di Zabbix è accessibile tramite HTTPS. Inoltre, potrebbe essere necessario modificare l'intestazione HTTP di risposta X-Frame-Options per visualizzare contenuti esterni al frontend di Zabbix.

Monitoraggio web

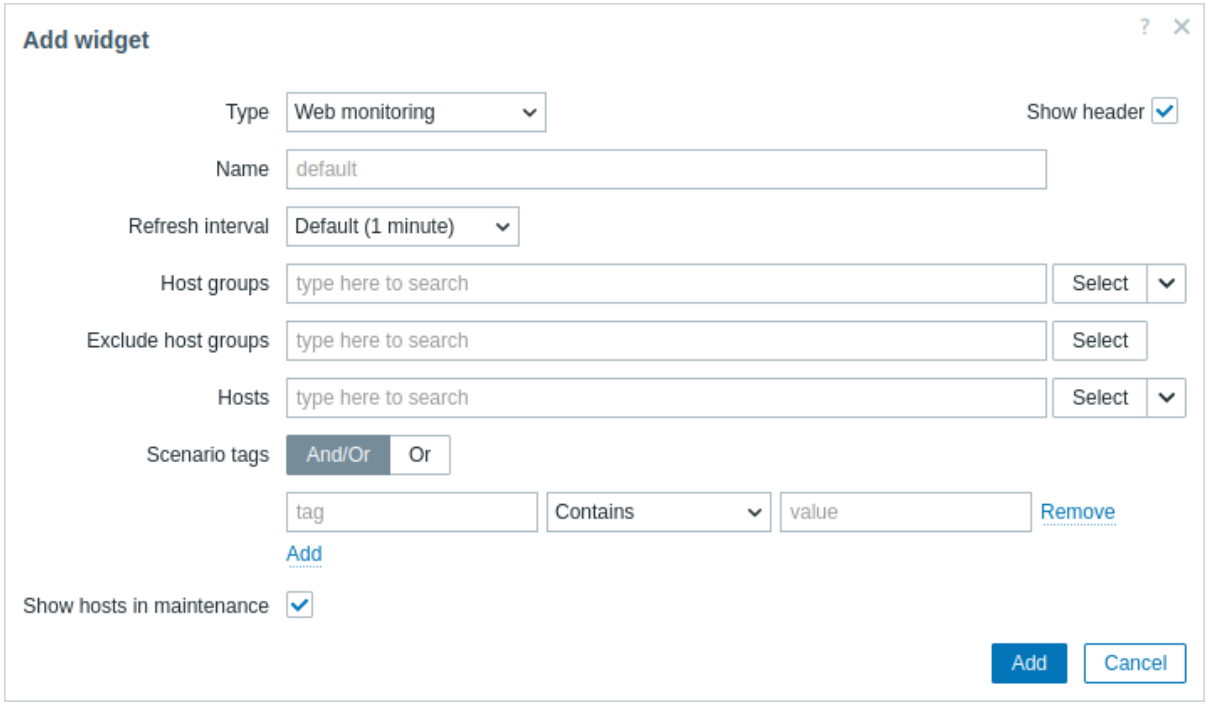
Panoramica

Il widget *Monitoraggio web* visualizza lo stato degli scenari web attivi.



Facendo clic sul nome di un gruppo di host si apre *Monitoraggio* > *Host* > *Scenari web*, dove è possibile visualizzare i dettagli dello scenario.

Configurazione



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Host groups	Selezionare i gruppi host da visualizzare nel widget. In alternativa, selezionare un widget compatibile come origine dati per i gruppi host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà proposto un menu a discesa con i gruppi corrispondenti. Specificando un gruppo host padre vengono selezionati implicitamente anche tutti i gruppi host nidificati. I dati degli host di questi gruppi host verranno visualizzati nel widget; se non viene inserito alcun gruppo host, verranno visualizzati tutti i gruppi host. Questo parametro non è disponibile durante la configurazione del widget in una template dashboard.
Exclude host groups	Selezionare i gruppi host da nascondere nel widget. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà proposto un menu a discesa con i gruppi corrispondenti. Specificando un gruppo host padre vengono selezionati implicitamente anche tutti i gruppi host nidificati. I dati degli host di questi gruppi host non verranno visualizzati nel widget. Ad esempio, gli host 001, 002, 003 possono trovarsi nel Gruppo A e gli host 002, 003 anche nel Gruppo B. Se si seleziona contemporaneamente di *mostrare* il Gruppo A e di *escludere* il Gruppo B, nella dashboard verranno visualizzati solo i dati dell'host 001. Questo parametro non è disponibile durante la configurazione del widget in una template dashboard.
Hosts	Selezionare gli host da visualizzare nel widget. In alternativa, selezionare un widget compatibile o la dashboard come origine dati per gli host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà proposto un menu a discesa con gli host corrispondenti. Se non viene inserito alcun host, verranno visualizzati tutti gli host. Questo parametro non è disponibile durante la configurazione del widget in una template dashboard.

Scenario tags	Specificare i tag per limitare il numero di scenari web visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi di tag e i valori specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Sono disponibili due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una sola condizione.
Show hosts in maintenance	Includere nelle statistiche gli host in manutenzione. Questo parametro è etichettato come *Show data in maintenance* durante la configurazione del widget in una template dashboard.

Monitoraggio

Panoramica

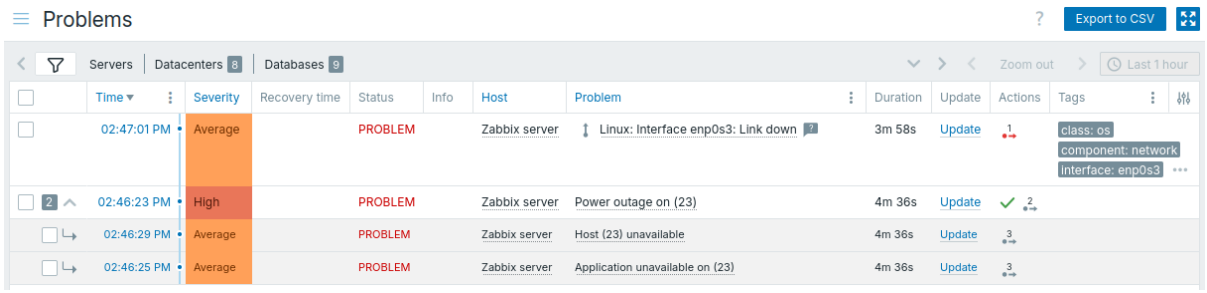
Il menu Monitoring riguarda la visualizzazione dei dati. Qualsiasi informazione che Zabbix è configurato per raccogliere, visualizzare e utilizzare per intraprendere azioni verrà mostrata nelle varie sezioni del menu *Monitoring*.

Problemi

Panoramica

In *Monitoring* > *Problems* puoi vedere quali problemi hai attualmente. I problemi sono quei trigger che si trovano nello stato "Problem".

Per impostazione predefinita, tutti i nuovi problemi sono classificati come problemi causa. È possibile riclassificare manualmente alcuni problemi come problemi sintomo del problema causa. Per maggiori dettagli, vedi eventi causa e sintomo.

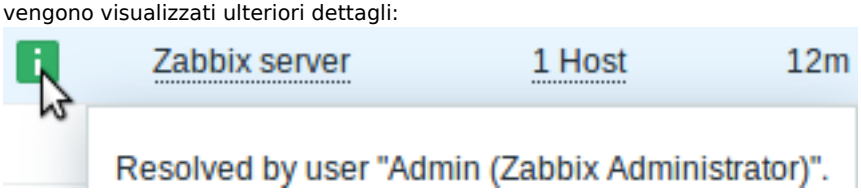
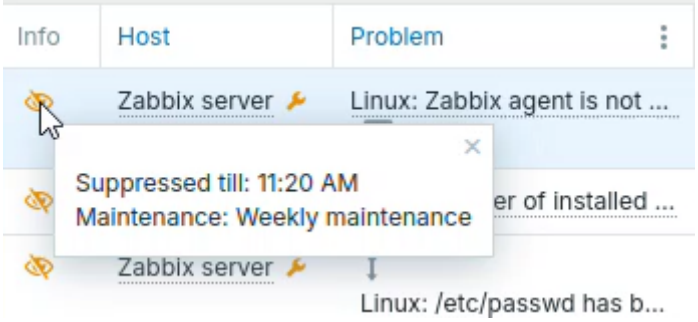


I dettagli del problema sono visualizzati in una tabella personalizzabile. Il pulsante *Personalizza tabella* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. Sono disponibili anche queste opzioni:

- *Vista compatta* - riduce l'altezza delle righe per visualizzare più problemi sullo schermo.
- *Evidenzia riga intera* - usa il colore della gravità del problema per evidenziare i problemi non risolti (non disponibile nei temi ad alto contrasto).

Puoi anche ridimensionare ogni colonna trascinando il suo bordo destro; fai doppio clic sul bordo per ripristinarlo. Tutte le personalizzazioni vengono applicate immediatamente e vengono sempre ricordate per l'utente corrente all'interno della scheda filtro attiva (predefinita o salvata).

Sono disponibili le seguenti colonne:

Colonna	Descrizione
Checkbox	Vengono visualizzate le caselle di selezione per la selezione dei problemi. Le icone accanto alle caselle di selezione hanno il seguente significato:  - il numero di eventi sintomo per il problema causa;  - espandi per mostrare gli eventi sintomo;  - comprimi per nascondere gli eventi sintomo;  - questo è un evento sintomo.
Time	Viene visualizzato l'orario di inizio del problema. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna puoi accedere all'opzione *Mostra timeline*, che attiva o disattiva la timeline visiva e il raggruppamento dei problemi (non disponibile quando *Vista compatta* è abilitata).
Severity	Viene visualizzata la gravità del problema. La gravità del problema si basa originariamente sulla gravità del trigger del problema sottostante; tuttavia, dopo che l'evento si è verificato, può essere aggiornata usando la schermata *Aggiorna problema* screen. Il colore della gravità del problema viene usato come sfondo della cella durante il tempo del problema.
Recovery time	Viene visualizzato il tempo di risoluzione del problema Questa colonna non è disponibile quando *Mostra* nel filtro è impostato su *Problems*.
Status	Viene visualizzato lo stato del problema: **Problem** - problema non risolto **Resolved** - problema risolto di recente. Puoi nascondere i problemi risolti di recente usando il filtro. I problemi nuovi e quelli risolti di recente lampeggiano per 2 minuti. I problemi risolti vengono visualizzati per 5 minuti in totale. Entrambi questi valori sono configurabili in *Administration > General > Trigger displaying options*. Questa colonna non è disponibile quando *Show* nel filtro è impostato su *Problems*.
Info	Viene visualizzata un'icona verde di informazione se un problema viene chiuso dalla correlazione globale o manualmente durante l'aggiornamento del problema. Passando il mouse sull'icona vengono visualizzati ulteriori dettagli:  Se viene mostrato un problema soppresso, viene visualizzata la seguente icona: . Quando l'evento è soppresso dalla manutenzione dell'host, l'icona viene mostrata in arancione: . Se si applicano sia la manutenzione sia la soppressione manuale, la soppressione manuale ha la priorità nel colore dell'icona. Passando il mouse sull'icona vengono visualizzati ulteriori dettagli: 
Host	Viene visualizzato l'host del problema. Facendo clic sul nome dell'host si apre il menu host.

Colonna	Descrizione
Problem	Viene visualizzato il nome del problema. Il nome del problema si basa sul nome del trigger del problema sottostante. Le macro nel nome del trigger vengono risolte nel momento in cui si verifica il problema e i valori risolti non vengono più aggiornati. Facendo clic sul nome del problema si apre il menu evento. Passando il mouse sull'icona dopo il nome del problema viene visualizzata la descrizione del trigger (per quei problemi che ne dispongono). Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna puoi accedere a due opzioni: - *Mostra dati operativi* che visualizza i dati operativi dopo il nome del problema tra parentesi (solo se il campo *Operational data* non è vuoto nella configurazione del trigger). - *Mostra espressione del trigger* che visualizza l'espressione del trigger del problema sotto il nome del problema (non disponibile quando *Vista compatta* è abilitata).
Operational data	Vengono visualizzati dati dinamici e in tempo reale per il problema. Il contenuto dipende dal campo *Operational data* nella configurazione del trigger: se è definita una stringa con macro, viene mostrata qui; se il campo viene lasciato vuoto, vengono invece mostrati gli ultimi valori di tutti gli item nell'espressione del trigger.
Duration	Viene visualizzata la durata del problema. Vedi anche: Durata negativa del problema
Update	Fai clic sul link *Update* per andare alla schermata di aggiornamento del problema, dove è possibile eseguire varie azioni sul problema, tra cui commentarlo e riconoscerlo.
Actions	La cronologia delle attività relative al problema viene visualizzata tramite icone simboliche:  - il problema è stato riconosciuto. Questa icona viene sempre visualizzata per prima.  - sono stati inseriti commenti. Viene visualizzato anche il numero di commenti.  - la gravità del problema è aumentata (ad esempio, Information > Warning).  - la gravità del problema è diminuita (ad esempio, Warning > Information).  - la gravità del problema è cambiata, ma è tornata al livello originale (ad esempio, Warning > Information > Warning).  - il problema è stato soppresso.  - la soppressione del problema è stata rimossa.  - il problema è stato soppresso a causa della manutenzione dell'host.  - la soppressione del problema è stata rimossa dopo la manutenzione dell'host.  - sono state eseguite azioni. Viene visualizzato anche il numero di azioni.  - sono state eseguite azioni, almeno una è in corso. Viene visualizzato anche il numero di azioni.  - sono state eseguite azioni, almeno una è fallita. Viene visualizzato anche il numero di azioni. Passando il mouse sulle icone, vengono visualizzati popup con i dettagli dell'attività. Vedi visualizzazione dei dettagli per saperne di più sulle icone usate nel popup per le azioni eseguite.
Tags	Vengono visualizzati i tag del problema. Inoltre, possono essere visualizzati anche i tag provenienti da un sistema esterno di ticketing (vedi l'opzione *Process tags* durante la configurazione dei webhook). Passando il mouse o facendo clic sull'icona con i tre puntini dopo i tag visualizzati si apre un pop-up con tutti i tag. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna puoi impostare il numero di tag visualizzati (1-3), scegliere come visualizzare i nomi dei tag (completi, abbreviati o nessuno), impostare la priorità di visualizzazione dei tag oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Tag value	Vengono visualizzati i valori dei tag del problema. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna puoi rinominare la colonna, scegliere quali valori dei tag visualizzare oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).

Colonna	Descrizione
Custom text	Viene visualizzato testo personalizzato, valori di macro o una combinazione di entrambi. Facendo clic sul pulsante ⋮ *Opzioni* nell'intestazione della colonna puoi rinominare la colonna, inserire il testo personalizzato o le macro, oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate). Macro supportate: `{HOST.*}` (eccetto `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (eccetto `{ITEM.STATE.ERROR}`), `{TRIGGER.ID}`, `{EVENT.ID}` e macro utente.

Durata negativa del problema

In alcune situazioni comuni è effettivamente possibile avere una durata negativa del problema, cioè quando l'ora di risoluzione del problema è precedente all'ora di creazione del problema, ad esempio:

- Se un host è monitorato da un proxy e si verifica un errore di rete, che comporta l'assenza di dati ricevuti dal proxy per un certo periodo, il trigger nodata(/host/key) verrà attivato dal server. Quando la connessione viene ripristinata, il server riceverà dati dell'item dal proxy con un timestamp nel passato. Quindi, il problema nodata(/host/key) verrà risolto e avrà una durata negativa;
- Quando i dati dell'item che risolvono l'evento del problema vengono inviati da Zabbix sender e contengono un timestamp precedente all'ora di creazione del problema, verrà visualizzata anche una durata negativa del problema.

Note:
La durata negativa del problema non influisce in alcun modo sul calcolo dello SLA o sul report di disponibilità di uno specifico trigger; non riduce né estende il tempo di problema.

Opzioni di modifica in massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica in massa:

- *Aggiornamento di massa* - aggiorna i problemi selezionati accedendo alla schermata aggiornamento dei problemi

Per usare questa opzione, seleziona le caselle di controllo accanto ai rispettivi problemi, quindi fai clic sul pulsante *Aggiornamento di massa*.

Pulsanti

Il pulsante a destra offre la seguente opzione:

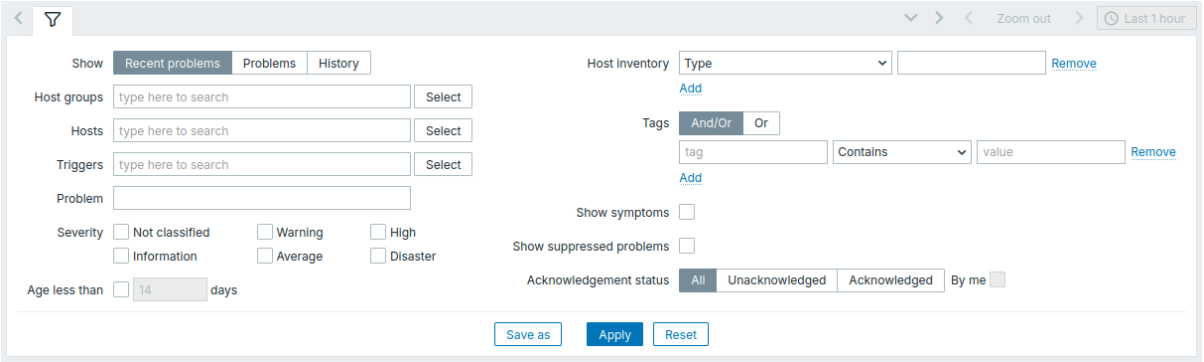
Export to CSV	Esporta il contenuto di tutte le pagine in un file CSV.

I pulsanti della modalità di visualizzazione, essendo comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo i problemi di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il filtro si trova sopra la tabella. Le impostazioni preferite del filtro possono essere salvate come schede e quindi richiamate rapidamente facendo clic sulle schede sopra il filtro.

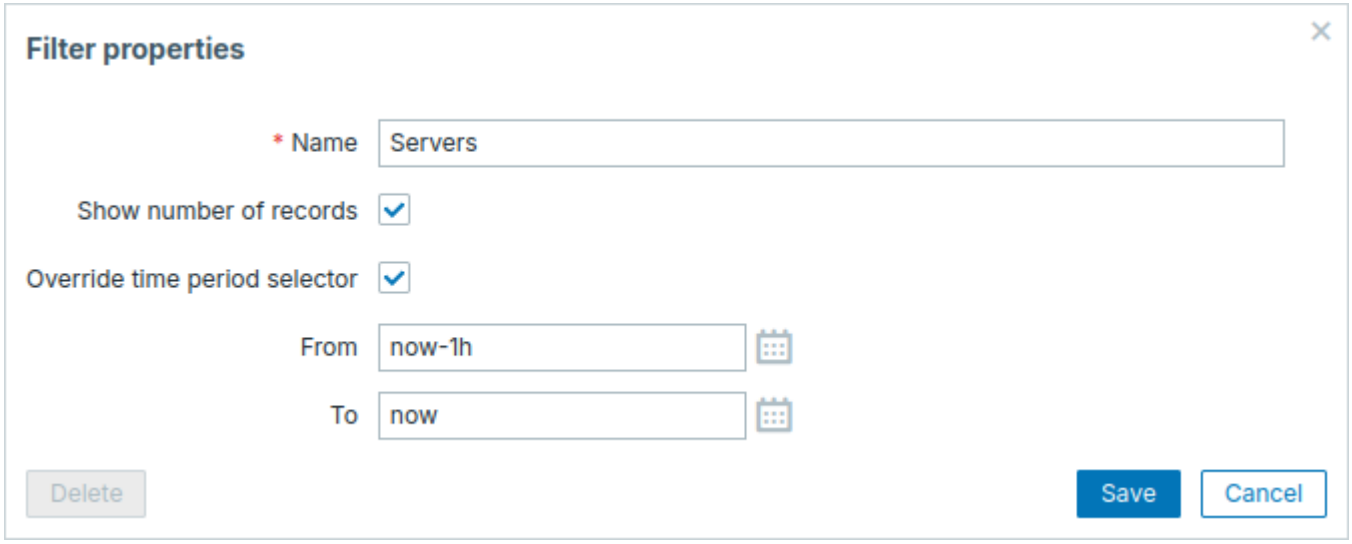


Parameter	Description
Show	Filtra per stato del problema: **Recent problems** - vengono visualizzati i problemi non risolti e quelli risolti di recente (predefinito) **Problems** - vengono visualizzati i problemi non risolti **History** - viene visualizzata la cronologia di tutti gli eventi
Host groups	Filtra per uno o più gruppi di host. Specificando un gruppo di host padre, vengono implicitamente selezionati tutti i gruppi di host nidificati.
Hosts	Filtra per uno o più host.
Triggers	Filtra per uno o più trigger.
Problem	Filtra per nome del problema.
Severity	Filtra per gravità del trigger (problema).
Age less than	Filtra in base all'età del problema.
Host inventory	Filtra per tipo e valore dell'inventario.
Tags	Filtra per nome e valore del tag dell'evento. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole/minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or **Or** - è sufficiente che una condizione sia soddisfatta Quando viene applicato il filtro, i tag specificati qui verranno visualizzati per primi con il problema, a meno che non vengano sovrascritti dall'elenco *Tag display priority* (vedere sotto).
Show symptoms	Selezionare la casella di controllo per visualizzare su una riga separata i problemi classificati come sintomi.
Show suppressed problems	Selezionare la casella di controllo per visualizzare i problemi che altrimenti verrebbero soppressi e nascosti a causa della manutenzione dell'host o soppressi manualmente da un utente.
Acknowledgement status	Filtra per visualizzare tutti i problemi, solo i problemi non riconosciuti oppure solo i problemi riconosciuti. Selezionare la casella di controllo aggiuntiva per filtrare i problemi che sono stati riconosciuti da te in passato.

Schede per i filtri preferiti

I set di parametri di filtro usati frequentemente possono essere salvati in schede.

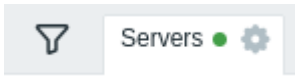
Per salvare un nuovo set di parametri di filtro, apri la scheda principale, configura le impostazioni del filtro e poi fai clic sul pulsante *Save as*. In una nuova finestra popup, definisci le *Filter properties*.



Parameter	Description
Name	Il nome del filtro da visualizzare nell'elenco delle schede.
Show number of records	Seleziona questa opzione se vuoi che il numero di problemi venga visualizzato accanto al nome della scheda.
Override time period selector	Seleziona questa opzione per impostare un periodo di tempo predefinito specifico per questo set di filtri. Se impostata, potrai modificare il periodo di tempo per questa scheda solo aggiornando le impostazioni del filtro. Per le schede senza un periodo di tempo personalizzato, l'intervallo temporale può essere modificato facendo clic sul pulsante di selezione del tempo nell'angolo in alto a destra (il nome del pulsante dipende dall'intervallo di tempo selezionato: This week, Last 30 minutes, Yesterday, ecc.). Questa opzione è disponibile solo per i filtri in *Monitoring* > *Problems*.
From/To	Inizio e fine del periodo di tempo in sintassi assoluta (`Y-m-d H:i:s`) o relativa (`now-1d`). Disponibile se *Set custom time period* è selezionato.

Una volta salvato, il filtro viene creato come scheda di filtro con nome e viene attivato immediatamente.

Per modificare le proprietà del filtro di un filtro esistente, fai clic sul simbolo dell'ingranaggio accanto al nome della scheda attiva.



Note:

- Per nascondere l'area del filtro, fai clic sul nome della scheda corrente. Fai di nuovo clic sul nome della scheda attiva per riaprire l'area del filtro.
- È supportata la navigazione da tastiera: usa le frecce per passare da una scheda all'altra, premi *Enter* per aprire.
- I pulsanti sinistra/destra sopra il filtro possono essere usati per passare tra i filtri salvati. In alternativa, il pulsante rivolto verso il basso apre un menu a discesa con tutti i filtri salvati e puoi fare clic su quello desiderato.
- Se le impostazioni di un filtro salvato sono state modificate (ma non salvate), viene visualizzato un punto verde dopo il nome del filtro. Per aggiornare il filtro in base alle nuove impostazioni, fai clic sul pulsante *Update*, visualizzato al posto del pulsante *Save as*.
- Il layout delle colonne della tabella viene salvato per ciascuna scheda filtro. Per personalizzare il layout di un filtro salvato, apri prima quella scheda filtro, quindi personalizza le colonne.
- Le schede filtro, le impostazioni del filtro e il layout delle colonne della tabella per ciascuna scheda vengono memorizzati nel profilo utente e ripristinati quando l'utente apre di nuovo la pagina.

Note:
Per condividere i filtri, copia e invia ad altri un URL di un filtro attivo. Dopo aver aperto questo URL, gli altri utenti potranno salvare questo set di parametri come filtro permanente nel proprio account Zabbix.
 Vedi anche: Page parameters.

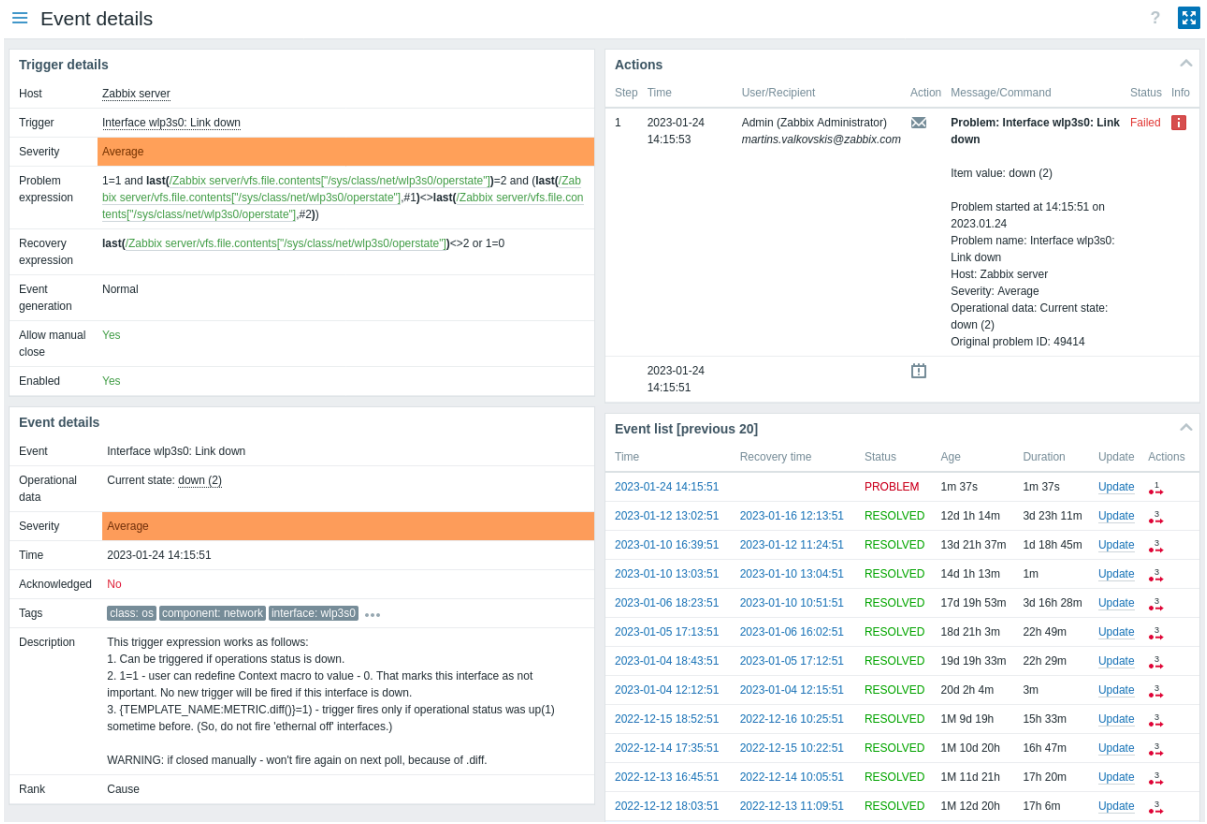
Pulsanti filtro

Apply	Applica i criteri di filtro specificati (senza salvare).

Reset	Ripristina il filtro corrente e torna ai parametri salvati della scheda corrente . Nella scheda principale, questo cancellerà il filtro.
Save as	Salva i parametri del filtro corrente in una nuova scheda. Disponibile solo nella scheda principale.
Update	Sostituisci i parametri della scheda con i parametri attualmente specificati. Non disponibile nella scheda principale.

Visualizzazione dei dettagli

Gli orari di inizio del problema e di ripristino in *Monitoring* > *Problems* sono collegamenti. Facendo clic su di essi si aprono ulteriori dettagli dell'evento.



Nei dettagli dell'evento, è possibile vedere se l'evento è stato soppresso a causa della manutenzione oppure soppresso manualmente da un utente.

Nota che la gravità del problema può differire per il trigger e per l'evento di problema, se è stata aggiornata per l'evento di problema usando la schermata *Update problem* screen.

Nell'elenco delle azioni, le seguenti icone vengono usate per indicare il tipo di attività:

- - evento di problema generato.
- - messaggio inviato.
- - evento di problema riconosciuto.
- - evento di problema non riconosciuto.
- - è stato aggiunto un commento.
- - la gravità del problema è aumentata (ad esempio, Information > Warning).
- - la gravità del problema è diminuita (ad esempio, Warning > Information).
- - la gravità del problema è stata modificata, ma è tornata al livello originale (ad esempio, Warning > Information > Warning).

-  - è stato eseguito un comando remoto.
-  - l'evento di problema è stato ripristinato.
-  - il problema è stato chiuso manualmente.
-  - il problema è stato soppresso.
-  - la soppressione del problema è stata rimossa.
-  - il problema è stato soppresso a causa della manutenzione dell'host.
-  - la soppressione del problema è stata rimossa dopo la manutenzione dell'host.
-  - il problema è stato convertito in un problema sintomo.
-  - il problema è stato convertito in un problema causa.

Cause e problemi di sintomo

Panoramica

Per impostazione predefinita, tutti i nuovi problemi sono classificati come problemi causa. È possibile riclassificare manualmente determinati problemi come problemi sintomo del problema causa.

Ad esempio, un'interruzione di corrente può essere la causa principale effettiva per cui un host non è raggiungibile o un servizio non è disponibile. In questo caso, i problemi "host non è raggiungibile" e "servizio non è disponibile" devono essere classificati come problemi sintomo di "interruzione di corrente", ovvero il problema causa.

La gerarchia causa-sintomo supporta solo due livelli. Un problema che è già un sintomo non può avere problemi sintomo "subordinati"; tutti i problemi assegnati come sintomi a un problema sintomo diventeranno sintomi dello stesso problema causa.

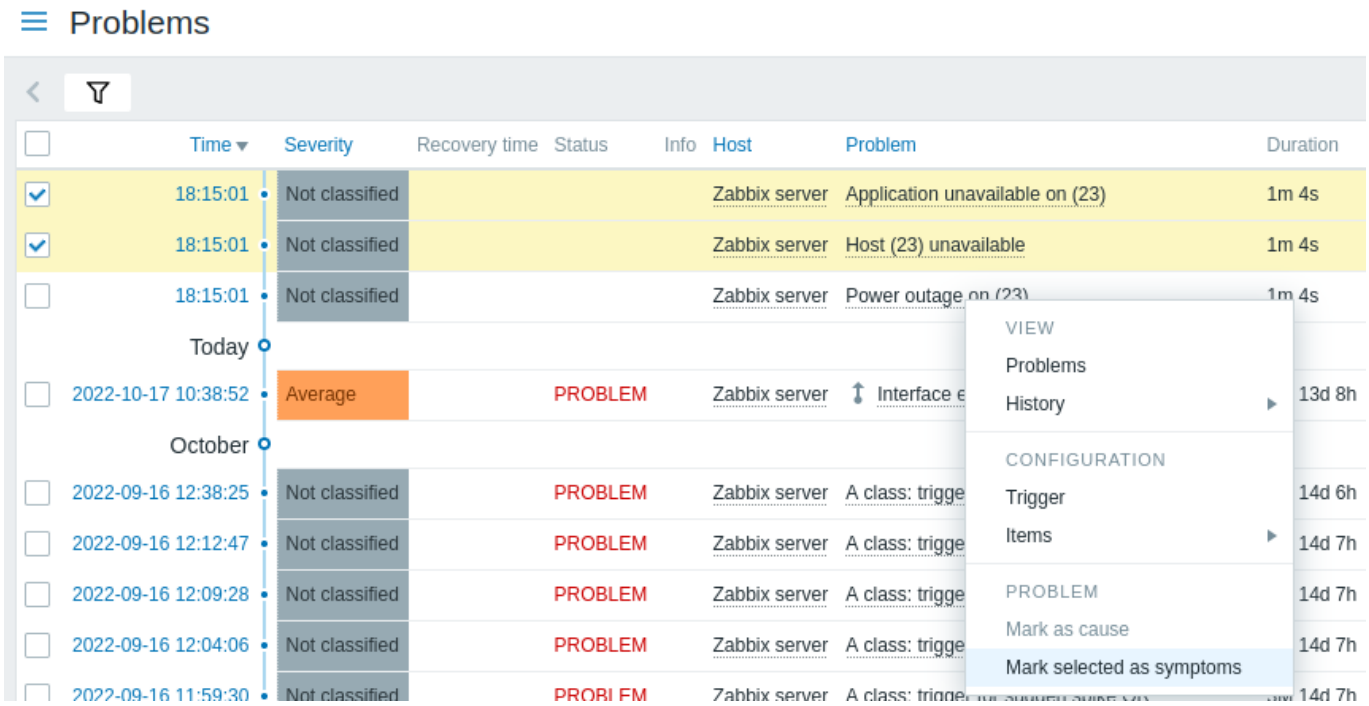
Solo i problemi causa vengono conteggiati nei totali dei problemi nelle mappe, nei widget della dashboard come *Problemi per gravità* o *Host con problemi*, ecc. Tuttavia, la classificazione dei problemi non influisce sui servizi.

Un problema sintomo può essere collegato a un solo problema causa. I problemi sintomo non vengono risolti automaticamente se il problema causa viene risolto o chiuso.

Configurazione

Per riclassificare un problema come problema sintomo, selezionalo prima nell'elenco dei problemi. È possibile selezionare uno o più problemi.

Quindi vai al problema causa e, nel relativo menu contestuale, fai clic sull'opzione *Mark selected as symptoms*.



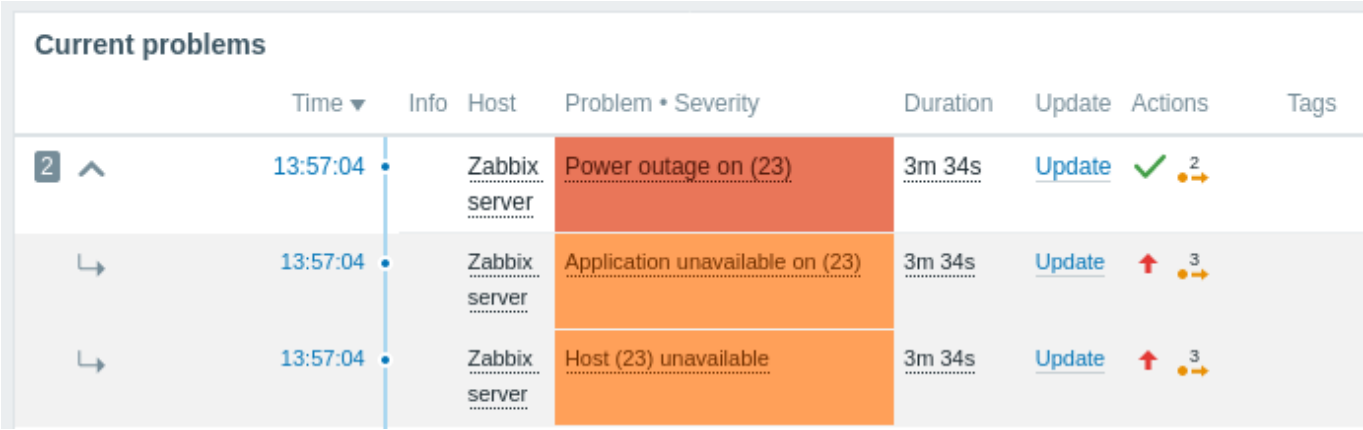
Dopo di ciò, i problemi selezionati verranno aggiornati dal server in problemi sintomo del problema causa.

Mentre lo stato del problema viene aggiornato, viene visualizzato in uno dei due modi seguenti:

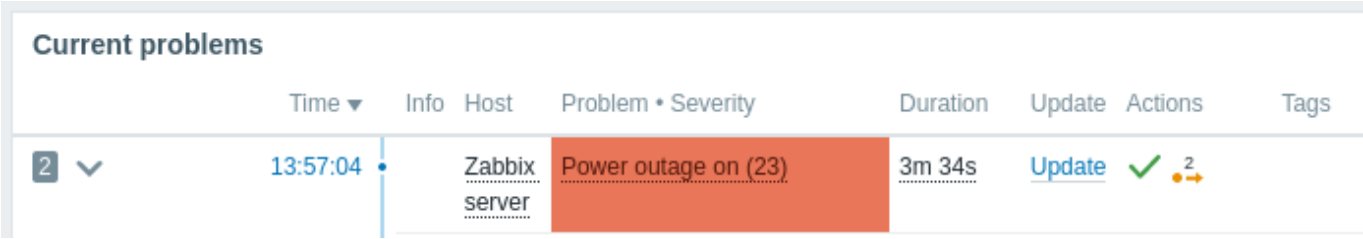
- Nella colonna Status viene visualizzato uno stato ”UPDATING” lampeggiante;
- Nella colonna Info viene visualizzata un’icona lampeggiante oppure (questo vale se nel filtro è selezionato solo *Problems* e quindi la colonna Status non viene mostrata).

Visualizzazione

I problemi sintomo vengono visualizzati sotto il problema causa e contrassegnati di conseguenza in *Monitoring* > *Problems* (e nel widget della dashboard *Problems*) - con un’icona, un carattere più piccolo e uno sfondo diverso.



Nella vista compressa, viene visualizzato solo il problema causa; l’esistenza dei problemi sintomo è indicata dal numero all’inizio della riga e dall’icona per espandere la vista.



È inoltre possibile visualizzare i problemi sintomo anche con carattere normale e su una riga separata. A tale scopo, selezionare *Show symptoms* nelle impostazioni del filtro o nella configurazione del widget.

Ripristino a problema causa

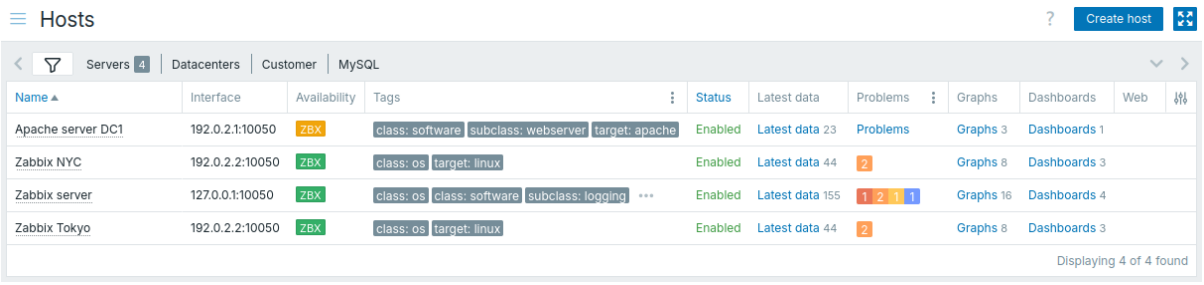
Un problema sintomo può essere ripristinato a un problema causa. Per farlo, è possibile:

- fare clic sull’opzione *Segna come causa* nel menu contestuale del problema sintomo;
- selezionare l’opzione *Converti in causa* nella schermata di aggiornamento del problema e fare clic su *Aggiorna* (questa opzione funzionerà anche se sono selezionati più problemi).

Host

Panoramica

La sezione *Monitoring* > *Hosts* mostra un elenco completo degli host monitorati con informazioni dettagliate sull’interfaccia dell’host, disponibilità, tag, problemi correnti, stato (abilitato/disabilitato) e collegamenti per navigare facilmente ai dati più recenti dell’host, alla cronologia dei problemi, ai grafici, ai dashboard e agli scenari web.



I dettagli dell'host sono visualizzati in una tabella personalizzabile. Il pulsante *Customize table* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinando il bordo destro; fare doppio clic sul bordo per ripristinarlo. Tutte le personalizzazioni vengono applicate immediatamente e vengono sempre ricordate per l'utente corrente all'interno della scheda filtro attiva (predefinita o salvata).

Sono disponibili le seguenti colonne:

Column	Description
Name	Il nome visibile dell'host. Facendo clic sul nome si apre il menu host. Un'icona arancione a forma di chiave inglese dopo il nome indica che questo host è in manutenzione. Fare clic sull'intestazione della colonna per ordinare gli host per nome in ordine crescente o decrescente.
Interface	Viene visualizzata l'interfaccia principale dell'host.
Availability	Viene visualizzata la disponibilità dell'host per ciascuna interfaccia configurata. Le icone di disponibilità rappresentano lo stato corrente dell'interfaccia dell'host su Zabbix server. Pertanto, se si disabilita un host nel frontend, la sua disponibilità verrà aggiornata dopo che Zabbix server avrà sincronizzato le modifiche di configurazione. Allo stesso modo, se si abilita un host, la sua disponibilità verrà aggiornata dopo che Zabbix server avrà sincronizzato le modifiche di configurazione e interrogato l'host. Le icone di disponibilità rappresentano solo i tipi di interfaccia configurati (Agent, SNMP, IPMI, JMX). Passando il mouse sull'icona viene visualizzato un pop-up con un elenco di tutte le interfacce dello stesso tipo, con dettagli, stato ed errori. Per l'interfaccia Agent, il pop-up mostra le interfacce (passive) e i controlli attivi. Se un host ha solo controlli attivi, viene visualizzata l'icona dell'interfaccia Agent anche se non è configurata alcuna interfaccia Agent per l'host. La colonna è vuota per gli host senza interfacce. Lo stato di una singola interfaccia dell'host è determinato dalla connessione tra un item abilitato che utilizza l'interfaccia e l'host. Lo stato può essere: **Available** - la connessione con l'host è riuscita; **Not available** - la connessione con l'host non è riuscita (timeout, problemi di firewall, ecc.); **Unknown** - il tentativo di connessione con l'host non è stato effettuato oppure il risultato è sconosciuto. Per ulteriori dettagli su come Zabbix server determina lo stato dell'interfaccia, vedere Unknown interface status e Unreachable/unavailable host interface settings. Lo stato di tutte le interfacce dell'host di un singolo tipo (Agent, SNMP, IPMI, JMX) è determinato da quelle interfacce che sono utilizzate da almeno un item abilitato. Lo stato è indicato dal colore dell'icona: **Green** - tutte le interfacce sono disponibili; **Yellow** - almeno un'interfaccia non è disponibile e almeno una è disponibile o sconosciuta; **Red** - tutte le interfacce non sono disponibili; **Gray** - almeno un'interfaccia è sconosciuta, ma nessuna non è disponibile. **Disponibilità dei controlli attivi.** Se sull'host è abilitato almeno un controllo attivo, anche i controlli attivi influiscono sulla disponibilità complessiva dell'interfaccia Agent come descritto sopra. Per determinare la disponibilità dei controlli attivi, i messaggi heartbeat vengono inviati nel thread dei controlli attivi dell'agent. La frequenza dei messaggi heartbeat è controllata dal parametro `HeartbeatFrequency` nella configurazione di Zabbix agent o agent 2 (predefinito 60 secondi, intervallo 0-3600). I controlli attivi sono considerati non disponibili quando l'heartbeat del controllo attivo è più vecchio di 2 x HeartbeatFrequency secondi. **Nota:** gli agent Zabbix precedenti alla versione 6.2.x non inviano heartbeat dei controlli attivi, quindi la disponibilità dei relativi host rimane sconosciuta.

Column	Description
Tags	Vengono visualizzati i tag dell'host. Passando il mouse o facendo clic sull'icona con i tre puntini dopo i tag visualizzati si apre un pop-up con tutti i tag. Facendo clic sul pulsante *Options* nell'intestazione della colonna è possibile impostare il numero di tag visualizzati (1–3), scegliere come visualizzare i nomi dei tag (completi, abbreviati o nessuno), impostare la priorità di visualizzazione dei tag oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Tag value	Vengono visualizzati i valori dei tag dell'host. Facendo clic sul pulsante *Options* nell'intestazione della colonna è possibile rinominare la colonna, scegliere quali valori dei tag visualizzare oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Status	Stato dell'host - *Enabled* o *Disabled*. Fare clic sull'intestazione della colonna per ordinare gli host per stato in ordine crescente o decrescente.
Latest data	Facendo clic sul collegamento si apre la sezione *Monitoring* > *Latest data* con tutti i dati più recenti raccolti dall'host. Il numero di item con dati più recenti è visualizzato in grigio.
Problems	Viene visualizzato il numero di problemi aperti per un host, ordinati per gravità. Il colore di ciascuna icona del problema indica la gravità, mentre il numero mostra il conteggio dei problemi per quella gravità. Facendo clic su un'icona si apre *Monitoring* > *Problems* per quell'host; se un host non ha problemi, viene invece mostrato un collegamento *Problems*. Facendo clic sul pulsante *Options* nell'intestazione della colonna si accede all'opzione *Show suppressed problems*, che include i problemi che altrimenti verrebbero soppressi (nascosti) a causa della manutenzione dell'host o della singola soppressione dei problemi.
Graphs	Facendo clic sul collegamento vengono visualizzati i grafici configurati per l'host. Il numero di grafici è visualizzato in grigio. Se un host non ha grafici, non viene visualizzato alcun collegamento.
Dashboards	Facendo clic sul collegamento vengono visualizzati i dashboard configurati per l'host. Il numero di dashboard è visualizzato in grigio. Se un host non ha dashboard, non viene visualizzato alcun collegamento.
Web	Facendo clic sul collegamento vengono visualizzati gli scenari web configurati per l'host. Il numero di scenari web è visualizzato in grigio. Se un host non ha scenari web, non viene visualizzato alcun collegamento.
Custom text	Vengono visualizzati testo personalizzato, valori di macro o una combinazione di entrambi. Facendo clic sul pulsante *Options* nell'intestazione della colonna è possibile rinominare la colonna, inserire il testo personalizzato o le macro, oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate). Macro supportate: `{HOST.*}` (eccetto `{HOST.METADATA}`), `{INVENTORY.*}` e macro utente.

Pulsanti

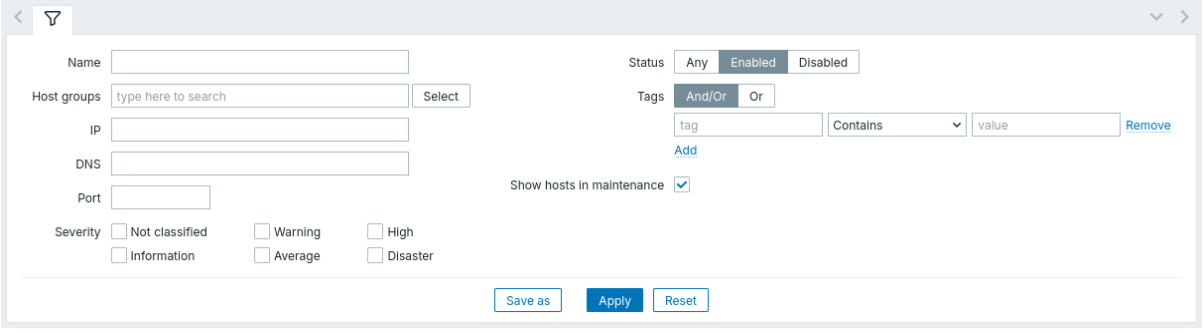
Crea host consente di creare un nuovo host. Questo pulsante è disponibile solo per gli utenti Admin e Super Admin.

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoraggio.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo gli host di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il filtro si trova sopra la tabella. È possibile filtrare gli host per nome, gruppo host, IP o DNS, porta dell'interfaccia, tag, gravità del problema, stato (abilitato/disabilitato/qualsiasi); è inoltre possibile scegliere se visualizzare gli host attualmente in manutenzione.



Parameter	Description
Name	Filtra per il nome visibile dell'host.
Host groups	Filtra per uno o più gruppi host. Specificare un gruppo host padre seleziona implicitamente tutti i gruppi host nidificati.
IP	Filtra per indirizzo IP.
DNS	Filtra per nome DNS.
Port	Filtra per numero di porta.
Severity	Filtra per gravità del problema. Per impostazione predefinita vengono visualizzati i problemi di tutte le gravità. I problemi vengono visualizzati se non sono soppressi.
Status	Filtra per stato dell'host.
Tags	Filtra per nome e valore del tag dell'host. Gli host possono essere filtrati sia in base ai tag a livello di host sia in base ai tag di tutti i template collegati, inclusi i template nidificati. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole/minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole/minuscole) **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole/minuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or **Or** - è sufficiente che una condizione sia soddisfatta
Show hosts in maintenance	Selezionare la casella di controllo per visualizzare gli host in manutenzione (visualizzati per impostazione predefinita).

Salvataggio del filtro

Le impostazioni del filtro preferito possono essere salvate come schede e quindi richiamate rapidamente facendo clic sulla scheda corrispondente sopra il filtro.

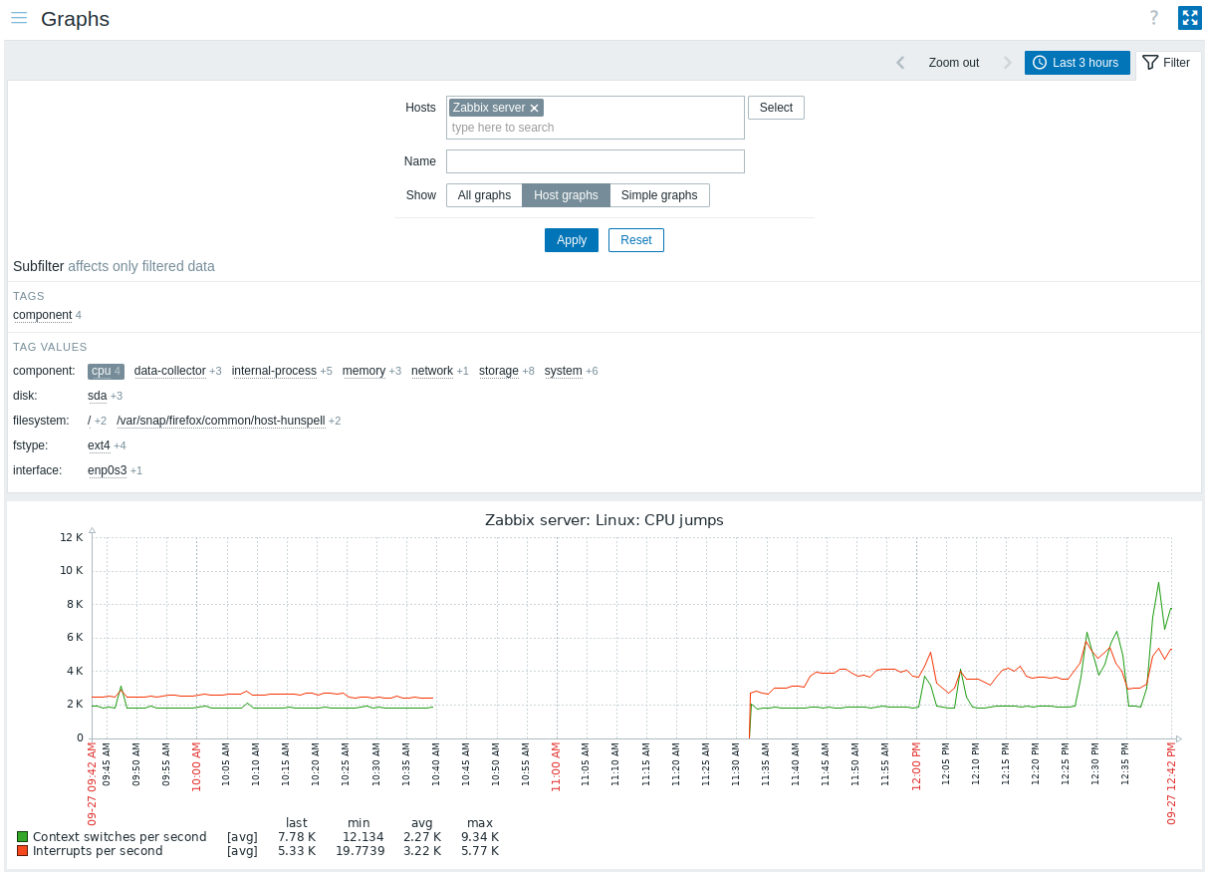
Per ulteriori dettagli, vedere salvataggio dei filtri.

Grafici

Panoramica

I grafici dell'host possono essere accessibili da *Monitoring* > *Hosts* facendo clic su Graphs per l'host corrispondente.

Può essere visualizzato qualsiasi custom graph configurato per l'host, così come qualsiasi simple graph.



I grafici sono ordinati in base a:

- nome del grafico (custom graphs)
- nome dell'item (simple graphs)

Sono accessibili anche i grafici degli host disabilitati.

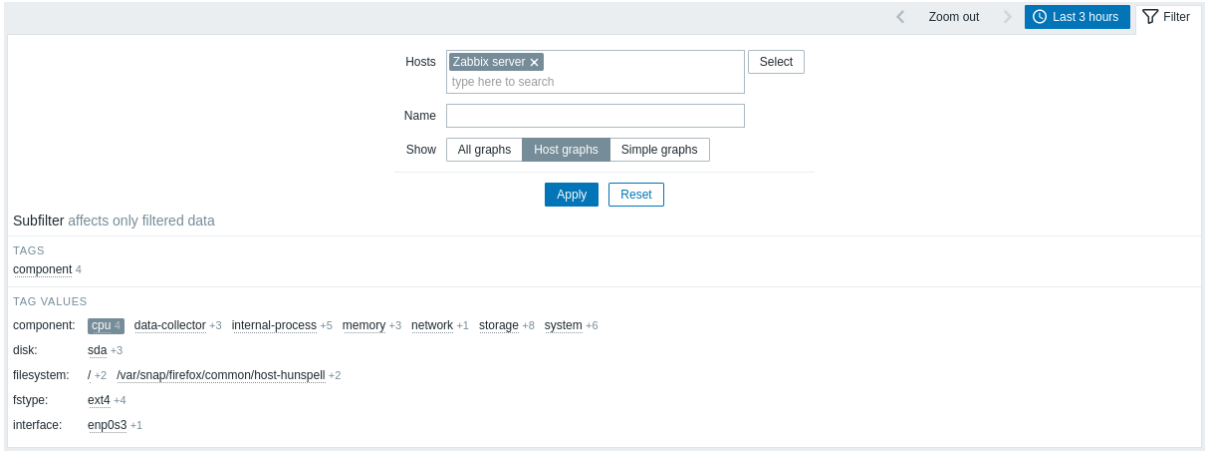
Selettore del periodo di tempo

Prendere nota del selettore del periodo di tempo sopra il grafico. Consente di selezionare i periodi richiesti più spesso con un clic del mouse.

Vedere anche: *Selettore del periodo di tempo*

Utilizzo del filtro

Per visualizzare un grafico specifico, selezionarlo nel filtro. Il filtro consente di specificare l'host, il nome del grafico e l'opzione *Mostra* (tutti/grafici dell'host/grafici semplici).



Se nel filtro non è selezionato alcun host, non viene visualizzato alcun grafico.

Utilizzo del subfilter

Il subfilter consente di modificare ulteriormente il filtraggio applicato dal filtro principale.

Contiene collegamenti cliccabili per accedere rapidamente ai grafici correlati. I grafici sono correlati da un'entità comune: il nome o il valore di un tag. Quando si fa clic sul nome o sul valore di un tag, questo viene evidenziato con uno sfondo grigio e i grafici vengono filtrati immediatamente (non è necessario fare clic su *Applica* nel filtro principale). Facendo clic su un altro nome o valore di tag, questo viene aggiunto ai risultati filtrati. Facendo nuovamente clic sul nome o sul valore del tag, il filtro viene rimosso.

I subfilter vengono generati in base ai dati filtrati, limitati a 1000 record. Se si desidera visualizzare più record nel subfilter, è necessario aumentare il valore del parametro *Limite per i risultati della ricerca e del filtro* (in *Amministrazione* > *Generale* > *GUI*).

A differenza del filtro principale, il subfilter viene aggiornato insieme a ogni richiesta di aggiornamento della tabella, per ottenere sempre informazioni aggiornate sulle opzioni di filtraggio disponibili e sui relativi conteggi.

Il numero di entità visualizzate orizzontalmente è limitato a 100. Se sono presenti più entità, alla fine viene visualizzata un'icona con tre punti, che non è cliccabile. Gli elenchi verticali (ad esempio, i tag con i relativi valori) sono limitati a 20 voci. Se sono presenti più voci, viene visualizzata un'icona con tre punti, che non è cliccabile.

Un numero accanto a ogni entità cliccabile indica il numero di grafici che contiene nei risultati del filtro principale.

Dopo aver selezionato un'entità, i numeri accanto alle altre entità disponibili vengono visualizzati con un segno più, che indica quanti grafici possono essere aggiunti alla selezione corrente.

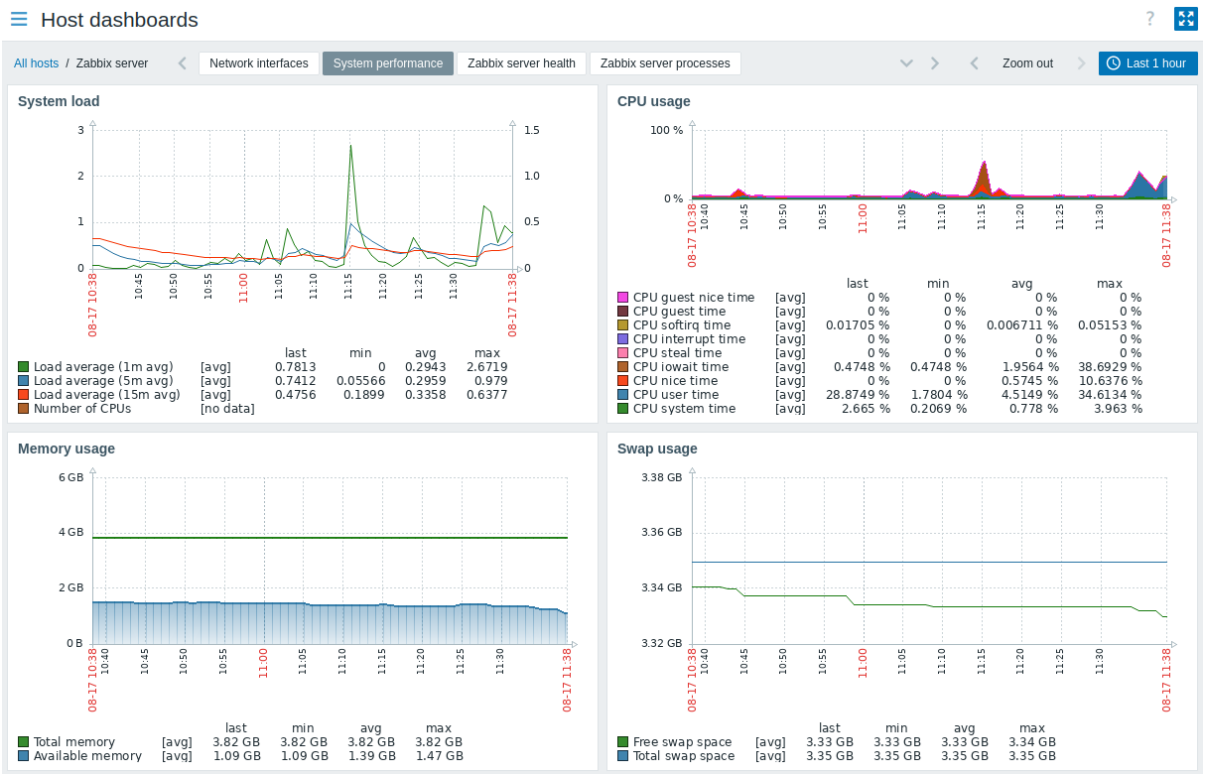
Pulsanti

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Dashboard host

Panoramica

I dashboard degli host sono simili ai dashboard globali; tuttavia, i dashboard degli host non hanno un proprietario e visualizzano i dati solo per l'host selezionato.



Quando si visualizzano i dashboard degli host, è possibile passare da un dashboard configurato all'altro facendo clic su:

- le schede dei dashboard;
- i pulsanti freccia sotto l'intestazione;
- il pulsante freccia sotto l'intestazione, che visualizza l'elenco completo dei dashboard degli host disponibili.

Per passare alla sezione *Monitoring* > *Hosts*, fare clic sul collegamento di navigazione *All hosts* sotto l'intestazione, nell'angolo superiore sinistro.

Configurazione

Le dashboard host sono configurate a livello di template. Una volta che un template è collegato a un host, le dashboard host vengono generate per quell'host. Si noti che le dashboard host *non possono* essere configurate nella sezione *Dashboard*, che è riservata alle dashboard globali.

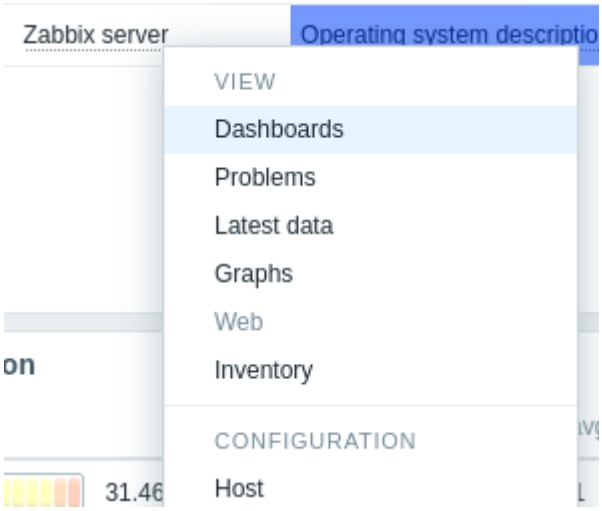
Anche i widget delle dashboard host possono essere configurati solo a livello di template, ad eccezione della modifica dell'intervallo di aggiornamento. Inoltre, i widget delle dashboard host possono essere copiati solo in altre dashboard host all'interno dello stesso template. Si noti che i widget delle dashboard globali *non possono* essere copiati nelle dashboard host.

Note:
Le dashboard host erano chiamate schermate host prima di Zabbix 5.2. Quando si importa un template meno recente che contiene schermate, l'importazione delle schermate verrà ignorata.

Accesso

Le dashboard dell'host possono essere accessibili:

- dopo aver cercato il nome di un host nella ricerca globale (fare clic sul link *Dashboards* fornito nei risultati della ricerca);
- dopo aver fatto clic sul nome di un host in *Inventory* > *Hosts* (fare clic sul link *Dashboards* fornito nella panoramica dell'host);
- dal menu dell'host facendo clic su *Dashboards*.

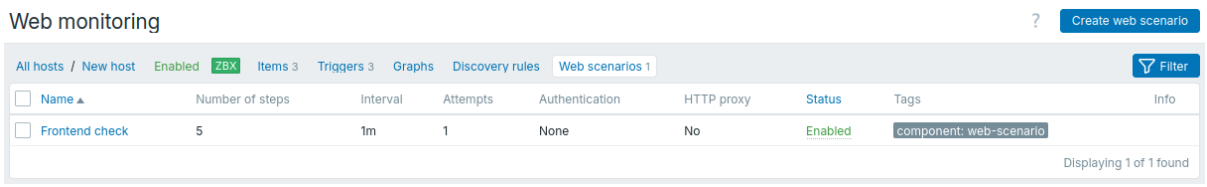


Si noti che le dashboard dell'host *non possono* essere accessibili direttamente nella sezione *Dashboards*, che è riservata alle dashboard globali.

Scenari web

Panoramica

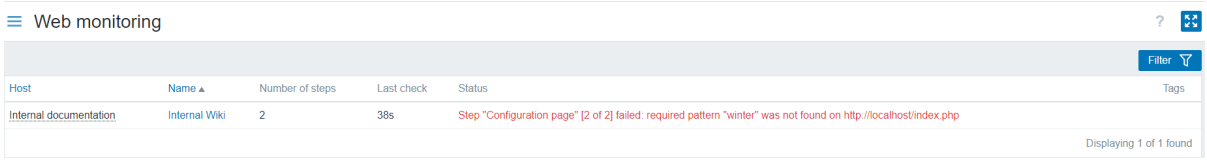
Le informazioni sul web scenario dell'host possono essere consultate da *Monitoring* > *Hosts* facendo clic su Web per l'host corrispondente.



Facendo clic sul nome dell'host si apre il menu dell'host. Sono accessibili anche i dati degli host disabilitati. Il nome di un host disabilitato è visualizzato in rosso.

Lo stato di un web scenario può essere:

- OK (verde) - tutti i passi sono stati completati con successo e Zabbix ha raccolto i dati dello scenario.
- Errore (rosso) - uno o più passi sono falliti (vedere il messaggio di errore per suggerimenti su quali parametri potrebbero richiedere una correzione).
- Sconosciuto (grigio) - non è ancora disponibile alcuno stato dei passi oppure non sono stati raccolti dati.



Il numero massimo di scenari visualizzati per pagina dipende dall'impostazione del profilo utente *Rows per page* setting.

Per impostazione predefinita, vengono visualizzati solo i valori che rientrano nelle ultime 24 ore. Questo limite è stato introdotto con l'obiettivo di migliorare i tempi di caricamento iniziali per le pagine di grandi dimensioni dei dati più recenti. È possibile estendere questo intervallo di tempo modificando il valore del parametro *Max history display period* nel menu *Administration > General > GUI*.

Il nome dello scenario è un collegamento a statistiche più dettagliate sullo stesso:



Utilizzo del filtro

La pagina mostra un elenco di tutti gli scenari web dell'host selezionato. Per visualizzare gli scenari web di un altro host o gruppo di host senza tornare alla pagina *Monitoring > Hosts*, seleziona quell'host o gruppo nel filtro. Puoi anche filtrare gli scenari in base ai tag.

Pulsanti

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Dati più recenti

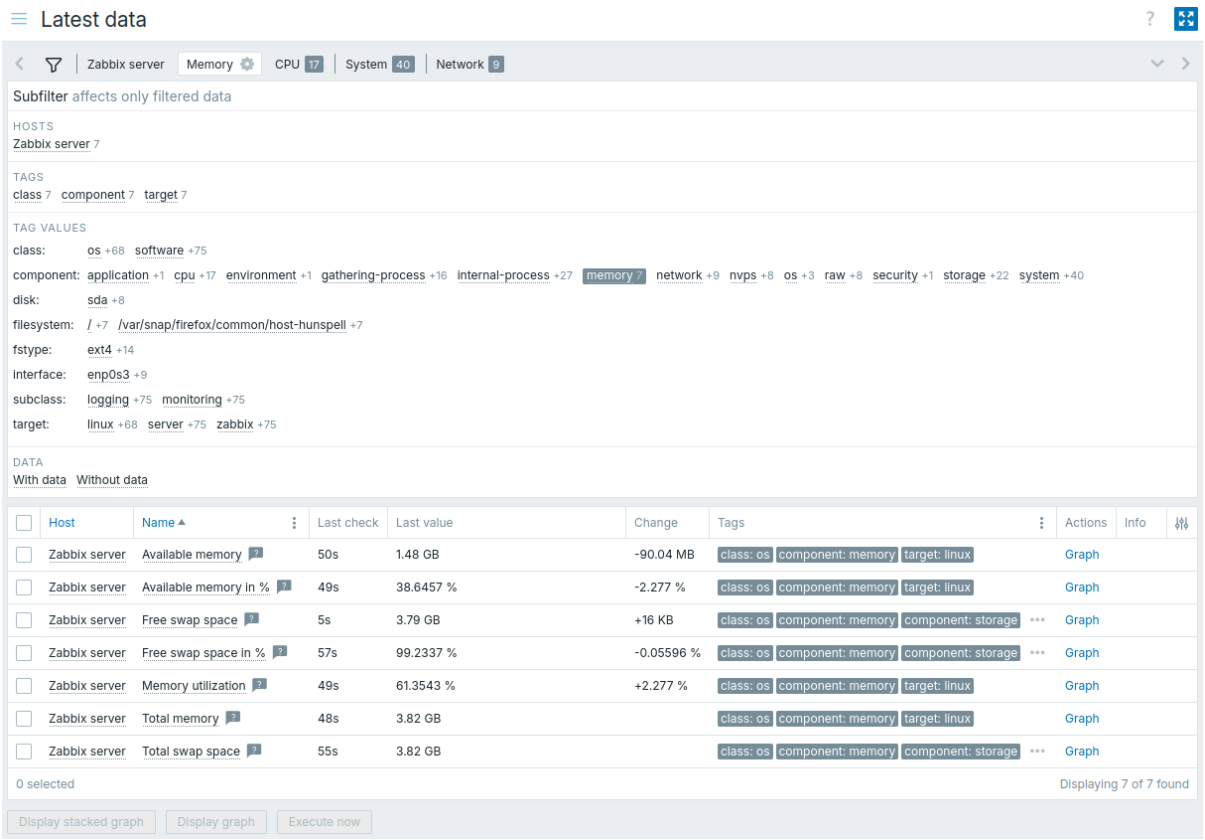
Panoramica

La sezione *Monitoring > Latest data* mostra gli ultimi valori raccolti dagli item.

Questa sezione contiene i seguenti elementi:

- Filtro
- Sottofiltro
- Elenco degli item

Note:
Il sottofiltro e l'elenco degli item vengono visualizzati solo se il filtro è impostato e sono presenti risultati da mostrare.



I dettagli dell'ultimo valore vengono visualizzati in una tabella personalizzabile. Il pulsante *Customize table* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinando il relativo bordo destro; fare doppio clic sul bordo per ripristinarlo. Tutte le personalizzazioni vengono applicate immediatamente e vengono sempre ricordate per l'utente corrente all'interno della scheda filtro attiva (predefinita o salvata).

Sono disponibili le seguenti colonne:

Colonna	Descrizione
Host	Nome dell'host a cui appartiene l'item. Facendo clic sul nome viene visualizzato il menu contestuale dell'host. Se un host è in manutenzione, dopo il nome dell'host viene visualizzata un'icona arancione a forma di chiave inglese . Se un host è disabilitato, il nome dell'host viene visualizzato in rosso. Si noti che i dati degli host disabilitati (inclusi grafici ed elenchi dei valori degli item) sono accessibili nella sezione *Latest data*.
Name	Nome dell'item. Facendo clic sul nome viene visualizzato il menu dell'item. Accanto al nome dell'item viene visualizzata un'icona con punto interrogativo per tutti gli item che hanno una descrizione. Passare il puntatore sull'icona per visualizzare un tooltip con la descrizione dell'item. Facendo clic sul pulsante *Options* nell'intestazione della colonna si accede all'opzione *Show item key*, che mostra la chiave dell'item sotto il nome di ciascun item.
Interval	Intervallo di aggiornamento dell'item.
History	Periodo di conservazione della cronologia dell'item.
Trends	Periodo di conservazione dei trend dell'item.
Type	Tipo dell'item.
Last check	Tempo trascorso dall'ultimo controllo dell'item.

Colonna	Descrizione
Last value	Valore più recente dell'item. I valori vengono visualizzati con la conversione delle unità e l'applicazione della mappatura dei valori. Passare il puntatore sul valore per visualizzare i dati grezzi (troncati a 8192 caratteri). I valori binari vengono visualizzati come segnaposto (*binary value*) invece del valore effettivo. Per impostazione predefinita, vengono visualizzati solo i valori ricevuti nelle ultime 24 ore. Questo limite migliora i tempi di caricamento iniziale per pagine di grandi dimensioni di dati recenti; per estenderlo, aggiornare il valore del parametro *Max history display period* in *Administration* > *General* > *GUI*.
Change	Differenza tra il valore precedente e il valore più recente. Per gli item con una frequenza di aggiornamento pari a 1 giorno o più, l'importo della variazione non verrà mai visualizzato (con l'impostazione predefinita). In questo caso, l'ultimo valore non verrà visualizzato affatto se è stato ricevuto più di 24 ore fa.
Tags	Tag associati all'item. Facendo clic su un tag, questo viene abilitato nel sottofiltro, filtrando così l'elenco degli item in base a quel tag (e a eventuali altri tag selezionati nel sottofiltro). Una volta fatto clic, tali tag non sono più cliccabili; per modificare il filtro in base ai tag, è necessario usare il sottofiltro. Facendo clic sul pulsante *Options* nell'intestazione della colonna è possibile impostare il numero di tag visualizzati (1-3), scegliere come vengono mostrati i nomi dei tag (completi, abbreviati o nessuno), impostare la priorità di visualizzazione dei tag oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Tag value	Vengono visualizzati i valori dei tag dell'item. Facendo clic sul pulsante *Options* nell'intestazione della colonna è possibile rinominare la colonna, scegliere quali valori dei tag visualizzare oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Actions	Collegamento al grafico semplice/storico dei valori dell'item.
Info	Informazioni aggiuntive sull'item. Se un item presenta errori (ad esempio, è diventato non supportato), viene visualizzata un'icona informativa. Passare il puntatore sull'icona per i dettagli.
Custom text	Viene visualizzato testo personalizzato, valori di macro o una combinazione di entrambi. Facendo clic sul pulsante *Options* nell'intestazione della colonna è possibile rinominare la colonna, inserire il testo personalizzato o le macro, oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate). Macro supportate: `{HOST.*}` (tranne `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (tranne `{ITEM.STATE.ERROR}`) e macro utente.

Pulsanti

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Azioni di massa

I pulsanti sotto l'elenco offrono azioni di massa con uno o più item selezionati:

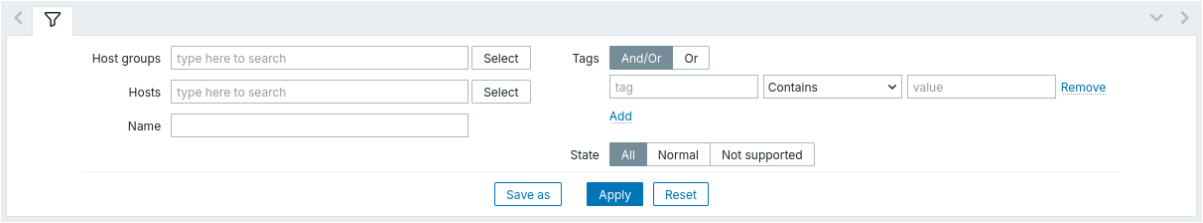
- *Display stacked graph* - visualizza un grafico ad hoc in pila.
- *Display graph* - visualizza un semplice grafico ad hoc.
- *Execute now* - esegue immediatamente un controllo per i nuovi valori degli item. Supportato solo per controlli **passivi** (vedi maggiori dettagli). Questa opzione è disponibile solo per gli host con accesso in lettura e scrittura. L'accesso a questa opzione per gli host con autorizzazioni di sola lettura dipende dall'opzione del ruolo utente chiamata *Invoke "Execute now" on read-only hosts*.

Per usare queste opzioni, seleziona le caselle di controllo accanto ai rispettivi item, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo gli item di interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

L'icona del filtro  si trova sopra l'elenco degli item e il sottofiltro. Fare clic su di essa per espandere il filtro.



Il filtro consente di restringere l'elenco degli item in base al gruppo di host, all'host, al nome dell'item, ai tag e allo stato. La specifica di un gruppo di host principale nel filtro seleziona implicitamente tutti i gruppi di host nidificati. Per informazioni dettagliate sul filtraggio in base ai tag, vedere *Monitoring* > *Problems*.

L'opzione del filtro *Show details* consente di espandere le informazioni visualizzate per gli item. Selezionarla per visualizzare dettagli quali l'intervallo di aggiornamento dell'item, le impostazioni della cronologia e dei trend, il tipo di item e gli errori dell'item (fine/non supportato).

Salvataggio del filtro

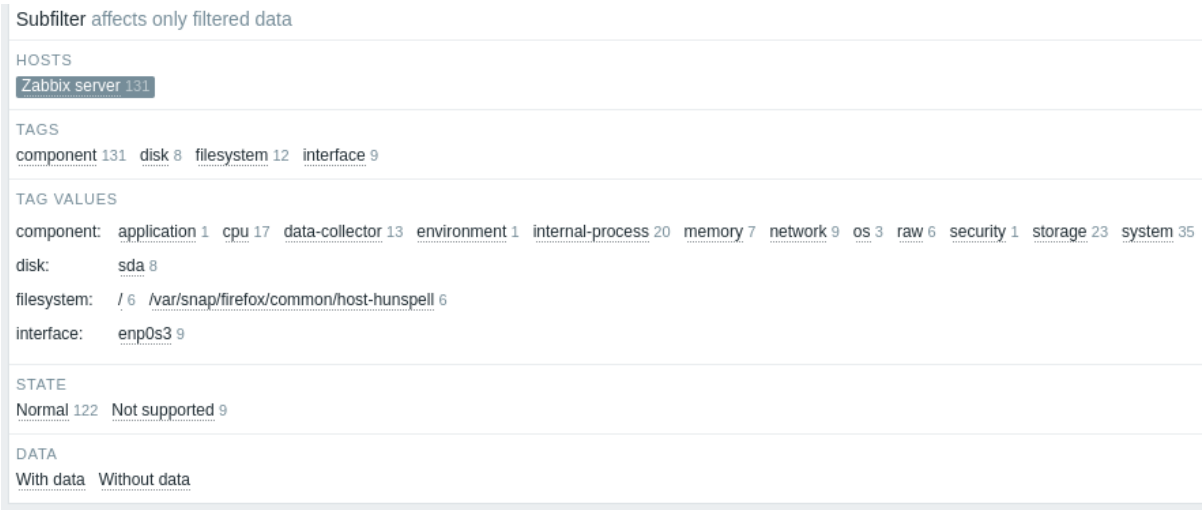
Le impostazioni del filtro preferito possono essere salvate come schede e quindi richiamate rapidamente facendo clic sulla scheda corrispondente sopra il filtro.

Per ulteriori dettagli, vedere salvataggio dei filtri.

Utilizzo del subfiltro

Il subfiltro consente di modificare ulteriormente il filtraggio del filtro principale.

Contiene collegamenti selezionabili per accedere rapidamente agli elementi correlati. Gli elementi sono correlati da un'entità comune: host, nome o valore del tag, stato dell'item o stato dei dati. Quando si fa clic su un'entità, questa viene evidenziata con uno sfondo grigio e gli elementi vengono filtrati immediatamente (non è necessario fare clic su *Applica* nel filtro principale). Facendo clic su un'altra entità, questa viene aggiunta ai risultati filtrati. Facendo nuovamente clic sull'entità, il filtro viene rimosso.



I subfiltri vengono generati in base ai dati filtrati, limitati a 1000 record. Se sono presenti 20 host, ciascuno con 100 item (per un totale di 2000 record), nel subfiltro sarà visibile solo la metà degli host. Per visualizzare più record nel subfiltro, è necessario aumentare il valore del parametro *Limit for search and filter results* (in *Administration* > *General* > *GUI*).

A differenza del filtro principale, il subfiltro viene aggiornato a ogni richiesta di aggiornamento della tabella, in modo da disporre sempre di informazioni aggiornate sulle opzioni di filtraggio disponibili e sui relativi conteggi.

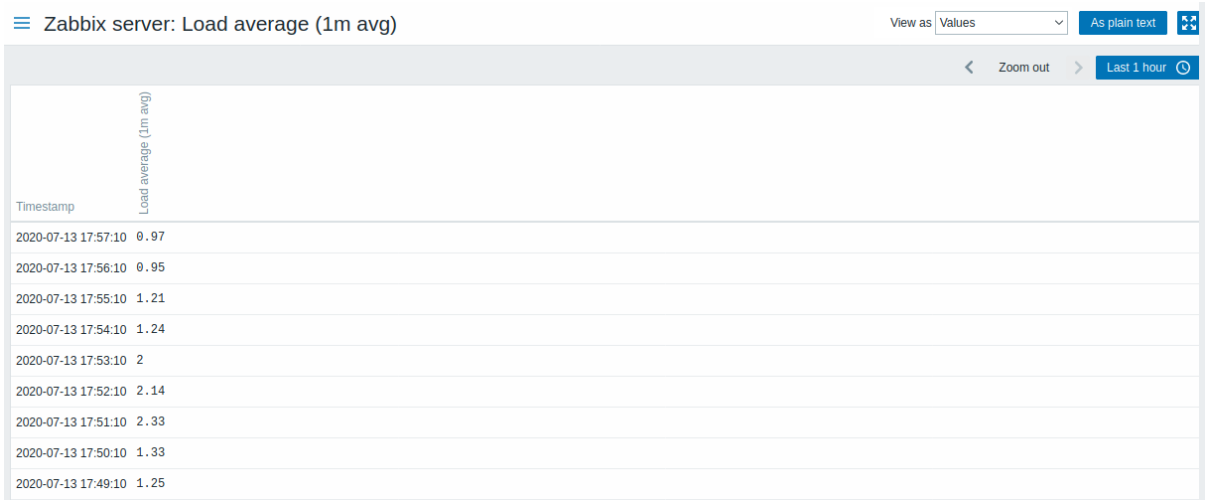
Per ogni gruppo di entità (host, tag, valori dei tag e così via) vengono visualizzate fino a 10 righe di entità. Se sono presenti più entità, l'elenco può essere espanso per visualizzare un massimo di 1000 voci (il valore di SUBFILTER_VALUES_PER_GROUP nelle definizioni del frontend) facendo clic sull'icona con tre punti alla fine dell'elenco. Per *Tag values*, l'elenco può essere espanso per visualizzare un massimo di 200 nomi di tag con i relativi valori. Una volta espanso completamente, l'elenco non può essere compresso.

Un numero accanto a ogni entità selezionabile indica il numero di item raggruppati al suo interno (in base ai risultati del filtro principale). Quando si fa clic su un'entità, i numeri associati alle altre entità disponibili vengono visualizzati con un segno più, che indica quanti item possono essere aggiunti alla selezione corrente. Le entità senza item non vengono visualizzate, a meno che non siano state precedentemente selezionate nel subfiltro.

Grafici e cronologia

La colonna *Graph/History* nell'elenco degli item offre i seguenti link:

- **History** - per tutti gli item testuali, porta agli elenchi (*Values/500 latest values*) che mostrano la cronologia dei valori precedenti dell'item.
- **Graph** - per tutti gli item numerici, porta a un grafico semplice. Nota che, quando il grafico viene visualizzato, un menu a discesa in alto a destra offre anche la possibilità di passare a *Values/500 latest values*.



I valori visualizzati in questo elenco sono grezzi, cioè non viene applicato alcun post-processing.

Note:
La quantità totale di valori visualizzati è definita dal valore del parametro *Limit for search and filter results*, impostato in *Administration* > *General* > *GUI*.

Mappe

Panoramica

Nella sezione *Monitoring* > *Maps* puoi configurare, gestire e visualizzare network maps.

Quando apri questa sezione, vedrai l'ultima mappa a cui hai acceduto oppure un elenco di tutte le mappe a cui hai accesso.

Tutte le mappe possono essere pubbliche o private. Le mappe pubbliche sono disponibili a tutti gli utenti, mentre le mappe private sono accessibili solo al loro proprietario e agli utenti con cui la mappa è condivisa.

Elenco delle mappe



Dati visualizzati:

Column	Description
Name	Nome della mappa. Fare clic sul nome per visualizzare la mappa.
Width	Viene visualizzata la larghezza della mappa.
Height	Viene visualizzata l'altezza della mappa.
Actions	Sono disponibili due azioni: **Properties** - imposta le proprietà generali della mappa **Edit** - accede alla griglia per aggiungere elementi della mappa

Per configurare una nuova mappa, fare clic sul pulsante *Create map* nell'angolo in alto a destra. Per importare una mappa da un file YAML, XML o JSON, fare clic sul pulsante *Import* nell'angolo in alto a destra. L'utente che importa la mappa ne verrà impostato come proprietario.

Due pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Export* - esporta le mappe in un file YAML, XML o JSON
- *Delete* - elimina le mappe

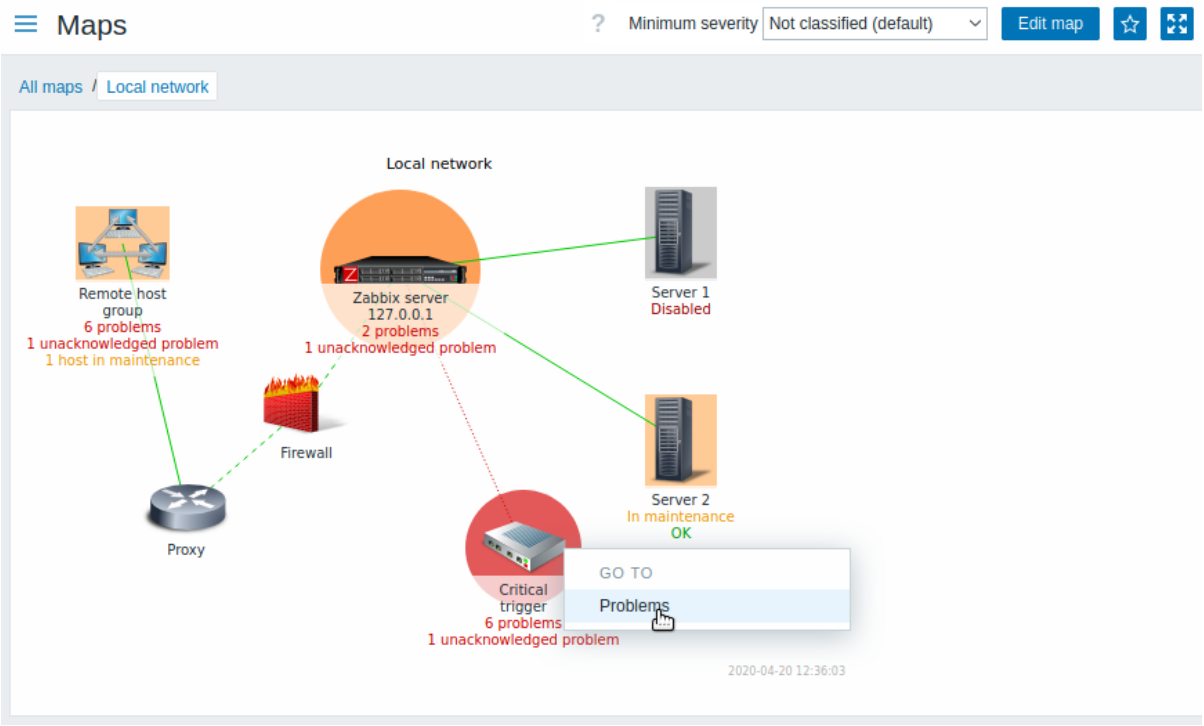
Per usare queste opzioni, selezionare le caselle di controllo davanti alle rispettive mappe, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

Puoi utilizzare il filtro per visualizzare solo le mappe che ti interessano. Per migliori prestazioni di ricerca, i dati vengono cercati con macro non risolte.

Visualizzazione delle mappe

Per visualizzare una mappa, fai clic sul suo nome nell'elenco di tutte le mappe.



Puoi usare il menu a discesa nella barra del titolo della mappa per selezionare il livello di gravità più basso dei trigger di problema da visualizzare. La gravità contrassegnata come *default* è il livello impostato nella configurazione della mappa. Se la mappa contiene una sotto-mappa, la navigazione alla sotto-mappa manterrà la gravità della mappa di livello superiore (tranne se è *Not classified*; in questo caso, non verrà passata alla sotto-mappa).

Evidenziazione delle icone

Se un elemento della mappa si trova in uno stato di problema, viene evidenziato con un cerchio. Il colore di riempimento del cerchio corrisponde al colore della gravità del problema. Solo i problemi con un livello di gravità pari o superiore a quello selezionato verranno visualizzati insieme all'elemento. Se tutti i problemi sono stati riconosciuti, intorno al cerchio viene visualizzato un bordo verde spesso.

Inoltre:

- un host in manutenzione viene evidenziato con un quadrato arancione pieno. La visualizzazione della manutenzione ha priorità rispetto all'evidenziazione della gravità del problema, se l'elemento della mappa è un host.
- un host disabilitato (non monitorato) viene evidenziato con un quadrato grigio pieno.

L'evidenziazione viene visualizzata se la casella di controllo *Evidenziazione delle icone* è selezionata nella configurazione della mappa.

Indicatori delle modifiche recenti

I triangoli rossi rivolti verso l'interno attorno a un elemento indicano una modifica recente dello stato del trigger, avvenuta negli ultimi 30 minuti. Questi triangoli vengono visualizzati se la casella di controllo *Contrassegna gli elementi alla modifica dello stato del trigger* è selezionata nella configurazione della mappa.

Collegamenti

Facendo clic su un elemento della mappa si apre un menu con alcuni collegamenti disponibili. Facendo clic sul nome dell'host viene visualizzato il menu host.

Pulsanti

I pulsanti a destra offrono le seguenti opzioni:

Edit map	Passa alla modifica del contenuto della mappa.
	Aggiunge la mappa al widget Preferiti in Dashboards.
	La mappa è nel widget Preferiti in Dashboards. Fare clic per rimuovere la mappa dal widget Preferiti.

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Riepilogo leggibile nelle mappe

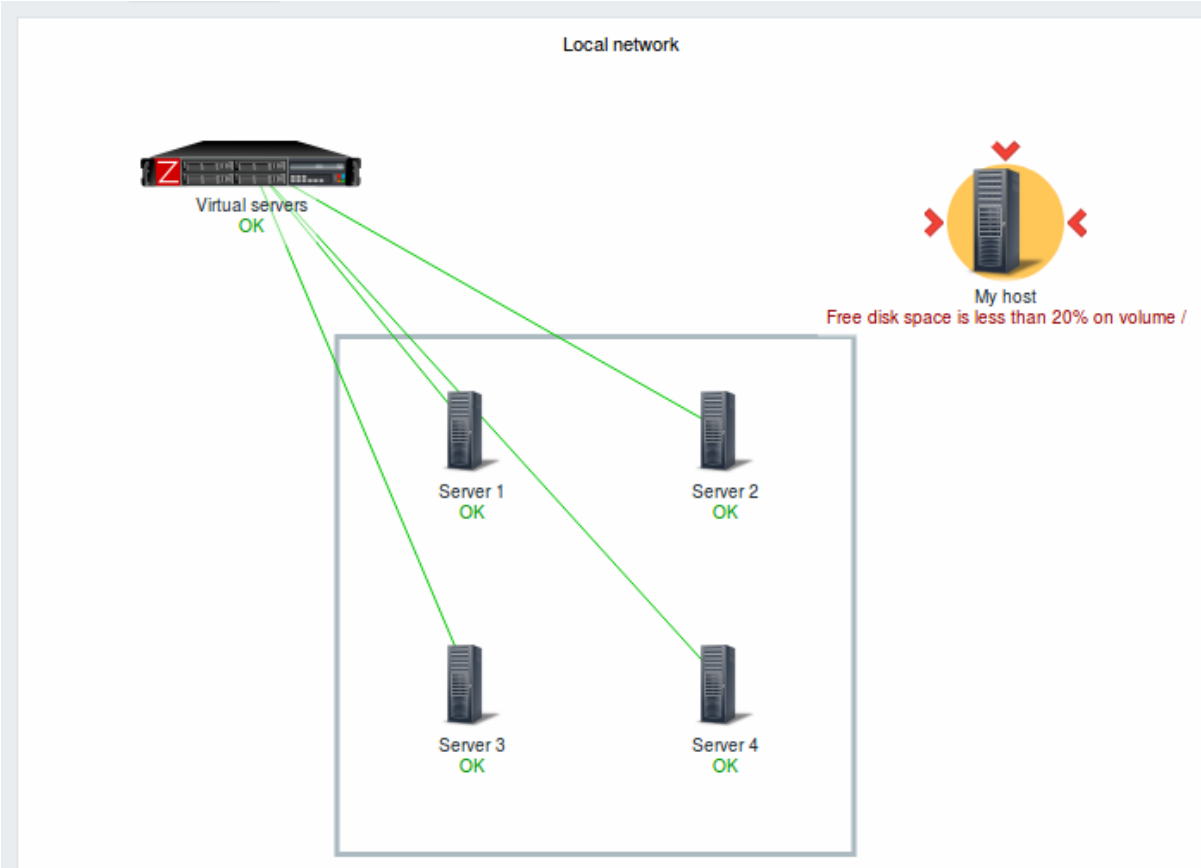
È disponibile una proprietà "aria-label" nascosta, che consente di leggere le informazioni della mappa con uno screen reader. Sono disponibili sia la descrizione generale della mappa sia la descrizione dei singoli elementi, nel formato seguente:

- per la descrizione della mappa: `<Nome mappa>, <* di * elementi nello stato di problema>, <* problemi in totale>.`
- per descrivere un elemento con un problema: `<Tipo di elemento>, Stato <Stato elemento>, <Nome elemento>, <Descrizione problema>.`
- per descrivere un elemento con più problemi: `<Tipo di elemento>, Stato <Stato elemento>, <Nome elemento>, <* problemi>.`
- per descrivere un elemento senza problemi: `<Tipo di elemento>, Stato <Stato elemento>, <Nome elemento>.`

Ad esempio, è disponibile la seguente descrizione:

```
'Rete locale, 1 di 6 elementi nello stato di problema, 1 problema in totale. Host, Stato problema, Il mio
```

per la seguente mappa:



Riferimento a una mappa di rete

Le mappe di rete possono essere referenziate sia da `sysmapid` che da `mapname` GET parametri. Per esempio,

```
http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network
```

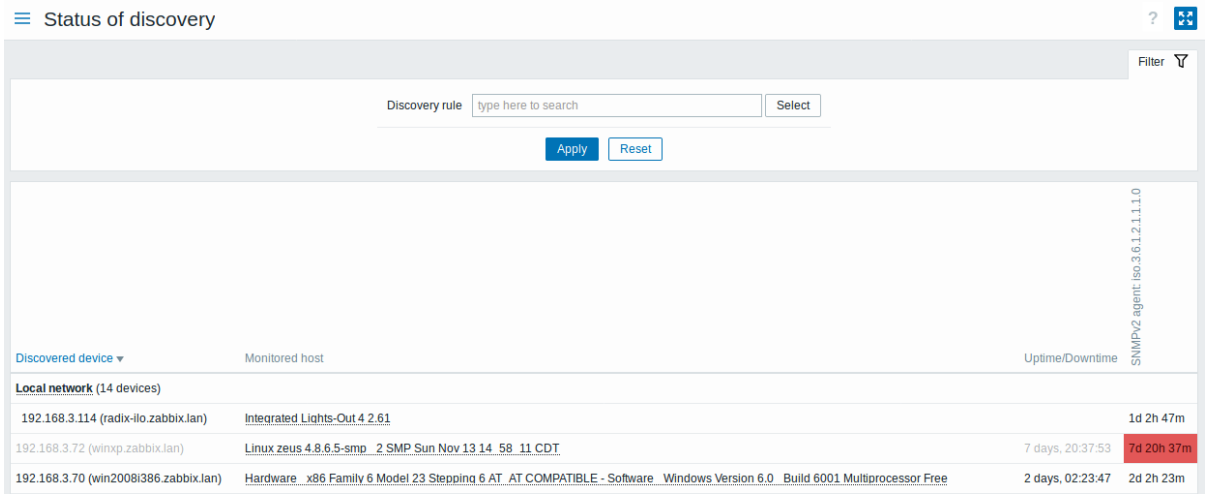
aprirà la mappa con quel nome (Rete locale).

Se sono specificati sia `sysmapid` (ID mappa) che `mapname` (nome mappa), `mapname` ha una priorità maggiore.

Scoperta

Panoramica

Nella sezione *Monitoraggio* > *Discovery* vengono mostrati i risultati della network discovery. I dispositivi rilevati sono ordinati in base alla regola di discovery.



Dati visualizzati:

Colonna	Descrizione
Dispositivo rilevato	I dispositivi rilevati sono elencati e raggruppati in base alla regola di discovery. Facendo clic sulla regola di discovery viene visualizzato il relativo menu, che contiene il collegamento al modulo di configurazione della regola di discovery. Se un dispositivo dispone di più indirizzi IP (come determinato dai *criteri di univocità del dispositivo* della regola di discovery), questi vengono elencati insieme: prima l'indirizzo IP principale del dispositivo, seguito dagli altri indirizzi IP (rientrati). L'indirizzo IP principale è l'indirizzo IP più basso che ha un host elencato nella colonna *Host monitorato* oppure, se non esiste un indirizzo di questo tipo, l'indirizzo IP più basso del dispositivo. I dispositivi possono essere ordinati in base al relativo indirizzo IP principale in ordine crescente o decrescente; gli altri indirizzi IP di un dispositivo sono sempre elencati sotto il relativo indirizzo IP principale, in ordine crescente.
Host monitorato	Se un dispositivo è già monitorato, in questa colonna viene elencato il nome dell'host. Facendo clic sul nome dell'host viene visualizzato il menu dell'host.
Tempo di attività/inattività	In questa colonna viene visualizzata la durata del periodo in cui il dispositivo è stato rilevato o perso dopo il rilevamento precedente. Per un dispositivo con più indirizzi IP, la durata viene visualizzata solo per il relativo indirizzo IP principale.
Controllo di discovery	Viene visualizzato lo stato del singolo servizio (controllo di discovery) per ogni dispositivo rilevato. Una cella rossa indica che il servizio è inattivo. Nella cella è inclusa la durata del periodo di attività o inattività del servizio. Questa colonna viene visualizzata solo se il servizio è stato trovato su almeno un dispositivo rilevato.

Note:
Il numero di regole di discovery e il numero totale di dispositivi rilevati visualizzati in questa sezione sono limitati dal parametro *Limite per i risultati della ricerca e dei filtri* in *Amministrazione > Generale > GUI*. Le regole o i dispositivi che superano il limite non vengono visualizzati e non viene mostrata alcuna indicazione al riguardo.

Pulsanti

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Uso del filtro

È possibile usare il filtro per visualizzare solo le regole di discovery di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Se nel filtro non è selezionato nulla, vengono visualizzate tutte le regole di discovery abilitate. Per selezionare una specifica regola di discovery da visualizzare, iniziare a digitare il suo nome nel filtro. Tutte le regole di discovery abilitate corrispondenti verranno elencate per la selezione. È possibile selezionare più di una regola di discovery.

APM

Panoramica

Il menu *APM* è destinato alle funzionalità di monitoraggio delle prestazioni delle applicazioni (APM) di Zabbix. Visualizza i dati di telemetria inviati dalle applicazioni e dai servizi tramite OpenTelemetry, uno standard aperto per la raccolta dei dati di telemetria. Questi dati consentono di vedere come le richieste attraversano le applicazioni e i servizi, misurarne le prestazioni e individuare la causa delle operazioni lente o non riuscite.

Il menu contiene le seguenti sezioni, corrispondenti ai tipi di segnali di OpenTelemetry:

- Tracce
- Metriche
- Log

A differenza di *Monitoraggio* > *Ultimi dati*, che visualizza i valori raccolti dagli item di Zabbix, le sezioni *APM* visualizzano i dati di telemetria direttamente dall'origine dati, quindi non è necessario creare alcun item. Per visualizzare i dati nelle sezioni *APM*:

1. Specificare un'origine dei dati di telemetria in *Amministrazione* > *Origine dati*.
2. Assicurarsi che i dati di telemetria vengano scritti nell'origine dati da un proxy Zabbix con APM abilitato oppure da un collector OpenTelemetry. Notare che il server Zabbix non può ricevere dati di telemetria; solo il proxy Zabbix può farlo.

Per utilizzare i dati di telemetria nei trigger o nei grafici, utilizzare l'item query di telemetria.

Tracce

Panoramica

La sezione *APM* > *Traces* visualizza le tracce raccolte dalle applicazioni e dai servizi.

Una **traccia** rappresenta il percorso seguito da una richiesta attraverso le applicazioni e i servizi. Ad esempio, quando un utente fa clic sul pulsante *Place order* in un negozio online, la richiesta può passare dal servizio frontend, attraverso i servizi di checkout e pagamento, fino al database.

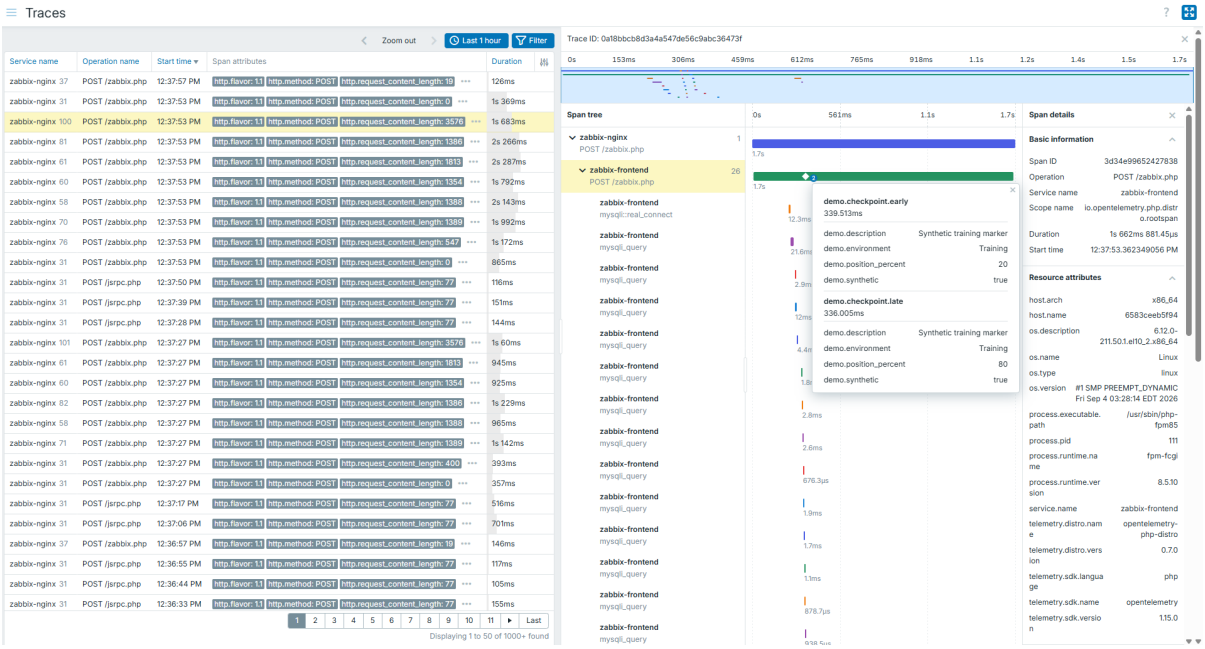
Una traccia è costituita da **span**. Ogni span rappresenta una singola operazione, ad esempio una richiesta HTTP inviata dal servizio frontend al servizio di checkout, una query al database che salva l'ordine o una chiamata di funzione che calcola il costo di spedizione.

Gli span formano una gerarchia. Lo **span radice** rappresenta la prima operazione della richiesta. Ogni altro span è uno **span figlio** dello span durante il quale è stata avviata la relativa operazione. Ad esempio, il servizio frontend che riceve la richiesta *Place order* costituisce lo span radice, mentre il servizio di checkout che elabora l'ordine è il relativo span figlio.

Ogni span viene prodotto da un **servizio** (ad esempio `zabbix-frontend`; denominato dall'attributo della risorsa OpenTelemetry `service.name`) e da un ambito di strumentazione, ovvero la parte di codice che lo ha creato (ad esempio una libreria che traccia le richieste HTTP).

Per fornire il contesto relativo a ogni operazione, gli span dispongono anche di **attributi** chiave-valore: gli attributi dello span descrivono l'operazione stessa (ad esempio `http.method: POST`), mentre gli attributi della risorsa descrivono il servizio e l'ambiente in cui viene eseguito (ad esempio `host.name: eabf8c2a5ddc`).

Per ulteriori informazioni sulle tracce, consultare la documentazione di OpenTelemetry.



La sezione *APM* > *Traces* contiene i seguenti elementi:

- Pulsante Modalità kiosk - visualizza solo il contenuto della pagina, senza il menu e l'intestazione.
- Selettore del periodo di tempo - seleziona il periodo di tempo per il quale visualizzare le tracce.
- Filtro - visualizza solo le tracce di interesse.
- Elenco delle tracce - visualizza le tracce, descritte dai relativi span radice.
- Pannello della traccia - visualizza tutti gli span della traccia selezionata su una sequenza temporale.

Elenco delle trace

Ogni riga nell'elenco delle trace rappresenta una trace; le colonne mostrano i dati del relativo span radice.

Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793	2s 900ms

Le trace vengono visualizzate in una tabella personalizzabile. Il pulsante *Personalizza tabella* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinandone il bordo destro; fare doppio clic sul bordo per ripristinare le dimensioni. Tutte le personalizzazioni vengono applicate immediatamente e memorizzate per l'utente corrente.

L'elenco delle trace può contenere le seguenti colonne:

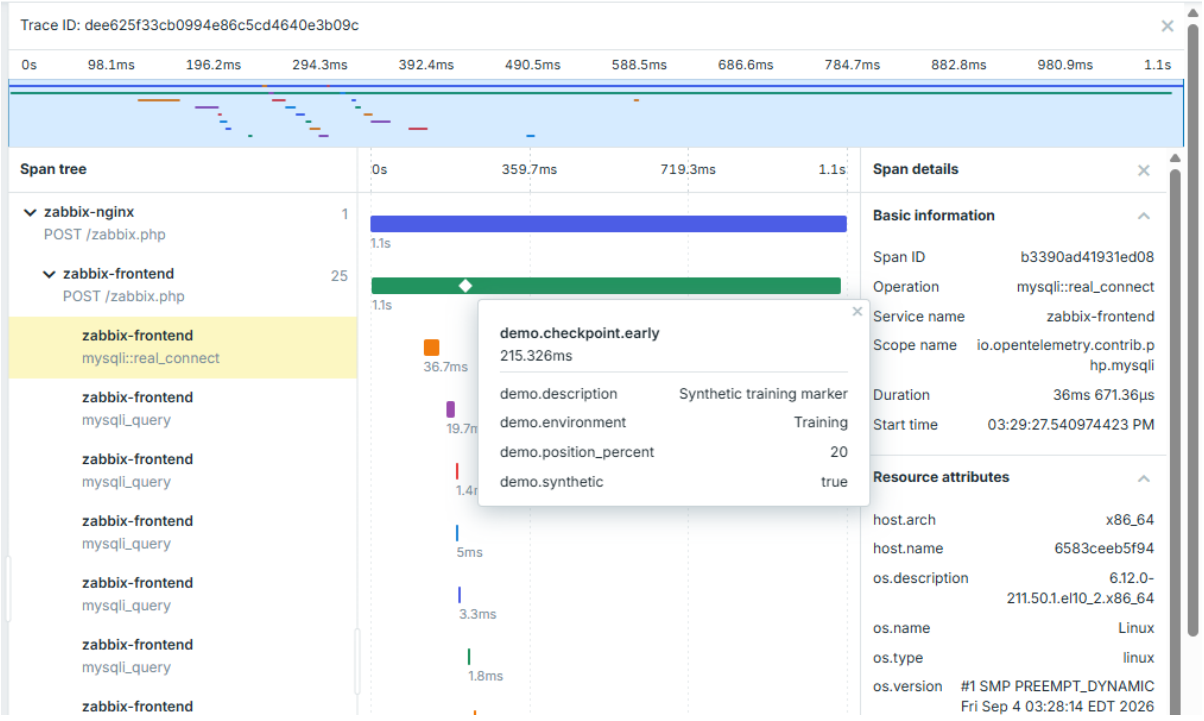
Note:
Ogni colonna corrisponde a una colonna nella tabella `otel_traces` della sorgente dati di telemetria; per l'elenco delle colonne, vedere Colonne di dati di telemetria supportate. La maggior parte delle colonne ha lo stesso nome utilizzato nella sorgente dati (ad esempio, *Trace ID* corrisponde alla colonna `TraceId`). Quando i nomi differiscono, nella tabella seguente viene specificata la colonna della sorgente dati.

Colonna	Descrizione
Trace ID	ID della trace, condiviso da tutti gli span appartenenti alla trace.
Span ID	ID dello span radice.

Colonna	Descrizione
Trace state	Dati aggiuntivi, se presenti, associati alla trace dagli strumenti di tracing (vedere W3C Trace Context).
Service name	Servizio che ha prodotto lo span radice, seguito dal numero di span nella trace. Se uno degli span ha il codice di stato *Error* (vedere di seguito), un'icona rossa mostra il relativo numero.
Operation name	Operazione rappresentata dallo span radice (ad esempio, `POST /zabbix.php`). Colonna della sorgente dati: `SpanName`.
Span kind	Ruolo dello span radice: **Unspecified** - non specificato. **Internal** - operazione eseguita all'interno di un'applicazione. **Server** - gestione di una richiesta in ingresso. **Client** - richiesta in uscita. **Producer** - creazione di un'operazione asincrona (ad esempio, l'invio di un messaggio). **Consumer** - elaborazione di un'operazione asincrona.
Scope name	Ambito di strumentazione che ha prodotto lo span radice.
Scope version	Versione dell'ambito di strumentazione.
Status code	Stato dell'operazione rappresentata dallo span radice: **Unset** - non è stato assegnato alcuno stato esplicito allo span radice. **Ok** - l'operazione è stata completata correttamente. **Error** - l'operazione è stata completata con un errore.
Status message	Informazioni aggiuntive, se presenti, sullo stato.
Start time	Data e ora di avvio dello span radice. Colonna della sorgente dati: `Timestamp`.
Span attributes	Attributi dello span radice. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.
Resource attributes	Attributi della risorsa dello span radice. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.
Duration	Durata della trace; la barra grigia mostra la durata relativa alle altre trace.

Pannello della traccia

Fare clic su una traccia nell'elenco delle tracce per aprire il pannello della traccia, che mostra tutti gli span della traccia su una sequenza temporale: quali operazioni sono state eseguite, in quale ordine e quanto tempo ha richiesto ciascuna.

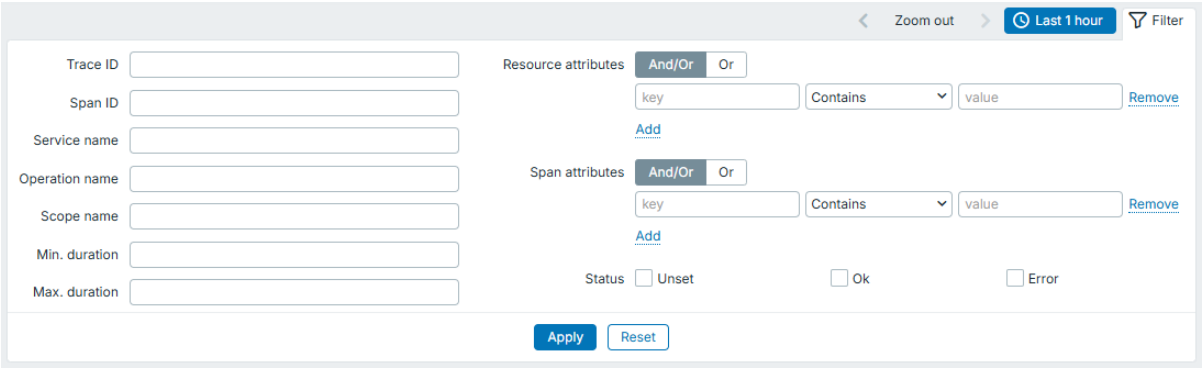


Il pannello della traccia può contenere i seguenti elementi:

Elemento	Descrizione
ID della traccia	ID della traccia selezionata.
Panoramica della sequenza temporale della traccia	L'intera traccia, con gli span disposti su un massimo di 8 righe (più span possono condividere una riga). Trascinare i bordi dell'intervallo selezionato, oppure l'intervallo stesso, per ingrandire la sequenza temporale degli span su quella parte della traccia.
Albero degli span	Gli span nella loro gerarchia; ciascuno mostra il proprio servizio (in grassetto), il nome dell'operazione e il numero di span figli diretti.
Sequenza temporale degli span	Gli span rappresentati come barre: la posizione e la lunghezza mostrano quando è iniziato ciascuno span e quanto tempo ha richiesto. I marcatori mostrano gli eventi (ad esempio, le eccezioni); passare il mouse su un marcatore o fare clic su di esso per visualizzarne i dettagli.
Dettagli dello span	Fare clic su uno span nell'albero degli span per visualizzarne i dettagli e gli attributi.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo le tracce di interesse.



Per filtrare in base agli attributi, immettere la chiave dell'attributo, selezionare l'operatore e immettere il valore dell'attributo.

Sono supportati i seguenti operatori:

- **Contiene** - *(predefinito)* il valore dell'attributo contiene la stringa immessa.
- **Uguale** - il valore dell'attributo è uguale alla stringa immessa.
- **Non contiene** - il valore dell'attributo non contiene la stringa immessa.
- **Non uguale** - il valore dell'attributo non è uguale alla stringa immessa.
- **Esiste** - esiste l'attributo con la chiave immessa.
- **Non esiste** - non esiste l'attributo con la chiave immessa.

Se si aggiungono più condizioni, selezionare come valutarle:

- **And/Or** - *(predefinito)* tutte le condizioni devono essere soddisfatte; le condizioni con la stessa chiave dell'attributo vengono combinate con *Or*.
- **Or** - deve essere soddisfatta almeno una condizione.

Metriche

Panoramica

La sezione *APM* > *Metriche* visualizza i punti dati delle metriche raccolti dalle applicazioni e dai servizi.

Una **metrica** è una misurazione (ad esempio, la durata di una richiesta, il numero di ordini elaborati o l'utilizzo della memoria). Le applicazioni inviano i relativi valori come **punti dati** che contengono il valore (o i valori aggregati) misurato in un momento specifico. Una metrica è simile a un item di Zabbix e un punto dati è simile a un valore di tale item.

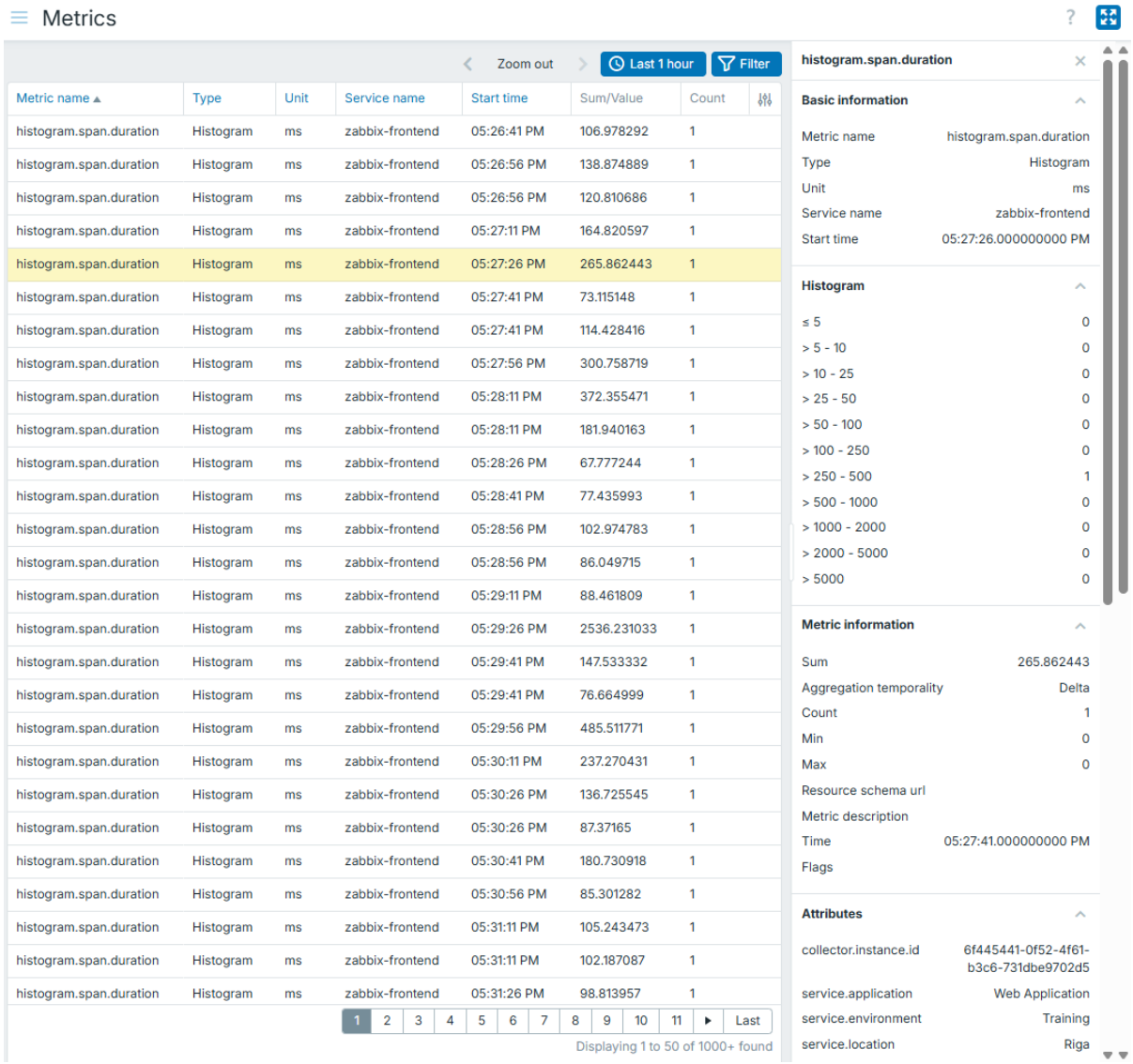
Un punto dati può essere di uno dei seguenti tipi:

- **Gauge** - un valore misurato in un momento specifico (ad esempio, l'utilizzo corrente della memoria).
- **Sum** - una somma delle misurazioni effettuate in un determinato periodo di tempo (ad esempio, il numero totale di richieste).
- **Histogram** - il numero di misurazioni effettuate in un determinato periodo di tempo che rientrano in ciascuno di diversi intervalli di valori fissi, chiamati bucket (ad esempio, quante richieste hanno richiesto 0-100 ms, 100-250 ms o 250-500 ms).
- **Exponential histogram** - come un istogramma, ma ogni bucket è più ampio del precedente dello stesso fattore (ad esempio, 1-2 ms, 2-4 ms, 4-8 ms).

Ogni punto dati è prodotto da un **servizio** (ad esempio, `zabbix-frontend`; denominato dall'attributo della risorsa OpenTelemetry `service.name`) e da un ambito di strumentazione, ovvero la parte di codice che lo ha creato (ad esempio, una libreria che misura la durata delle richieste HTTP).

Per fornire il contesto relativo a ogni misurazione, i punti dati hanno anche **attributi** chiave-valore: gli attributi dei punti dati descrivono cosa è stato misurato e distinguono i punti dati della stessa metrica (ad esempio, `http.request.method: POST`), mentre gli attributi delle risorse descrivono il servizio e l'ambiente in cui viene eseguito (ad esempio, `host.name: eabf8c2a5ddc`).

Per ulteriori informazioni sulle metriche, consultare la documentazione di OpenTelemetry.

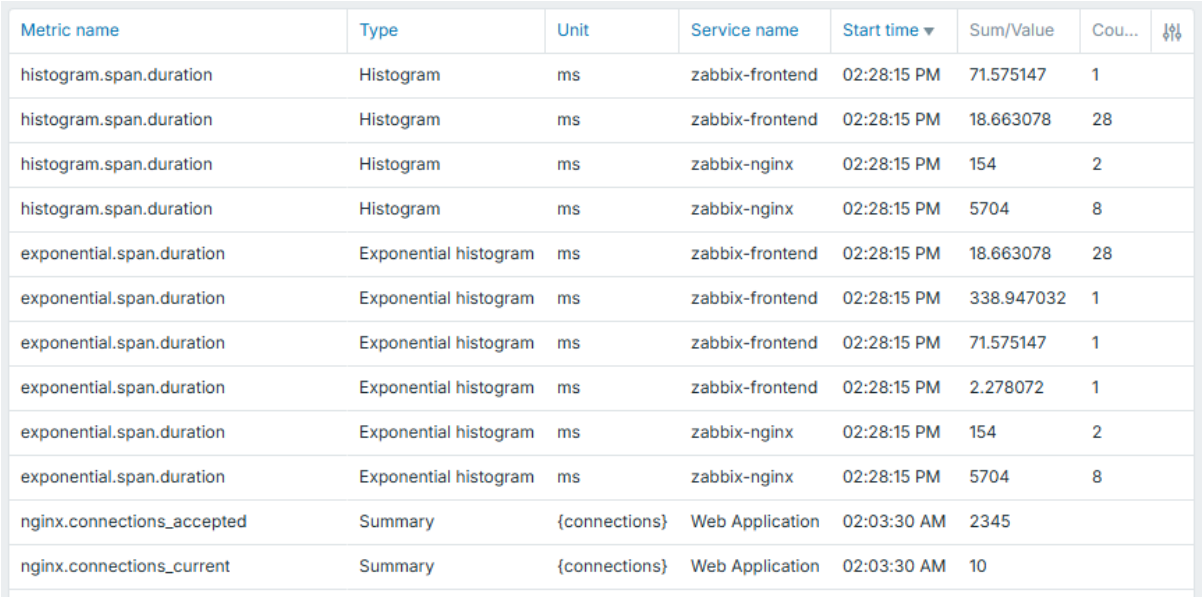


La sezione *APM* > *Metriche* contiene i seguenti elementi:

- Pulsante Modalità chiosco - visualizza solo il contenuto della pagina, senza il menu e l'intestazione.
- Selettore del periodo di tempo - seleziona il periodo di tempo per il quale visualizzare i punti dati.
- Filtro - visualizza solo i punti dati di interesse.
- Elenco delle metriche - visualizza i punti dati.
- Dettagli della metrica - visualizza i dettagli del punto dati selezionato.

Elenco delle metriche

Ogni riga nell'elenco delle metriche rappresenta un punto dati.



Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

I punti dati vengono visualizzati in una tabella personalizzabile. Il pulsante *Personalizza tabella* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinandone il bordo destro; fare doppio clic sul bordo per ripristinare le dimensioni. Tutte le personalizzazioni vengono applicate immediatamente e memorizzate per l'utente corrente.

L'elenco delle metriche può contenere le seguenti colonne:

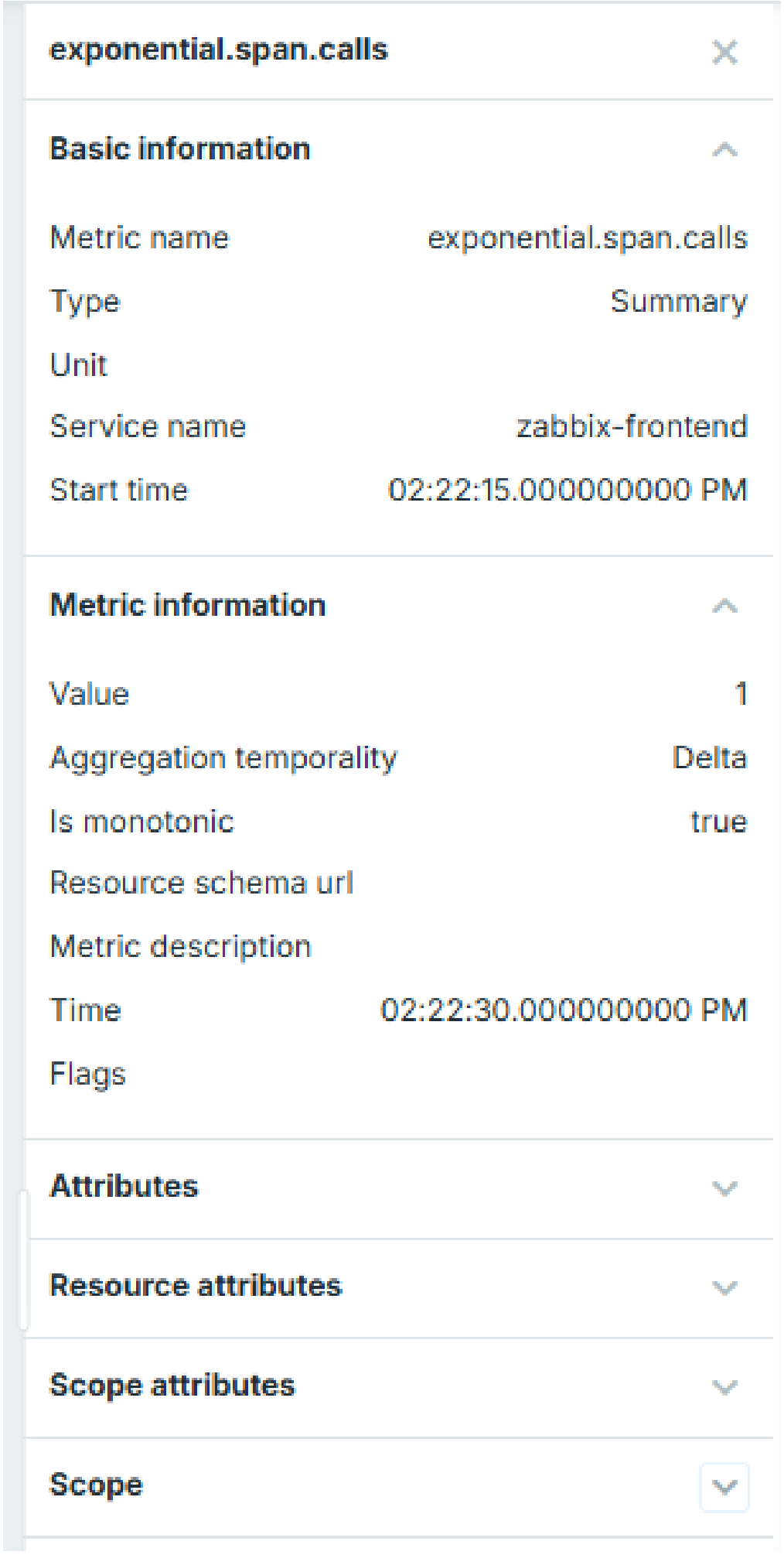
Note:
Ogni colonna, ad eccezione di *Tipo*, corrisponde a una colonna nelle tabelle delle metriche (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` e `otel_metrics_exponential_histogram`) dell'origine dati di telemetria; per l'elenco delle colonne, vedere Colonne di dati di telemetria supportate. La maggior parte delle colonne ha lo stesso nome utilizzato nell'origine dati (ad esempio, *Nome metrica* corrisponde alla colonna `MetricName`). Quando i nomi differiscono, nella tabella seguente viene specificata la colonna dell'origine dati.

Colonna	Descrizione
Nome metrica	Metrica a cui appartiene il punto dati (ad esempio, `http.server.duration`).
Tipo	Tipo di punto dati: *Gauge*, *Sum*, *Histogram* o *Exponential histogram*. Determinato dalla tabella delle metriche che contiene il punto dati.
Unità	Unità, se presente, dei valori della metrica (ad esempio, `s` o `By`; vedere UCUM). Colonna dell'origine dati: `MetricUnit`.
Nome del servizio	Servizio che ha prodotto il punto dati.
Nome dell'ambito	Ambito di strumentazione che ha prodotto il punto dati.
Ora di inizio	Data e ora di inizio del periodo di misurazione del punto dati. Colonna dell'origine dati: `StartTimeUnix`.
Somma/Valore	Somma di tutte le misurazioni (*Histogram*, *Exponential histogram*) o valore (*Gauge*, *Sum*) del punto dati.
Conteggio	Numero di misurazioni in un punto dati *Histogram* o *Exponential histogram*.
Ora	Data e ora del punto dati; per i punti dati *Sum*, *Histogram* ed *Exponential histogram*, indica la fine del periodo di misurazione. Colonna dell'origine dati: `TimeUnix`.
Flag	Flag del punto dati: **0** - non è impostato alcun flag. **1** - non è stato registrato alcun valore per il punto dati.
Temporaneità dell'aggregazione	Modalità di aggregazione nel tempo dei valori di un punto dati *Sum*, *Histogram* o *Exponential histogram*: **Unspecified** - non specificata. **Delta** - aggregati nel periodo trascorso dal punto dati precedente. **Cumulative** - aggregati nel periodo trascorso dall'ora di inizio.
Descrizione della metrica	Descrizione, se presente, della metrica.
Attributi	Attributi del punto dati. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.

Colonna	Descrizione
Attributi della risorsa	Attributi della risorsa del punto dati. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.
Attributi dell'ambito	Attributi dell'ambito di strumentazione. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.

Dettagli della metrica

Fare clic su un punto dati nell'elenco delle metriche per aprire il pannello dei dettagli della metrica, che mostra tutte le informazioni sul punto dati.

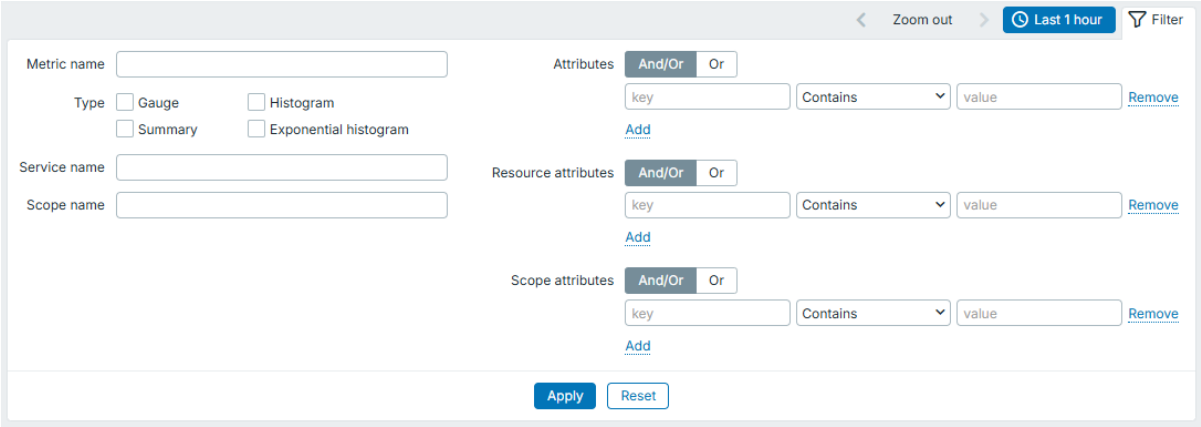

exponential.span.calls
Basic information
Metric name exponential.span.calls
Type Summary
Unit
Service name zabbix-frontend
Start time 02:22:15.000000000 PM
Metric information
Value 1
Aggregation temporality Delta
Is monotonic true
Resource schema url
Metric description
Time 02:22:30.000000000 PM
Flags
Attributes
Resource attributes
Scope attributes
Scope

Il pannello dei dettagli della metrica può contenere le seguenti sezioni:

Sezione	Descrizione
Informazioni di base	Nome della metrica, tipo, unità, nome del servizio e ora di inizio.
Istogramma	Bucket (ad esempio, `> 5 – 10`), ciascuno con il numero di misurazioni contenute. Visualizzato solo per i punti dati di tipo *Istogramma*.
Istogramma esponenziale	Bucket (ad esempio, `>= 249.7885 – 249.7887`), ciascuno con il numero di misurazioni contenute. Visualizzato solo per i punti dati di tipo *Istogramma esponenziale*.
Informazioni sulla metrica	Valori del punto dati, che dipendono dal relativo tipo, nonché descrizione della metrica, ora, flag e URL dello schema della risorsa.
Attributi	Attributi del punto dati.
Attributi della risorsa	Attributi della risorsa.
Attributi dell'ambito	Attributi, se presenti, dell'ambito di strumentazione.
Ambito	Nome, versione e URL dello schema dell'ambito di strumentazione.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo i punti dati di interesse.



Per filtrare in base agli attributi, immettere la chiave dell'attributo, selezionare l'operatore e immettere il valore dell'attributo.

Sono supportati i seguenti operatori:

- **Contiene** - *(predefinito)* il valore dell'attributo contiene la stringa immessa.
- **È uguale a** - il valore dell'attributo è uguale alla stringa immessa.
- **Non contiene** - il valore dell'attributo non contiene la stringa immessa.
- **Non è uguale a** - il valore dell'attributo non è uguale alla stringa immessa.
- **Esiste** - esiste l'attributo con la chiave immessa.
- **Non esiste** - non esiste l'attributo con la chiave immessa.

Se si aggiungono più condizioni, selezionare come valutarle:

- **And/Or** - *(predefinito)* tutte le condizioni devono essere soddisfatte; le condizioni con la stessa chiave dell'attributo vengono combinate con *Or*.
- **Or** - deve essere soddisfatta almeno una condizione.

Log

Panoramica

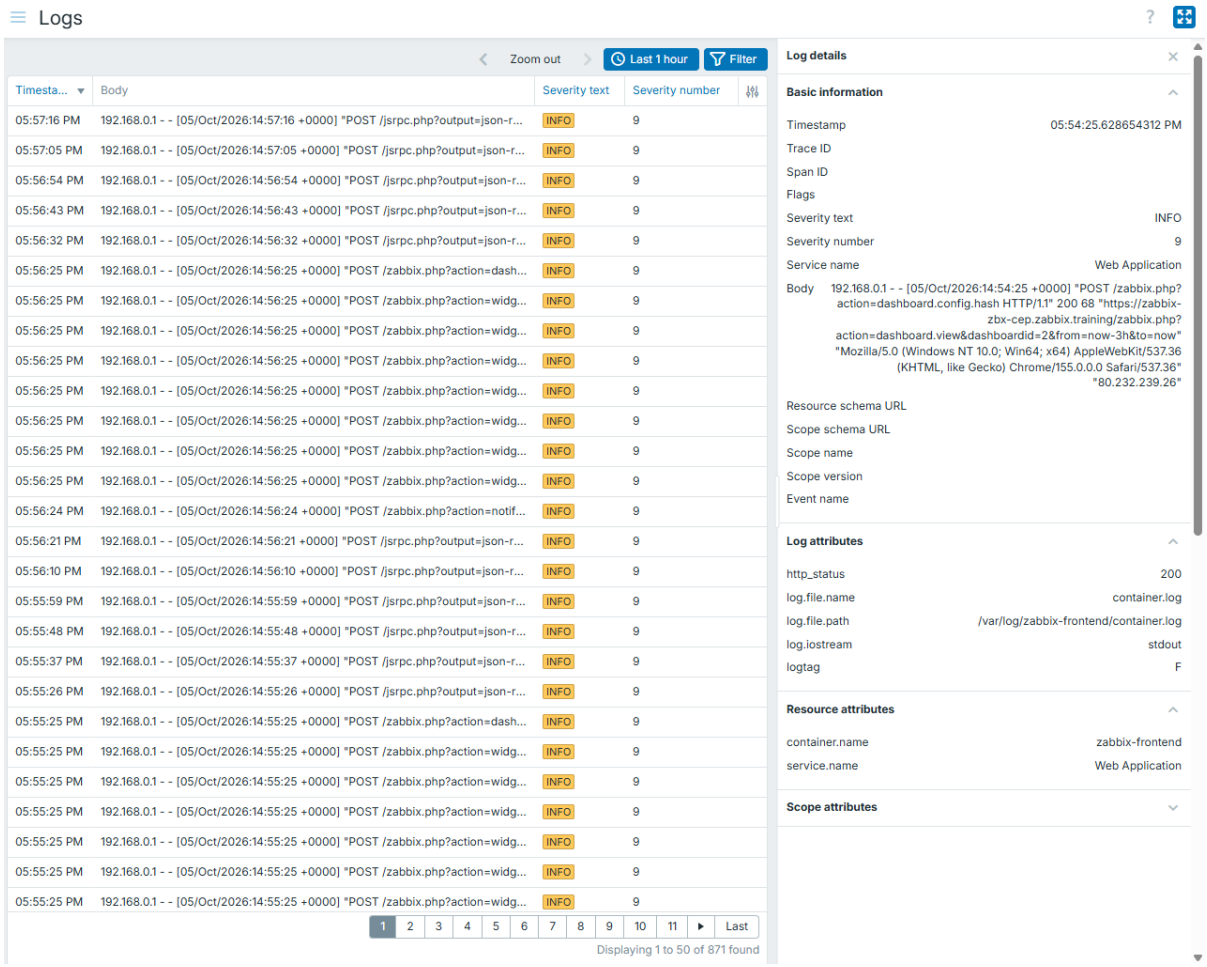
La sezione *APM* > *Logs* visualizza i record di log raccolti dalle applicazioni e dai servizi.

Un **record di log** è un record testuale di un evento (ad esempio, un pagamento non riuscito o un timeout del database). Ogni record di log ha una gravità e un messaggio (corpo). Se un record di log è stato prodotto durante un'operazione che fa parte di una traccia, contiene anche gli ID di tale traccia e dello span.

Ogni record di log viene prodotto da un **servizio** (ad esempio, `zabbix-frontend`; denominato dall'attributo della risorsa Open-Telemetry `service.name`) e da un ambito di strumentazione, ovvero la parte di codice che lo ha creato (ad esempio, una libreria di logging).

Per fornire il contesto relativo a ogni evento, i record di log hanno anche **attributi** chiave-valore: gli attributi di log descrivono l'evento stesso (ad esempio, `log.file.path: /var/log/zabbix/zabbix_frontend.log`), mentre gli attributi della risorsa descrivono il servizio e l'ambiente in cui viene eseguito (ad esempio, `host.name: eabf8c2a5ddc`).

Per ulteriori informazioni sui log, consultare la documentazione di OpenTelemetry.

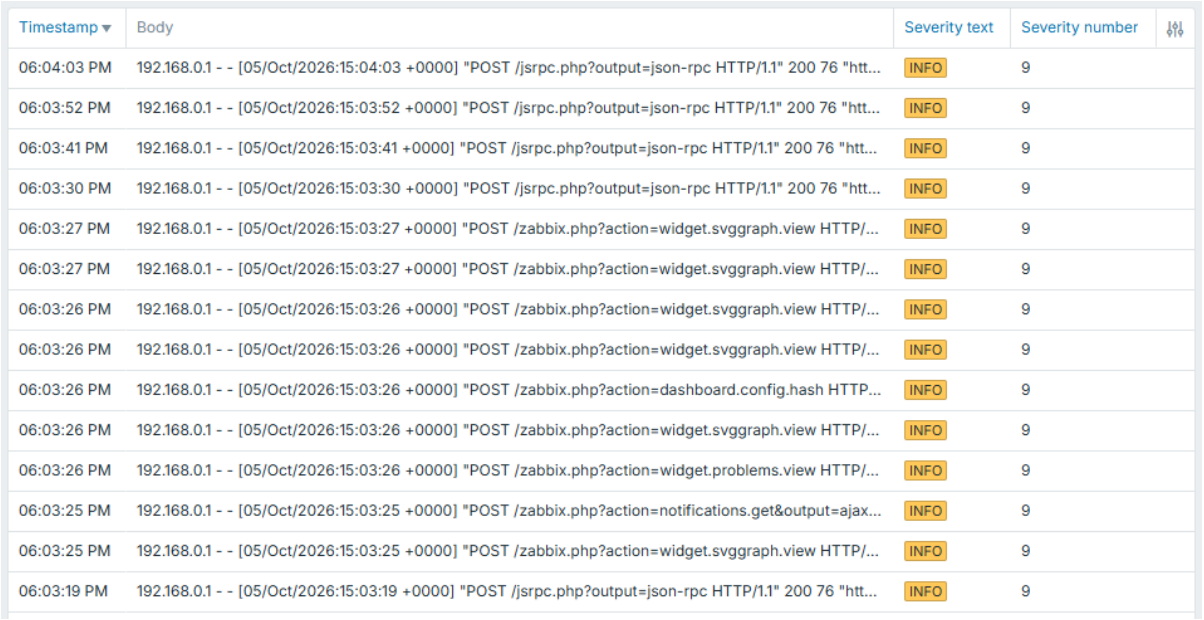


La sezione *APM* > *Logs* contiene i seguenti elementi:

- Pulsante Modalità kiosk - visualizza solo il contenuto della pagina, senza il menu e l'intestazione.
- Selettore del periodo di tempo - seleziona il periodo di tempo per il quale visualizzare i record di log.
- Filtro - visualizza solo i record di log di interesse.
- Elenco dei log - visualizza i record di log.
- Dettagli del log - visualizza i dettagli del record di log selezionato.

Elenco dei log

Ogni riga nell'elenco dei log rappresenta un record di log.



Timestamp ▾	Body	Severity text	Severity number
06:04:03 PM	192.168.0.1 - - [05/Oct/2026:15:04:03 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:52 PM	192.168.0.1 - - [05/Oct/2026:15:03:52 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:41 PM	192.168.0.1 - - [05/Oct/2026:15:03:41 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:30 PM	192.168.0.1 - - [05/Oct/2026:15:03:30 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=dashboard.config.hash HTTP...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.problems.view HTTP/...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=notifications.get&output=ajax...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:19 PM	192.168.0.1 - - [05/Oct/2026:15:03:19 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9

I record di log vengono visualizzati in una tabella personalizzabile. Il pulsante *Personalizza tabella* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinandone il bordo destro; fare doppio clic sul bordo per ripristinare le dimensioni. Tutte le personalizzazioni vengono applicate immediatamente e memorizzate per l'utente corrente.

L'elenco dei log può contenere le seguenti colonne:

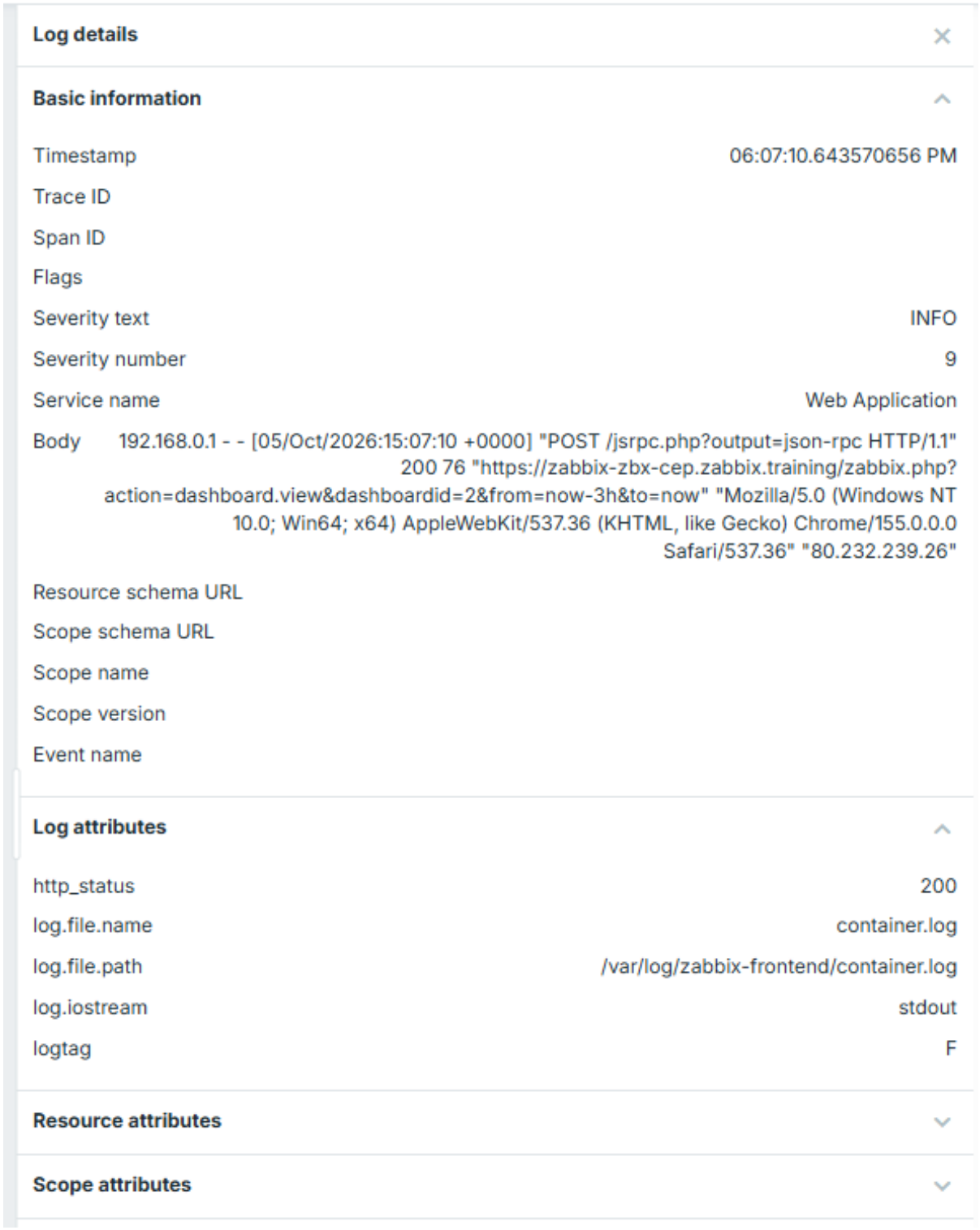
Note:
Ogni colonna corrisponde a una colonna nella tabella `otel_logs` della sorgente dati di telemetria; per l'elenco delle colonne, vedere Colonne di dati di telemetria supportate. La maggior parte delle colonne ha lo stesso nome utilizzato nella sorgente dati (ad esempio, *Testo della gravità* corrisponde alla colonna `SeverityText`). Quando i nomi differiscono, la colonna della sorgente dati è specificata nella tabella seguente.

Colonna	Descrizione
Timestamp	Data e ora in cui si è verificato l'evento.
Corpo	Messaggio del record di log.
Testo della gravità	Livello di gravità fornito dall'applicazione (ad esempio, `INFO`, `WARN` o `ERROR`). Il colore dell'etichetta dipende dalla gravità.
Numero della gravità	Livello di gravità espresso come numero: **0** - NON SPECIFICATO. **1-4** - TRACE. **5-8** - DEBUG. **9-12** - INFO. **13-16** - WARN. **17-20** - ERROR. **21-24** - FATAL.
ID trace	ID, se presente, della trace durante la quale è stato prodotto il record di log.
ID span	ID, se presente, dello span durante il quale è stato prodotto il record di log.
Flag	Flag della trace del record di log: **0** - la trace non è campionata. **1** - la trace è campionata (registrata). Colonna della sorgente dati: `TraceFlags`.
Nome del servizio	Servizio che ha prodotto il record di log.
Nome dell'ambito	Ambito di strumentazione che ha prodotto il record di log.
Versione dell'ambito	Versione dell'ambito di strumentazione.
Nome dell'evento	Nome, se presente, del tipo di evento rappresentato dal record di log.
Attributi del log	Attributi del log del record di log. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.
Attributi della risorsa	Attributi della risorsa del record di log. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.

Colonna	Descrizione
Attributi dell'ambito	Attributi dell'ambito di strumentazione. Fare clic sul pulsante con tre puntini ••• per visualizzare tutti gli attributi.

Dettagli del log

Fare clic su un record di log nell'elenco dei log per aprire il pannello *Dettagli del log*, che mostra tutte le informazioni relative al record di log.



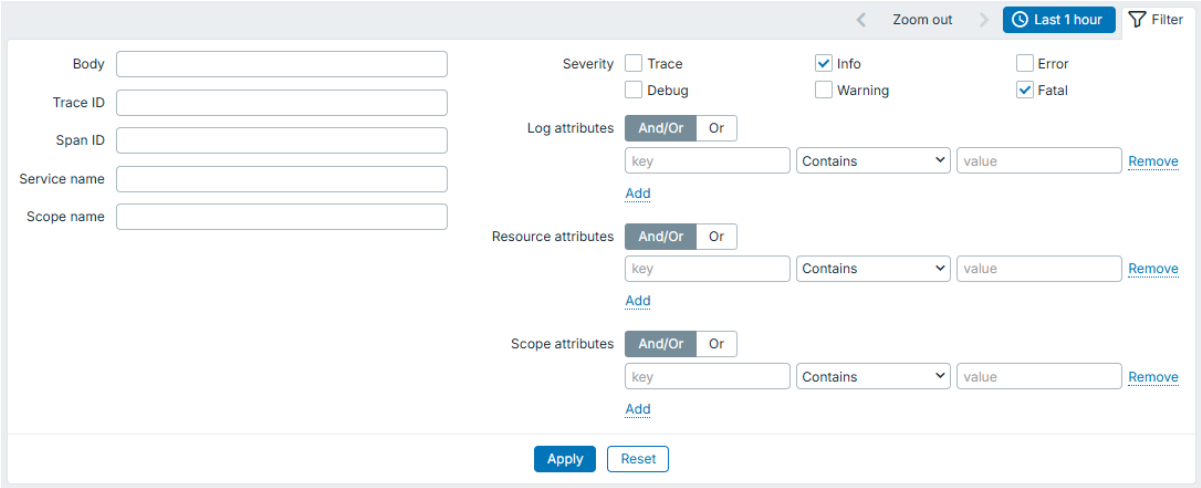
Il pannello *Dettagli del log* può contenere le seguenti sezioni:

Sezione	Descrizione
Informazioni di base	ID della traccia, ID dello span e flag della traccia del record di log.
Attributi del log	Attributi del log del record di log.

Sezione	Descrizione
Attributi della risorsa	URL dello schema e attributi della risorsa.
Attributi dell'ambito	URL dello schema, nome, versione e attributi dell'ambito di strumentazione.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo i record di log di interesse.



Per filtrare in base agli attributi, immettere la chiave dell'attributo, selezionare l'operatore e immettere il valore dell'attributo.

Sono supportati i seguenti operatori:

- **Contiene** - *(predefinito)* il valore dell'attributo contiene la stringa immessa.
- **Uguale** - il valore dell'attributo è uguale alla stringa immessa.
- **Non contiene** - il valore dell'attributo non contiene la stringa immessa.
- **Diverso da** - il valore dell'attributo è diverso dalla stringa immessa.
- **Esiste** - esiste l'attributo con la chiave immessa.
- **Non esiste** - non esiste l'attributo con la chiave immessa.

Se si aggiungono più condizioni, selezionare come valutarle:

- **And/Or** - *(predefinito)* devono essere soddisfatte tutte le condizioni; le condizioni con la stessa chiave dell'attributo vengono combinate con *Or*.
- **Or** - deve essere soddisfatta almeno una condizione.

Servizi

Panoramica

Il menu Servizi è per le funzioni monitoraggio servizi di Zabbix.

Servizi

Panoramica

In questa sezione è possibile visualizzare lo stato generale di tutti i servizi configurati in Zabbix, in base alla propria infrastruttura.

Un servizio può essere una gerarchia composta da diversi livelli di altri servizi, chiamati servizi "figlio", che contribuiscono allo stato complessivo del servizio (vedere anche una panoramica della funzionalità di monitoraggio dei servizi).

Le principali categorie dello stato del servizio sono *OK* o *Problema*, dove lo stato *Problema* viene espresso dal nome e dal colore della gravità del problema corrispondente.

La modalità di visualizzazione consente di monitorare i servizi insieme al loro stato e ad altri dettagli. In questa sezione è inoltre possibile configurare la gerarchia dei servizi (aggiungere/modificare servizi e servizi figlio) passando alla modalità di modifica.

Per passare dalla modalità di visualizzazione a quella di modifica (e viceversa), fare clic sul pulsante corrispondente nell'angolo superiore destro:

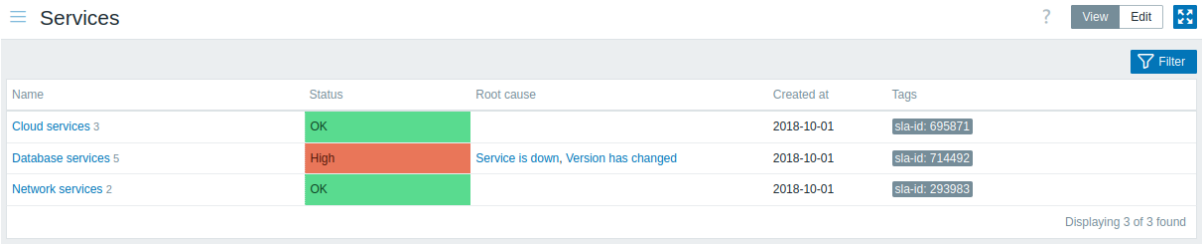
- 

- visualizza i servizi

- Edit - aggiunge/modifica servizi e servizi figlio

L'accesso alla modifica dipende dalle impostazioni del ruolo utente.

Visualizzazione dei servizi



Viene visualizzato un elenco dei servizi esistenti.

Dati visualizzati:

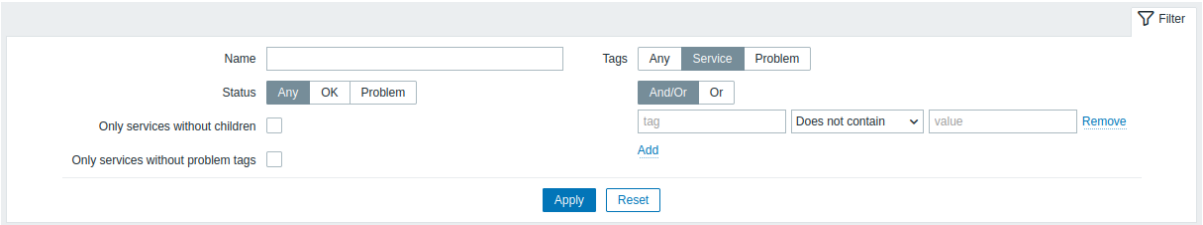
Parameter	Description
Name	Nome del servizio. Il nome del servizio è un collegamento a service details. Il numero dopo il nome indica quanti child services ha il servizio.
Status	Stato del servizio: **OK** - nessun problema **<problem color and severity>** - indica un problema e la sua gravità. In caso di più problemi, vengono visualizzati il colore e la gravità del problema più critico.
Root cause	Vengono elencati i problemi sottostanti che influenzano direttamente o indirettamente lo stato del servizio. Gli stessi problemi sono elencati come restituiti dalla macro {SERVICE.ROOTCAUSE} macro. Fare clic sul nome del problema per visualizzare ulteriori dettagli in *Monitoring* > *Problems*. I problemi che non influenzano lo stato del servizio non sono inclusi nell'elenco.
Created at	Viene visualizzato l'orario in cui il servizio è stato creato.
Tags	Vengono visualizzati i Tags del servizio. I tag vengono utilizzati per identificare un servizio nelle actions del servizio e negli SLAs.

Pulsanti

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Uso del filtro

È possibile utilizzare il filtro per visualizzare solo i servizi di interesse.



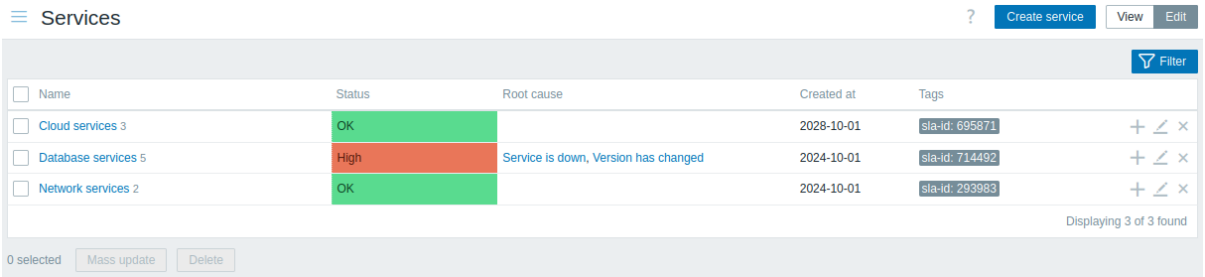
Parametro	Descrizione
Name	Filtra per nome del servizio.
Status	Filtra per stato del servizio.
Only services without children	Selezionare la casella per visualizzare solo i servizi senza servizi figli. Questo parametro è disponibile solo nella modalità di modifica del servizio.
Only services without problem tags	Selezionare la casella per visualizzare solo i servizi senza tag di problema. Questo parametro è disponibile solo nella modalità di modifica del servizio.

Parametro	Descrizione
Tags	Filtra per nome e valore del tag del servizio, oppure per nome e valore del tag di problema del servizio (nella modalità di modifica del servizio). È possibile sia includere sia escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi di tag specificati; **Equals** - include i nomi e i valori di tag specificati (con distinzione tra maiuscole e minuscole); **Contains** - include i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole); **Does not exist** - esclude i nomi di tag specificati; **Does not equal** - esclude i nomi e i valori di tag specificati (con distinzione tra maiuscole e minuscole); **Does not contain** - esclude i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, senza distinzione tra maiuscole e minuscole). Sono disponibili due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte; le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or; **Or** - è sufficiente che sia soddisfatta una sola condizione.

Modifica dei servizi

Fai clic sul pulsante *Edit* per accedere alla modalità di modifica. Quando sei in modalità di modifica, l'elenco viene integrato con caselle di controllo prima delle voci e anche con queste opzioni aggiuntive:

-  - aggiungi un servizio figlio a questo servizio
-  - modifica questo servizio
-  - elimina questo servizio

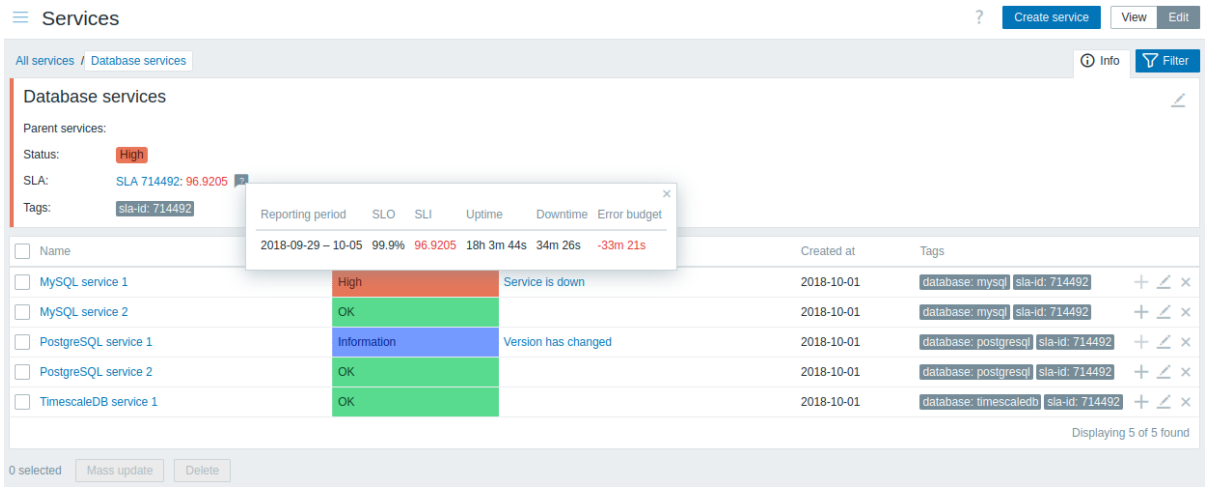


Per configurare un nuovo servizio, fai clic sul pulsante *Create service* nell'angolo in alto a destra.

Dettagli del servizio

Per accedere ai dettagli del servizio, fai clic sul nome del servizio. Per tornare all'elenco di tutti i servizi, fai clic su *Tutti i servizi*.

I dettagli del servizio includono il riquadro informazioni e l'elenco dei servizi figlio.



Per accedere al riquadro informazioni, fai clic sulla scheda *Info*. Il riquadro informazioni contiene le seguenti voci:

- Nomi dei servizi padre (se presenti)
- Stato corrente di questo servizio
- SLA corrente/i di questo servizio, nel formato `SLA name:service level indicator`. 'SLA name' è anche un collegamento al report SLA per questo servizio. Se posizioni il mouse sul riquadro informazioni accanto all'indicatore del livello di servizio (SLI), viene visualizzato un elenco informativo a comparsa con i dettagli dello SLI. L'indicatore del livello di servizio mostra il livello di servizio corrente, in percentuale.
- Tag del servizio

Il riquadro informazioni contiene anche un collegamento alla configurazione del servizio.

Per usare il filtro per i servizi figlio, fai clic sulla scheda *Filtro*.

Quando si è in modalità di modifica, l'elenco dei servizi figlio è completato con ulteriori opzioni di modifica:

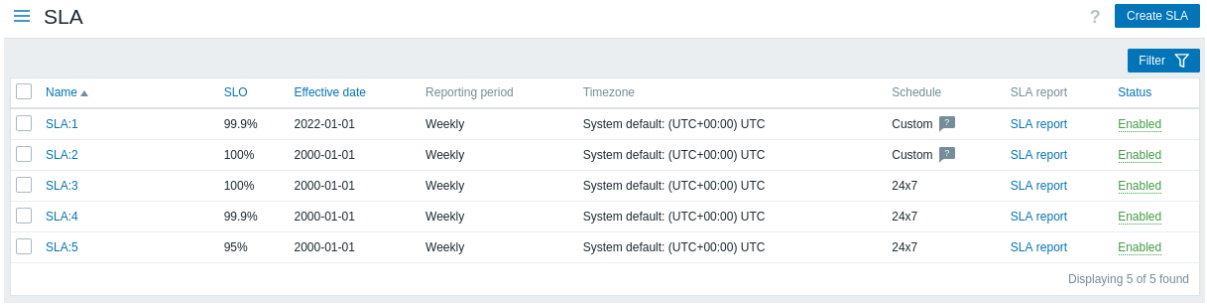
-  - aggiungi un servizio figlio a questo servizio
-  - modifica questo servizio
-  - elimina questo servizio

SLA

Panoramica

Questa sezione consente di visualizzare e configurare gli SLA.

SLA



Viene visualizzato un elenco degli SLA configurati. *Nota* che verranno visualizzati solo gli SLA relativi ai servizi accessibili all'utente (in sola lettura, a meno che *Manage SLA* non sia abilitato per il ruolo utente).

Dati visualizzati:

Parameter	Description
Name	Viene visualizzato il nome dello SLA. Il nome è un collegamento alla configurazione dello SLA.
SLO	Viene visualizzato l'obiettivo di livello di servizio (SLO).
Effective date	Viene visualizzata la data di inizio del calcolo dello SLA.

Parameter	Description
Reporting period	Viene visualizzato il periodo utilizzato nel report SLA - *daily*, *weekly*, *monthly*, *quarterly* o *annually*.
Time zone	Viene visualizzato il fuso orario dello SLA.
Schedule	Viene visualizzato il programma dello SLA - 24x7 o personalizzato.
SLA report	Fare clic sul collegamento per visualizzare il report SLA per questo SLA.
Status	Viene visualizzato lo stato dello SLA - abilitato o disabilitato.

Report SLA

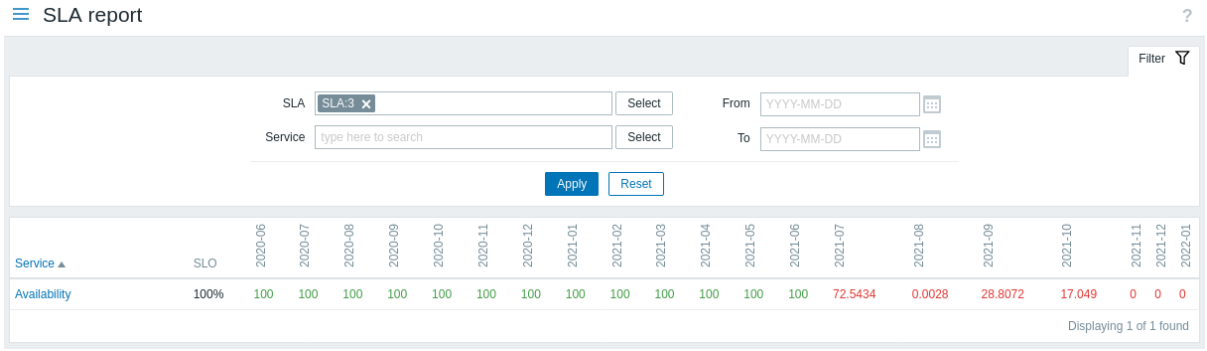
Panoramica

Questa sezione consente di visualizzare i report SLA in base ai criteri selezionati nel filtro.

I report SLA possono essere visualizzati anche come widget del dashboard.

Report

Il filtro consente di selezionare il report in base al nome dello SLA e al nome del servizio. È inoltre possibile limitare il periodo visualizzato.

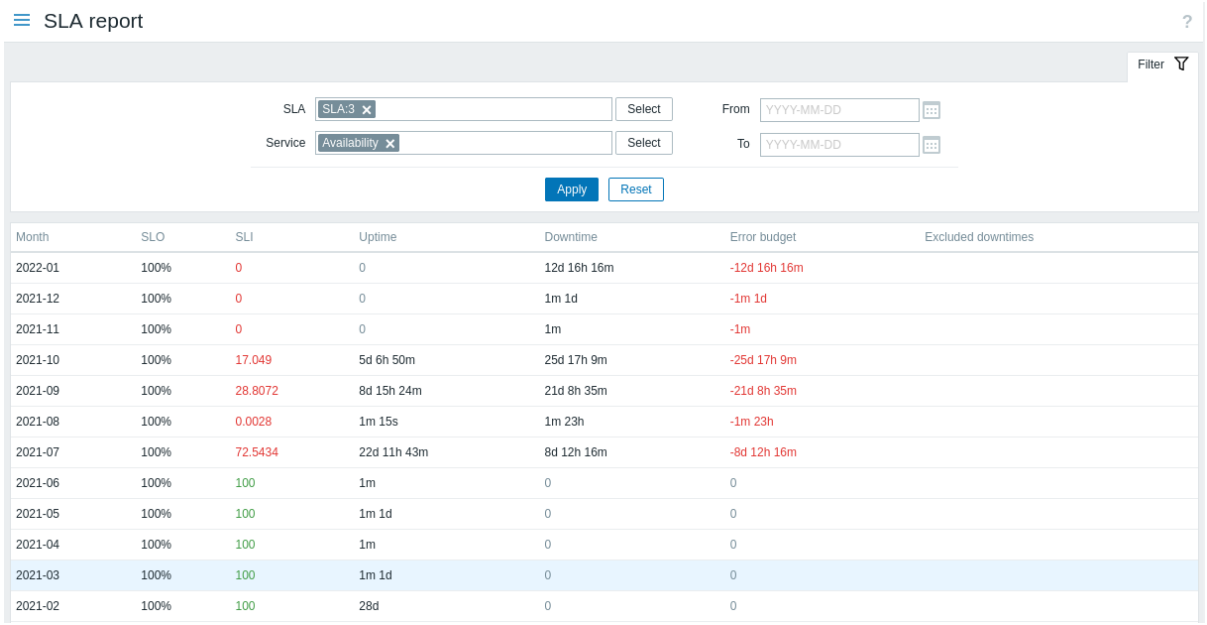


Ogni colonna (periodo) visualizza lo SLI relativo a tale periodo. Gli SLI che violano l'SLO impostato sono evidenziati in rosso.

Nel report vengono visualizzati 20 periodi. È possibile visualizzare un massimo di 100 periodi se vengono specificate sia la data *Da* sia la data *A*.

Dettagli del report

Se fai clic sul nome del service nel report, puoi accedere a un altro report che mostra una vista più dettagliata.



Tieni presente che la durata negativa del problema non influisce sul calcolo o sulla reportistica dell'SLA.

Inventario

Panoramica

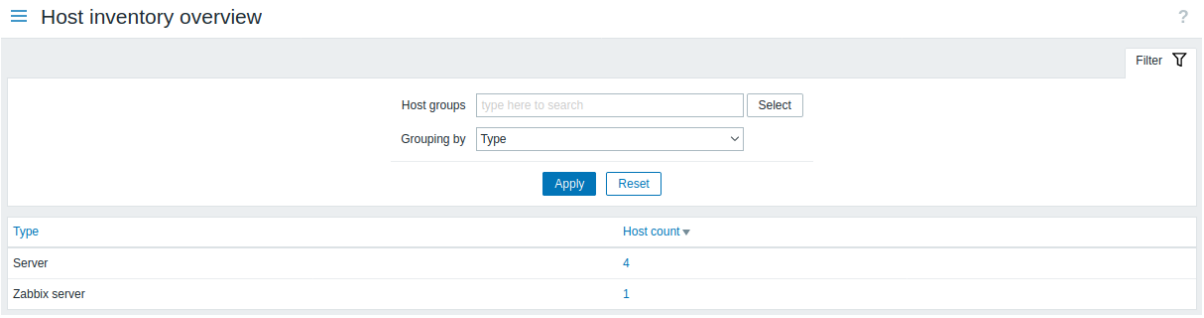
Il menu Inventario presenta sezioni che forniscono una panoramica dei dati dell'host in base a un parametro scelto, nonché la possibilità di visualizzare di dettagli dell'inventario dell'host.

Panoramica

Panoramica

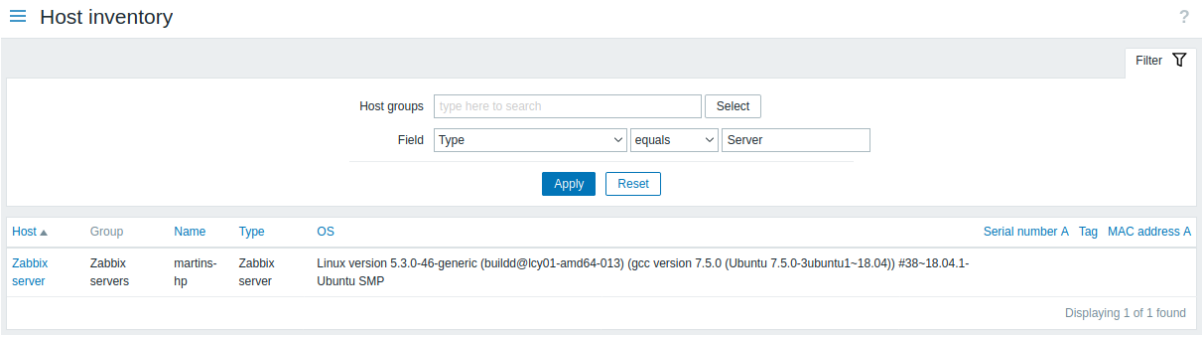
La sezione *Inventory* > *Overview* fornisce modi per ottenere una panoramica dei dati di host inventory.

Per visualizzare una panoramica, scegli i gruppi di host (oppure nessuno) e il campo di inventory in base al quale visualizzare i dati. Verrà mostrato il numero di host corrispondenti a ciascuna voce del campo scelto.



La completezza di una panoramica dipende da quante informazioni di inventory vengono mantenute per gli host.

I numeri nella colonna *Host count* sono collegamenti; portano alla visualizzazione di questi host filtrati nella tabella *Host Inventories*.

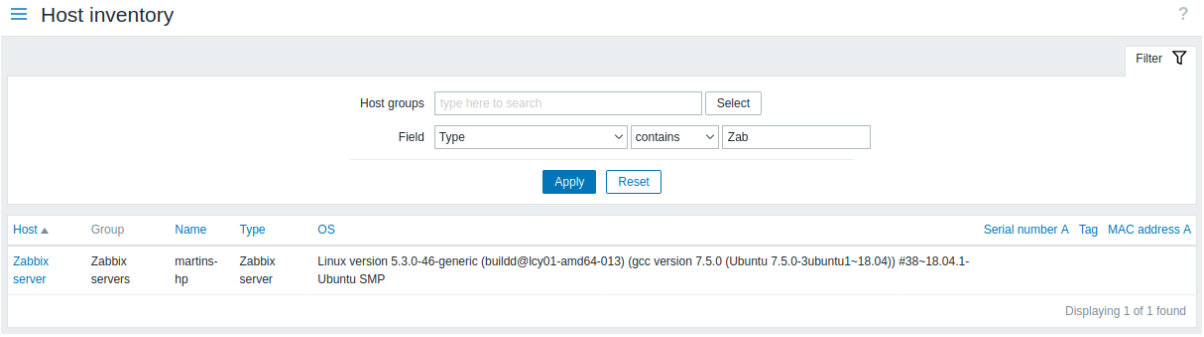


Host

Panoramica

Nella sezione *Inventory* > *Hosts* vengono visualizzati i dati di inventario degli host.

È possibile filtrare gli host per uno o più gruppi di host e per qualsiasi campo di inventario, in modo da visualizzare solo gli host di interesse.

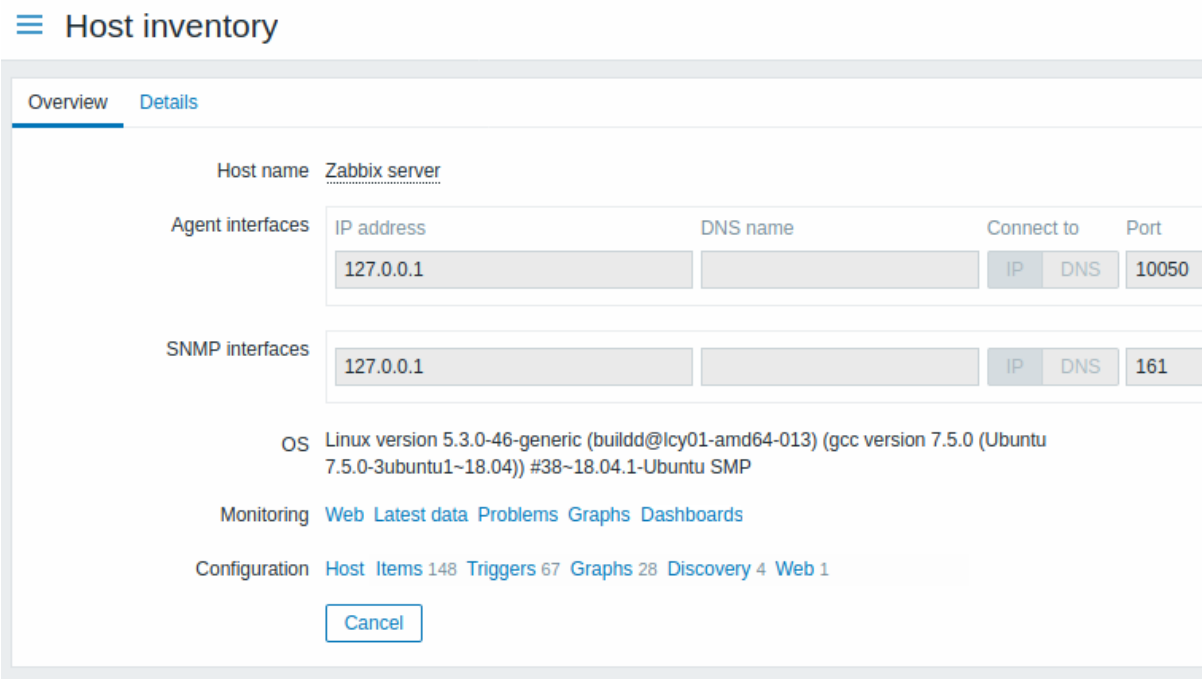


Per visualizzare tutti gli inventari degli host, fare clic sul pulsante *Reset*.

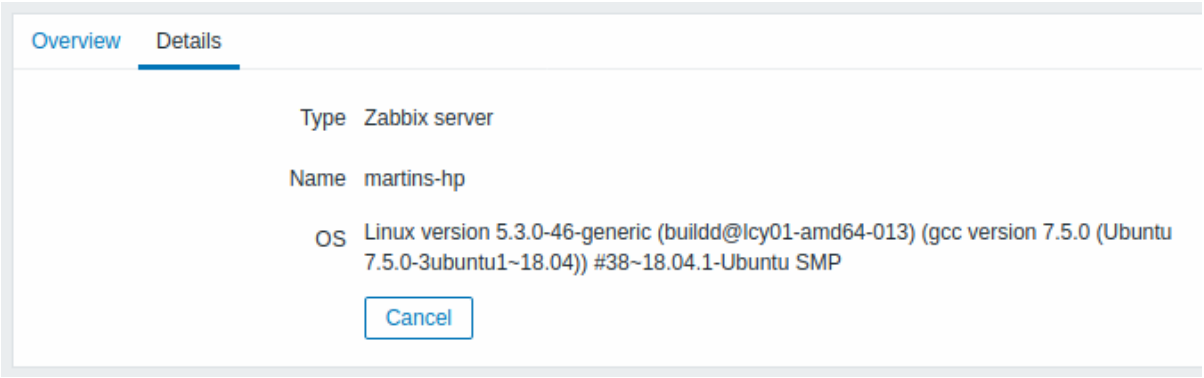
Sebbene nella tabella vengano mostrati solo alcuni campi chiave dell'inventario, è anche possibile visualizzare tutte le informazioni di inventario disponibili per quell'host. Per farlo, fare clic sul nome dell'host nella prima colonna.

Dettagli dell'inventario

La scheda **Panoramica** contiene alcune informazioni generali sul host, gli ultimi dati di monitoraggio e le opzioni di configurazione del host:



La scheda **Dettagli** contiene tutti i dettagli di inventario disponibili per il host:



La completezza dei dati di inventario dipende dalla quantità di informazioni di inventario mantenute per il host. Se non viene mantenuta alcuna informazione, la scheda *Dettagli* è disabilitata.

Rapporti

Panoramica

Il menu Rapporti presenta diverse sezioni che ne contengono una varietà report predefiniti e personalizzabili dall'utente incentrati sulla visualizzazione di una panoramica di parametri quali informazioni di sistema, trigger e dati raccolti.

Informazioni di sistema

Panoramica

In *Report* > *Informazioni di sistema* viene visualizzato un riepilogo dei dati principali del server Zabbix e del sistema. I dati di sistema vengono raccolti utilizzando gli item interni.

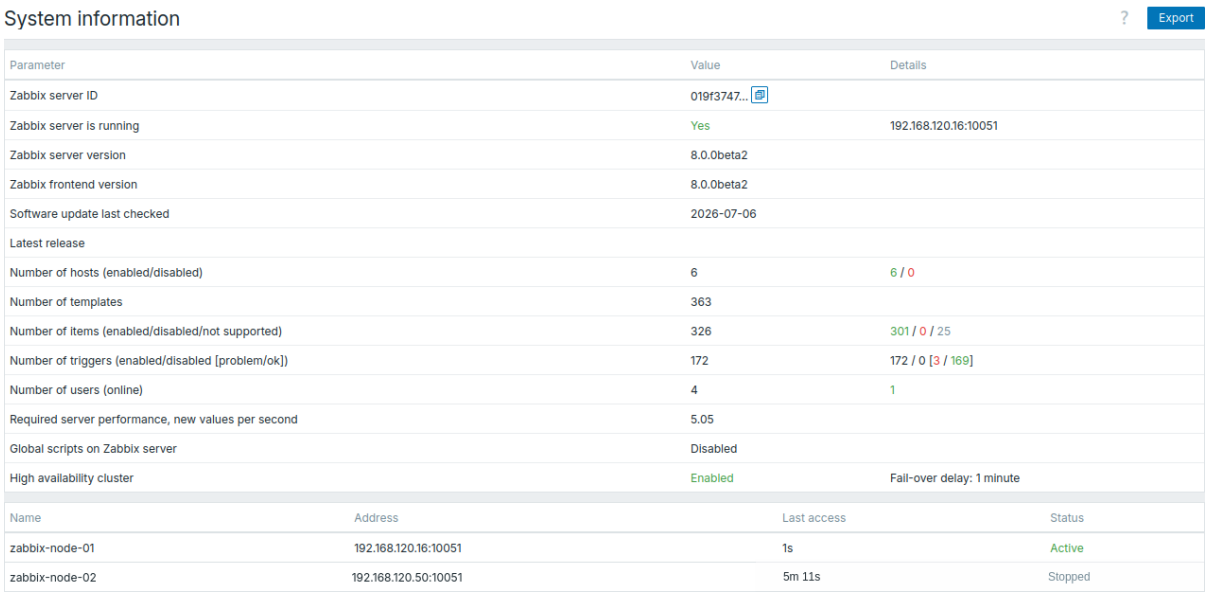
Il report delle informazioni di sistema è visibile solo agli utenti Zabbix *Super Admin* e Zabbix *Admin*.

In una configurazione ad alta disponibilità, è possibile reindirizzare l'origine delle informazioni di sistema (istanza del server). A tale scopo, modificare il file di configurazione del frontend - rimuovere il commento e impostare `$ZBX_SERVER` oppure sia `$ZBX_SERVER` sia `$ZBX_SERVER_PORT` su un server diverso da quello attivo visualizzato. Se si imposta solo `$ZBX_SERVER`, verrà utilizzato il valore predefinito (10051) per `$ZBX_SERVER_PORT`.

Con la configurazione ad alta disponibilità abilitata, sotto le statistiche di sistema viene visualizzato un blocco separato con i dettagli dei nodi ad alta disponibilità.

Le *Informazioni di sistema* sono disponibili anche come widget del dashboard.

Statistiche del sistema



Dati visualizzati:

Parametro	Valore	Dettagli
ID del server Zabbix	Identificatore univoco del server Zabbix (UUIDv7).	L'ID del server viene generato al primo avvio del server. Vengono visualizzati i primi 8 caratteri, seguiti da un'ellissi. Passando il mouse sul valore dell'ID del server o facendo clic su di esso, viene visualizzato un tooltip con il valore completo. Facendo clic sull'icona accanto al valore, l'ID viene copiato negli appunti. Se è abilitata la disponibilità elevata, ogni nodo dispone di un proprio ID del nodo per l'indirizzamento dalla riga di comando, ma tutti i nodi sono raggruppati sotto un unico ID del server generato dal nodo avviato per primo (o che in precedenza era un server autonomo), indipendentemente dal nodo attualmente attivo.
Il server Zabbix è in esecuzione	Stato del server Zabbix: **Sì** - il server è in esecuzione **No** - il server non è in esecuzione *Nota:* per visualizzare le informazioni rimanenti, il web frontend necessita che il server sia in esecuzione e che sul server sia stato avviato almeno un processo trapper (il parametro StartTrappers nel file `zabbix_server.conf` > 0).	Posizione e porta del server Zabbix.

Parametro	Valore	Dettagli
Versione del server Zabbix	Viene visualizzato il numero di versione corrente del server. *Nota:* viene visualizzato solo quando il server Zabbix è in esecuzione.	Viene visualizzato lo stato della versione del server: **Aggiornata** - è in uso la versione più recente; **Nuovo aggiornamento disponibile** - è disponibile una versione più recente; **Obsoleta** - il periodo di supporto completo per questa versione è terminato. Queste informazioni sono disponibili solo se il controllo degli aggiornamenti software è abilitato nella configurazione del server Zabbix. Non viene visualizzato nulla se l'ultimo controllo degli aggiornamenti software è stato eseguito più di una settimana fa o se non sono disponibili dati sulla versione corrente.
Versione del frontend Zabbix	Viene visualizzato il numero di versione del frontend Zabbix.	Viene visualizzato lo stato della versione del frontend Zabbix: **Aggiornata** - è in uso la versione più recente; **Nuovo aggiornamento disponibile** - è disponibile una versione più recente; **Obsoleta** - il periodo di supporto completo per questa versione è terminato. Queste informazioni sono disponibili solo se il controllo degli aggiornamenti software è abilitato nella configurazione del server Zabbix. Non viene visualizzato nulla se l'ultimo controllo degli aggiornamenti software è stato eseguito più di una settimana fa o se non sono disponibili dati sulla versione corrente.
Ultimo controllo degli aggiornamenti software	Viene visualizzata la data dell'ultimo controllo degli aggiornamenti software di Zabbix. Queste informazioni sono disponibili solo se il controllo degli aggiornamenti software è abilitato nella configurazione del server.	
Ultima release	Viene visualizzato il numero di una release più recente (se disponibile) per la versione corrente di Zabbix. Queste informazioni sono disponibili solo se il controllo degli aggiornamenti software è abilitato nella configurazione del server. Non viene visualizzato nulla se l'ultimo controllo degli aggiornamenti software è stato eseguito più di una settimana fa o se non sono disponibili dati sulla versione corrente.	Viene visualizzato un collegamento alle note di rilascio dell'ultima release disponibile di Zabbix.
Numero di host	Viene visualizzato il numero totale di host configurati.	Numero di host monitorati/non monitorati.
Numero di template	Viene visualizzato il numero totale di template.	
Numero di item	Viene visualizzato il numero totale di item.	Numero di item a livello di host monitorati/disabilitati/non supportati. Gli item sugli host disabilitati vengono conteggiati come disabilitati.
Numero di trigger	Viene visualizzato il numero totale di trigger.	Numero di trigger a livello di host abilitati/disabilitati; suddivisione dei trigger abilitati in base agli stati "Problema"/"OK". I trigger elencati nello stato "OK" includono i trigger con stato "Sconosciuto". I trigger che dipendono da item disabilitati o assegnati a host disabilitati vengono conteggiati come disabilitati.
Numero di utenti	Viene visualizzato il numero totale di utenti configurati.	Numero di utenti online.

Parametro	Valore	Dettagli
Prestazioni richieste del server, nuovi valori al secondo	Viene visualizzato il numero previsto di nuovi valori elaborati dal server Zabbix al secondo.	Le *prestazioni richieste del server* sono una stima e possono essere utili come riferimento. Per ottenere il numero preciso di valori elaborati, utilizzare l'item interno `zabbix[wcache,values,all]`. Nel calcolo sono inclusi gli item abilitati degli host monitorati. Gli item di log vengono conteggiati come un valore per ogni intervallo di aggiornamento dell'item. Vengono conteggiati i valori con intervallo regolare; i valori con intervallo flessibile e pianificato non vengono conteggiati. Il calcolo non viene modificato durante un periodo di manutenzione "nodata". Gli item trapper non vengono conteggiati.
Script globali sul server Zabbix	In questo campo viene visualizzato **Disabilitato** se gli script globali sono disabilitati sul server Zabbix impostando EnableGlobalScripts=0 nella configurazione del server.	
Cluster ad alta disponibilità	Stato del cluster ad alta disponibilità per il server Zabbix: **Disabilitato** - server autonomo **Abilitato** - esiste almeno un nodo ad alta disponibilità	Se abilitato, viene visualizzato il ritardo di failover.

Informazioni di sistema visualizzerà inoltre un messaggio di errore nelle seguenti condizioni:

- Il database utilizzato non dispone del set di caratteri o della collation richiesti (UTF-8).
- La versione del database è inferiore o superiore all'intervallo supportato (disponibile solo per gli utenti con il tipo di *ruolo Super admin*).
- La gestione automatica per TimescaleDB è configurata in modo errato (le tabelle della cronologia o dei trend contengono chunk compressi, ma le opzioni *Sovrascrivi il periodo della cronologia degli item* o *Sovrascrivi il periodo dei trend degli item* sono disabilitate).

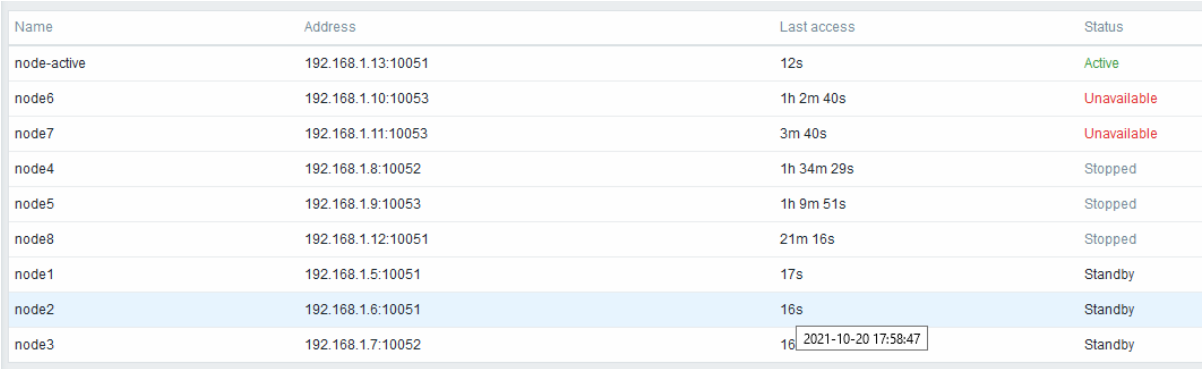
Pulsanti

Il pulsante nell'angolo in alto a destra della pagina offre la seguente opzione:

Export	Esporta le informazioni di sistema in un file JSON. Il file viene salvato come `Zabbix-system-information-{time}.json`, dove `{time}` è il timestamp Unix che indica quando i dati sono stati preparati sul server. Se l'alta disponibilità è abilitata, vengono esportati anche i dettagli dei nodi ad alta disponibilità.

Nodi ad alta disponibilità

Se high availability cluster è abilitato, viene visualizzato un blocco aggiuntivo di dati che mostra lo stato di ciascun nodo ad alta disponibilità.



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

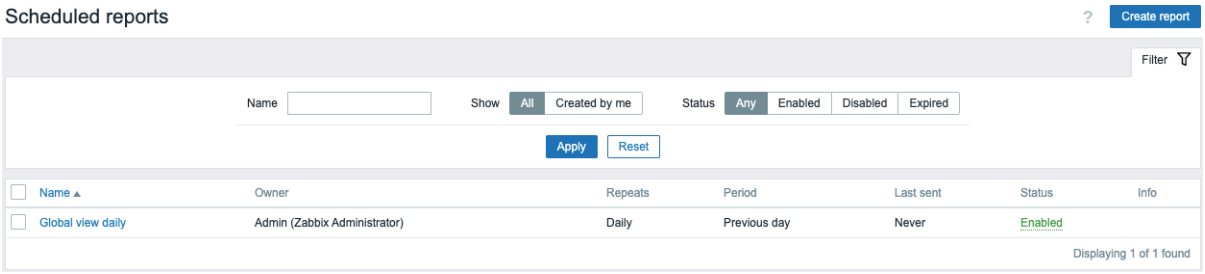
Dati visualizzati:

Column	Description
Name	Nome del nodo, come definito nella configurazione del server.
Address	Indirizzo IP e porta del nodo.
Last access	Ora dell'ultimo accesso al nodo. Passando il mouse sulla cella viene mostrato il timestamp dell'ultimo accesso in formato esteso.
Status	Stato del nodo. Le righe della tabella sono ordinate in base a questi stati nel seguente ordine di priorità: **Active** - il nodo è attivo e funzionante **Unavailable** - il nodo non è stato rilevato per più del ritardo di failover (potrebbe essere utile capire il motivo) **Stopped** - il nodo è stato arrestato o non è riuscito ad avviarsi (potrebbe essere utile avviarlo o eliminarlo) **Standby** - il nodo è attivo e in attesa

Report pianificati

Panoramica

In *Report* > *Report pianificati*, gli utenti con autorizzazioni sufficienti possono configurare la generazione pianificata di versioni PDF dei dashboard, che verranno inviate tramite email ai destinatari specificati.



La schermata iniziale mostra le informazioni sui report pianificati, che possono essere filtrate per facilitare la navigazione - vedere la sezione Utilizzo del filtro di seguito.

Dati visualizzati:

Colonna	Descrizione
Nome	Nome del report. Facendo clic su di esso si apre il modulo di configurazione del report.
Proprietario	Utente che ha creato il report.
Ripetizione	Frequenza di generazione del report (giornaliera/settimanale/mensile/annuale).
Periodo	Periodo di tempo per il quale viene preparato il report (giorno, settimana, mese o anno precedente).
Ultimo invio	Data e ora in cui è stato inviato il report più recente.
Stato	Stato attuale del report (abilitato/disabilitato/scaduto). Gli utenti con autorizzazioni sufficienti possono modificare lo stato facendo clic su di esso: da "Abilitato" a "Disabilitato" e viceversa; da "Scaduto" a "Disabilitato" e viceversa. Per gli utenti senza diritti sufficienti, lo stato non è selezionabile.
Informazioni	Visualizza icone informative: Un'icona rossa indica che la generazione del report non è riuscita; passando il mouse sopra di essa viene visualizzato un tooltip con le informazioni sull'errore. Un'icona gialla indica che un report è stato generato, ma l'invio ad alcuni o a tutti i destinatari non è riuscito, oppure che un report è scaduto; passando il mouse sopra di essa viene visualizzato un tooltip con informazioni aggiuntive.

Utilizzo del filtro

È possibile utilizzare il filtro per restringere l'elenco dei report. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Sono disponibili le seguenti opzioni di filtro:

- *Nome* - è consentita la corrispondenza parziale del nome
- *Mostra* - creati dall'utente corrente oppure tutti i report
- *Stato* - selezionare tra "Qualsiasi" (mostra tutti i report), "Abilitato", "Disabilitato" oppure "Scaduto"

Il filtro si trova sotto il nome della sezione *Report pianificati*. Può essere aperto e compresso facendo clic sulla scheda *Filtro* nell'angolo superiore destro.

Aggiornamento di massa

A volte potresti voler eliminare o modificare lo stato di più report contemporaneamente. Invece di aprire ogni singolo report per modificarlo, puoi utilizzare la funzione di aggiornamento di massa.

Per aggiornare in massa alcuni report, procedi come segue:

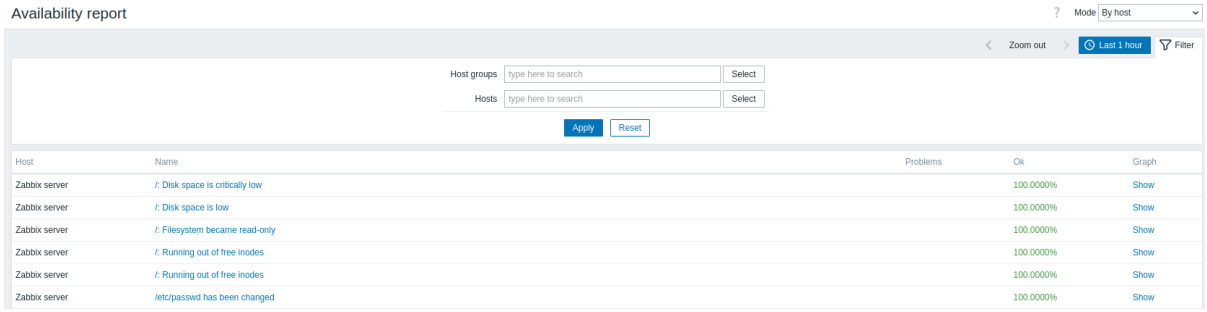
- Seleziona le caselle di controllo dei report da aggiornare nell'elenco
- Fai clic sul pulsante richiesto sotto l'elenco per apportare le modifiche (*Abilita*, *Disabilita* oppure *Elimina*)

Rapporto di disponibilità

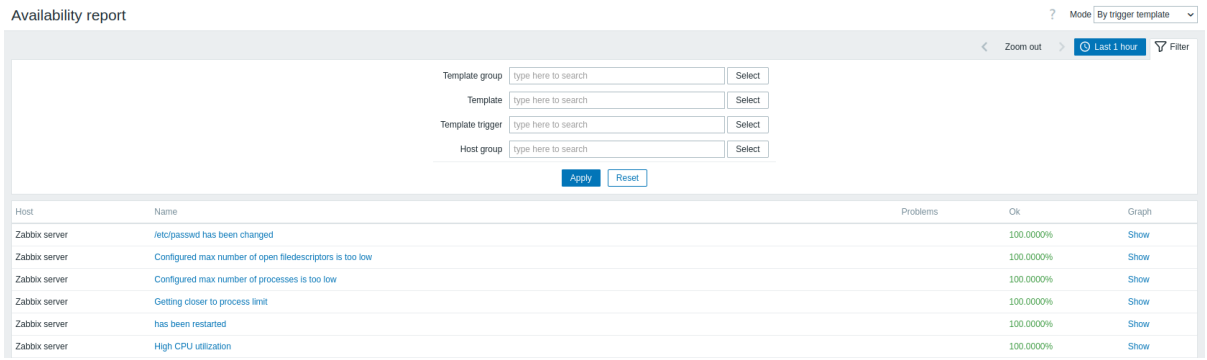
Panoramica

In *Report* > *Report di disponibilità* puoi vedere per quale proporzione di tempo ciascun trigger è stato nello stato di problema/OK.

Per ciascuno stato viene visualizzata una percentuale di tempo, rendendo semplice determinare la disponibilità di vari elementi del sistema.



Nel menu a discesa nell'angolo superiore destro puoi scegliere la modalità di selezione, ossia se visualizzare i trigger per host oppure i trigger appartenenti a un template.



Il nome del trigger è un collegamento agli eventi più recenti di quel trigger.

Uso del filtro

Il filtro può aiutare a restringere il numero di host e/o trigger visualizzati. Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Il filtro si trova sotto il nome della sezione *Availability report*. Può essere aperto e chiuso facendo clic sulla scheda *Filter* a destra.

Filtraggio per template di trigger

Nella modalità *Per template di trigger*, i risultati possono essere filtrati in base a uno o più dei parametri elencati di seguito.

Parametro	Descrizione
Gruppo di template	Filtra gli host in base ai trigger ereditati dai template che appartengono al gruppo di template selezionato. La specifica di un gruppo di template padre seleziona implicitamente tutti i gruppi di template nidificati.
Template	Filtra gli host in base ai trigger ereditati dal template selezionato, inclusi i template nidificati. Se un template nidificato ha trigger propri, tali trigger non verranno visualizzati.

Parametro	Descrizione
Trigger del template	Filtra gli host in base al trigger selezionato. Gli altri trigger degli host filtrati non verranno visualizzati.
Gruppo di host	Filtra gli host che appartengono al gruppo di host selezionato.

Filtraggio per host

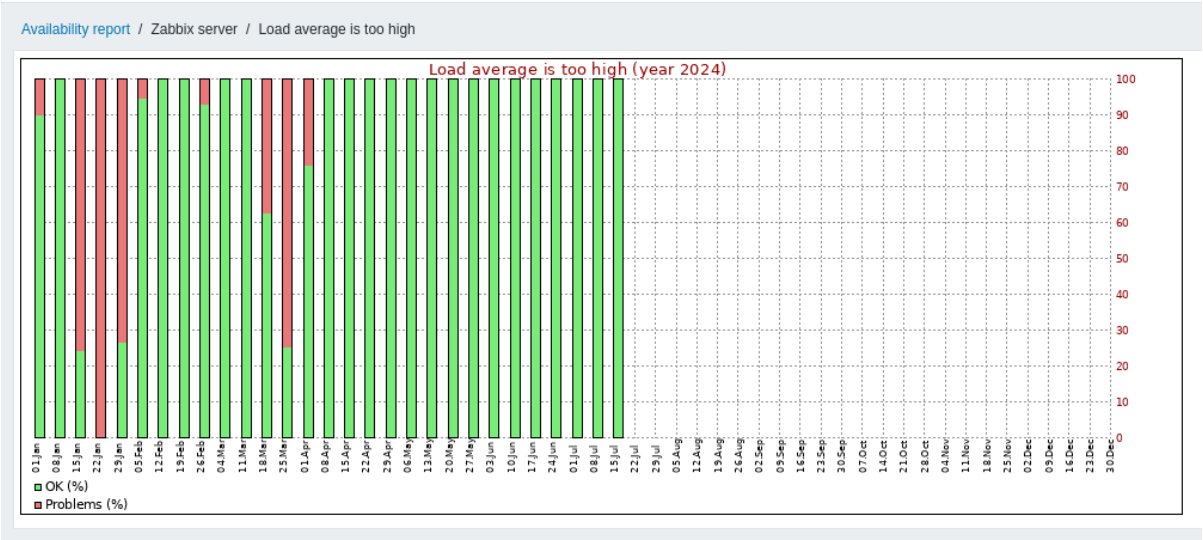
Nella modalità *Per host*, i risultati possono essere filtrati per host o gruppo di host. Specificando un gruppo di host padre, vengono selezionati implicitamente tutti i gruppi di host nidificati.

Selettore del periodo di tempo

Il selettore del periodo di tempo consente di selezionare i periodi di tempo utilizzati più frequentemente con un solo clic. Il selettore può essere espanso e compresso facendo clic sulla scheda *Periodo di tempo* accanto al filtro.

Facendo clic su *Mostra* nella colonna *Grafico* vengono visualizzate le informazioni sulla disponibilità in un grafico a barre, in cui ogni barra rappresenta una settimana trascorsa dell'anno corrente.

Availability report graph



Il verde di una barra indica il tempo OK, mentre il rosso indica il tempo di problema.

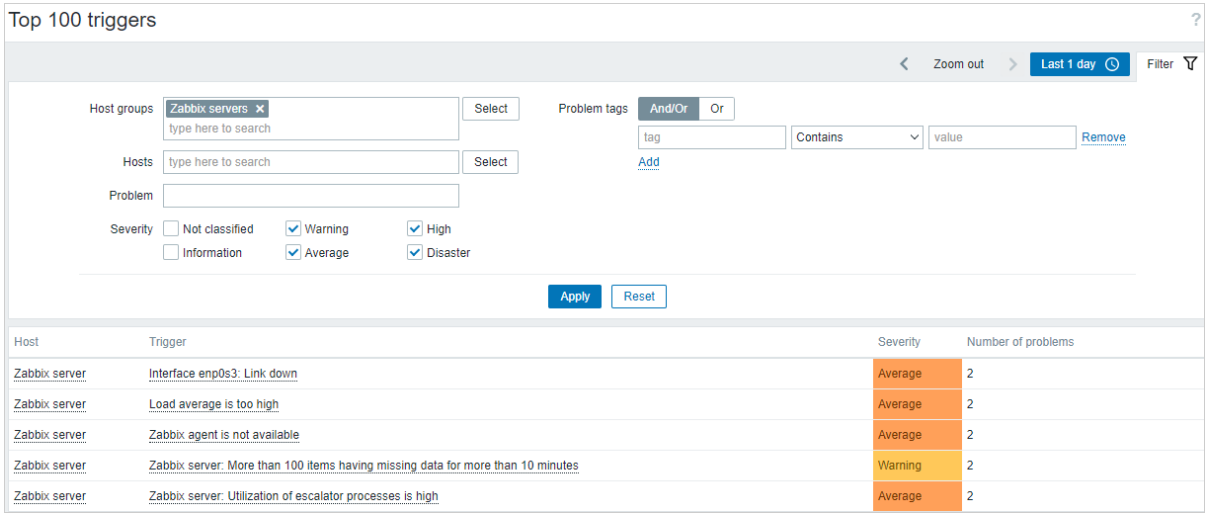
Effetto dei periodi di manutenzione

Maintenance non esclude automaticamente il tempo dal report di disponibilità. Solo la maintenance configurata come *No data collection* interrompe il polling (quindi non vengono generati problemi) e, di conseguenza, lascia invariata la percentuale di disponibilità per i trigger interessati.

I 100 trigger principali

Panoramica

In *Report* > *Top 100 trigger*, è possibile visualizzare i trigger con il maggior numero di problemi rilevati durante il periodo selezionato.



Le voci delle colonne host e trigger sono entrambe collegamenti che offrono alcune opzioni utili:

- per l'host - facendo clic sul nome dell'host viene visualizzato il menu dell'host
- per il trigger - facendo clic sul nome del trigger vengono visualizzati i collegamenti agli eventi più recenti, al grafico semplice per ogni item del trigger e ai moduli di configurazione del trigger stesso e di ogni item del trigger

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare i trigger in base al gruppo di host, all'host, al nome del problema, ai tag o alla gravità del trigger. Specificando un gruppo di host padre, vengono selezionati implicitamente tutti i gruppi di host nidificati. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Il filtro si trova sotto il nome della sezione *Top 100 trigger*. È possibile aprirlo e comprimerlo facendo clic sulla scheda *Filtro* a destra.

Selettore del periodo di tempo

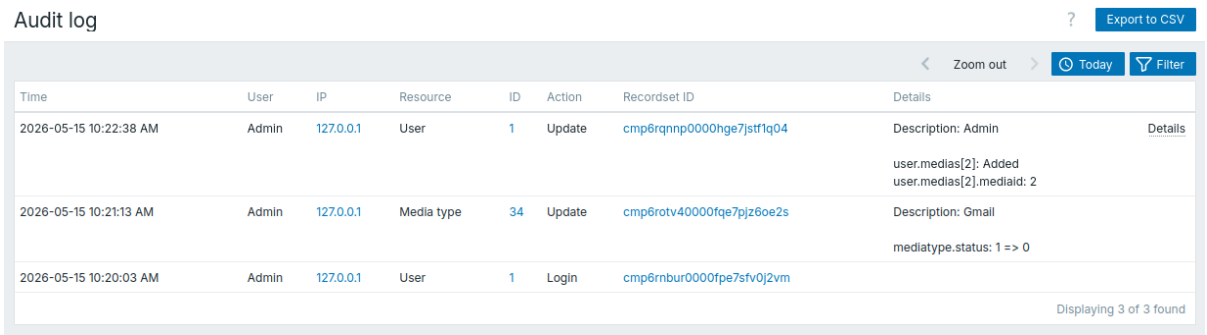
Il selettore del periodo di tempo accanto al filtro consente di selezionare con un solo clic del mouse i periodi utilizzati più frequentemente. Per ulteriori informazioni, vedere Selettori del periodo di tempo e dell'host.

Registro di audit

Panoramica

Nella sezione *Reports* > *Audit log* è possibile visualizzare i record delle attività degli utenti e del sistema.

Note:
Per raccogliere e visualizzare i record di audit, la casella di controllo *Enable audit logging* deve essere selezionata nella sezione *Administration* > *Audit log*. Se questa impostazione non è abilitata, la cronologia delle attività non verrà registrata nel database e non verrà mostrata nell'audit log.



L'audit log mostra i seguenti dati:

Column	Description
Time	Timestamp del record di audit.
User	Utente che ha eseguito l'attività.

Column	Description
IP	IP da cui è stata avviata l'attività. Facendo clic sul collegamento ipertestuale, i record dell'audit log verranno filtrati in base a questo IP.
Resource	Tipo della risorsa interessata (*API token*, *Action*, *Authentication*, *Autoregistration*, ecc.).
ID	ID della risorsa interessata. Facendo clic sul collegamento ipertestuale, i record dell'audit log verranno filtrati in base a questo ID risorsa.
Action	Tipo dell'attività (*Add*, *Configuration refresh*, *Delete*, *Execute*, *Failed login*, *History clear*, *Login*, *Logout*, *Push*, *Update*).
Recordset ID	ID condiviso per tutti i record dell'audit log creati come risultato della stessa operazione. Ad esempio, quando si collega un template a un host, viene creato un record separato dell'audit log per ogni entità del template ereditata (item, trigger, ecc.) - tutti questi record avranno lo stesso *Recordset ID*. Facendo clic sul collegamento ipertestuale, i record dell'audit log verranno filtrati in base a questo *Recordset ID*.
Details	Descrizione della risorsa e informazioni dettagliate sull'attività eseguita. Se un record contiene più di due righe, verrà visualizzato un collegamento aggiuntivo *Details*. Fare clic su questo collegamento per visualizzare l'elenco completo delle modifiche.

Note:
Quando un trapper item o un HTTP agent item (con trapping abilitato) ha ricevuto alcuni dati, una voce nell'audit log verrà aggiunta solo se i dati sono stati inviati utilizzando il metodo API `history.push`, e non l'utilità Zabbix sender.

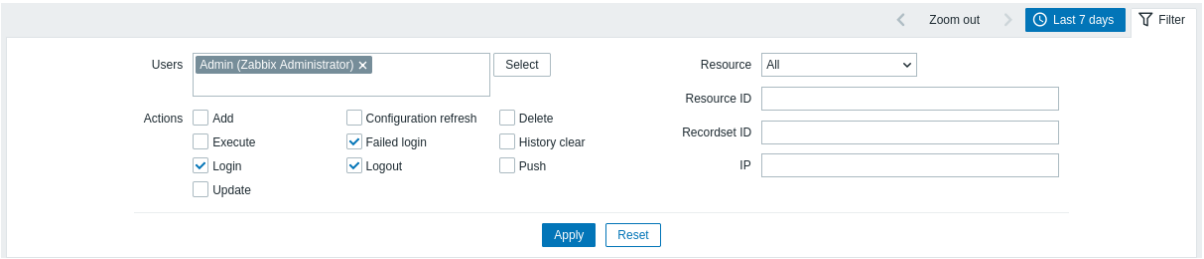
Pulsanti

Il pulsante nell'angolo in alto a destra della pagina offre la seguente opzione:

Export to CSV	Esporta i record del registro di audit da tutte le pagine in un file CSV. Se viene applicato un filtro, verranno esportati solo i record filtrati.

Utilizzo del filtro

Il filtro si trova sotto la barra del *Log di audit*. Può essere aperto e compresso facendo clic sulla scheda *Filtro* nell'angolo superiore destro.



È possibile utilizzare il filtro per restringere i record in base a utente, risorsa interessata, ID risorsa, operazione eseguita (*ID del set di record*) e IP. A seconda della risorsa, nel filtro è possibile selezionare una o più azioni specifiche.

Per ottenere prestazioni di ricerca migliori, tutti i dati vengono cercati con le macro non risolte.

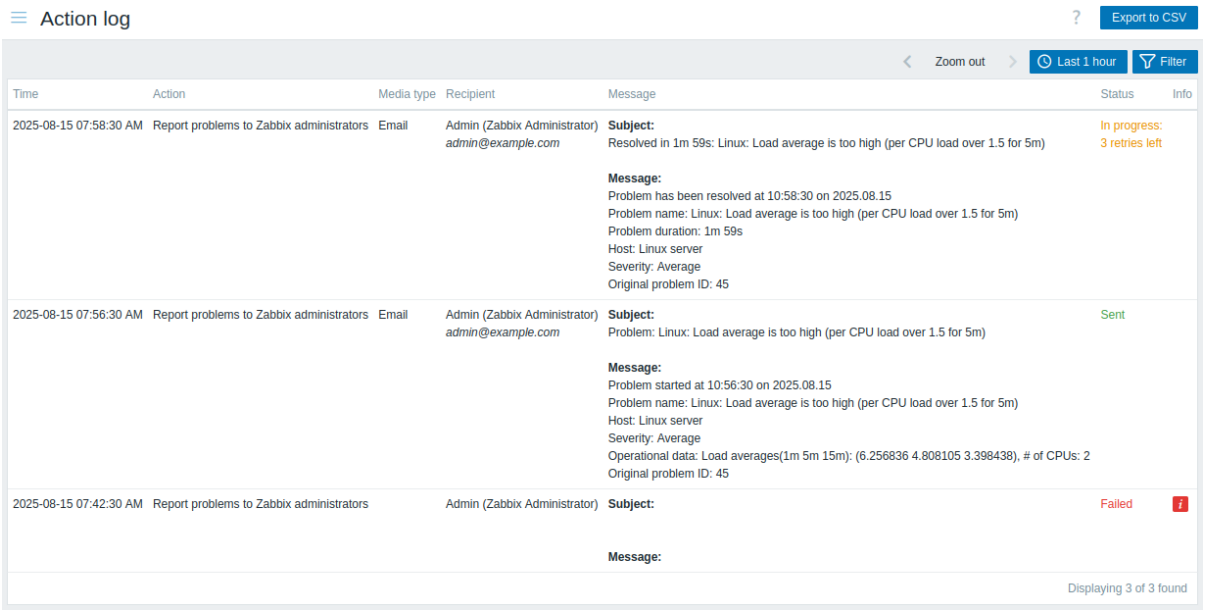
Selettore del periodo di tempo

Il selettore del periodo di tempo accanto al filtro consente di selezionare con un solo clic del mouse i periodi utilizzati più frequentemente. Per ulteriori informazioni, vedere Selettori del periodo di tempo e dell'host.

Registro delle azioni

Panoramica

In *Reports* > *Action log*, puoi visualizzare i dettagli delle operazioni (notifiche, comandi remoti) eseguite all'interno di un'azione.



Dati visualizzati:

Column	Description
Time	Timestamp dell'operazione.
Action	Nome dell'azione che causa le operazioni.
Media type	Tipo di media (ad es. Email, Jira, ecc.) usato per inviare una notifica. Per le operazioni che hanno eseguito comandi remoti, questa colonna sarà vuota.
Recipient	Informazioni sul destinatario della notifica - username, nome e cognome (tra parentesi), e informazioni aggiuntive a seconda del tipo di media (email, username, ecc.). Per le operazioni che hanno eseguito comandi remoti, questa colonna sarà vuota.
Message	Il contenuto del messaggio, del comando remoto o del nome dello script globale. Un comando remoto è separato dall'host di destinazione con il simbolo dei due punti: `<host>:<command>`. Ad esempio, se il comando remoto è stato eseguito su Zabbix server, le informazioni avranno il seguente formato: `Zabbix server:<command>`.
Status	Stato dell'operazione: *In progress* - l'operazione di invio di una notifica è in corso (viene inoltre visualizzato il numero rimanente di tentativi che il server effettuerà per inviare la notifica) *Sent* - la notifica è stata inviata *Executed* - il comando remoto è stato eseguito *Failed* - l'operazione non è stata completata
Info	Informazioni sugli errori (se presenti) relative all'esecuzione dell'operazione.

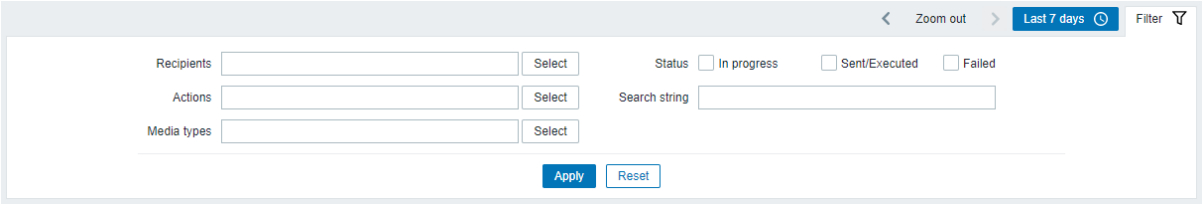
Pulsanti

Il pulsante nell'angolo superiore destro della pagina offre la seguente opzione:

Export to CSV	Esporta i record del log delle azioni dalla pagina corrente in un file CSV. Se è applicato un filtro, verranno esportati solo i record filtrati. Nel file CSV esportato, le colonne "Recipient" e "Message" sono suddivise in più colonne: "Recipient's Zabbix username", "Recipient's name", "Recipient's surname", "Recipient" e "Subject", "Message", "Command".

Utilizzo del filtro

Il filtro si trova sotto la barra *Action log*. Può essere aperto e compresso facendo clic sulla scheda *Filter* nell'angolo in alto a destra della pagina.



È possibile usare il filtro per restringere i record in base ai destinatari delle notifiche, alle azioni, ai tipi di media, allo stato, oppure in base al contenuto del messaggio/comando remoto (*Search string*). Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Selettore del periodo di tempo

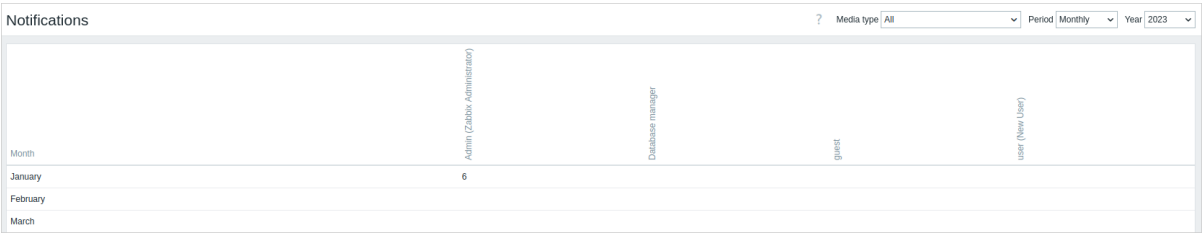
Il selettore del periodo di tempo accanto al filtro consente di selezionare con un clic del mouse i periodi utilizzati più frequentemente. Per ulteriori informazioni, vedere Selettori del periodo di tempo e dell'host.

Notifiche

Panoramica

Nella sezione *Reports* > *Notifications* viene visualizzato un report sul numero di notifiche inviate a ciascun utente.

Dai menu a discesa nell'angolo in alto a destra è possibile scegliere il tipo di media (oppure tutti), il periodo (dati per ogni giorno/settimana/mese/anno) e l'anno per le notifiche inviate.



Ogni colonna mostra i totali per un singolo utente di sistema.

Raccolta dati

Panoramica

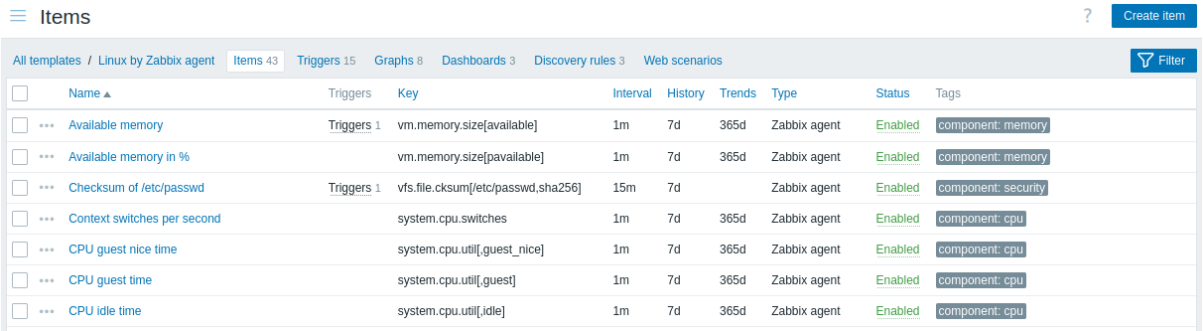
Questo menu include sezioni relative alla configurazione della raccolta dati.

Item

Panoramica

L'elenco degli item per un template è accessibile da *Raccolta dati* > *Template*, facendo clic su *Item* per il template desiderato.

Viene visualizzato un elenco degli item esistenti.



Dati visualizzati:

Colonna	Descrizione
Menu item	Fare clic sull'icona con tre puntini per aprire il menu item.

Colonna	Descrizione
Template	Template a cui appartiene l'item. Facendo clic sul nome del template si apre il modulo di configurazione del template. Questa colonna viene visualizzata solo se nel filtro sono selezionati più template o nessun template.
Nome	Nome dell'item visualizzato come collegamento blu ai dettagli dell'item. Facendo clic sul collegamento con il nome dell'item si apre il modulo di configurazione dell'item. Se l'item è ereditato da un altro template, il nome del template viene visualizzato prima del nome dell'item come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco degli item a quel livello del template.
Trigger	Passando il mouse su Trigger viene visualizzato un riquadro informativo con i trigger associati all'item. Il numero di trigger viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item.
Intervallo	Viene visualizzata la frequenza del controllo.
Cronologia	Viene visualizzato per quanti giorni verrà conservata la cronologia dei dati dell'item.
Trend	Viene visualizzato per quanti giorni verrà conservata la cronologia dei trend dell'item.
Tipo	Viene visualizzato il tipo di item (Zabbix agent, SNMP agent, controllo semplice, ecc.).
Stato	Viene visualizzato lo stato dell'item: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo, da Abilitato a Disabilitato e viceversa.
Tag	Vengono visualizzati i tag dell'item. È possibile visualizzare fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di visualizzare tutti i tag passando il mouse su di esso.

Per configurare un nuovo item, fare clic sul pulsante *Crea item* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

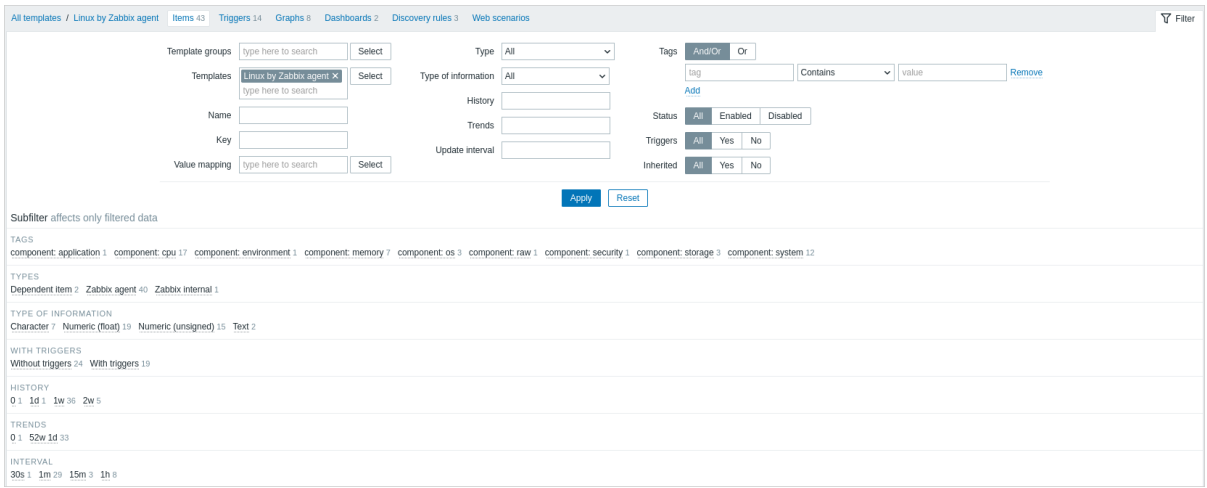
- *Enable* - cambia lo stato dell'item in *Enabled*.
- *Disable* - cambia lo stato dell'item in *Disabled*.
- *Copy* - copia gli item su altri host o template.
- *Mass update* - aggiorna diverse proprietà di più item contemporaneamente.
- *Delete* - elimina gli item.

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi item, quindi fai clic sul pulsante desiderato.

Uso del filtro

L'elenco degli item può contenere molti item. Utilizzando il filtro, è possibile filtrare alcuni di essi per individuare rapidamente gli item che si stanno cercando. Per prestazioni di ricerca migliori, i dati vengono cercati con i macro non risolti.

L'icona *Filter* è disponibile nell'angolo in alto a destra. Facendovi clic si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.

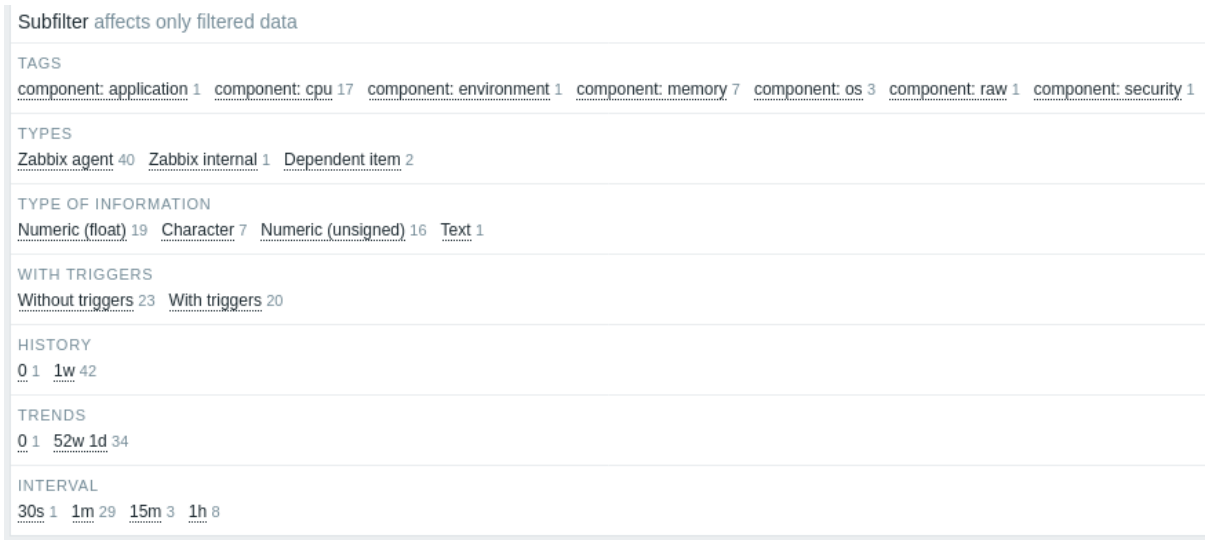


Parameter	Description
Template groups	Filtra per uno o più gruppi di template. Specificando un gruppo di template padre, vengono implicitamente selezionati tutti i gruppi nidificati.
Templates	Filtra per uno o più template.
Name	Filtra per nome dell'item.
Key	Filtra per chiave dell'item.
Value mapping	Filtra in base alla mappa dei valori utilizzata. Questo parametro non viene visualizzato se l'opzione *Templates* è vuota.
Type	Filtra per tipo di item (Zabbix agent, SNMP agent, ecc.).
Type of information	Filtra per tipo di informazione (Numeric unsigned, float, ecc.).
History	Filtra in base al tempo per cui viene conservata la cronologia dell'item.
Trends	Filtra in base al tempo per cui vengono conservate le tendenze dell'item.
Update interval	Filtra in base all'intervallo di aggiornamento dell'item.
Tags	Specifica i tag per limitare il numero di item visualizzati. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole) **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole) **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or **Or** - è sufficiente che una condizione sia soddisfatta
Status	Filtra per stato dell'item - *Enabled* o *Disabled*.
Triggers	Filtra gli item con (o senza) trigger.
Inherited	Filtra gli item ereditati (o non ereditati) dai template collegati.

Utilizzo del subfiltro

Il subfiltro consente di modificare ulteriormente il filtraggio applicato dal filtro principale.

Contiene collegamenti cliccabili per accedere rapidamente agli item correlati. Gli item sono correlati da un'entità comune, ad esempio tag, tipo di item, intervallo di aggiornamento dell'item e così via. Quando si fa clic su un'entità, questa viene evidenziata con uno sfondo grigio e gli item vengono filtrati immediatamente (non è necessario fare clic su *Applica* nel filtro principale). Facendo clic su un'altra entità, questa viene aggiunta ai risultati filtrati. Facendo nuovamente clic sull'entità, il filtro viene rimosso.



I subfiltri vengono generati in base ai dati filtrati, limitati a 1000 record. Se si desidera visualizzare più record nel subfiltro, è necessario aumentare il valore del parametro *Limite per i risultati della ricerca e del filtro* (in *Amministrazione* > *Generale* > *GUI*).

A differenza del filtro principale, il subfiltro viene aggiornato a ogni richiesta di aggiornamento della tabella, in modo da mostrare sempre informazioni aggiornate sulle opzioni di filtraggio disponibili e sui relativi conteggi.

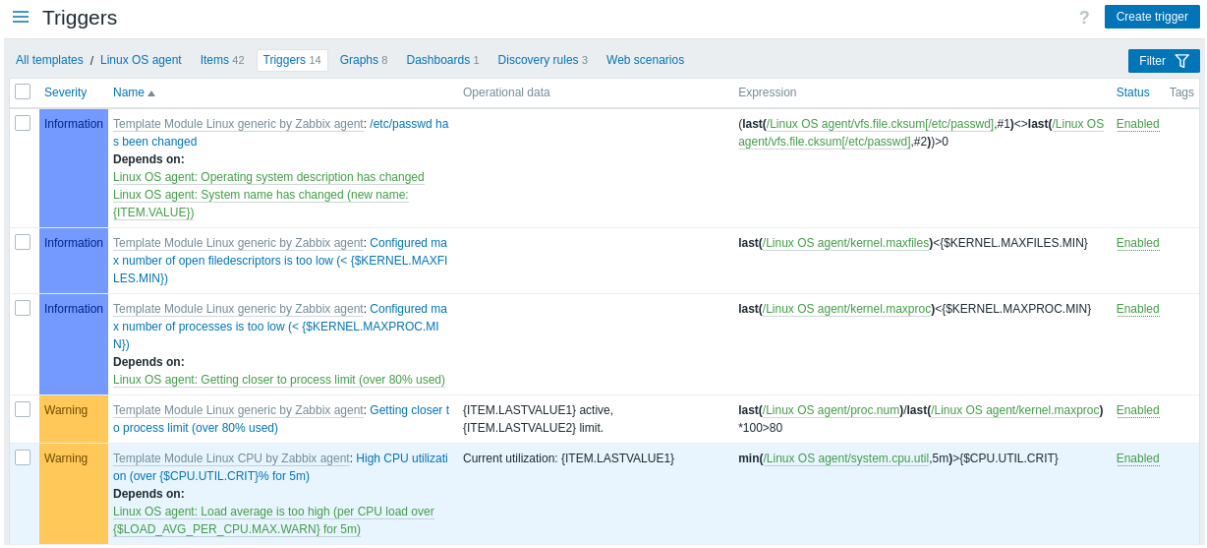
Il numero di entità visualizzate è limitato a 100 orizzontalmente. Se sono presenti più entità, alla fine viene visualizzata un'icona con tre punti, che non è cliccabile.

Il numero accanto a ogni entità cliccabile indica il numero di item raggruppati al suo interno (in base ai risultati del filtro principale). Quando si fa clic su un'entità, accanto alle altre entità disponibili vengono visualizzati dei numeri preceduti da un segno più, che indicano quanti item possono essere aggiunti alla selezione corrente. Le entità senza item non vengono visualizzate, a meno che non siano già state selezionate nel subfiltro.

Trigger

Panoramica

L'elenco dei trigger per un template è accessibile da *Raccolta dati* > *Template*, facendo clic su *Trigger* per il template desiderato.



Dati visualizzati:

Colonna	Descrizione
Gravità	La gravità del trigger viene visualizzata sia tramite il nome sia tramite il colore di sfondo della cella.
Template	Template a cui appartiene il trigger. Facendo clic sul nome del template si apre il modulo di configurazione del template. Questa colonna viene visualizzata solo se nel filtro sono selezionati più template o nessun template.
Nome	Nome del trigger visualizzato come collegamento blu ai dettagli del trigger. Facendo clic sul collegamento del nome del trigger si apre il modulo di configurazione del trigger. Se il trigger è ereditato da un altro template, il nome del template viene visualizzato prima del nome del trigger come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei trigger a quel livello del template.
Dati operativi	Definizione dei dati operativi del trigger, contenente stringhe arbitrarie e macro che verranno risolte dinamicamente in *Monitoraggio* > *Problemi*.
Espressione	Viene visualizzata l'espressione del trigger. La parte template-item dell'espressione viene visualizzata come collegamento che conduce al modulo di configurazione dell'item.
Stato	Viene visualizzato lo stato del trigger: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo, da Abilitato a Disabilitato e viceversa.
Tag	Se un trigger contiene tag, in questa colonna vengono visualizzati il nome e il valore del tag.

Per configurare un nuovo trigger, fare clic sul pulsante *Crea trigger* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Enable* - cambia lo stato del trigger in *Enabled*

- *Disable* - cambia lo stato del trigger in *Disabled*
- *Copy* - copia i trigger su altri host o template
- *Mass update* - aggiorna più proprietà per un numero di trigger contemporaneamente
- *Delete* - elimina i trigger

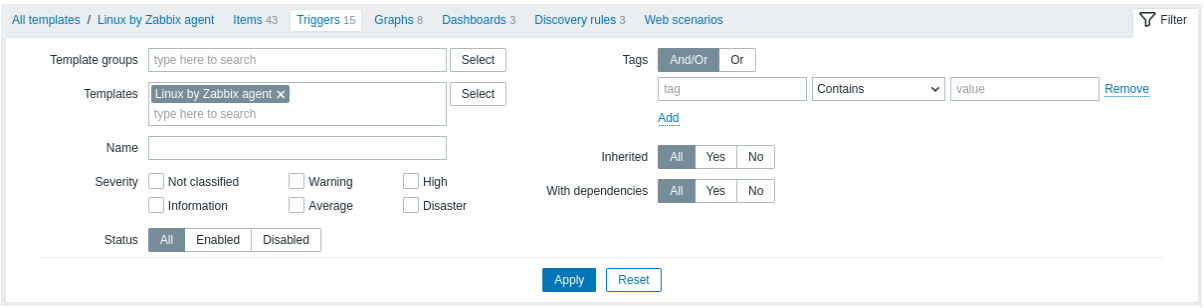
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi trigger, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo i trigger di interesse.

Per migliorare le prestazioni della ricerca, i dati vengono cercati con i macro non risolti.

L'icona *Filter* è disponibile nell'angolo in alto a destra. Facendo clic su di essa si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.



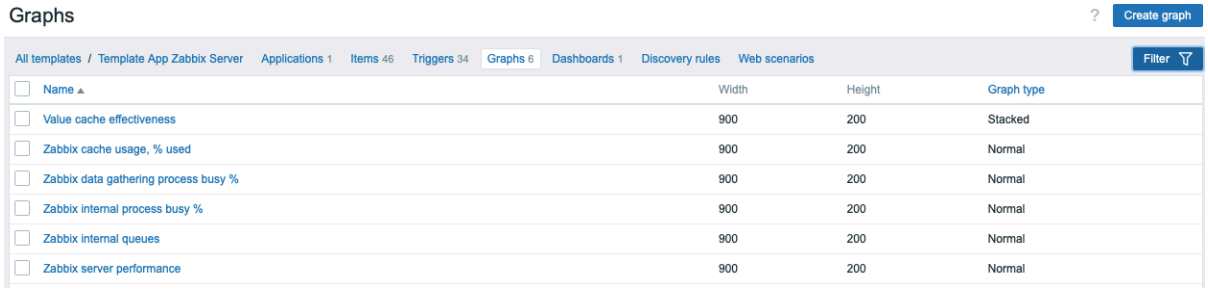
Parameter	Description
Template groups	Filtra per uno o più gruppi di template. La specifica di un gruppo di template padre seleziona implicitamente tutti i gruppi nidificati.
Templates	Filtra per uno o più template. Se i gruppi di template sono già selezionati sopra, la selezione dei template è limitata a tali gruppi.
Name	Filtra per nome del trigger.
Severity	Seleziona per filtrare per una o più severità del trigger.
Status	Filtra per lo stato di attivazione del trigger (Enabled/Disabled).
Tags	Filtra per nome e valore del tag del trigger. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole) **Contains** - include i nomi dei tag specificati in cui i valori dei tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole) **Does not contain** - esclude i nomi dei tag specificati in cui i valori dei tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or **Or** - è sufficiente che una condizione sia soddisfatta Nei campi nome del tag e valore del tag sono supportati macro e macro functions.
Inherited	Filtra i trigger ereditati (o non ereditati) dai template collegati.
With dependencies	Filtra i trigger con (o senza) dipendenze.

Grafici

Panoramica

L'elenco dei grafici personalizzati per un template è accessibile da *Raccolta dati* > *Template*, facendo clic su *Grafici* per il template desiderato.

Viene visualizzato un elenco dei grafici esistenti.



Dati visualizzati:

Colonna	Descrizione
Template	Template a cui appartiene il grafico. Facendo clic sul nome del template si apre il modulo di configurazione del template. Questa colonna viene visualizzata solo se nel filtro sono selezionati più template o nessun template.
Nome	Nome del grafico personalizzato, visualizzato come collegamento blu ai dettagli del grafico. Facendo clic sul collegamento con il nome del grafico si apre il modulo di configurazione del grafico. Se il grafico è ereditato da un altro template, il nome del template viene visualizzato prima del nome del grafico come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei grafici a quel livello del template.
Larghezza	Viene visualizzata la larghezza del grafico.
Altezza	Viene visualizzata l'altezza del grafico.
Tipo di grafico	Viene visualizzato il tipo di grafico: *Normale*, *Impilato*, *A torta* o *Esploso*.

Per configurare un nuovo grafico, fare clic sul pulsante *Crea grafico* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Copia* - copia i graph su altri host o template
- *Elimina* - elimina i graph

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi graph, quindi fai clic sul pulsante desiderato.

Uso del filtro

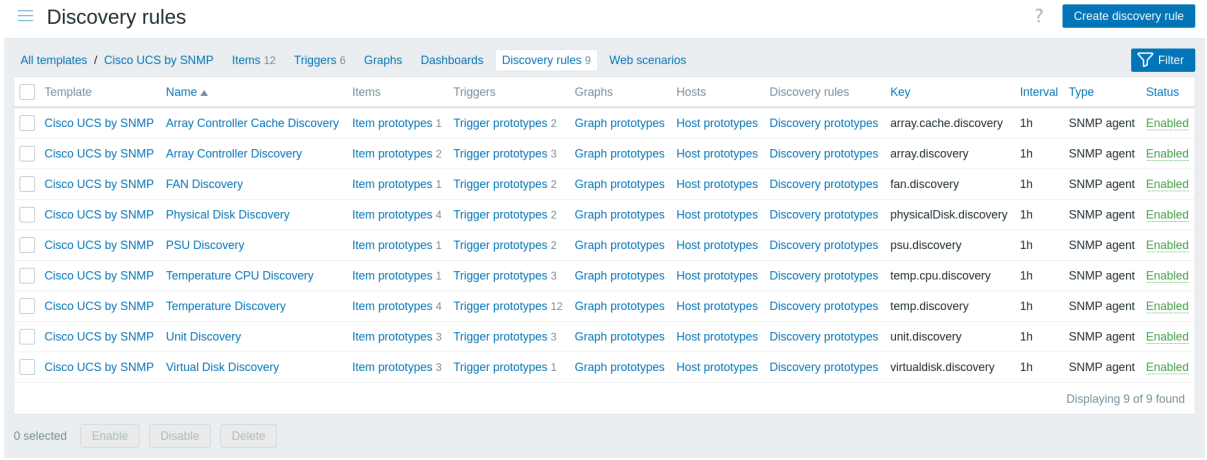
È possibile filtrare i grafici per gruppo di template e template. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Regole di discovery

Panoramica

L'elenco delle regole di discovery di basso livello per un template è accessibile da *Raccolta dati* > *Template*, facendo clic su *Discovery* per il template corrispondente.

Viene visualizzato un elenco delle regole di discovery di basso livello esistenti. È inoltre possibile visualizzare tutte le regole di discovery indipendentemente dal template oppure tutte le regole di discovery di uno specifico gruppo di template modificando le impostazioni del filtro.



Dati visualizzati:

Colonna	Descrizione
Template	Il template a cui appartiene la regola di discovery. Facendo clic sul nome del template si apre il modulo di configurazione del template.
Nome	Nome della regola, visualizzato come collegamento blu. Facendo clic sul nome della regola si apre il modulo di configurazione della regola di discovery di basso livello. Se la regola di discovery è ereditata da un altro template, il nome del template viene visualizzato prima del nome della regola come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco delle regole di discovery a quel livello del template.
Item	Viene visualizzato un collegamento all'elenco dei prototipi di item. Il numero di prototipi di item esistenti viene visualizzato in grigio.
Trigger	Viene visualizzato un collegamento all'elenco dei prototipi di trigger. Il numero di prototipi di trigger esistenti viene visualizzato in grigio.
Grafici	Viene visualizzato un collegamento all'elenco dei prototipi di grafico. Il numero di prototipi di grafico esistenti viene visualizzato in grigio.
Host	Viene visualizzato un collegamento all'elenco dei prototipi di host. Il numero di prototipi di host esistenti viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item utilizzata per la discovery.
Intervallo	Viene visualizzata la frequenza di esecuzione della discovery.
Tipo	Viene visualizzato il tipo di item utilizzato per la discovery (Zabbix agent, SNMP agent, ecc.).
Stato	Viene visualizzato lo stato della regola di discovery: *Abilitata* o *Disabilitata*. Facendo clic sullo stato è possibile modificarlo, da Abilitata a Disabilitata e viceversa.

Per configurare una nuova regola di discovery di basso livello, fare clic sul pulsante *Crea regola di discovery* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

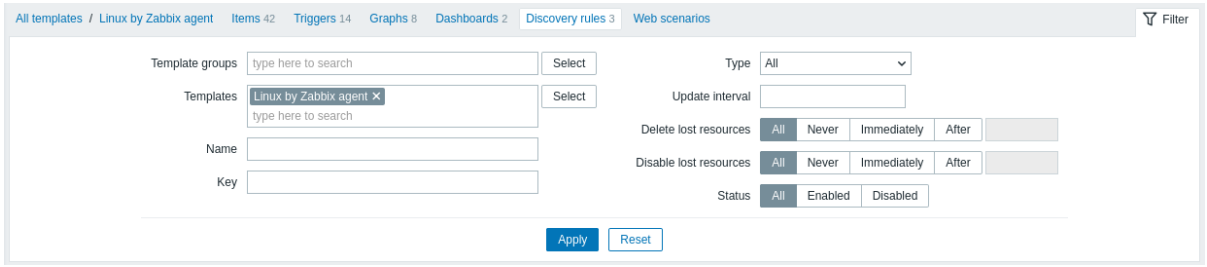
- *Enable* - cambia lo stato della regola di discovery a basso livello in *Enabled*
- *Disable* - cambia lo stato della regola di discovery a basso livello in *Disabled*
- *Delete* - elimina le regole di discovery a basso livello

Per usare queste opzioni, seleziona le caselle di controllo davanti alle rispettive regole di discovery, quindi fai clic sul pulsante desiderato.

Uso del filtro

È possibile usare il filtro per visualizzare solo le regole di discovery di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

L'icona *Filtro* è disponibile nell'angolo in alto a destra. Facendo clic su di essa si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati, come template, nome della regola di discovery, chiave dell'item, tipo di item, ecc.



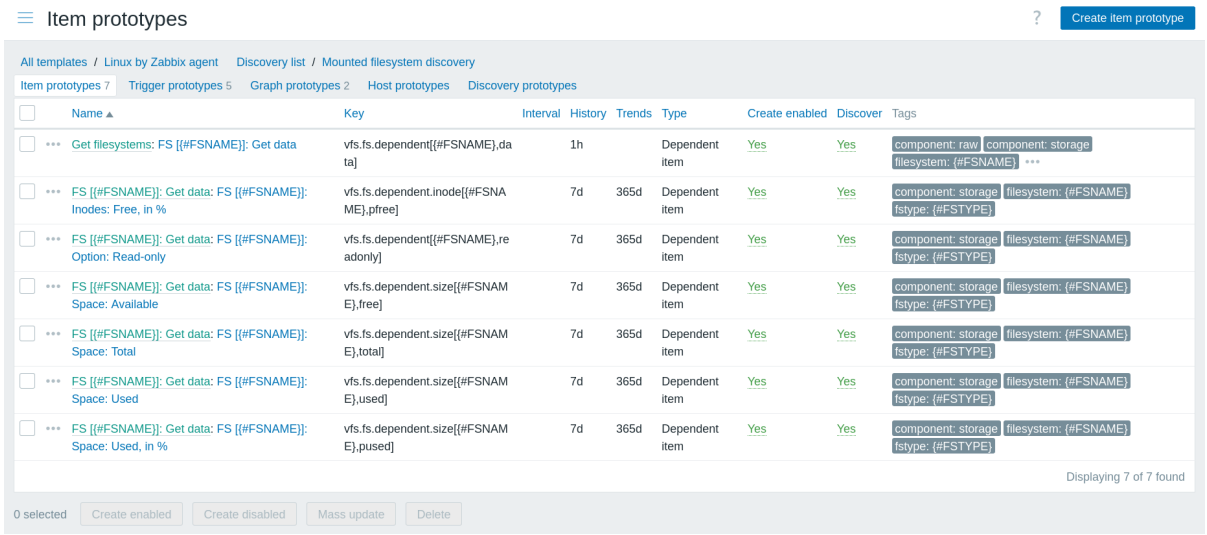
Parameter	Description
Template groups	Filtra per uno o più gruppi di template. La specifica di un gruppo di template padre seleziona implicitamente tutti i gruppi nidificati.
Templates	Filtra per uno o più template.
Name	Filtra per nome della regola di discovery.
Key	Filtra per chiave dell'item di discovery.
Type	Filtra per tipo dell'item di discovery.
Update interval	Filtra per intervallo di aggiornamento. Non disponibile per Zabbix trapper e item dipendenti.
Delete lost resources	Filtra per il periodo *Delete lost resources*.
Disable lost resources	Filtra per il periodo *Disable lost resources*.
Status	Filtra per lo stato della regola di discovery (All/Enabled/Disabled).

Prototipi di item

Panoramica

In questa sezione vengono visualizzati i prototipi di item configurati di una regola di low-level discovery nel template.

Se il template è collegato all'host, i prototipi di item costituiranno la base per la creazione di item reali sull'host durante la low-level discovery.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di item, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di item. Se il prototipo di item appartiene a un template collegato, il nome del template viene visualizzato prima del nome del prototipo di item come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di item a livello del template collegato. Se il prototipo di item è di tipo "Dependent item", il nome dell'item master viene visualizzato prima del nome del prototipo di item in verde (come nel caso dei prototipi di item a livello dell'host).
Key	Viene visualizzata la chiave del prototipo di item.

Column	Description
Interval	Viene visualizzata la frequenza del controllo.
History	Viene visualizzato per quanti giorni conservare lo storico dei dati dell'item.
Trends	Viene visualizzato per quanti giorni conservare lo storico dei trend dell'item.
Type	Viene visualizzato il tipo di prototipo di item (Zabbix agent, SNMP agent, controllo semplice, ecc.).
Create enabled	Crea l'item in base a questo prototipo come: **Yes** - abilitato **No** - disabilitato. È possibile passare da "Yes" a "No" e viceversa facendo clic su di essi.
Discover	Rileva l'item in base a questo prototipo: **Yes** - rileva **No** - non rilevare. È possibile passare da "Yes" a "No" e viceversa facendo clic su di essi.
Tags	Vengono visualizzati i tag del prototipo di item.

Per configurare un nuovo prototipo di item, fare clic sul pulsante *Create item prototype* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Crea abilitati* - crea questi item come *Abilitati*
- *Crea disabilitati* - crea questi item come *Disabilitati*
- *Aggiornamento massivo* - aggiorna in massa questi prototipi di item
- *Elimina* - elimina questi prototipi di item

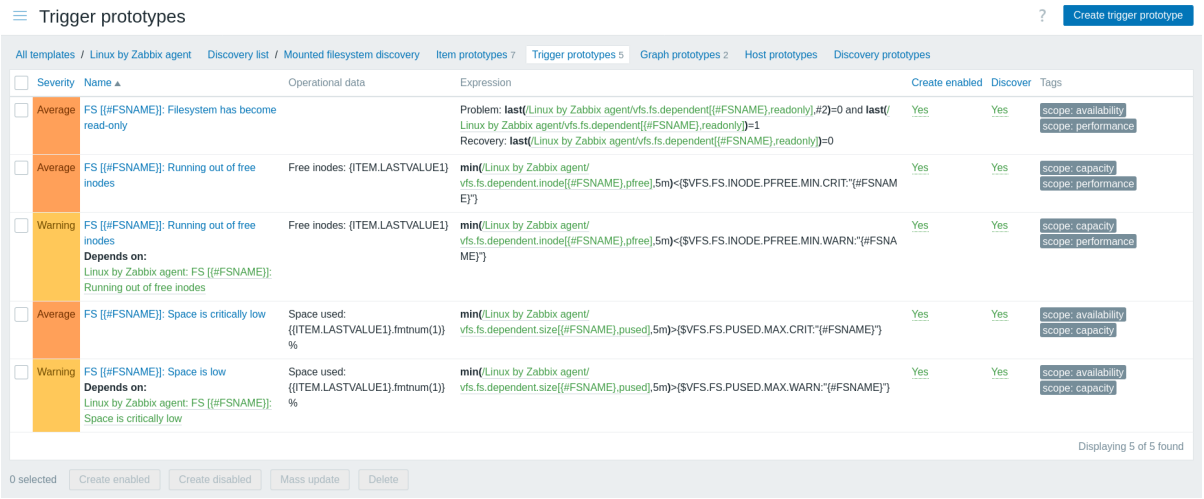
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di item, quindi fai clic sul pulsante richiesto.

Prototipi di trigger

Panoramica

In questa sezione vengono visualizzati i prototipi di trigger configurati di una regola di discovery di basso livello sul template.

Se il template è collegato all'host, i prototipi di trigger costituiranno la base per la creazione dei trigger reali dell'host durante la discovery di basso livello.



Dati visualizzati:

Column	Description
Severity	La gravità del trigger viene visualizzata sia tramite il nome sia tramite il colore di sfondo della cella.
Name	Nome del prototipo di trigger, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di trigger. Se il prototipo di trigger appartiene a un template collegato, il nome del template viene visualizzato prima del nome del trigger come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di trigger a livello del template collegato.
Expression	Viene visualizzata l'espressione del trigger. La parte dell'espressione relativa all'item del template viene visualizzata come collegamento al modulo di configurazione dell'item.

Column	Description
Operational data	Viene visualizzato il formato dei dati operativi del trigger, contenente stringhe arbitrarie e macro che verranno risolte dinamicamente in *Monitoring* > *Problems*.
Create enabled	Crea il trigger in base a questo prototipo come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' e viceversa facendo clic sul valore.
Discover	Esegue la discovery del trigger in base a questo prototipo: **Yes** - esegui la discovery **No** - non eseguire la discovery. È possibile passare da 'Yes' a 'No' e viceversa facendo clic sul valore.
Tags	Vengono visualizzati i tag del prototipo di trigger.

Per configurare un nuovo prototipo di trigger, fare clic sul pulsante *Create trigger prototype* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Crea abilitati* - crea questi trigger come *Abilitati*
- *Crea disabilitati* - crea questi trigger come *Disabilitati*
- *Aggiornamento massivo* - aggiorna in massa questi prototipi di trigger
- *Elimina* - elimina questi prototipi di trigger

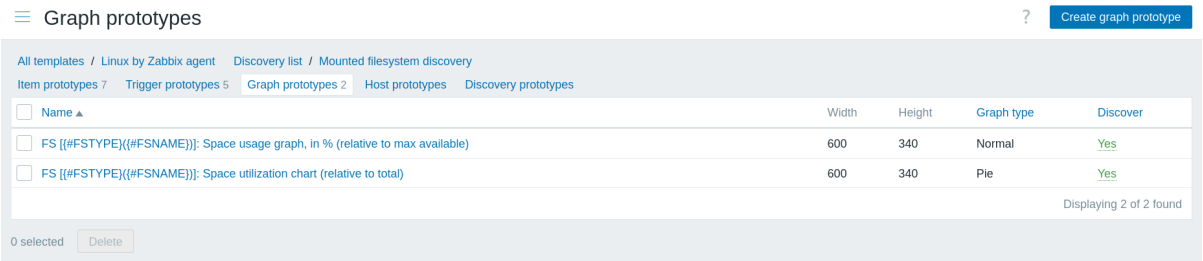
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di trigger, quindi fai clic sul pulsante richiesto.

Prototipi di grafico

Panoramica

In questa sezione vengono visualizzati i prototipi di grafico configurati di una regola di discovery di basso livello nel template.

Se il template è collegato all'host, i prototipi di grafico costituiranno la base per la creazione dei grafici reali dell'host durante la discovery di basso livello.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del prototipo di grafico, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di grafico. Se il prototipo di grafico appartiene a un template collegato, il nome del template viene visualizzato prima del nome del grafico come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di grafico a livello del template collegato.
Larghezza	Viene visualizzata la larghezza del prototipo di grafico.
Altezza	Viene visualizzata l'altezza del prototipo di grafico.
Tipo	Viene visualizzato il tipo di prototipo di grafico: *Normale*, *Impilato*, *Torta* o *Esploso*.
Discovery	Esegue la discovery del grafico in base a questo prototipo: **Sì** - esegue la discovery **No** - non esegue la discovery. È possibile alternare tra "Sì" e "No" facendo clic su di essi.

Per configurare un nuovo prototipo di grafico, fare clic sul pulsante *Crea prototipo di grafico* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Elimina* - elimina questi prototipi di grafico

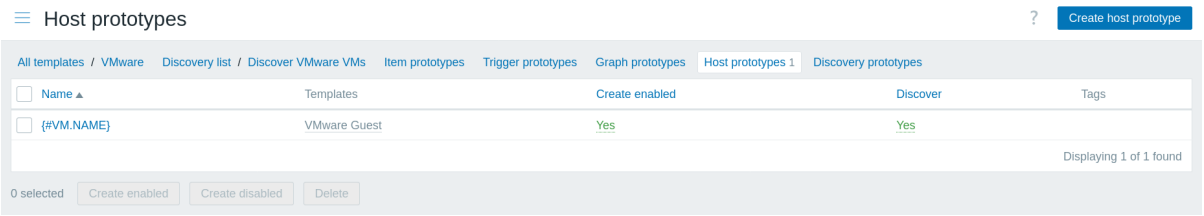
Per usare queste opzioni, seleziona le caselle di controllo accanto ai rispettivi prototipi di grafico, quindi fai clic sul pulsante desiderato.

Prototipi host

Panoramica

In questa sezione vengono visualizzati i prototipi di host configurati di una regola di discovery di basso livello sul template.

Se il template è collegato all'host, i prototipi di host costituiranno la base per la creazione di host reali durante la discovery di basso livello.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del prototipo di host, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di host. Se il prototipo di host appartiene a un template collegato, il nome del template viene visualizzato prima del nome dell'host come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di host a livello del template collegato.
Template	Vengono visualizzati i template del prototipo di host.
Creazione abilitata	Crea l'host in base a questo prototipo come: **Sì** - abilitata **No** - disabilitata. È possibile passare da "Sì" a "No" e viceversa facendo clic su di essi.
Discovery	Esegue la discovery dell'host in base a questo prototipo: **Sì** - esegui la discovery **No** - non eseguire la discovery. È possibile passare da "Sì" a "No" e viceversa facendo clic su di essi.
Tag	Vengono visualizzati i tag del prototipo di host.

Per configurare un nuovo prototipo di host, fare clic sul pulsante *Crea prototipo di host* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Crea abilitati* - crea questi host come *Abilitati*
- *Crea disabilitati* - crea questi host come *Disabilitati*
- *Elimina* - elimina questi prototipi di host

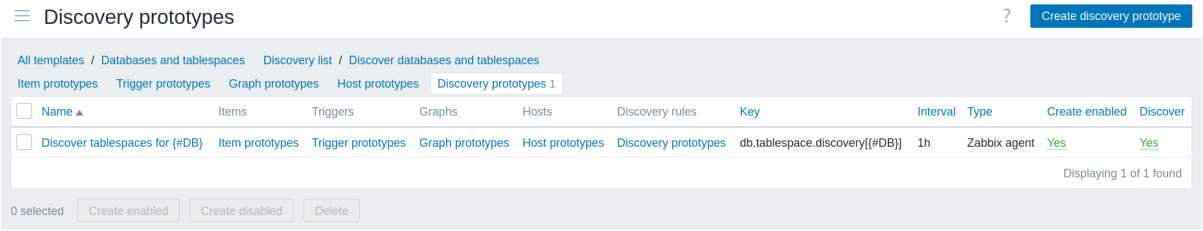
Per usare queste opzioni, seleziona le caselle di controllo accanto ai rispettivi prototipi di host, quindi fai clic sul pulsante desiderato.

Prototipi di discovery

Panoramica

In questa sezione vengono visualizzati i prototipi di discovery configurati di una regola di discovery di basso livello nel template.

Se il template è collegato all'host, i prototipi di discovery costituiranno la base per la creazione dei veri prototipi di discovery della regola di discovery principale.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di discovery, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di discovery. Se il prototipo di discovery appartiene a un template collegato, il nome del template viene visualizzato prima del nome del prototipo di discovery come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di discovery a livello del template collegato.
Items	Viene visualizzato un collegamento all'elenco dei prototipi di item. Il numero di prototipi di item esistenti viene visualizzato in grigio.
Triggers	Viene visualizzato un collegamento all'elenco dei prototipi di trigger. Il numero di prototipi di trigger esistenti viene visualizzato in grigio.
Graphs	Viene visualizzato un collegamento all'elenco dei prototipi di grafici. Il numero di prototipi di grafici esistenti viene visualizzato in grigio.
Hosts	Viene visualizzato un collegamento all'elenco dei prototipi di host. Il numero di prototipi di host esistenti viene visualizzato in grigio.
Discovery rules	Viene visualizzato un collegamento all'elenco dei prototipi di discovery. Il numero di prototipi di discovery esistenti viene visualizzato in grigio.
Key	Viene visualizzata la chiave dell'item utilizzata per la discovery.
Interval	Viene visualizzata la frequenza di esecuzione della discovery.
Type	Viene visualizzato il tipo di item utilizzato per la discovery (Zabbix agent, SNMP agent, ecc.).
Create enabled	Viene visualizzato lo stato di creazione del prototipo di discovery: creazione abilitata (*Yes*) o creazione disabilitata (*No*). Facendo clic sullo stato è possibile modificarlo, da Yes a No e viceversa.
Discover	Viene visualizzato lo stato di discovery del prototipo di discovery: esegui discovery (*Yes*) o non eseguire discovery (*No*). Facendo clic sullo stato è possibile modificarlo, da Yes a No e viceversa.

Per configurare un nuovo prototipo di discovery, fare clic sul pulsante *Create discovery prototype* nell'angolo superiore destro.

Opzioni di modifica in massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica in massa:

- *Crea abilitati* - crea questi prototipi di discovery come *Abilitati*
- *Crea disabilitati* - crea questi prototipi di discovery come *Disabilitati*
- *Elimina* - elimina questi prototipi di discovery

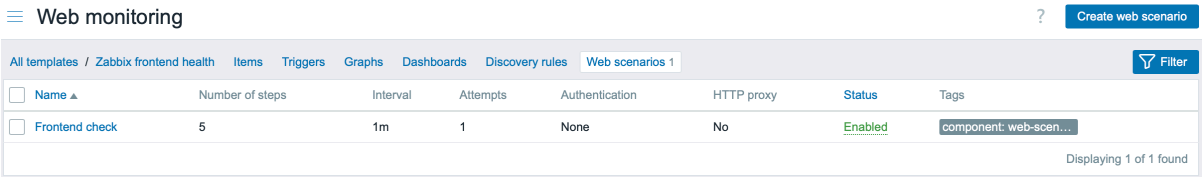
Per usare queste opzioni, seleziona le caselle di controllo accanto ai rispettivi prototipi di discovery, quindi fai clic sul pulsante desiderato.

Scenari web

Panoramica

L'elenco degli scenari web per un template è accessibile da *Raccolta dati* > *Template*, facendo clic su *Web* per il template desiderato.

Viene visualizzato un elenco degli scenari web esistenti.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome dello scenario web. Facendo clic sul nome dello scenario web si apre il modulo di configurazione dello scenario web. Se lo scenario web è ereditato da un altro template, il nome del template viene visualizzato prima del nome dello scenario web come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco degli scenari web a quel livello del template.
Numero di passaggi	Numero di passaggi contenuti nello scenario.
Intervallo di aggiornamento	Frequenza di esecuzione dello scenario.
Tentativi	Numero di tentativi effettuati per eseguire i passaggi dello scenario web.
Autenticazione	Viene visualizzato il metodo di autenticazione: Basic, NTLM, Kerberos, Digest o None.
Proxy HTTP	Visualizza il proxy HTTP oppure "No" se non viene utilizzato.
Stato	Viene visualizzato lo stato dello scenario web: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.
Tag	Vengono visualizzati i tag dello scenario web. È possibile visualizzare fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di visualizzare tutti i tag al passaggio del mouse.

Per configurare un nuovo scenario web, fare clic sul pulsante *Crea scenario web* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Enable* - cambia lo stato dello scenario in *Enabled*
- *Disable* - cambia lo stato dello scenario in *Disabled*
- *Delete* - elimina gli scenari web

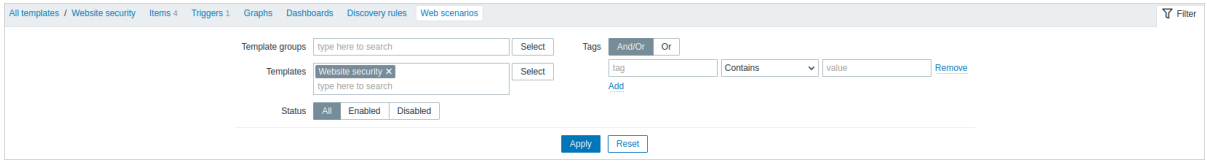
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi scenari web, quindi fai clic sul pulsante desiderato.

Uso del filtro

È possibile usare il filtro per visualizzare solo gli scenari che interessano.

Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Il link *Filter* è disponibile sopra l'elenco degli scenari web. Se si fa clic, viene visualizzato un filtro che consente di filtrare gli scenari per gruppo di template, template, stato e tag.

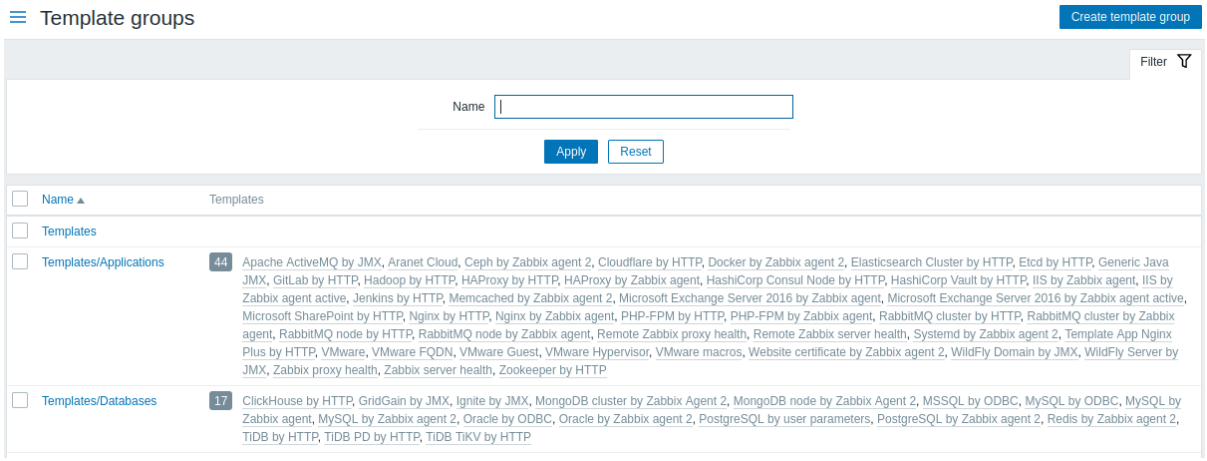


Gruppi di template

Panoramica

Nella sezione *Raccolta dati* > *Gruppi di template*, gli utenti possono configurare e gestire i gruppi di template.

Viene visualizzato un elenco dei gruppi di template esistenti con i relativi dettagli. È possibile cercare e filtrare i gruppi di template in base al nome.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del gruppo di template. Facendo clic sul nome del gruppo si apre il modulo di configurazione del gruppo di template.
Template	Numero di template nel gruppo (visualizzato in grigio), seguito dall'elenco dei membri del gruppo. Facendo clic sul nome di un template si apre il modulo di configurazione del template. Facendo clic sul numero si apre l'elenco dei template presenti in questo gruppo.

Opzioni di modifica in massa

Per eliminare più gruppi di template contemporaneamente, seleziona le caselle di controllo accanto ai rispettivi gruppi, quindi fai clic sul pulsante Delete sotto l'elenco.

Utilizzo del filtro

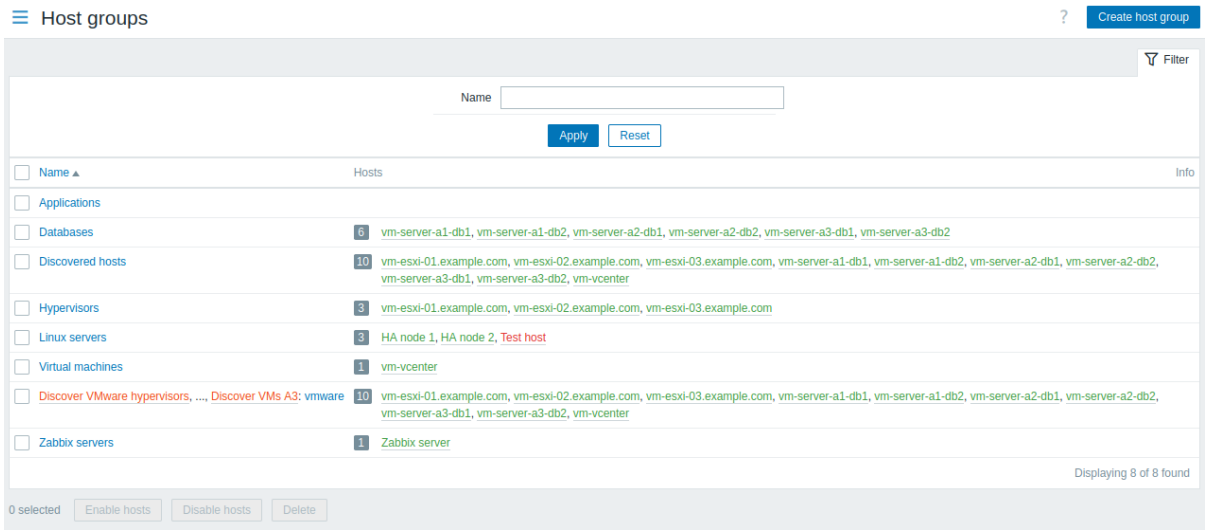
È possibile utilizzare il filtro per visualizzare solo i gruppi di template di proprio interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Gruppi di host

Panoramica

Nella sezione *Raccolta dati* > *Gruppi di host*, gli utenti possono configurare e gestire i gruppi di host.

Viene visualizzato un elenco dei gruppi di host esistenti con i relativi dettagli. È possibile cercare e filtrare i gruppi di host per nome.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del gruppo di host. Facendo clic sul nome del gruppo si apre il modulo di configurazione del gruppo di host. I gruppi di host rilevati vengono visualizzati con i nomi delle regole di rilevamento di basso livello come prefisso. Facendo clic sul nome della regola LLD si apre il modulo di configurazione del prototipo di host. I gruppi di host rilevati vengono eliminati quando vengono eliminate tutte le regole LLD che li hanno rilevati.
Host	Numero di host nel gruppo (visualizzato in grigio), seguito dall'elenco dei membri del gruppo. Facendo clic sul nome di un host si apre il modulo di configurazione dell'host. Facendo clic sul numero, nell'elenco completo degli host vengono filtrati quelli appartenenti al gruppo.
Informazioni	Vengono visualizzate le informazioni sugli errori, se presenti, relative al gruppo di host.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Abilita host* - cambia lo stato di tutti gli host nel gruppo in "Monitored"
- *Disabilita host* - cambia lo stato di tutti gli host nel gruppo in "Not monitored"
- *Elimina* - elimina i gruppi di host

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi gruppi di host, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

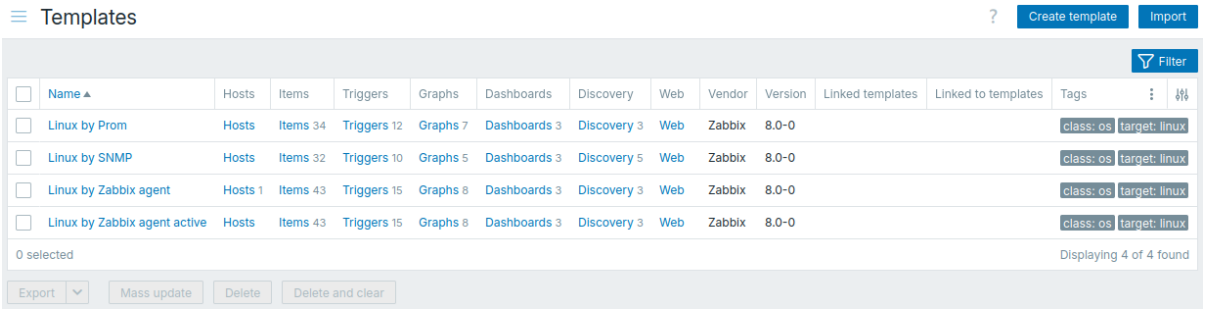
Puoi utilizzare il filtro per visualizzare solo i gruppi host che sei interessati. Per migliorare le prestazioni della ricerca, i dati vengono cercati con macro irrisolte.

Template

Panoramica

Nella sezione *Raccolta dati* > *Template*, gli utenti possono configurare e gestire i template.

Viene visualizzato un elenco dei template esistenti con i relativi dettagli.



I dettagli dei template vengono visualizzati in una tabella personalizzabile. Il pulsante *Personalizza tabella* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinandone il bordo destro; fare doppio clic sul bordo per ripristinare le dimensioni. Tutte le personalizzazioni vengono applicate immediatamente e vengono sempre ricordate per l'utente corrente.

Sono disponibili le seguenti colonne:

Colonna	Descrizione
Nome	Nome del template. Facendo clic sul nome del template si apre il modulo di configurazione del template.
Host	Numero di host modificabili a cui è collegato il template; gli host di sola lettura non sono inclusi. Facendo clic su *Host* si apre l'elenco degli host, filtrato in modo da mostrare solo quelli collegati al template.
Entità (Item, Trigger, Grafici, Dashboard, Rilevamento, Web)	Numero delle rispettive entità nel template (visualizzato in grigio). Facendo clic sul nome dell'entità, nell'elenco completo di tale entità vengono filtrate quelle appartenenti al template.
Template collegati	Template collegati al template.
Collegato ai template	Template a cui il template è collegato.

Colonna	Descrizione
Vendor, *Versione*	Vendor e versione del template; visualizzati se la configurazione del template contiene tali informazioni e solo per template pronti all'uso, template importati o template modificati tramite la Template API. Per i template pronti all'uso, la versione viene visualizzata come segue: versione principale di Zabbix, delimitatore ("-"), numero di revisione (incrementato a ogni nuova versione del template e reimpostato a ogni versione principale di Zabbix). Ad esempio, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Tag	Vengono visualizzati i tag del template, con le macro non risolte. Passando il mouse sopra o facendo clic sull'icona con i tre puntini dopo i tag visualizzati, si apre una finestra pop-up con tutti i tag. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna è possibile impostare il numero di tag visualizzati (1–3), scegliere come visualizzare i nomi dei tag (completi, abbreviati o nessuno), impostare la priorità di visualizzazione dei tag o duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Valore del tag	Vengono visualizzati i valori dei tag del template, con le macro non risolte. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna è possibile rinominare la colonna, scegliere quali valori dei tag visualizzare o duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Testo personalizzato	Vengono visualizzati testo personalizzato, valori di macro o una combinazione di entrambi. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna è possibile rinominare la colonna, inserire il testo personalizzato o le macro oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate). Macro supportate: `{HOST.ID}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.DESCRIPTION}` e macro utente.

Per configurare un nuovo template, fare clic sul pulsante *Crea template* nell'angolo superiore destro.

Per importare un template da un file YAML, XML o JSON, fare clic sul pulsante *Importa* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Export* - esporta il template in un file YAML, XML o JSON;
- *Mass update* - aggiorna diverse proprietà per più template contemporaneamente;
- *Delete* - elimina il template lasciando le entità collegate (item, trigger, ecc.) sugli host;
- *Delete and clear* - elimina il template e le sue entità collegate dagli host.

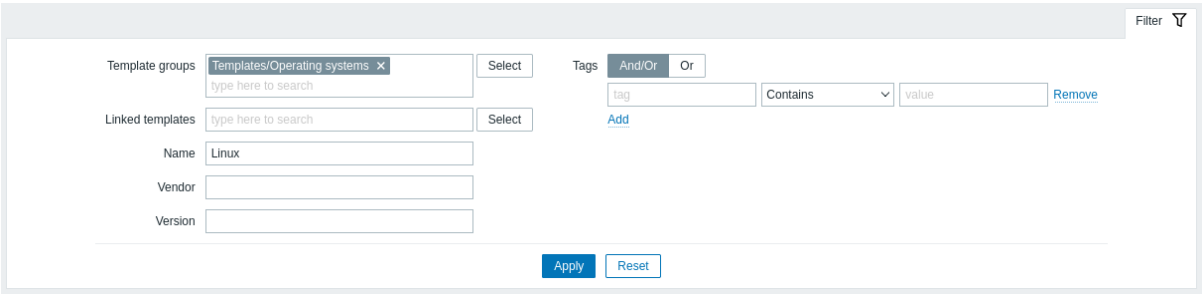
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi template, quindi fai clic sul pulsante richiesto.

Uso del filtro

È possibile usare il filtro per visualizzare solo i template di interesse.

Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Il link *Filter* è disponibile sotto i pulsanti *Create template* e *Import*. Se lo si fa clic, diventa disponibile un filtro con cui è possibile filtrare i template per gruppo di template, template collegati, nome e tag.



Parameter	Description
Template groups	Filtra per uno o più gruppi di template. Specificare un gruppo di template padre seleziona implicitamente tutti i gruppi nidificati.
Linked templates	Filtra per template collegati direttamente.
Name	Filtra per nome del template.
Vendor	Filtra per vendor del template.

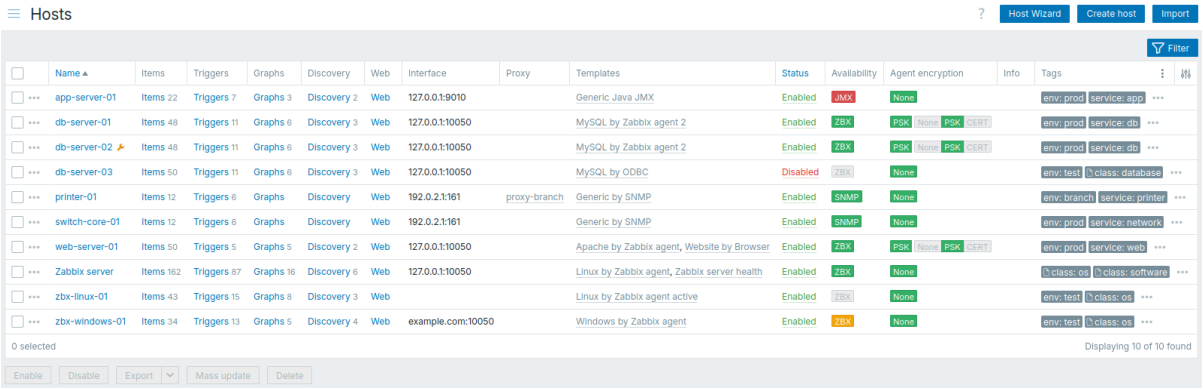
Parameter	Description
Version	Filtra per versione del template.
Tags	Filtra per nome e valore del tag del template. Il filtraggio è possibile solo per i tag a livello di template (non per quelli ereditati). È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole). Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or; **Or** - è sufficiente che una condizione sia soddisfatta.

Host

Panoramica

Nella sezione *Raccolta dati* > *Host*, gli utenti possono configurare e gestire gli host.

Viene visualizzato un elenco degli host esistenti con i relativi dettagli.



I dettagli degli host vengono visualizzati in una tabella personalizzabile. Il pulsante *Personalizza tabella* nell'intestazione della tabella consente di mostrare, nascondere o riordinare le colonne e di ripristinare il layout predefinito. È inoltre possibile ridimensionare ogni colonna trascinandone il bordo destro; fare doppio clic sul bordo per ripristinare le dimensioni. Tutte le personalizzazioni vengono applicate immediatamente e vengono sempre ricordate per l'utente corrente.

Sono disponibili le seguenti colonne:

Colonna	Descrizione
Nome	Nome dell'host. Facendo clic sul nome dell'host si apre il modulo di configurazione dell'host. Un'icona a forma di chiave inglese arancione dopo il nome dell'host indica che l'host è in manutenzione. I dettagli della manutenzione vengono visualizzati posizionando il puntatore del mouse sull'icona.
Entità (item, trigger, grafici, discovery, Web)	Facendo clic sul nome dell'entità vengono visualizzati gli item, i trigger e così via dell'host. Il numero delle rispettive entità viene visualizzato in grigio.
Interfaccia	Viene visualizzata l'interfaccia principale dell'host.

Colonna	Descrizione
Proxy	In questa colonna vengono visualizzati i proxy assegnati: <Nome del proxy> - l'host è monitorato da un proxy autonomo (anche se il proxy fa parte di un gruppo di proxy); <Nome del gruppo di proxy: nome del proxy> - l'host è monitorato da un gruppo di proxy e il server Zabbix ha assegnato un proxy per monitorare l'host; <Nome del gruppo di proxy> - l'host è monitorato da un gruppo di proxy senza proxy oppure il server Zabbix non ha assegnato un proxy per monitorare l'host; Niente - l'host non è monitorato né da un proxy né da un gruppo di proxy. Questa colonna viene visualizzata solo se l'opzione di filtro *Monitorato da* è impostata su 'Qualsiasi', 'Proxy' o 'Gruppo di proxy'.
Template	Vengono visualizzati i template collegati all'host. Se il template collegato contiene altri template, questi vengono visualizzati tra parentesi e separati da una virgola. Facendo clic sul nome di un template si apre il relativo modulo di configurazione.
Stato	Viene visualizzato lo stato dell'host: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo manualmente. Gli host rilevati che sono andati persi vengono contrassegnati con un'icona informativa. Il testo del suggerimento fornisce dettagli sul loro stato.
Disponibilità	Viene visualizzata la disponibilità dell'host per ogni interfaccia configurata. Le icone di disponibilità rappresentano lo stato corrente dell'interfaccia dell'host sul server Zabbix. Pertanto, se si disabilita un host nel frontend, la sua disponibilità verrà aggiornata dopo che il server Zabbix avrà sincronizzato le modifiche alla configurazione. Analogamente, se si abilita un host, la sua disponibilità verrà aggiornata dopo che il server Zabbix avrà sincronizzato le modifiche alla configurazione ed eseguito il polling dell'host. Le icone di disponibilità rappresentano solo i tipi di interfaccia (Agent, SNMP, IPMI, JMX) configurati. Passando il mouse sull'icona viene visualizzato un pop-up con l'elenco di tutte le interfacce dello stesso tipo, con dettagli, stato ed errori. Per l'interfaccia Agent, il pop-up visualizza le interfacce (passive) e i controlli attivi. Se un host dispone solo di controlli attivi, viene visualizzata l'icona dell'interfaccia Agent anche se per l'host non è configurata alcuna interfaccia Agent. La colonna è vuota per gli host senza interfacce. Lo stato di una singola interfaccia dell'host è determinato dalla connessione tra un item abilitato che utilizza l'interfaccia e l'host. Lo stato può essere: **Disponibile** - la connessione all'host è riuscita; **Non disponibile** - la connessione all'host non è riuscita (timeout, problemi del firewall e così via); **Sconosciuto** - non è stato tentato alcun collegamento all'host oppure il risultato è sconosciuto. Per ulteriori dettagli su come il server Zabbix determina lo stato dell'interfaccia, vedere Stato dell'interfaccia sconosciuto e Impostazioni dell'interfaccia dell'host non raggiungibile/non disponibile. Lo stato di tutte le interfacce dell'host di un singolo tipo (Agent, SNMP, IPMI, JMX) è determinato dalle interfacce utilizzate da almeno un item abilitato. Lo stato è indicato dal colore dell'icona: **Verde** - tutte le interfacce sono disponibili; **Giallo** - almeno un'interfaccia non è disponibile e almeno un'altra è disponibile o sconosciuta; **Rosso** - nessuna interfaccia è disponibile; **Grigio** - almeno un'interfaccia è sconosciuta, ma nessuna è non disponibile. **Disponibilità dei controlli attivi.** Se sull'host è abilitato almeno un controllo attivo, anche i controlli attivi influiscono sulla disponibilità totale dell'interfaccia Agent come descritto sopra. Per determinare la disponibilità dei controlli attivi, nel thread dei controlli attivi dell'agent vengono inviati messaggi heartbeat. La frequenza dei messaggi heartbeat è controllata dal parametro `HeartbeatFrequency` nella configurazione dell'agent o dell'agent 2 Zabbix (predefinita: 60 secondi, intervallo: 0-3600). I controlli attivi sono considerati non disponibili quando il heartbeat del controllo attivo è più vecchio di 2 x HeartbeatFrequency secondi. **Nota:** gli agent Zabbix precedenti alla versione 6.2.x non inviano heartbeat dei controlli attivi, pertanto la disponibilità dei relativi host rimane sconosciuta.

Colonna	Descrizione
Crittografia dell'agent	Viene visualizzato lo stato della crittografia per le connessioni verso l'host e dall'host: **Nessuna** - nessuna crittografia; **PSK** - viene utilizzata una chiave precondivisa; **Cert** - viene utilizzato un certificato.
Informazioni	Vengono visualizzate le informazioni sugli errori, se presenti, relative all'host.
Tag	Vengono visualizzati i tag dell'host, con le macro non risolte. Passando il mouse sull'icona con tre punti o facendo clic su di essa dopo i tag visualizzati, si apre un pop-up con tutti i tag. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna è possibile impostare il numero di tag visualizzati (1–3), scegliere come visualizzare i nomi dei tag (completi, abbreviati o nessuno), impostare la priorità di visualizzazione dei tag o duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Valore del tag	Vengono visualizzati i valori dei tag dell'host, con le macro non risolte. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna è possibile rinominare la colonna, scegliere quali valori dei tag visualizzare o duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate).
Testo personalizzato	Vengono visualizzati testo personalizzato, valori di macro o una combinazione di entrambi. Facendo clic sul pulsante *Opzioni* nell'intestazione della colonna è possibile rinominare la colonna, inserire il testo personalizzato o le macro oppure duplicare la colonna (le colonne duplicate possono essere rinominate o eliminate). Macro supportate: `{HOST.*}` (ad eccezione di `{HOST.METADATA}`), `{INVENTORY.*}` e macro utente.

Per aprire il menu dell'host, fare clic sul pulsante *Opzioni* prima del nome dell'host.

Per configurare un nuovo host, fare clic sul pulsante *Procedura guidata host* o *Crea host* nell'angolo superiore destro.

Per importare un host da un file YAML, XML o JSON, fare clic sul pulsante *Importa* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

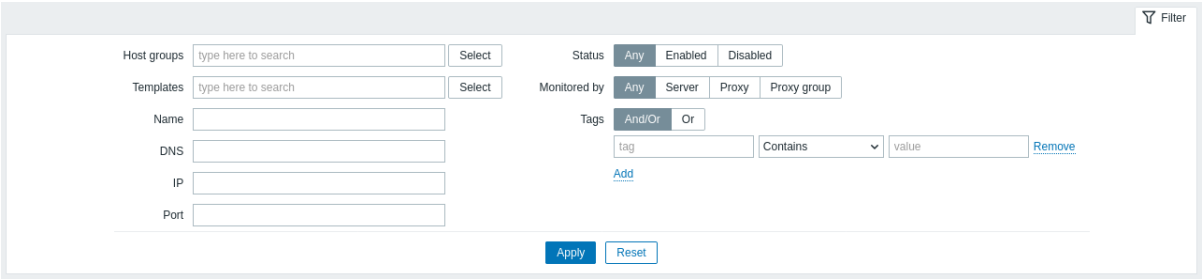
- *Enable* - cambia lo stato dell'host in *Monitored*;
- *Disable* - cambia lo stato dell'host in *Not monitored*;
- *Export* - esporta gli host in un file YAML, XML o JSON;
- *Mass update* - aggiorna diverse proprietà per più host contemporaneamente;
- *Delete* - elimina gli host.

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi host, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

Puoi usare il filtro per visualizzare solo gli host che ti interessano. Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

L'icona *Filter* è disponibile nell'angolo in alto a destra. Facendoci clic si aprirà un filtro in cui puoi specificare i criteri di filtraggio desiderati.



Parameter	Description
Host groups	Filtra per uno o più gruppi di host. Specificare un gruppo di host padre seleziona implicitamente tutti i gruppi di host nidificati.
Templates	Filtra per template collegati.
Name	Filtra per il nome visibile dell'host.
DNS	Filtra per nome DNS.

Parameter	Description
IP	Filtra per indirizzo IP.
Port	Filtra per numero di porta.
Status	Filtra per stato dell'host.
Monitored by	Filtra gli host monitorati da Zabbix server, proxy o gruppo di proxy.
Proxies	Filtra gli host monitorati dai proxy specificati qui. Questo campo è disponibile solo se in *Monitored by* è selezionato "Proxy".
Proxy groups	Filtra gli host monitorati dai gruppi di proxy specificati qui. Questo campo è disponibile solo se in *Monitored by* è selezionato "Proxy group".
Tags	Filtra per nome e valore del tag dell'host. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole); **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole). Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or; **Or** - è sufficiente che una condizione sia soddisfatta.

Stato interfaccia sconosciuto

Zabbix server visualizza lo stato "Unknown" per un'interfaccia host (Agent, SNMP, IPMI, JMX) nei seguenti casi:

- L'host è disabilitato.
- L'host è impostato per essere monitorato da proxy, da un proxy diverso o dal server se in precedenza era monitorato da proxy.
- L'host è monitorato da un proxy che sembra essere offline (nessun aggiornamento ricevuto dal proxy durante il massimo intervallo di heartbeat - 1 ora).
- Tutti gli item dell'host con quel tipo di interfaccia sono disabilitati.
- Non sono stati configurati poller per quel tipo di interfaccia (ad esempio, il parametro di configurazione del server `StartAgentPollers` o `StartSNMPPollers` è impostato su 0).

La disponibilità dell'interfaccia viene impostata su "Unknown" dopo la sincronizzazione della cache di configurazione di Zabbix server.

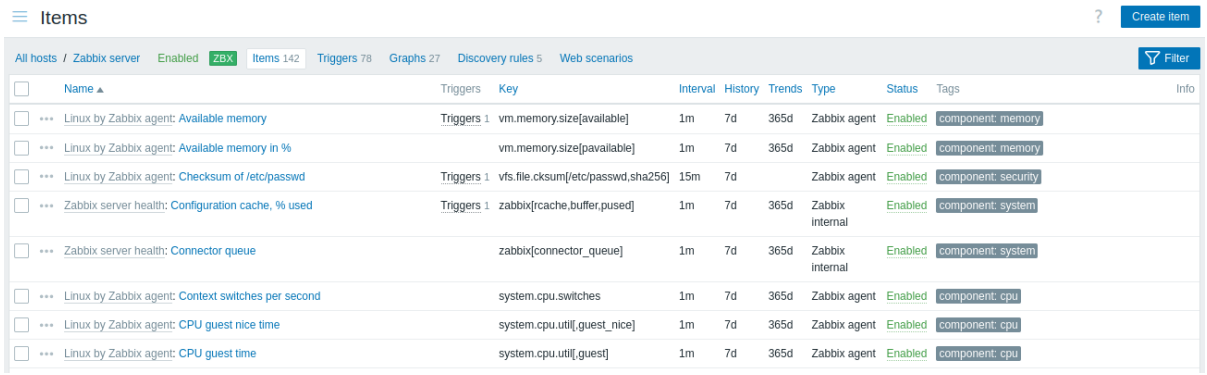
La disponibilità dell'interfaccia (Available/Not available) sugli host monitorati dai proxy viene ripristinata dopo la sincronizzazione della cache di configurazione del proxy.

Item

Panoramica

È possibile accedere all'elenco degli item di un host da *Raccolta dati* > *Host*, facendo clic su *Item* per l'host desiderato.

Viene visualizzato un elenco degli item esistenti.



Dati visualizzati:

Colonna	Descrizione
Menu item	Fare clic sull'icona con tre puntini per aprire il menu item.
Host	Host dell'item. Facendo clic sul nome dell'host si apre il modulo di configurazione dell'host. Questa colonna viene visualizzata solo se nel filtro sono selezionati più host o nessun host.
Nome	Nome dell'item visualizzato come collegamento blu ai dettagli dell'item. Facendo clic sul collegamento del nome dell'item si apre il modulo di configurazione dell'item. Se l'item dell'host appartiene a un template, il nome del template viene visualizzato prima del nome dell'item come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco degli item a livello di template. Se l'item è stato creato da un prototipo di item, il suo nome è preceduto dal nome della regola di discovery di basso livello, in arancione. Facendo clic sul nome della regola di discovery si apre l'elenco dei prototipi di item.
Trigger	Passando il mouse su Trigger viene visualizzato un riquadro informativo con i trigger associati all'item. Il numero di trigger viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item.
Intervallo	Viene visualizzata la frequenza del controllo. *Nota*: anche gli item passivi possono essere controllati immediatamente premendo il pulsante *Esegui ora*.
Cronologia	Viene visualizzato per quanti giorni verrà conservata la cronologia dei dati dell'item.
Trend	Viene visualizzato per quanti giorni verrà conservata la cronologia dei trend dell'item.
Tipo	Viene visualizzato il tipo di item (Zabbix agent, SNMP agent, controllo semplice, ecc.).
Stato	Viene visualizzato lo stato dell'item: *Abilitato*, *Disabilitato* o *Non supportato*. È possibile modificare manualmente lo stato facendo clic su di esso: da Abilitato a Disabilitato e viceversa; da Non supportato a Disabilitato e viceversa. Gli item rilevati che sono stati persi vengono contrassegnati con un'icona informativa. Il testo del suggerimento fornisce dettagli sul loro stato.
Tag	Vengono visualizzati i tag dell'item. È possibile visualizzare fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di visualizzare tutti i tag passando il mouse su di esso.
Informazioni	Se l'item funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". Passare il mouse sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare un nuovo item, fare clic sul pulsante *Crea item* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Enable* - cambia lo stato dell'item in *Enabled*
- *Disable* - cambia lo stato dell'item in *Disabled*
- *Execute now* - esegue immediatamente un controllo per i nuovi valori dell'item. Supportato solo per i controlli **passive** (vedi maggiori dettagli). Nota che, quando si controllano i valori immediatamente, la cache di configurazione non viene aggiornata; quindi i valori non rifletteranno le modifiche molto recenti alla configurazione dell'item.
- *Clear history and trends* - elimina i dati di history e trend per gli item.
- *Copy* - copia gli item su altri host o template.

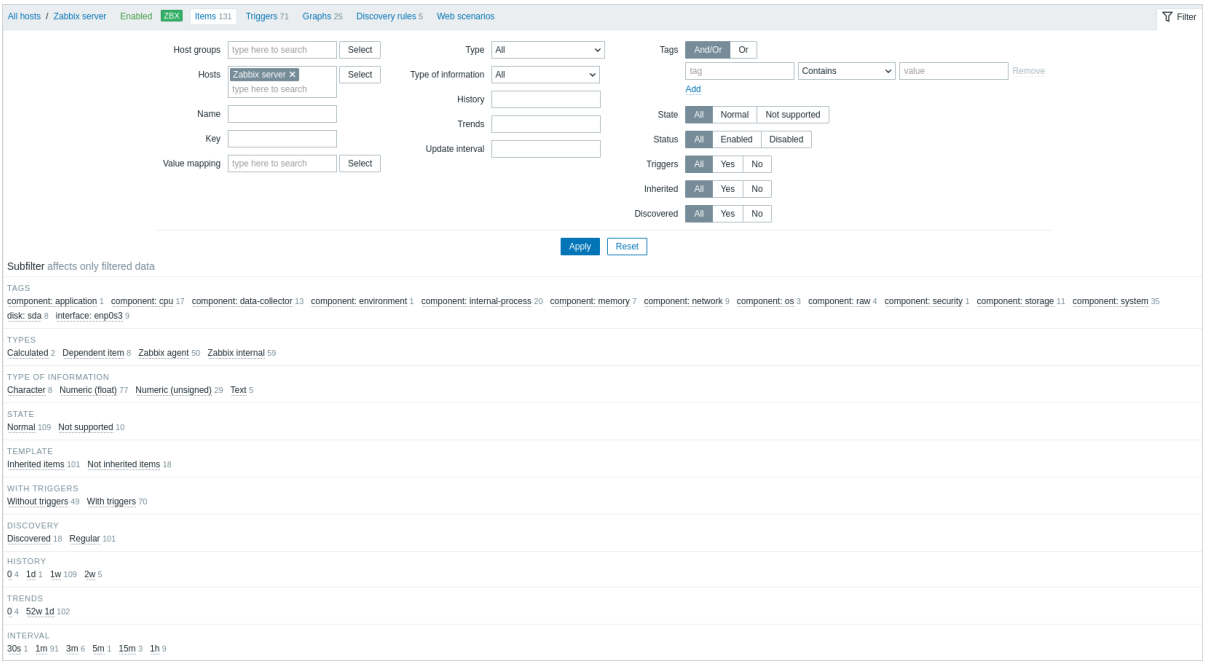
- *Mass update* - aggiorna diverse proprietà per più item contemporaneamente.
- *Delete* - elimina gli item.

Per usare queste opzioni, seleziona le caselle di controllo accanto ai rispettivi item, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo gli item di interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

L'icona *Filter* è disponibile nell'angolo in alto a destra. Facendo clic su di essa si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.



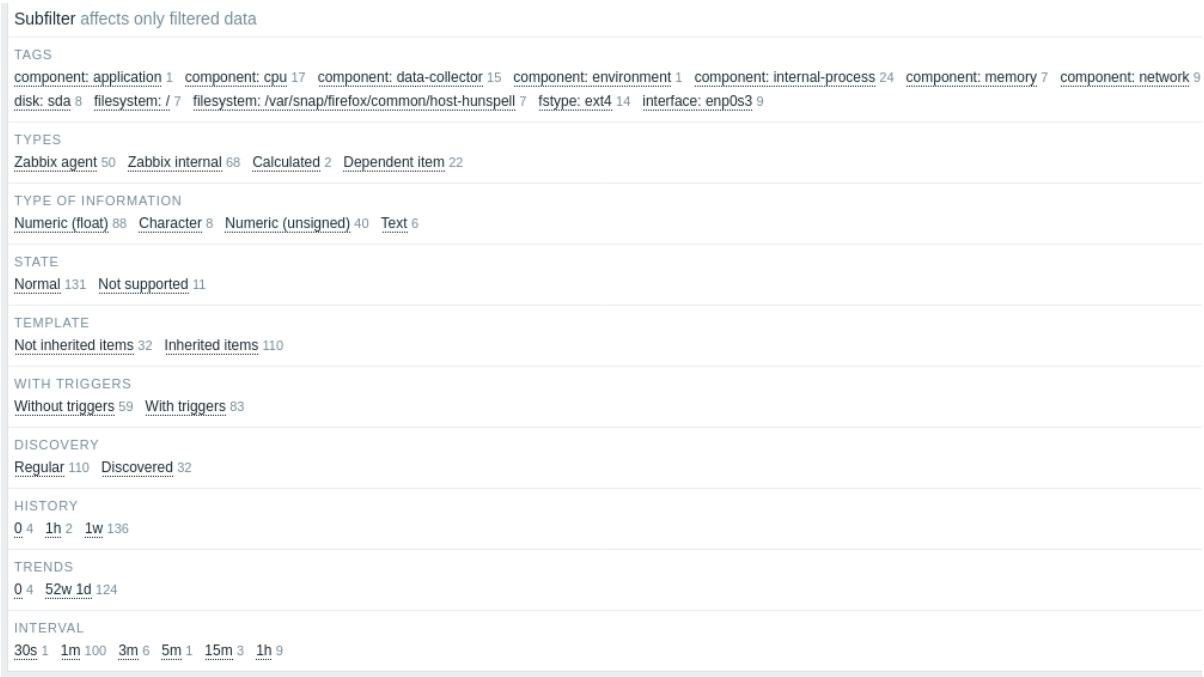
Parameter	Description
Host groups	Filtra per uno o più host group. La specifica di un host group padre seleziona implicitamente tutti gli host group nidificati. Non è possibile selezionare gli host group che contengono solo template.
Hosts	Filtra per uno o più host.
Name	Filtra per nome dell'item.
Key	Filtra per chiave dell'item.
Value mapping	Filtra in base alla value map utilizzata. Questo parametro non viene visualizzato se l'opzione *Hosts* è vuota.
Type	Filtra per tipo di item (Zabbix agent, SNMP agent, ecc.).
Type of information	Filtra per tipo di informazione (Numeric unsigned, float, ecc.).
History	Filtra in base al periodo di conservazione della history dell'item.
Trends	Filtra in base al periodo di conservazione delle trends dell'item.
Update interval	Filtra in base all'intervallo di aggiornamento dell'item.

Parameter	Description
Tags	Specificare i tag per limitare il numero di item visualizzati. È possibile includere ed escludere tag e valori dei tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole e minuscole. Per ogni condizione sono disponibili diversi operatori: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole) **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole e minuscole) **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza di sottostringa, non sensibile alle maiuscole e minuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate tramite la condizione Or **Or** - è sufficiente che una condizione sia soddisfatta
State	Filtra per stato dell'item - *Normal* o *Not supported*.
Status	Filtra per stato dell'item - *Enabled* o *Disabled*.
Triggers	Filtra gli item con (o senza) trigger.
Inherited	Filtra gli item ereditati (o non ereditati) da un template.
Discovery	Filtra gli item individuati (o non individuati) dalla low-level discovery.

Utilizzo del subfiltro

Il subfiltro consente di modificare ulteriormente il filtraggio applicato dal filtro principale.

Contiene collegamenti cliccabili per accedere rapidamente agli item correlati. Gli item sono correlati da un'entità comune, ad esempio tag, tipo di item, stato dell'item, intervallo di aggiornamento dell'item e così via. Quando si fa clic su un'entità, questa viene evidenziata con uno sfondo grigio e gli item vengono filtrati immediatamente (non è necessario fare clic su *Applica* nel filtro principale). Facendo clic su un'altra entità, questa viene aggiunta ai risultati filtrati. Facendo nuovamente clic sull'entità, il filtro viene rimosso.



I subfiltri vengono generati in base ai dati filtrati, limitati a 1000 record. Se si desidera visualizzare più record nel subfiltro, è necessario aumentare il valore del parametro *Limite per i risultati della ricerca e del filtro* (in *Amministrazione* > *Generale* > *GUI*).

A differenza del filtro principale, il subfiltro viene aggiornato a ogni richiesta di aggiornamento della tabella, in modo da mostrare sempre informazioni aggiornate sulle opzioni di filtraggio disponibili e sui relativi conteggi.

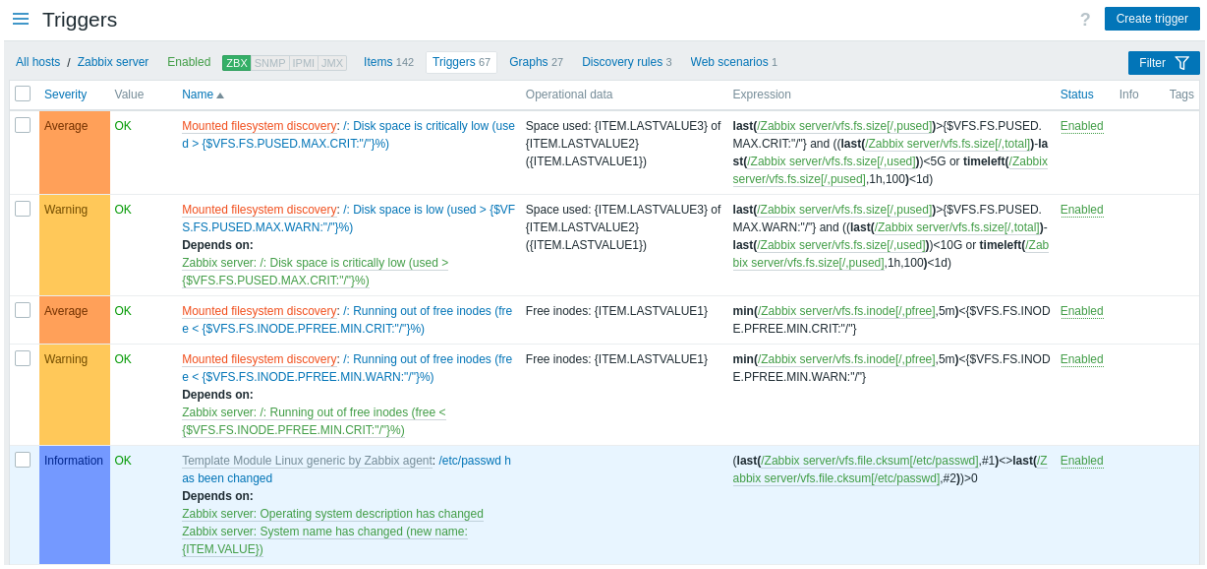
Il numero di entità visualizzate è limitato a 100 in orizzontale. Se sono presenti più entità, alla fine viene visualizzata un'icona con tre punti; l'icona non è cliccabile.

Un numero accanto a ogni entità cliccabile indica il numero di item raggruppati al suo interno (in base ai risultati del filtro principale). Quando si fa clic su un'entità, i numeri delle altre entità disponibili vengono visualizzati con un segno più, che indica quanti item possono essere aggiunti alla selezione corrente. Le entità senza item non vengono visualizzate, a meno che non siano già state selezionate nel subfiltro.

Trigger

Panoramica

L'elenco dei trigger per un host è accessibile da *Raccolta dati* > *Host*, facendo clic su *Trigger* per l'host desiderato.



Dati visualizzati:

Colonna	Descrizione
Gravità	La gravità del trigger viene visualizzata sia tramite il nome sia tramite il colore di sfondo della cella.
Valore	Viene visualizzato lo stato del trigger: **OK** - stato OK **PROBLEM** - stato Problema
Host	Host del trigger. Facendo clic sul nome dell'host si apre il modulo di configurazione dell'host. Questa colonna viene visualizzata solo se nel filtro sono selezionati più host o nessun host.
Nome	Nome del trigger, visualizzato come collegamento blu ai dettagli del trigger. Facendo clic sul collegamento con il nome del trigger si apre il modulo di configurazione del trigger. Se il trigger dell'host appartiene a un template, il nome del template viene visualizzato prima del nome del trigger come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei trigger a livello di template. Se il trigger è stato creato da un prototipo di trigger, il suo nome è preceduto dal nome della regola di discovery di basso livello, in arancione. Facendo clic sul nome della regola di discovery si apre l'elenco dei prototipi di trigger.
Dati operativi	Definizione dei dati operativi del trigger, contenente stringhe arbitrarie e macro che verranno risolte dinamicamente in *Monitoraggio* > *Problemi*.
Espressione	Viene visualizzata l'espressione del trigger. La parte host-item dell'espressione viene visualizzata come collegamento che conduce al modulo di configurazione dell'item.
Stato	Viene visualizzato lo stato del trigger: *Abilitato*, *Disabilitato* o *Sconosciuto*. Facendo clic sullo stato è possibile modificarlo manualmente: da Abilitato a Disabilitato (e viceversa); da Sconosciuto a Disabilitato (e viceversa). I problemi di un trigger disabilitato non vengono più visualizzati nel frontend, ma non vengono eliminati. I trigger rilevati che sono andati persi vengono contrassegnati con un'icona informativa. Il testo del tooltip fornisce dettagli sul loro stato.

Colonna	Descrizione
Informazioni	Se tutto funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". Passare il mouse sull'icona per visualizzare un tooltip con la descrizione dell'errore.
Tag	Se un trigger contiene tag, in questa colonna vengono visualizzati il nome e il valore del tag.

Per configurare un nuovo trigger, fare clic sul pulsante *Crea trigger* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

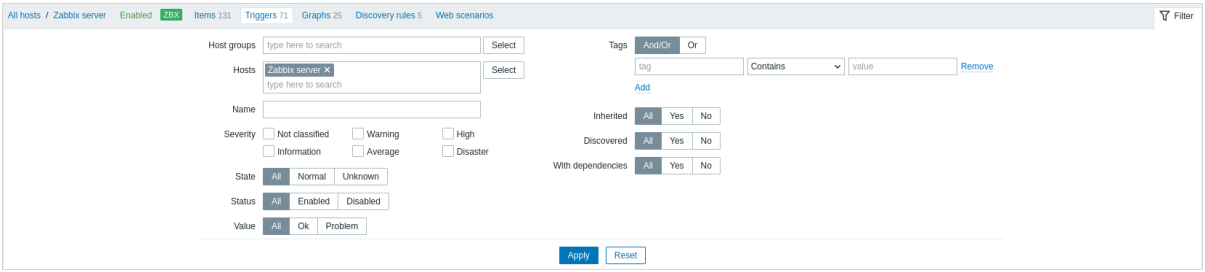
- *Enable* - cambia lo stato del trigger in *Enabled*.
- *Disable* - cambia lo stato del trigger in *Disabled*.
- *Copy* - copia i trigger su altri host o template.
- *Mass update* - aggiorna più proprietà per un numero di trigger contemporaneamente.
- *Delete* - elimina i trigger.

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi trigger, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

Puoi usare il filtro per visualizzare solo i trigger di tuo interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con i macro non risolti.

L'icona *Filter* è disponibile nell'angolo in alto a destra. Facendoci clic si aprirà un filtro in cui puoi specificare i criteri di filtraggio desiderati.



Parametro	Descrizione
Host groups	Filtra per uno o più host groups. Specificare un host group padre seleziona implicitamente tutti gli host groups nidificati. Non è possibile selezionare host groups che contengono solo template.
Hosts	Filtra per uno o più host. Se gli host groups sono già selezionati sopra, la selezione degli host è limitata a quei gruppi.
Name	Filtra per nome del trigger.
Severity	Seleziona per filtrare per una o più severità del trigger.
State	Filtra per lo stato dell'espressione del trigger (Normal/Unknown).
Status	Filtra per lo stato di attivazione del trigger (Enabled/Disabled).
Value	Filtra per il valore del trigger.

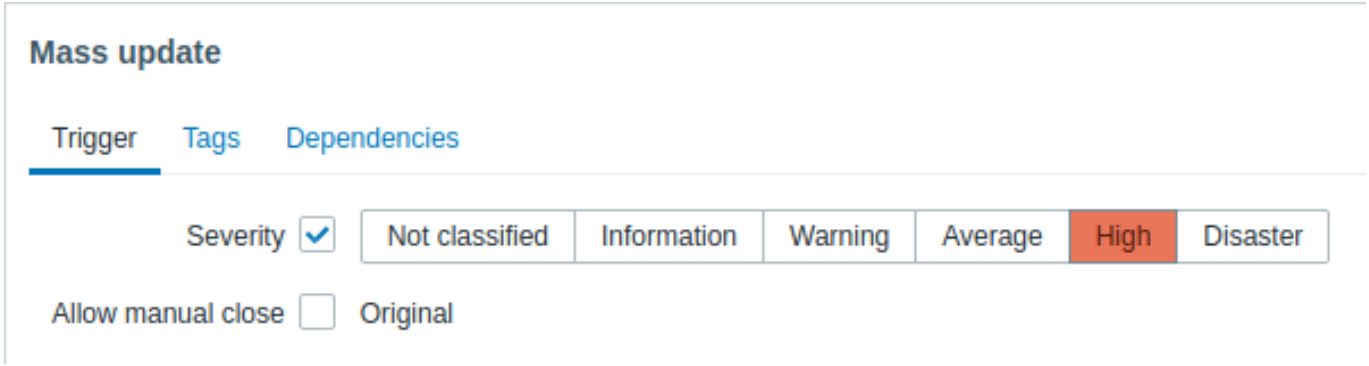
Parametro	Descrizione
Tags	Filtra per nome e valore del tag del trigger. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi dei tag specificati **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole) **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole) **Does not exist** - esclude i nomi dei tag specificati **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole) **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole) Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate tramite la condizione Or **Or** - è sufficiente che una condizione sia soddisfatta Le macro e le macro functions sono supportate sia nei campi del nome del tag sia in quelli del valore del tag.
Inherited	Filtra i trigger ereditati (o non ereditati) da un template.
Discovered	Filtra i trigger scoperti (o non scoperti) dalla low-level discovery.
With dependencies	Filtra i trigger con (o senza) dipendenze.

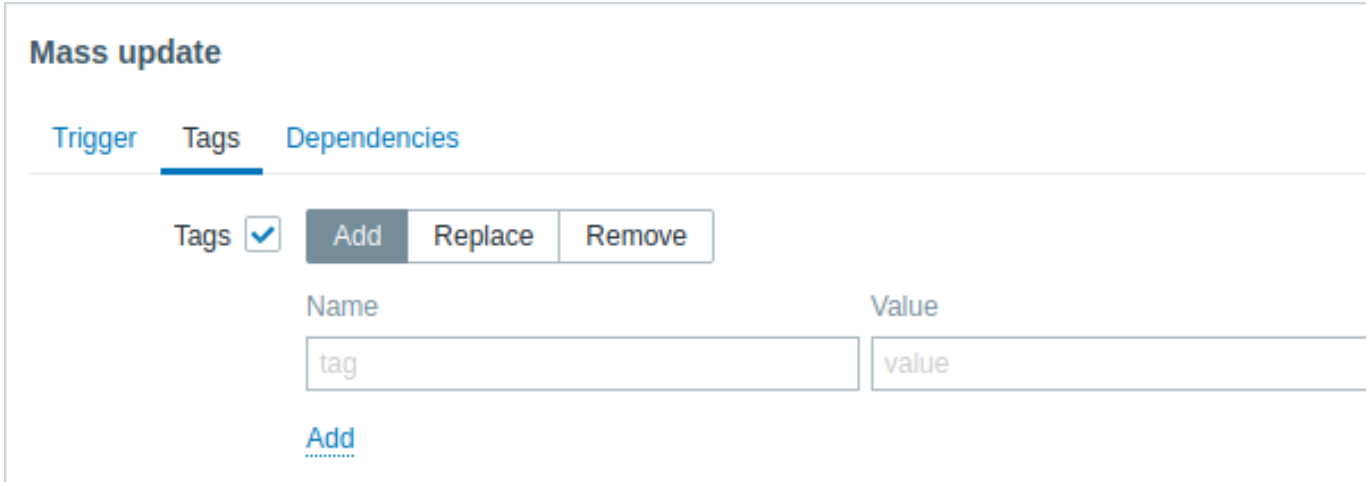
Utilizzo dell'aggiornamento di massa

Con l'aggiornamento di massa è possibile modificare contemporaneamente alcuni attributi di più trigger, evitando di dover aprire ogni singolo trigger per modificarlo.

Per aggiornare più trigger contemporaneamente, procedere come segue:

- Selezionare le caselle di controllo dei trigger che si desidera aggiornare nell'elenco
- Fare clic su *Aggiornamento di massa* sotto l'elenco
- Accedere alla scheda con gli attributi richiesti (*Trigger*, *Tag* o *Dipendenze*)
- Selezionare le caselle di controllo degli attributi da aggiornare

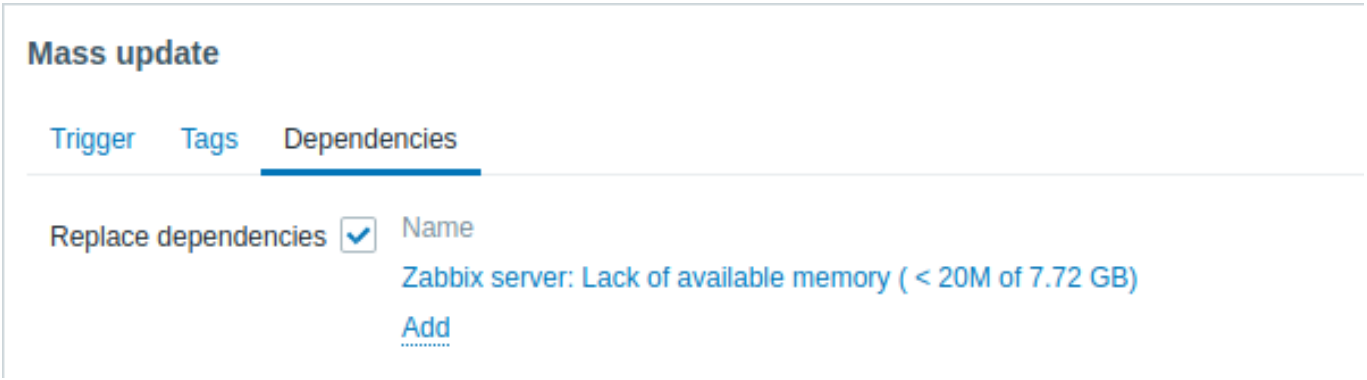




Quando si seleziona il pulsante corrispondente per l'aggiornamento dei tag, sono disponibili le seguenti opzioni:

- *Aggiungi* - aggiunge nuovi tag ai trigger;
- *Sostituisci* - rimuove tutti i tag esistenti dal trigger e li sostituisce con quelli specificati di seguito;
- *Rimuovi* - rimuove dai trigger i tag specificati.

I tag con lo stesso nome ma valori diversi non sono considerati ”duplicati” e possono essere aggiunti allo stesso trigger.



Sostituisci dipendenze - rimuove tutte le dipendenze esistenti dal trigger e le sostituisce con quelle specificate.

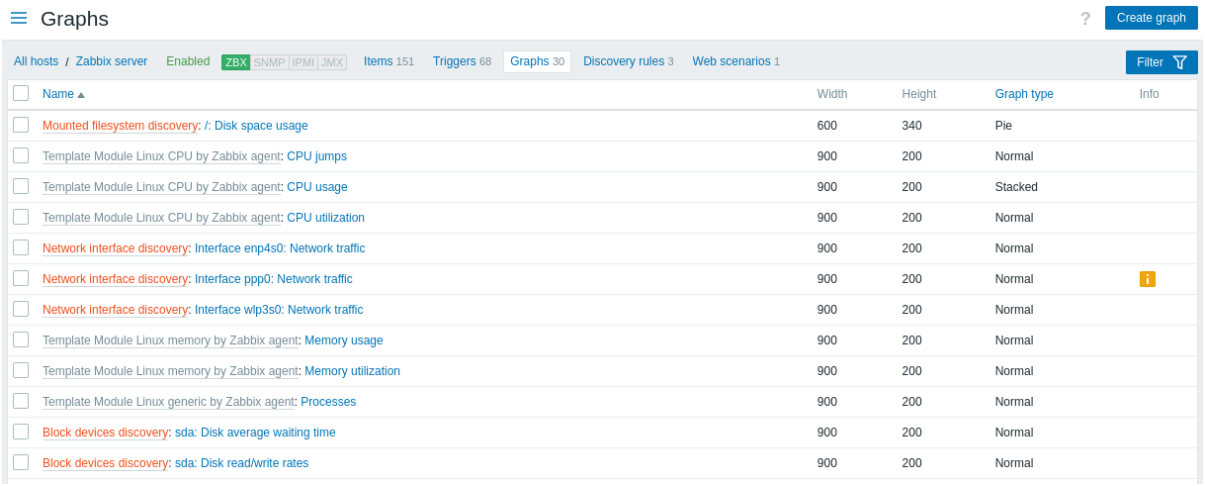
Fare clic su *Aggiorna* per applicare le modifiche.

Grafici

Panoramica

L’elenco dei grafici personalizzati per un host è accessibile da *Raccolta dati* > *Host*, facendo clic su *Grafici* per l’host desiderato.

Viene visualizzato un elenco dei grafici esistenti.



Dati visualizzati:

Colonna	Descrizione
Host	Host del grafico. Facendo clic sul nome dell’host si apre il modulo di configurazione dell’host. Questa colonna viene visualizzata solo se nel filtro sono selezionati più host o nessun host.
Nome	Nome del grafico personalizzato, visualizzato come collegamento blu ai dettagli del grafico. Facendo clic sul collegamento con il nome del grafico si apre il modulo di configurazione del grafico. Se il grafico dell’host appartiene a un template, il nome del template viene visualizzato prima del nome del grafico come collegamento grigio. Facendo clic sul collegamento del template si apre l’elenco dei grafici a livello di template. Se il grafico è stato creato da un prototipo di grafico, il suo nome è preceduto dal nome della regola di low-level discovery, in arancione. Facendo clic sul nome della regola di discovery si apre l’elenco dei prototipi di grafico.
Larghezza	Viene visualizzata la larghezza del grafico.
Altezza	Viene visualizzata l’altezza del grafico.
Tipo di grafico	Viene visualizzato il tipo di grafico: *Normale*, *Impilato*, *A torta* o *Esploso*.

Colonna	Descrizione
Informazioni	Se il grafico funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". Passare il mouse sull'icona per visualizzare un tooltip con la descrizione dell'errore.

Per configurare un nuovo grafico, fare clic sul pulsante *Crea grafico* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Copia* - copia i graph su altri host o template
- *Elimina* - elimina i graph

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi graph, quindi fai clic sul pulsante desiderato.

Uso del filtro

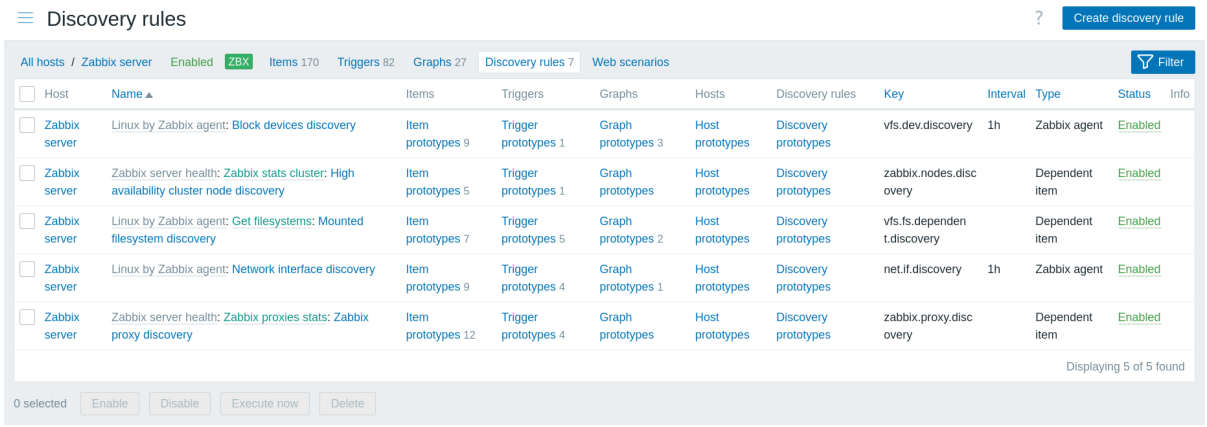
È possibile filtrare i grafici per gruppo host e host. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Regole di discovery

Panoramica

L'elenco delle regole di discovery di basso livello per un host è accessibile da *Raccolta dati* > *Host*, facendo clic su *Discovery* per l'host desiderato.

Viene visualizzato un elenco delle regole di discovery di basso livello esistenti. È inoltre possibile visualizzare tutte le regole di discovery indipendentemente dall'host oppure tutte le regole di discovery di uno specifico gruppo di host modificando le impostazioni del filtro.



Dati visualizzati:

Colonna	Descrizione
Host	Viene visualizzato il nome visibile dell'host. Facendo clic sul nome dell'host si apre il modulo di configurazione dell'host. Se non è disponibile un nome visibile dell'host, viene visualizzato il nome tecnico dell'host.
Nome	Nome della regola, visualizzato come collegamento blu. Facendo clic sul nome della regola si apre il modulo di configurazione della regola di discovery di basso livello. Se la regola di discovery appartiene a un template, il nome del template viene visualizzato prima del nome della regola come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco delle regole a livello di template.
Item	Viene visualizzato un collegamento all'elenco dei prototipi di item. Il numero di prototipi di item esistenti viene visualizzato in grigio.
Trigger	Viene visualizzato un collegamento all'elenco dei prototipi di trigger. Il numero di prototipi di trigger esistenti viene visualizzato in grigio.
Grafici	Viene visualizzato un collegamento all'elenco dei prototipi di grafico. Il numero di prototipi di grafico esistenti viene visualizzato in grigio.

Colonna	Descrizione
Host	Viene visualizzato un collegamento all'elenco dei prototipi di host. Il numero di prototipi di host esistenti viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item utilizzata per la discovery.
Intervallo	Viene visualizzata la frequenza di esecuzione della discovery. *Nota*: la discovery può essere eseguita anche immediatamente premendo il pulsante *Esegui ora* sotto l'elenco.
Tipo	Viene visualizzato il tipo di item utilizzato per la discovery (Zabbix agent, SNMP agent, ecc.).
Stato	Viene visualizzato lo stato della regola di discovery: *Abilitata*, *Disabilitata* o *Non supportata*. Facendo clic sullo stato è possibile modificarlo: da Abilitata a Disabilitata (e viceversa); da Non supportata a Disabilitata (e viceversa).
Informazioni	Se tutto funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". Passare il mouse sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare una nuova regola di discovery di basso livello, fare clic sul pulsante *Crea regola di discovery* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

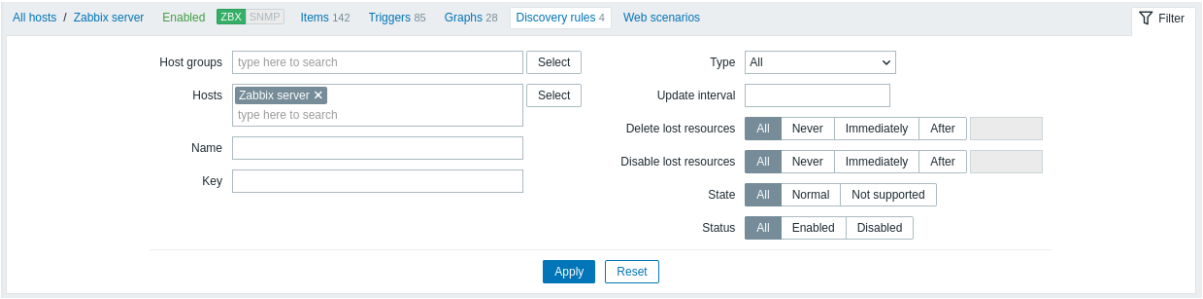
- *Enable* - cambia lo stato della regola di discovery di basso livello in *Enabled*.
- *Disable* - cambia lo stato della regola di discovery di basso livello in *Disabled*.
- *Execute now* - esegue immediatamente il discovery in base alle regole di discovery. Vedere maggiori dettagli. Si noti che quando il discovery viene eseguito immediatamente, la cache di configurazione non viene aggiornata, quindi il risultato non rifletterà le modifiche molto recenti alla configurazione della regola di discovery.
- *Delete* - elimina le regole di discovery di basso livello.

Per usare queste opzioni, selezionare le caselle di controllo prima delle rispettive regole di discovery, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

Puoi usare il filtro per visualizzare solo le regole di discovery che ti interessano. Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il link *Filter* è disponibile sopra l'elenco delle regole di discovery. Se fai clic su di esso, diventa disponibile un filtro con cui puoi filtrare le regole di discovery per gruppo host, host, nome, chiave dell'item, tipo di item e altri parametri.



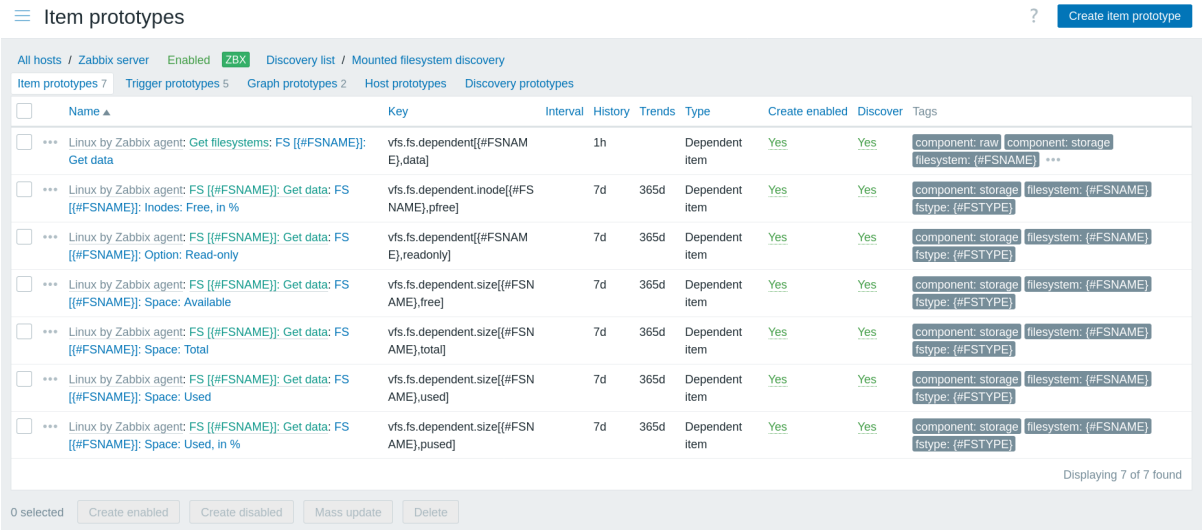
Parameter	Description
Host groups	Filtra per uno o più gruppi host. La specifica di un gruppo host padre seleziona implicitamente tutti i gruppi host nidificati.
Hosts	Filtra per uno o più host.
Name	Filtra per il nome della regola di discovery.
Key	Filtra per la chiave dell'item di discovery.
Type	Filtra per il tipo dell'item di discovery.
Update interval	Filtra per l'intervallo di aggiornamento. Non disponibile per Zabbix trapper e item dipendenti.
Delete lost resources	Filtra per il periodo *Delete lost resources*.
Disable lost resources	Filtra per il periodo *Disable lost resources*.
SNMP OID	Filtra per SNMP OID. Disponibile solo se *SNMP agent* è selezionato come tipo.
State	Filtra per lo stato della regola di discovery (All/Normal/Not supported).

Parameter	Description
Status	Filtra per lo stato della regola di discovery (All/Enabled/Disabled).

Prototipi di item

Panoramica

In questa sezione vengono visualizzati i prototipi di item di una regola di discovery di basso livello sul host. I prototipi di item sono la base degli item reali del host che vengono creati durante la discovery di basso livello.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di item, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di item. Se il prototipo di item appartiene a un template, il nome del template viene visualizzato prima del nome della regola come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di item a livello del template. Se il prototipo di item è di tipo "Dependent item", il nome del master item viene visualizzato prima del nome del prototipo di item in verde.
Key	Viene visualizzata la chiave del prototipo di item.
Interval	Viene visualizzata la frequenza del controllo.
History	Viene visualizzato per quanti giorni conservare lo storico dei dati dell'item.
Trends	Viene visualizzato per quanti giorni conservare lo storico dei trend dell'item.
Type	Viene visualizzato il tipo di prototipo di item (Zabbix agent, SNMP agent, controllo semplice, ecc.).
Create enabled	Crea l'item in base a questo prototipo come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Discover	Esegue la discovery dell'item in base a questo prototipo: **Yes** - esegui la discovery **No** - non eseguire la discovery. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Tags	Vengono visualizzati i tag del prototipo di item.

Per configurare un nuovo prototipo di item, fare clic sul pulsante *Create item prototype* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

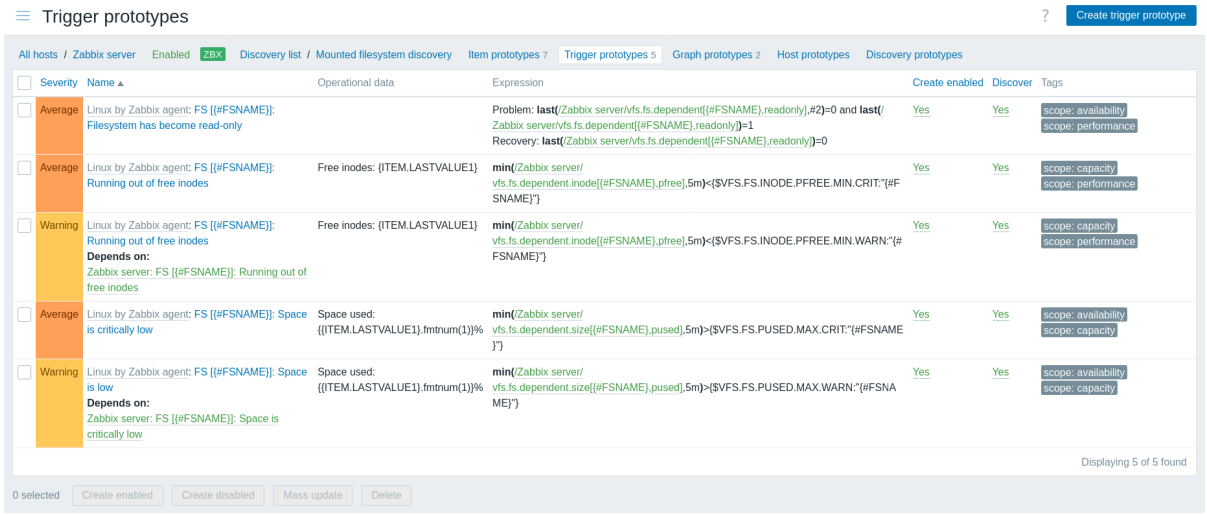
- *Crea abilitati* - crea questi item come *Abilitati*
- *Crea disabilitati* - crea questi item come *Disabilitati*
- *Aggiornamento di massa* - aggiorna in massa questi prototipi di item
- *Elimina* - elimina questi prototipi di item

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di item, quindi fai clic sul pulsante richiesto.

Prototipi di trigger

Panoramica

In questa sezione vengono visualizzati i prototipi di trigger di una regola di discovery di basso livello sul host. I prototipi di trigger sono la base dei trigger reali del host, creati durante la discovery di basso livello.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del prototipo di trigger, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di trigger. Se il prototipo di trigger appartiene a un template collegato, il nome del template viene visualizzato prima del nome del trigger come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di trigger a livello del template collegato.
Dati operativi	Viene visualizzato il formato dei dati operativi del trigger, contenente stringhe arbitrarie e macro che verranno risolte dinamicamente in *Monitoraggio* > *Problemi*.
Creazione abilitata	Crea il trigger in base a questo prototipo come: **Sì** - abilitato **No** - disabilitato. È possibile passare da "Sì" a "No" e viceversa facendo clic sui valori.
Discovery	Esegue la discovery del trigger in base a questo prototipo: **Sì** - esegui la discovery **No** - non eseguire la discovery. È possibile passare da "Sì" a "No" e viceversa facendo clic sui valori.
Tag	Vengono visualizzati i tag del prototipo di trigger.

Per configurare un nuovo prototipo di trigger, fare clic sul pulsante *Crea prototipo di trigger* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

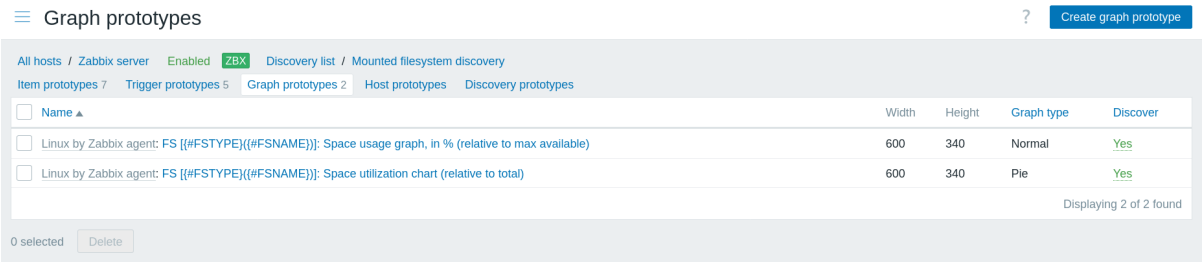
- *Crea abilitati* - crea questi trigger come *Abilitati*
- *Crea disabilitati* - crea questi trigger come *Disabilitati*
- *Aggiornamento di massa* - aggiorna in massa questi prototipi di trigger
- *Elimina* - elimina questi prototipi di trigger

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di trigger, quindi fai clic sul pulsante richiesto.

Prototipi di grafico

Panoramica

In questa sezione vengono visualizzati i prototipi di grafico di una regola di discovery di basso livello sul host. I prototipi di grafico sono la base dei grafici reali del host creati durante la discovery di basso livello.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di grafico, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di grafico. Se il prototipo di grafico appartiene a un template collegato, il nome del template viene visualizzato prima del nome del grafico come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di grafico a livello del template collegato.
Width	Viene visualizzata la larghezza del prototipo di grafico.
Height	Viene visualizzata l'altezza del prototipo di grafico.
Type	Viene visualizzato il tipo di prototipo di grafico: *Normal*, *Stacked*, *Pie* o *Exploded*.
Discover	Esegue la discovery del grafico in base a questo prototipo: **Yes** - esegui la discovery **No** - non eseguire la discovery. È possibile passare da 'Yes' a 'No' e viceversa facendo clic sui valori.

Per configurare un nuovo prototipo di grafico, fare clic sul pulsante *Create graph prototype* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

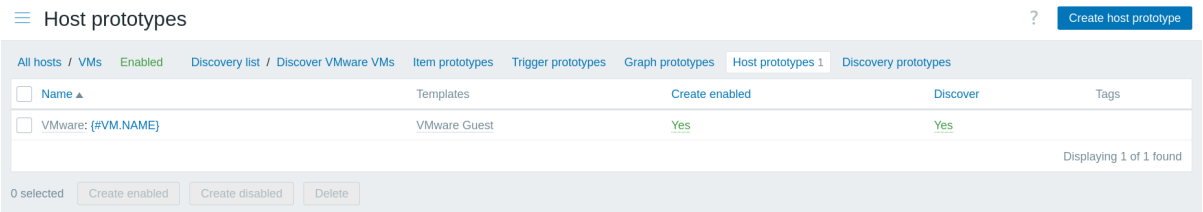
- *Elimina* - elimina questi prototipi di grafico

Per usare queste opzioni, seleziona le caselle di controllo accanto ai rispettivi prototipi di grafico, quindi fai clic sul pulsante desiderato.

Prototipi host

Panoramica

In questa sezione vengono visualizzati i prototipi di host di una regola di discovery di basso livello sull'host. I prototipi di host sono la base degli host reali creati durante la discovery di basso livello.



Dati visualizzati:

Column	Description
Nome	Nome del prototipo di host, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di host. Se il prototipo di host appartiene a un template collegato, il nome del template viene visualizzato prima del nome dell'host come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di host a livello del template collegato.
Template	Vengono visualizzati i template del prototipo di host.

Column	Description
Creazione abilitata	Crea l'host in base a questo prototipo come: **Sì** - abilitata **No** - disabilitata. È possibile passare da "Sì" a "No" e viceversa facendo clic su di essi.
Discovery	Esegue la discovery dell'host in base a questo prototipo: **Sì** - esegui la discovery **No** - non eseguire la discovery. È possibile passare da "Sì" a "No" e viceversa facendo clic su di essi.
Tag	Vengono visualizzati i tag del prototipo di host.

Per configurare un nuovo prototipo di host, fare clic sul pulsante *Crea prototipo di host* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

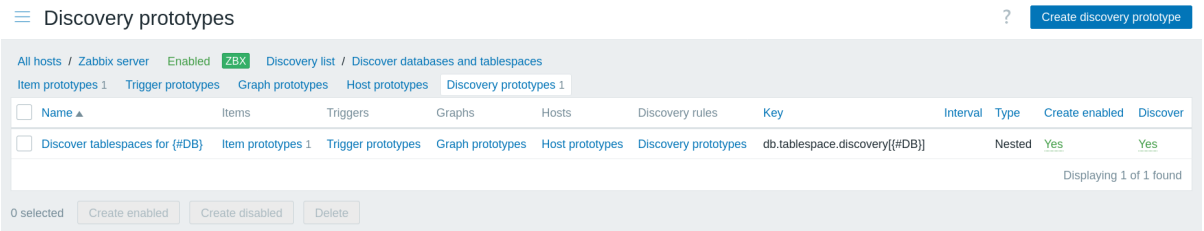
- *Crea abilitati* - crea questi host come *Abilitati*
- *Crea disabilitati* - crea questi host come *Disabilitati*
- *Elimina* - elimina questi prototipi di host

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di host, quindi fai clic sul pulsante richiesto.

Prototipi di discovery

Panoramica

In questa sezione vengono visualizzati i prototipi di discovery di una regola di discovery low-level sul host. I prototipi di discovery sono regole di discovery annidate nella regola di discovery principale.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del prototipo di discovery, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di discovery. Se il prototipo di discovery appartiene a un template collegato, il nome del template viene visualizzato prima del nome del host come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco dei prototipi di discovery a livello del template collegato.
Item	Viene visualizzato un collegamento all'elenco dei prototipi di item. Il numero di prototipi di item esistenti viene visualizzato in grigio.
Trigger	Viene visualizzato un collegamento all'elenco dei prototipi di trigger. Il numero di prototipi di trigger esistenti viene visualizzato in grigio.
Grafici	Viene visualizzato un collegamento all'elenco dei prototipi di grafico. Il numero di prototipi di grafico esistenti viene visualizzato in grigio.
Host	Viene visualizzato un collegamento all'elenco dei prototipi di host. Il numero di prototipi di host esistenti viene visualizzato in grigio.
Regole di discovery	Viene visualizzato un collegamento all'elenco dei prototipi di discovery. Il numero di prototipi di discovery esistenti viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item utilizzata per la discovery.
Intervallo	Viene visualizzata la frequenza di esecuzione della discovery. *Nota*: la discovery può essere eseguita anche immediatamente premendo il pulsante *Esegui ora* sotto l'elenco.
Tipo	Viene visualizzato il tipo di item utilizzato per la discovery (Zabbix agent, SNMP agent, ecc.).

Colonna	Descrizione
Creazione abilitata	Viene visualizzato lo stato di creazione del prototipo di discovery: creazione abilitata (*Sì*) o creazione disabilitata (*No*). Facendo clic sullo stato è possibile modificarlo, da Sì a No e viceversa.
Rileva	Viene visualizzato lo stato di rilevamento del prototipo di discovery: rileva (*Sì*) o non rilevare (*No*). Facendo clic sullo stato è possibile modificarlo, da Sì a No e viceversa.

Per configurare un nuovo prototipo di discovery, fare clic sul pulsante *Crea prototipo di discovery* nell'angolo superiore destro.

Opzioni di modifica in massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica in massa:

- *Crea abilitato* - crea questi prototipi di discovery come *Abilitato*
- *Crea disabilitato* - crea questi prototipi di discovery come *Disabilitato*
- *Elimina* - elimina questi prototipi di discovery

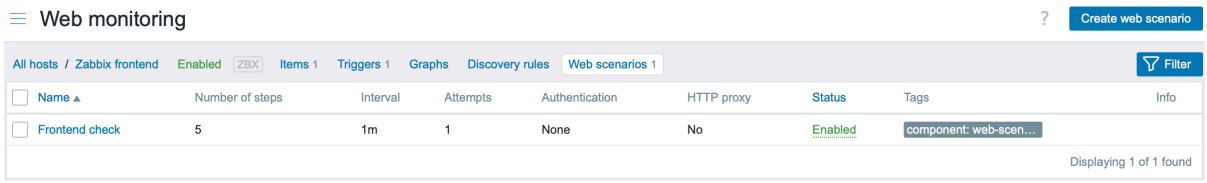
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di discovery, quindi fai clic sul pulsante desiderato.

Scenari web

Panoramica

L'elenco degli scenari web per un host è accessibile da *Raccolta dati* > *Host*, facendo clic su *Web* per l'host desiderato.

Viene visualizzato un elenco degli scenari web esistenti.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome dello scenario web. Facendo clic sul nome dello scenario web si apre il modulo di configurazione dello scenario web. Se lo scenario web dell'host appartiene a un template, il nome del template viene visualizzato prima del nome dello scenario web come collegamento grigio. Facendo clic sul collegamento del template si apre l'elenco degli scenari web a livello di template.
Numero di passaggi	Numero di passaggi contenuti nello scenario.
Intervallo di aggiornamento	Frequenza di esecuzione dello scenario.
Tentativi	Numero di tentativi effettuati per eseguire i passaggi dello scenario web.
Autenticazione	Viene visualizzato il metodo di autenticazione: Basic, NTLM, Kerberos, Digest o None.
Proxy HTTP	Visualizza il proxy HTTP oppure "No" se non viene utilizzato.
Stato	Viene visualizzato lo stato dello scenario web: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.
Tag	Vengono visualizzati i tag dello scenario web. È possibile visualizzare fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di visualizzare tutti i tag al passaggio del mouse.
Informazioni	Se tutto funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". Passare il mouse sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare un nuovo scenario web, fare clic sul pulsante *Crea scenario web* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

- *Enable* - cambia lo stato dello scenario in *Enabled*
- *Disable* - cambia lo stato dello scenario in *Disabled*

- *Clear history and trends* - cancella i dati di history e trend per gli scenari
- *Delete* - elimina gli web scenario

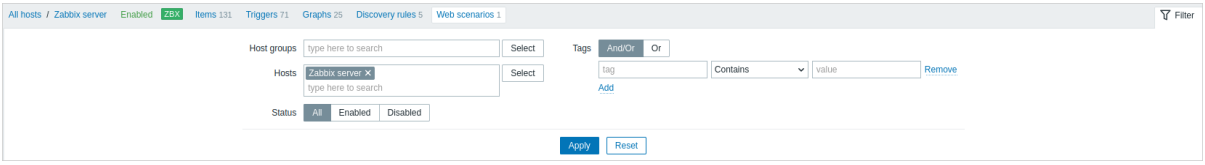
Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi web scenario, quindi fai clic sul pulsante desiderato.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo gli scenari di interesse.

Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il link *Filter* è disponibile sopra l'elenco degli scenari web. Facendovi clic si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.



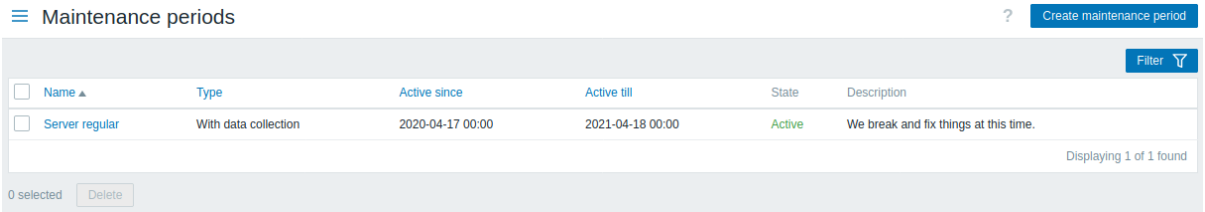
Parameter	Description
Host groups	Filtra per uno o più gruppi di host. La specifica di un gruppo di host padre seleziona implicitamente tutti i gruppi di host nidificati.
Hosts	Filtra per uno o più host. Se sopra sono già stati selezionati dei gruppi di host, la selezione degli host è limitata a tali gruppi.
Status	Filtra in base allo stato dello scenario web.
Tags	Filtra per nome e valore del tag dello scenario web. È possibile includere ed escludere tag e valori di tag specifici. È possibile impostare più condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole. Sono disponibili diversi operatori per ciascuna condizione: **Exists** - include i nomi dei tag specificati; **Equals** - include i nomi e i valori dei tag specificati (sensibile alle maiuscole); **Contains** - include i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole); **Does not exist** - esclude i nomi dei tag specificati; **Does not equal** - esclude i nomi e i valori dei tag specificati (sensibile alle maiuscole); **Does not contain** - esclude i nomi dei tag specificati in cui i valori del tag contengono la stringa inserita (corrispondenza di sottostringa, non sensibile alle maiuscole). Esistono due tipi di calcolo per le condizioni: **And/Or** - tutte le condizioni devono essere soddisfatte, le condizioni con lo stesso nome di tag verranno raggruppate dalla condizione Or; **Or** - è sufficiente che una condizione sia soddisfatta.

Manutenzione

Panoramica

Nella sezione *Raccolta dati* > *Manutenzione*, gli utenti possono configurare e gestire i periodi di manutenzione per gli host.

Viene visualizzato un elenco dei periodi di manutenzione esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del periodo di manutenzione. Facendo clic sul nome del periodo di manutenzione si apre il modulo di configurazione del periodo di manutenzione.
Tipo	Viene visualizzato il tipo di manutenzione: *Con raccolta dati* o *Senza raccolta dati*
Attivo dal	Data e ora a partire dalle quali l'esecuzione dei periodi di manutenzione diventa attiva. Nota: questo orario non attiva un periodo di manutenzione; i periodi di manutenzione devono essere impostati separatamente.
Attivo fino al	Data e ora fino alle quali l'esecuzione dei periodi di manutenzione rimane attiva.
Stato	Stato del periodo di manutenzione: **In arrivo** - diventerà attivo a breve **Attivo** - è attivo **Scaduto** - non è più attivo
Descrizione	Viene visualizzata la descrizione del periodo di manutenzione.

Per configurare un nuovo periodo di manutenzione, fare clic sul pulsante *Crea periodo di manutenzione* nell'angolo superiore destro.

Opzioni di modifica in massa

Un pulsante sotto l'elenco offre una opzione di modifica in massa:

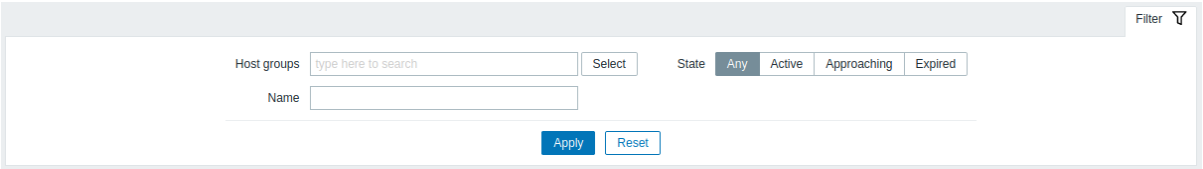
- *Delete* - elimina i periodi di manutenzione

Per usare questa opzione, seleziona le caselle di controllo davanti ai rispettivi periodi di manutenzione e fai clic su *Delete*.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo i periodi di manutenzione di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Il collegamento *Filter* è disponibile sopra l'elenco dei periodi di manutenzione. Se si fa clic su di esso, diventa disponibile un filtro con cui è possibile filtrare i periodi di manutenzione per host group, nome e stato.



Calcolo delle code durante la manutenzione

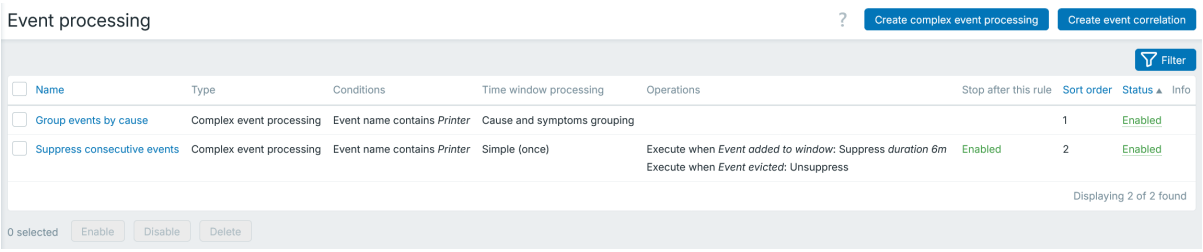
Attention:
Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calcolo delle code durante la manutenzione per i dettagli.

Event processing

Overview

In the *Data collection* > *Event processing* section users can configure and maintain:

- Complex event processing rules (CEP rules)
- Global event correlation rules (deprecated)



Displayed data:

Column	Description
Name	Name of the rule. Clicking the name opens the rule configuration form - for CEP rule or global correlation rule respectively.
Type	Type of the rule - *Complex event processing* or *Correlation*.
Conditions	Rule conditions are displayed.
Time window processing	Time window type is displayed.
Operations	Rule operations are displayed.
Stop after this rule	Stop execution after this rule - *Enabled* or *Disabled*.
Sort order	Sort order of the CEP rule.
Status	Rule status is displayed - *Enabled* or *Disabled*. By clicking the status you can change it.
Info	If everything is working correctly, no icon is displayed in this column. In case of errors, a square icon with the letter "i" is displayed. Hover over the icon to see a tooltip with the error description.

To configure a new CEP rule, click the *Create complex event processing* button in the upper-right corner.

To configure a new correlation rule, click the *Create event correlation* button in the upper-right corner.

Mass editing options

Buttons below the list offer some mass-editing options:

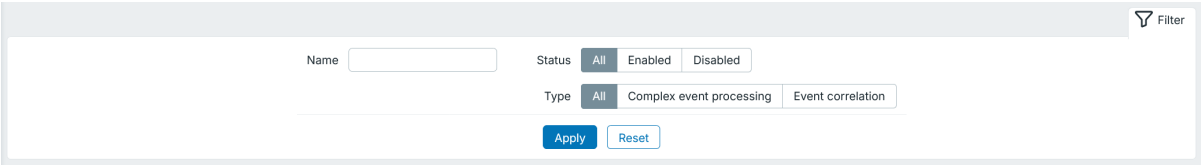
- *Enable* - change the rule status to *Enabled*
- *Disable* - change the rule status to *Disabled*
- *Delete* - delete the rules

To use these options, mark the checkboxes before the respective rules, then click the required button.

Using filter

You can use the filter to display only the rules you are interested in. For better search performance, data is searched with macros unresolved.

The *Filter* link is available above the list of CEP/correlation rules. If you click it, a filter becomes available where you can filter rules by name and status.

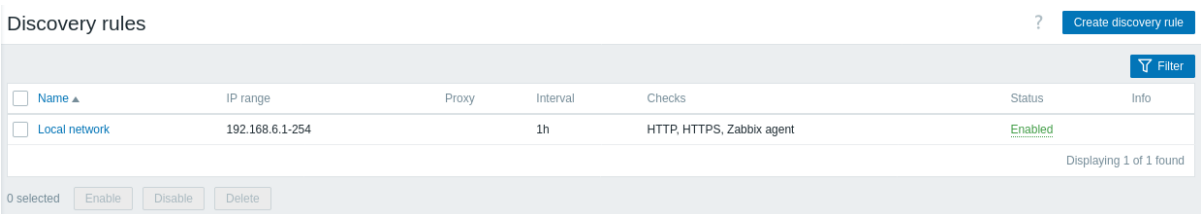


Scoperta

Panoramica

Nella sezione *Raccolta dati* > *Discovery*, gli utenti possono configurare e gestire le regole di discovery.

Viene visualizzato un elenco delle regole di discovery esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome della regola di discovery. Facendo clic sul nome della regola di discovery si apre il modulo di configurazione della regola di discovery.
Intervallo IP	Viene visualizzato l'intervallo di indirizzi IP da utilizzare per la scansione della rete.
Proxy	Viene visualizzato il nome del proxy, se la discovery viene eseguita dal proxy.
Intervallo	Viene visualizzata la frequenza con cui viene eseguita la discovery.
Controlli	Vengono visualizzati i tipi di controlli utilizzati per la discovery.

Colonna	Descrizione
Stato	Viene visualizzato lo stato della regola di discovery: *Abilitata* o *Disabilitata*. Facendo clic sullo stato è possibile modificarlo.
Informazioni	Se tutto funziona correttamente, questa colonna non visualizza nulla. In caso di errori, viene visualizzata un'icona rossa con la lettera "i". Passare il mouse sull'icona per visualizzare un tooltip con la descrizione dell'errore.

Per configurare una nuova regola di discovery, fare clic sul pulsante *Crea regola di discovery* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

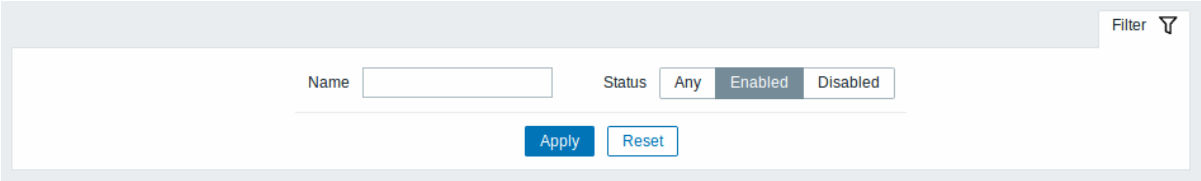
- *Abilita* - cambia lo stato della regola di discovery in *Abilitato*
- *Disabilita* - cambia lo stato della regola di discovery in *Disabilitato*
- *Elimina* - elimina le regole di discovery

Per usare queste opzioni, seleziona le caselle di controllo davanti alle rispettive regole di discovery, quindi fai clic sul pulsante desiderato.

Utilizzo del filtro

È possibile usare il filtro per visualizzare solo le regole di discovery che interessano. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con i macro non risolti.

Il link *Filter* è disponibile sopra l'elenco delle regole di discovery. Se lo si fa clic, diventa disponibile un filtro con cui è possibile filtrare le regole di discovery per nome e stato.



Avvisi

Panoramica

Questo menu presenta sezioni relative alla configurazione avvisi in Zabbix.

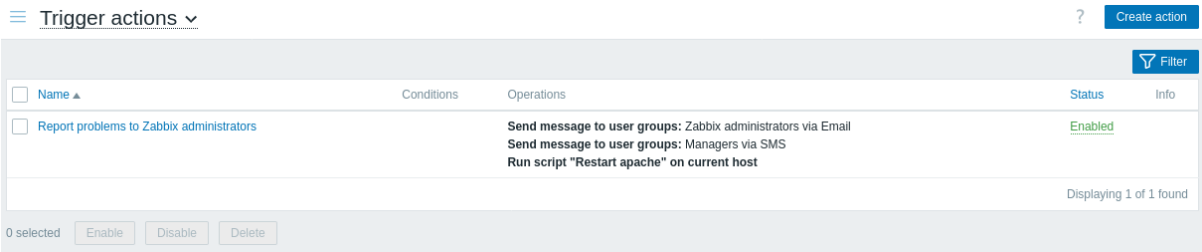
Azioni

Panoramica

Nella sezione *Avvisi* > *Azioni* gli utenti possono configurare e gestire le azioni.

Le azioni visualizzate sono quelle assegnate alla sorgente di eventi selezionata (trigger, servizi, discovery, autoregistrazione, azioni interne).

Per passare a una sorgente di eventi diversa, fare clic su *Azioni* nella sezione del menu *Avvisi*. È anche possibile passare da una sorgente all'altra utilizzando il menu a discesa del titolo nell'angolo superiore sinistro.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome dell'azione. Facendo clic sul nome dell'azione si apre il modulo di configurazione dell'azione.
Condizioni	Vengono visualizzate le condizioni dell'azione.
Operazioni	Vengono visualizzate le operazioni dell'azione. L'elenco delle operazioni mostra anche il tipo di supporto (e-mail, SMS o script) utilizzato per la notifica, nonché il nome e il cognome (tra parentesi dopo il nome utente) del destinatario della notifica. L'operazione dell'azione può essere una notifica o un comando remoto, a seconda del tipo di operazione selezionato.
Stato	Viene visualizzato lo stato dell'azione: *Abilitata* o *Disabilitata*. Facendo clic sullo stato è possibile modificarlo. Per ulteriori informazioni su cosa accade se un'azione viene disabilitata durante un'escalation in corso, vedere la sezione Escalation.
Informazioni	Se tutto funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, ad esempio operazioni o condizioni dell'azione mancanti dopo un aggiornamento o una discovery di basso livello, viene visualizzata un'icona di avviso. Passare il mouse sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare una nuova azione, fare clic sul pulsante *Crea azione* nell'angolo superiore destro.

Per gli utenti senza diritti di Super admin, le azioni vengono visualizzate in base alle impostazioni delle autorizzazioni. Ciò significa che in alcuni casi un utente senza diritti di Super admin non può visualizzare l'elenco completo delle azioni a causa di determinate limitazioni delle autorizzazioni. Un'azione viene visualizzata all'utente senza diritti di Super admin se sono soddisfatte le seguenti condizioni:

- L'utente dispone dell'accesso in lettura-scrittura ai gruppi di host, agli host, ai template e ai trigger nelle condizioni dell'azione
- L'utente dispone dell'accesso in lettura-scrittura ai gruppi di host, agli host e ai template nelle operazioni dell'azione, nelle operazioni di ripristino e nelle operazioni di aggiornamento
- L'utente dispone dell'accesso in lettura ai gruppi di utenti e agli utenti nelle operazioni dell'azione, nelle operazioni di ripristino e nelle operazioni di aggiornamento

Opzioni di modifica in massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica in massa:

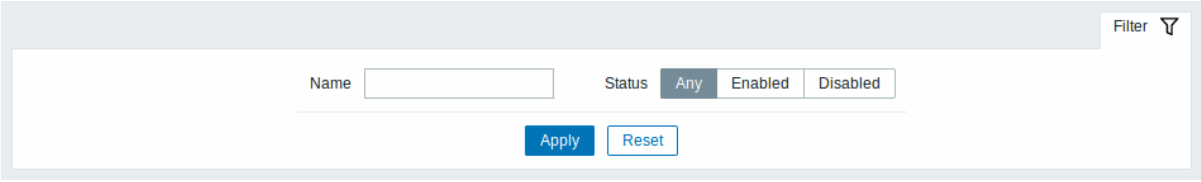
- *Enable* - cambia lo stato dell'azione in *Enabled*
- *Disable* - cambia lo stato dell'azione in *Disabled*
- *Delete* - elimina le azioni

Per usare queste opzioni, seleziona le caselle di controllo prima delle rispettive azioni, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

Puoi utilizzare il filtro per visualizzare solo le azioni che ti interessano. Per migliori prestazioni di ricerca, i dati vengono cercati con le macro irrisolto.

Il link *Filtro* è disponibile sopra l'elenco delle azioni. Se clicchi su di esso, diventa disponibile un filtro in cui è possibile filtrare le azioni per nome e stato.



Tipi di media

Panoramica

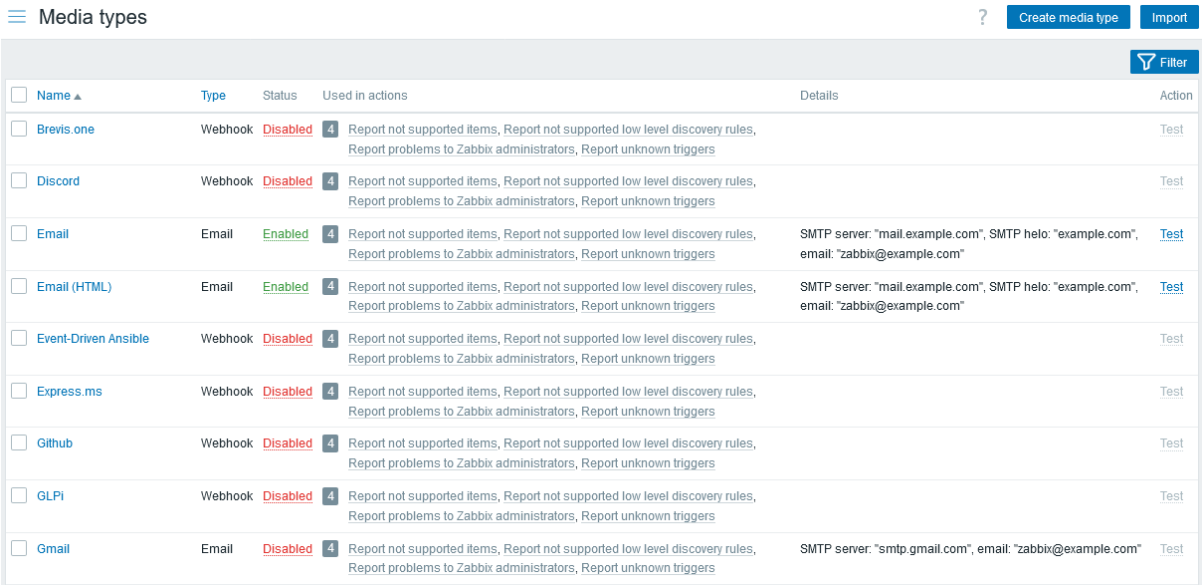
Nella sezione *Avvisi* > *Tipi di supporto*, gli utenti possono configurare e gestire le informazioni sui tipi di supporto.

Questo menu è disponibile solo per gli utenti di tipo SuperAdmin.

Le informazioni sui tipi di supporto contengono istruzioni generali per utilizzare un supporto come canale di distribuzione per le notifiche. I dettagli specifici, come i singoli indirizzi email a cui inviare una notifica, vengono conservati nei singoli utenti.

Viene visualizzato un elenco dei tipi di supporto esistenti con i relativi dettagli.

Note:
Se i metodi di distribuzione (*Email*, *SMS*, *Webhook*, *Script* o *Push*) sono specificati nel flag delle funzionalità `$ZBX_FEATURE_FLAGS['media_type_denylist']` nel file di configurazione del frontend, i tipi di supporto preconfigurati corrispondenti vengono disabilitati in questa sezione.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del tipo di supporto. Facendo clic sul nome si apre il modulo di configurazione del tipo di supporto.
Tipo	Viene visualizzato il tipo di supporto (email, SMS e così via).
Stato	Viene visualizzato lo stato del tipo di supporto: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.
Utilizzato nelle azioni	Vengono visualizzate le azioni in cui viene utilizzato il tipo di supporto, precedute dal numero totale di tali azioni. Facendo clic sul nome dell'azione si apre il modulo di configurazione dell'azione. Se l'utente non dispone delle autorizzazioni per l'azione, il nome non è selezionabile.
Dettagli	Vengono visualizzate informazioni dettagliate sul tipo di supporto.
Azioni	È disponibile la seguente azione: **Test** - fare clic per aprire un modulo di test in cui è possibile inserire i parametri del tipo di supporto (ad esempio, un indirizzo del destinatario con oggetto e corpo del test) e inviare un messaggio di test per verificare che il tipo di supporto configurato funzioni. Vedere anche: Test del tipo di supporto.

Per configurare un nuovo tipo di supporto, fare clic sul pulsante *Crea tipo di supporto* nell'angolo superiore destro.

Per importare un tipo di supporto, fare clic sul pulsante *Importa* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

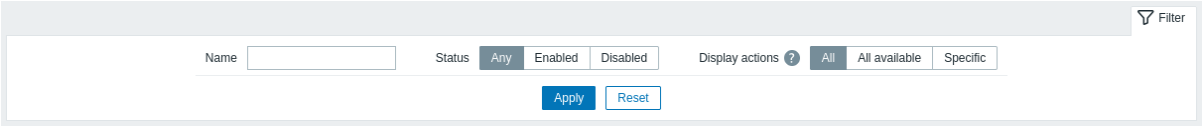
- *Enable* - cambia lo stato del media type in *Enabled*
- *Disable* - cambia lo stato del media type in *Disabled*
- *Export* - esporta i media type in un file YAML, XML o JSON
- *Delete* - elimina i media type

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi media type, quindi fai clic sul pulsante desiderato.

Uso del filtro

È possibile usare il filtro per visualizzare solo i tipi di media di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Il link *Filter* è disponibile sopra l'elenco dei tipi di media. Facendo clic su di esso, diventa disponibile un filtro con cui è possibile filtrare i tipi di media per nome e stato. Inoltre, è possibile usare il filtro per visualizzare le azioni nella colonna *Used in actions* in base all'ambito di utilizzo del relativo tipo di media (definito dal parametro *Send to media type* nei dettagli dell'operazione dell'azione).



Script

Panoramica

Nella sezione *Avvisi* > *Script* è possibile configurare e gestire gli script globali definiti dall'utente.

Questo menu è disponibile solo per gli utenti di tipo SuperAdmin.

Gli script globali, in base all'ambito configurato e alle autorizzazioni dell'utente, sono disponibili per l'esecuzione:

- Dal menu dell'host in varie sezioni del frontend (*Dashboard*, *Problemi*, *Ultimi dati*, *Mappe*, ecc.).
- Dal menu dell'evento.
- Possono essere eseguiti come operazione di un'azione.

Gli script vengono eseguiti da Zabbix agent, Zabbix server (proxy) o solo da Zabbix server. Vedere anche Esecuzione dei comandi.

I comandi remoti sono disabilitati per impostazione predefinita sia su Zabbix agent sia su Zabbix proxy. Possono essere abilitati come segue:

- Per i comandi remoti eseguiti da Zabbix agent, aggiungere un parametro `AllowKey=system.run[<command>,*]` o `AllowKeyRegexp=^system\.run\[<command>,.*\]$` per ogni comando consentito nella configurazione dell'agent, dove `*` corrisponde a una delle modalità di esecuzione (`wait` o `nowait`).
- Per i comandi remoti eseguiti da Zabbix proxy, impostare il parametro `EnableRemoteCommands` su 1 nella configurazione del proxy. Si noti che non è necessario abilitare i comandi remoti su Zabbix proxy quando vengono eseguiti da Zabbix agent su un host monitorato dal proxy.

Attention:
Non utilizzare `AllowKey=system.run[*]` o `AllowKeyRegexp=^system\.run\[.*\]$` se non necessario, poiché consentono l'esecuzione di qualsiasi comando sull'host. Per motivi di sicurezza, si consiglia di consentire solo comandi esatti, mentre `system.run[*]` è negato per impostazione predefinita (`DenyKey=system.run[*]`).

L'esecuzione degli script globali su Zabbix server può essere disabilitata impostando `EnableGlobalScripts=0` nella configurazione del server. Per le nuove installazioni a partire da Zabbix 7.0, l'esecuzione degli script globali su Zabbix server è disabilitata per impostazione predefinita.

Viene visualizzato un elenco degli script esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome dello script. Facendo clic sul nome dello script si apre il modulo di configurazione dello script.

Colonna	Descrizione
Ambito	Ambito dello script: operazione di un'azione, azione manuale sull'host o azione manuale sull'evento. Questa impostazione determina dove è disponibile lo script: nei comandi remoti delle operazioni di un'azione, dal menu dell'host o dal menu dell'evento, rispettivamente.
Utilizzato nelle azioni	Vengono visualizzate tutte le azioni in cui viene utilizzato lo script, precedute dal numero totale di tali azioni. Facendo clic sul nome dell'azione si apre il modulo di configurazione dell'azione. Se l'utente non dispone delle autorizzazioni per l'azione, il nome non è selezionabile.
Tipo	Viene visualizzato il tipo di script: comando *URL*, *Webhook*, *Script*, *SSH*, *Telnet* o *IPMI*.
Esegui su	Indica se lo script verrà eseguito su Zabbix agent, Zabbix proxy o server, oppure solo su Zabbix server.
Comandi	Vengono visualizzati tutti i comandi da eseguire nello script. Per i webhook non viene visualizzato nulla.
Gruppo utenti	Viene visualizzato il gruppo di utenti per il quale è disponibile lo script (oppure *Tutti* per tutti i gruppi di utenti).
Gruppo di host	Viene visualizzato il gruppo di host per il quale è disponibile lo script (oppure *Tutti* per tutti i gruppi di host).
Accesso all'host	Viene visualizzato il livello di autorizzazione per il gruppo di host: *Lettura* o *Scrittura*. Solo gli utenti con il livello di autorizzazione richiesto potranno eseguire lo script.

Per configurare un nuovo script, fare clic sul pulsante *Crea script* nell'angolo superiore destro.

Opzioni di modifica massiva

Un pulsante sotto l'elenco offre una opzione di modifica massiva:

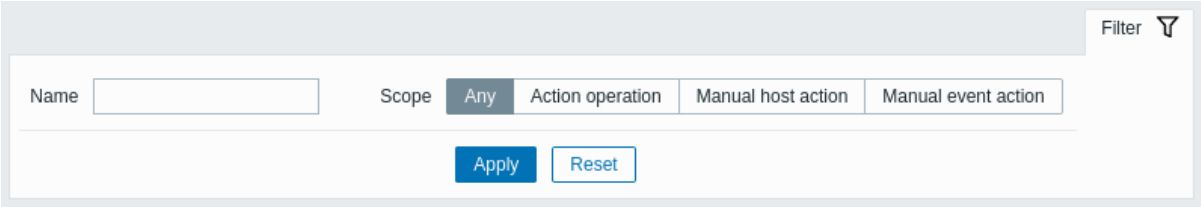
- *Elimina* - elimina gli script

Per usare questa opzione, seleziona le caselle di controllo accanto ai rispettivi script e fai clic su *Elimina*.

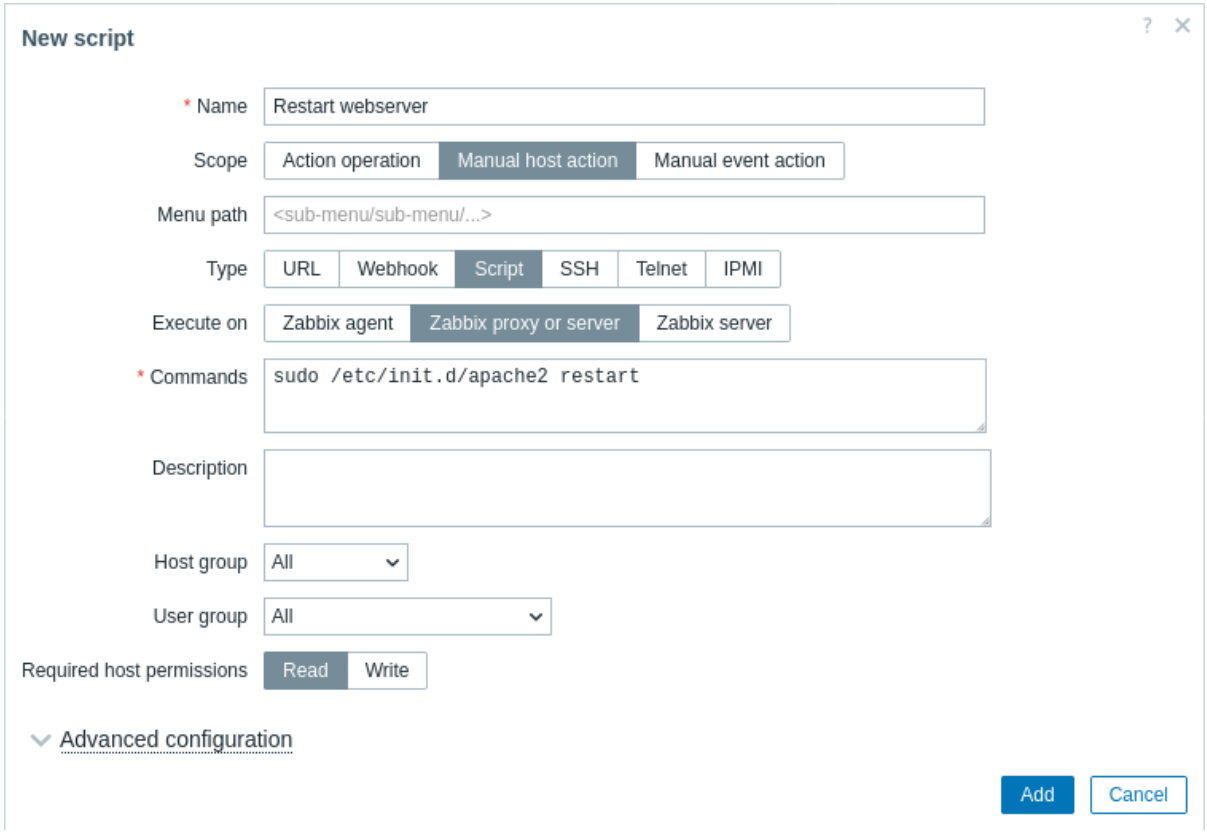
Uso del filtro

È possibile usare il filtro per visualizzare solo gli script di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il link *Filter* è disponibile sopra l'elenco degli script. Se lo si fa clic, diventa disponibile un filtro in cui è possibile filtrare gli script per nome e ambito.



Configurazione di uno script globale



Note:
Si consiglia di utilizzare le funzioni macro invece delle macro semplici negli script globali per migliorare la sicurezza, poiché le macro non vengono sottoposte automaticamente a escape.

Attributi dello script:

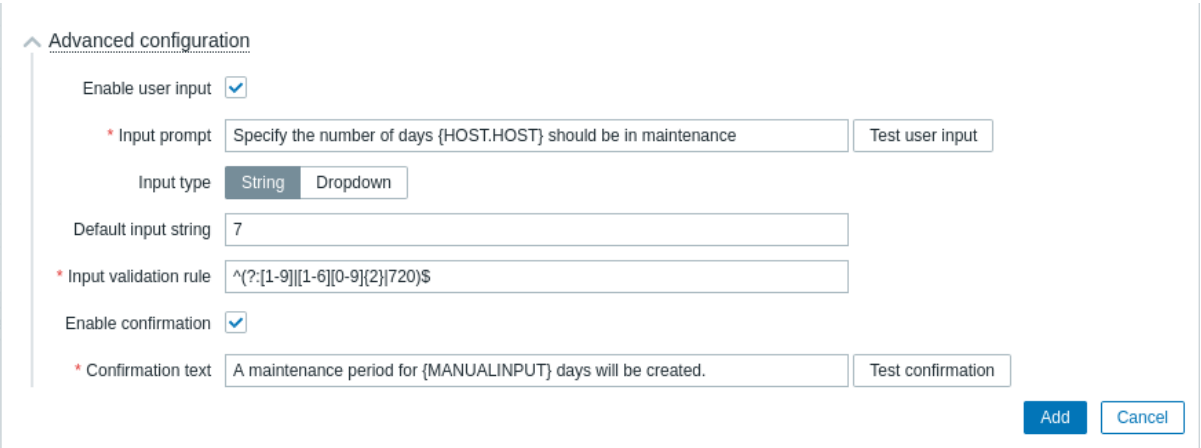
Parametro	Descrizione
Nome	Nome univoco dello script. Ad esempio, `Clear /tmp filesystem`
Ambito	Ambito dello script: operazione di azione, azione manuale sull'host o azione manuale sull'evento. Questa impostazione determina dove può essere utilizzato lo script: nei comandi remoti delle operazioni dell'azione, dal menu dell'host o dal menu dell'evento, rispettivamente. Impostando l'ambito su "Operazione di azione", lo script diventa disponibile per tutti gli utenti con accesso a *Avvisi* > *Azioni*. Se uno script viene effettivamente utilizzato in un'azione, il suo ambito non può essere modificato da "operazione di azione". **Supporto delle macro** L'ambito influisce sull'insieme di macro disponibili. Ad esempio, le macro relative all'utente (`{USER.*}`) sono supportate negli script per consentire il passaggio di informazioni sull'utente che ha avviato lo script. Tuttavia, non sono supportate se l'ambito dello script è "operazione di azione", poiché le operazioni di azione vengono eseguite automaticamente. Una macro `{MANUALINPUT}` consente di specificare un input manuale al momento dell'esecuzione dello script. È supportata dagli script di azione manuale sull'host e di azione manuale sull'evento. Per sapere quali altre macro sono supportate, cercare "Notifiche e comandi basati sui trigger/Comandi basati sui trigger", "Script di azione manuale sull'host" e "Script di azione manuale sull'evento" nella tabella delle macro supportate. Se una macro può risolversi in un valore contenente spazi, ad esempio il nome dell'host, non dimenticare di racchiuderla tra virgolette quando necessario.

Parametro	Descrizione
Percorso del menu	Percorso del menu desiderato per lo script. Ad esempio, `Default` o `Default/` visualizzerà lo script nella directory corrispondente. I menu possono essere nidificati, ad esempio `Menu principale/Sottomenu1/Sottomenu2`. Quando si accede agli script tramite il menu dell'host/evento nelle sezioni di monitoraggio, saranno organizzati in base alle directory specificate. Questo campo viene visualizzato solo se come *Ambito* è selezionato "Azione manuale sull'host" o "Azione manuale sull'evento".
Tipo	Fare clic sul pulsante corrispondente per selezionare il tipo di script: **URL**, **Webhook**, **Script**, **SSH**, **Telnet** o comando **IPMI**. Il tipo **URL** è disponibile solo quando come *Ambito* è selezionato "Azione manuale sull'host" o "Azione manuale sull'evento".
Tipo di script: **URL**	
URL	Specificare l'URL per l'accesso rapido dal menu dell'host o dal menu dell'evento. Le macro e le macro utente personalizzate sono supportate. Il supporto delle macro dipende dall'ambito dello script (vedere *Ambito* sopra). Utilizzare la macro {MANUALINPUT} in questo campo per poter specificare un input manuale al momento dell'esecuzione dello script, ad esempio: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` I valori delle macro non devono essere codificati nell'URL.
Apri in una nuova finestra	Determina se l'URL deve essere aperto in una scheda del browser nuova o corrente.
Tipo di script: **Webhook**	
Parametri	Specificare le variabili del webhook come coppie attributo-valore. Vedere anche: configurazione del supporto Webhook. Nei valori dei parametri sono supportate le macro e le macro utente personalizzate. Il supporto delle macro dipende dall'ambito dello script (vedere *Ambito* sopra).
Script	Inserire il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona a forma di matita accanto a esso. Il supporto delle macro dipende dall'ambito dello script (vedere *Ambito* sopra). Vedere anche: configurazione del supporto Webhook, Oggetti JavaScript aggiuntivi.
Timeout	Timeout di esecuzione JavaScript (1-60 s, valore predefinito 30 s). Sono supportati i suffissi temporali, ad esempio 30s, 1m.
Tipo di script: **Script**	
Esegui su	Fare clic sul pulsante corrispondente per eseguire lo script shell su: **Zabbix agent**: lo script verrà eseguito da Zabbix agent (se l'item system.run è consentito) sull'host **Zabbix proxy o server**: lo script verrà eseguito da Zabbix proxy o server, a seconda che l'host sia monitorato da un proxy o dal server. Verrà eseguito sul proxy se abilitato da EnableRemoteCommands. Verrà eseguito sul server se gli script globali sono abilitati dal parametro del server EnableGlobalScripts. **Zabbix server**: lo script verrà eseguito solo da Zabbix server. Questa opzione non sarà disponibile se gli script globali sono disabilitati dal parametro del server EnableGlobalScripts.
Comandi	Inserire il percorso completo dei comandi da eseguire nello script. Il supporto delle macro dipende dall'ambito dello script (vedere *Ambito* sopra). Sono supportate le macro utente personalizzate.
Tipo di script: **SSH**	
Metodo di autenticazione	Selezionare il metodo di autenticazione: password o chiave pubblica.
Nome utente	Inserire il nome utente.
Password	Inserire la password. Questo campo è disponibile se come metodo di autenticazione è selezionato "Password".
File della chiave pubblica	Inserire il percorso del file della chiave pubblica. Questo campo è disponibile se come metodo di autenticazione è selezionato "Chiave pubblica".
File della chiave privata	Inserire il percorso del file della chiave privata. Questo campo è disponibile se come metodo di autenticazione è selezionato "Chiave pubblica".

Parametro	Descrizione
Passphrase	Inserire la passphrase. Questo campo è disponibile se come metodo di autenticazione è selezionato ”Chiave pubblica”.
Porta	Inserire la porta del servizio SSH remoto sull’host di destinazione a cui Zabbix si connetterà.
Comandi	Inserire i comandi. Il supporto delle macro dipende dall’ambito dello script (vedere *Ambito* sopra). Sono supportate le macro utente personalizzate.
Tipo di script: **Telnet**	
Nome utente	Inserire il nome utente.
Password	Inserire la password.
Porta	Inserire la porta del servizio Telnet remoto sull’host di destinazione a cui Zabbix si connetterà.
Comandi	Inserire i comandi. Il supporto delle macro dipende dall’ambito dello script (vedere *Ambito* sopra). Sono supportate le macro utente personalizzate.
Tipo di script: **IPMI**	
Comando	Inserire il comando IPMI. Il supporto delle macro dipende dall’ambito dello script (vedere *Ambito* sopra). Sono supportate le macro utente personalizzate.
Descrizione	Inserire una descrizione dello script.
Gruppo di host	Selezionare il gruppo di host per il quale lo script sarà disponibile, oppure *Tutti* per tutti i gruppi di host.
Gruppo di utenti	Selezionare il gruppo di utenti per il quale lo script sarà disponibile, oppure *Tutti* per tutti i gruppi di utenti. Questo campo viene visualizzato solo se come *Ambito* è selezionato ”Azione manuale sull’host” o ”Azione manuale sull’evento”.
Autorizzazioni host richieste	Selezionare il livello di autorizzazione per il gruppo di host: *Lettura* o *Scrittura*. Solo gli utenti con il livello di autorizzazione richiesto potranno eseguire lo script. Questo campo viene visualizzato solo se come *Ambito* è selezionato ”Azione manuale sull’host” o ”Azione manuale sull’evento”.
Configurazione avanzata	Fare clic sull’intestazione *Configurazione avanzata* per visualizzare le opzioni di configurazione avanzata. Questo campo viene visualizzato solo se come *Ambito* è selezionato ”Azione manuale sull’host” o ”Azione manuale sull’evento”.

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata*:



Parametro	Descrizione
Abilita input dell’utente	Selezionare la casella di controllo per abilitare l’input manuale dell’utente prima dell’esecuzione dello script. L’input manuale dell’utente sostituirà il valore della macro {MANUALINPUT} nello script. Vedere anche: Input manuale dell’utente.

Parametro	Descrizione
Prompt di input	Inserire il testo personalizzato che richiede l'input dell'utente. Questo testo verrà visualizzato sopra il campo di input nella finestra popup *Input manuale*. Per visualizzare un'anteprima della finestra popup *Input manuale*, fare clic su *Testa input dell'utente*. L'anteprima consente inoltre di verificare se la stringa di input è conforme alla regola di convalida dell'input (vedere i parametri seguenti). Il supporto per macro e macro utente dipende dall'ambito dello script (vedere *Ambito* nei parametri generali di configurazione dello script).
Tipo di input	Selezionare il tipo di input manuale: **Stringa** - stringa singola; **Elenco a discesa** - il valore viene selezionato tra più opzioni dell'elenco a discesa.
Opzioni dell'elenco a discesa	Inserire i valori univoci per l'elenco a discesa dell'input dell'utente in un elenco delimitato da virgole. Per includere un'opzione vuota nell'elenco a discesa, aggiungere una virgola aggiuntiva all'inizio, al centro o alla fine dell'elenco. Questo campo viene visualizzato solo se *Elenco a discesa* è selezionato come *Tipo di input*.
Stringa di input predefinita	Inserire la stringa predefinita per l'input dell'utente (oppure nessuna). Questo campo verrà convalidato rispetto all'espressione regolare fornita nel campo *Regola di convalida dell'input*. Il valore inserito qui verrà visualizzato per impostazione predefinita nella finestra popup *Input manuale*. Questo campo viene visualizzato solo se *Stringa* è selezionato come *Tipo di input*.
Regola di convalida dell'input	Inserire un'espressione regolare per convalidare la stringa di input dell'utente. Sono supportate le espressioni regolari globali. Questo campo viene visualizzato solo se *Stringa* è selezionato come *Tipo di input*.
Abilita conferma	Selezionare la casella di controllo per visualizzare un messaggio di conferma prima dell'esecuzione dello script. Questa funzionalità può essere particolarmente utile per operazioni potenzialmente pericolose, come uno script di riavvio, o per operazioni che potrebbero richiedere molto tempo.
Testo di conferma	Inserire il testo personalizzato per la finestra popup di conferma abilitata con la casella di controllo precedente (ad esempio, *Il sistema remoto verrà riavviato. Continuare?*). Per visualizzare l'aspetto del testo, fare clic su *Testa conferma* accanto al campo. Le macro e le macro utente personalizzate sono supportate. *Nota:* le macro non verranno espanse durante il test del messaggio di conferma.

Se sono configurati sia l'input manuale dell'utente sia un messaggio di conferma, verranno visualizzati in finestre popup consecutive.

Input manuale dell'utente

L'input manuale dell'utente consente di fornire un parametro personalizzato a ogni esecuzione dello script. In questo modo non è necessario creare più script utente simili, che differiscono solo per un parametro.

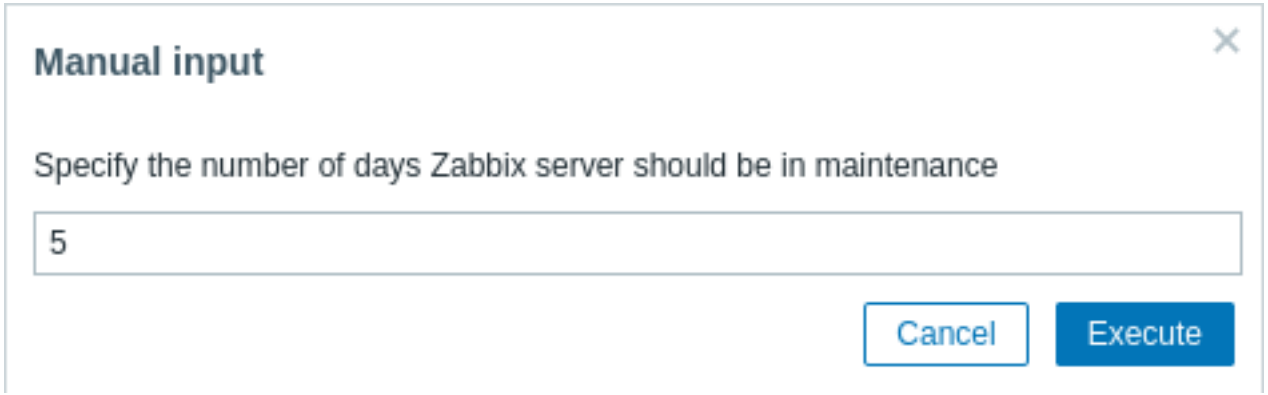
Ad esempio, è possibile fornire allo script un intero diverso o un indirizzo URL diverso durante l'esecuzione.

Per abilitare l'input manuale dell'utente:

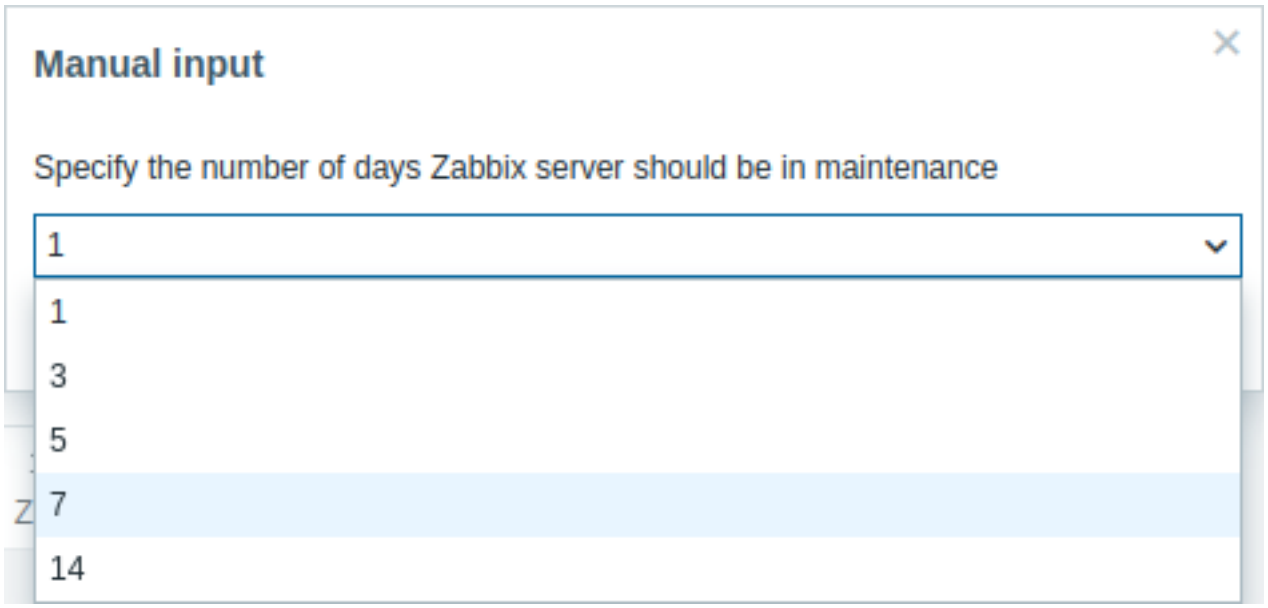
- utilizzare la macro {MANUALINPUT} nello script (comandi, script, parametro dello script) dove necessario; oppure nel campo URL degli script URL;
- nella configurazione avanzata dello script, abilitare l'input manuale dell'utente e configurare le opzioni di input.

Con l'input dell'utente abilitato, prima dell'esecuzione dello script, all'utente verrà mostrata una finestra popup *Input manuale* che richiede di fornire un valore personalizzato. Il valore fornito sostituirà {MANUALINPUT} nello script.

A seconda della configurazione, all'utente verrà richiesto di inserire un valore stringa:



Oppure di selezionare il valore da un elenco a discesa di opzioni predefinite:



L'input manuale dell'utente è disponibile solo per gli script il cui ambito è "Azione manuale sull'host" o "Azione manuale sull'evento".

Esecuzione dello script e risultato

Gli script eseguiti da Zabbix server vengono eseguiti nell'ordine descritto nella pagina sull'esecuzione dei comandi.

Il risultato dello script viene visualizzato in una finestra pop-up che appare dopo l'esecuzione dello script. Il valore restituito dallo script è l'output standard:

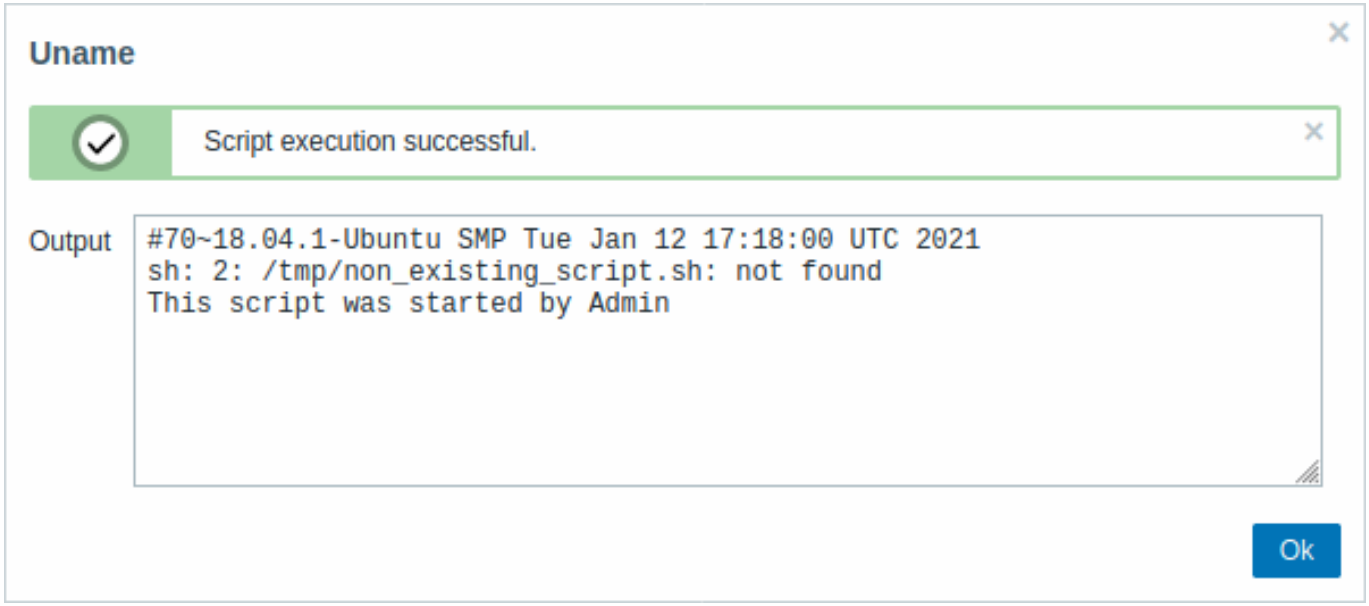
- Se lo script termina correttamente (codice di uscita 0), il valore restituito è limitato a 16 MB (inclusi gli spazi finali che vengono troncati).
- Se lo script termina con un errore (codice di uscita diverso da zero), il valore restituito è l'errore standard limitato a 2 KB.

Per impostazione predefinita, Zabbix non memorizza l'output esteso dello script. Per conservare i dettagli completi dell'output, è possibile implementare la registrazione all'interno dello script stesso (ad esempio, reindirizzando l'output a un file di log locale).

Si noti che per gli script eseguiti su Zabbix server o Zabbix proxy si applicano anche i limiti del database.

Di seguito è riportato un esempio di script e della finestra del risultato:

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



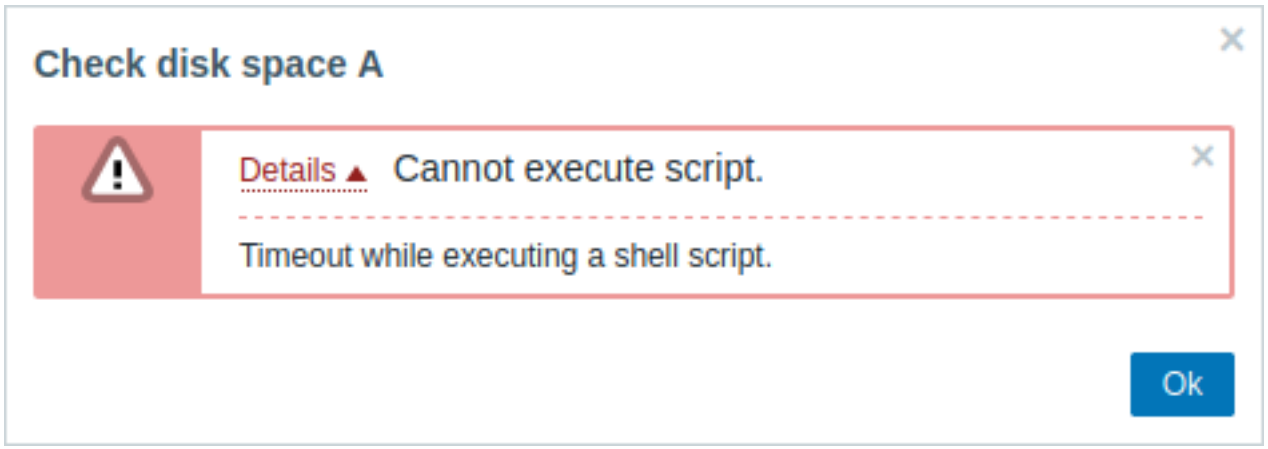
Il risultato dello script non mostra lo script stesso.

Timeout dello script sull'agent Zabbix

È possibile che si verifichi un timeout durante l'esecuzione di uno script.

Di seguito è riportato un esempio di script in esecuzione sull'agent Zabbix e della relativa finestra dei risultati:

```
sleep 5
df -h
```



In questo caso, il messaggio di errore è il seguente:

```
Timeout while executing a shell script.
```

Per evitare situazioni di questo tipo, si consiglia di ottimizzare lo script stesso (nell'esempio precedente, "5") invece di modificare il parametro `Timeout` nella configurazione dell'agent Zabbix e nella configurazione del server Zabbix.

Se il parametro `Timeout` è stato modificato nella configurazione dell'agent Zabbix, verrà visualizzato il seguente messaggio di errore:

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Ciò significa che la modifica è stata apportata nella configurazione dell'agent Zabbix, ma è necessario modificare il parametro `Timeout` anche nella configurazione del server Zabbix.

Timeout dello script sugli agent attivi

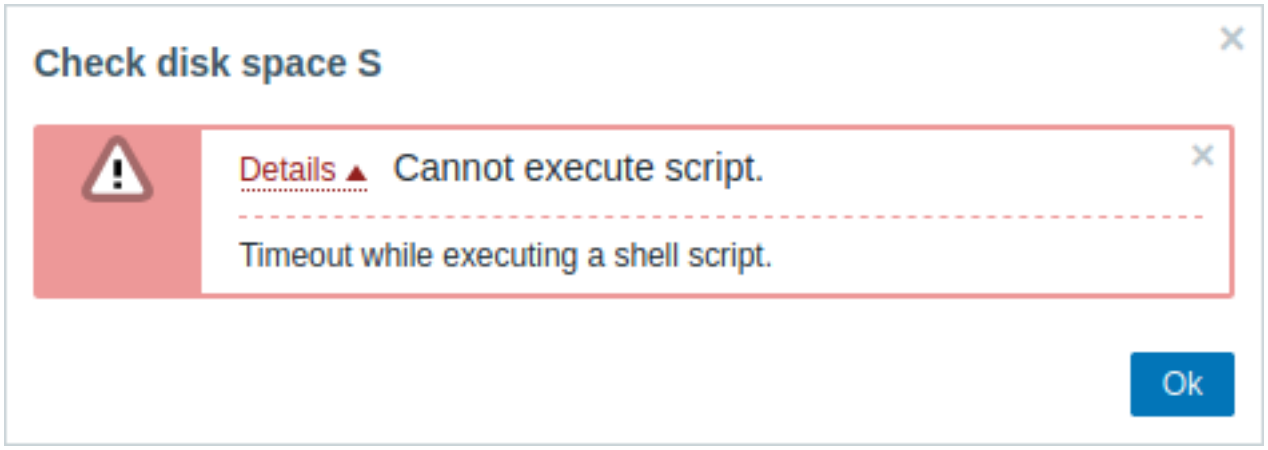
È possibile eseguire script solo sugli agent in modalità attiva. Una volta che l'esecuzione dello script viene avviata dall'operazione dell'azione o dall'esecuzione manuale dello script, il comando viene incluso nella configurazione dei controlli attivi ed eseguito non appena l'agent attivo lo riceve.

Gli script manuali vengono inviati all'agent attivo insieme al timeout di server/proxy per l'esecuzione dello script. Aumentare il timeout predefinito di server/proxy per l'esecuzione dello script. Il timeout deve essere superiore alla frequenza di aggiornamento dei controlli attivi, altrimenti il timeout viene superato prima che l'agent attivo riceva lo script e possa restituire il risultato.

Timeout dello script sul server/proxy Zabbix

Di seguito è riportato un esempio di script in esecuzione sul server Zabbix e la relativa finestra dei risultati:

```
sleep 11
df -h
```



È inoltre consigliabile ottimizzare lo script stesso, invece di modificare il parametro `TrapperTimeout` impostandolo su un valore corrispondente (nel nostro caso, > 11) modificando la configurazione del server Zabbix.

Utenti

Panoramica

Questo menu include sezioni relative alla configurazione degli utenti in Zabbix. Questo menu è disponibile solo per gli utenti del tipo SuperAdmin.

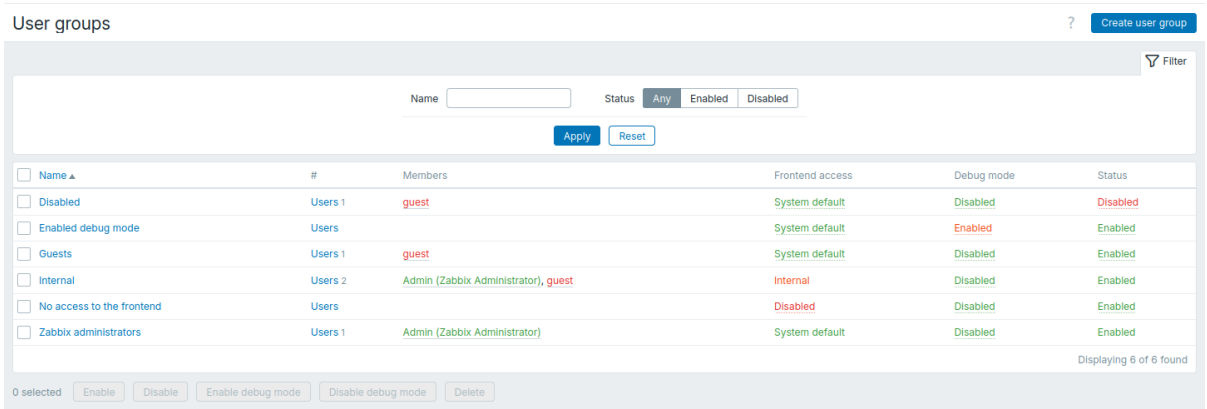
Gruppi di utenti

Panoramica

Nella sezione *Users* > *User groups* vengono gestiti i gruppi di utenti del sistema.

Gruppi di utenti

Viene visualizzato un elenco dei gruppi di utenti esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del gruppo di utenti. Facendo clic sul nome del gruppo di utenti si apre il modulo di configurazione del gruppo di utenti.
#	Numero di utenti nel gruppo. Facendo clic su *Utenti* verranno visualizzati i rispettivi utenti filtrati nell'elenco degli utenti.
Membri	Nomi utente dei singoli utenti nel gruppo di utenti (con nome e cognome tra parentesi). Facendo clic sul nome utente si apre il modulo di configurazione dell'utente. Gli utenti appartenenti a gruppi disabilitati vengono visualizzati in rosso.

Colonna	Descrizione
Accesso al frontend	Viene visualizzato il livello di accesso al frontend: **Predefinito di sistema** - gli utenti vengono autenticati da Zabbix, LDAP o HTTP (a seconda del metodo di autenticazione impostato globalmente); **Interno** - gli utenti vengono autenticati da Zabbix; ignorato se l'autenticazione HTTP è il valore predefinito globale; **LDAP** - gli utenti vengono autenticati da LDAP; ignorato se l'autenticazione HTTP è il valore predefinito globale; **Disabilitato** - l'accesso al frontend di Zabbix è vietato per questo gruppo. Facendo clic sul livello corrente, è possibile modificarlo.
Modalità debug	Viene visualizzato lo stato della modalità debug - *Abilitata* o *Disabilitata*. Facendo clic sullo stato, è possibile modificarlo.
Stato	Viene visualizzato lo stato del gruppo di utenti - *Abilitato* o *Disabilitato*. Facendo clic sullo stato, è possibile modificarlo.

Per configurare un nuovo gruppo di utenti, fare clic sul pulsante *Crea gruppo di utenti* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

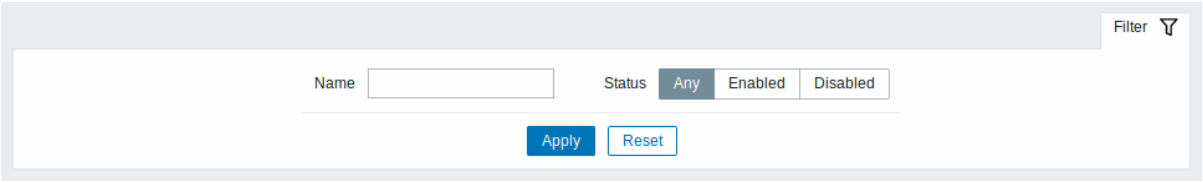
- *Enable* - cambia lo stato del gruppo utente in *Enabled*
- *Disable* - cambia lo stato del gruppo utente in *Disabled*
- *Enable debug mode* - abilita la modalità di debug per i gruppi utente
- *Disable debug mode* - disabilita la modalità di debug per i gruppi utente
- *Delete* - elimina i gruppi utente

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi gruppi utente, quindi fai clic sul pulsante richiesto.

Uso del filtro

È possibile usare il filtro per visualizzare solo i gruppi utente di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

Il collegamento *Filter* è disponibile sopra l'elenco dei gruppi utente. Se si fa clic su di esso, diventa disponibile un filtro con cui è possibile filtrare i gruppi utente per nome e stato.



Ruoli utente

Panoramica

Nella sezione *Utenti* > *Ruoli utente* è possibile creare ruoli utente.

I ruoli utente consentono di creare autorizzazioni dettagliate basate sul tipo di utente selezionato inizialmente (*Utente*, *Amministratore*, *Super amministratore*).

Dopo aver selezionato un tipo di utente, vengono concesse tutte le autorizzazioni disponibili per questo tipo di utente (selezionate per impostazione predefinita).

Le autorizzazioni possono essere revocate solo dal sottoinsieme disponibile per il tipo di utente; non possono essere estese oltre quelle disponibili per il tipo di utente.

Le caselle di controllo relative alle autorizzazioni non disponibili sono disabilitate; gli utenti non potranno accedere all'elemento nemmeno inserendo nel browser un URL diretto a tale elemento.

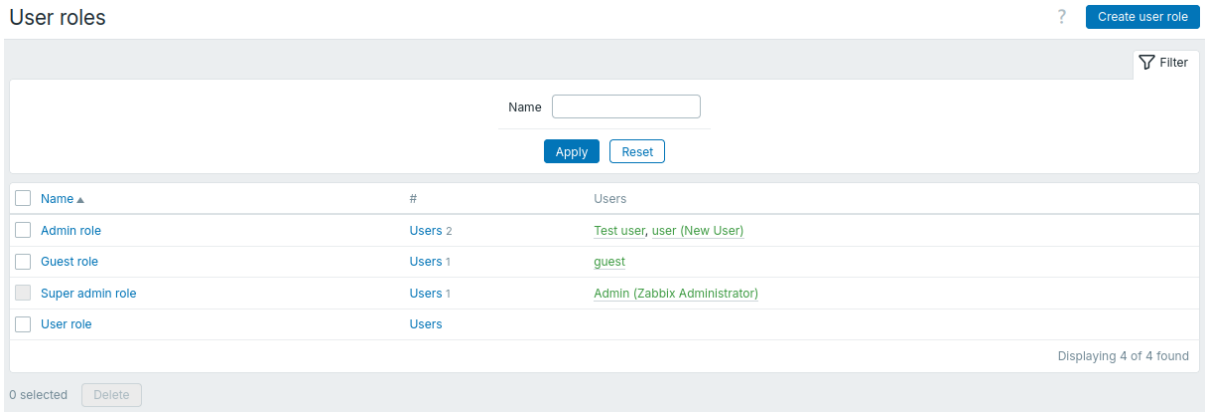
> **Note:**
> La limitazione dell'accesso ad alcuni elementi dell'interfaccia utente impedisce solo l'apertura della pagina; non rimuove la possibilità di accedere ai dati sottostanti da altre parti dell'interfaccia.

I ruoli utente possono essere assegnati agli utenti del sistema. A ogni utente può essere assegnato un solo ruolo.

Ruoli utente predefiniti

Per impostazione predefinita, Zabbix è configurato con quattro ruoli utente, che dispongono di un insieme predefinito di autorizzazioni:

- Ruolo Guest
- Ruolo User
- Ruolo Admin
- Ruolo Super admin



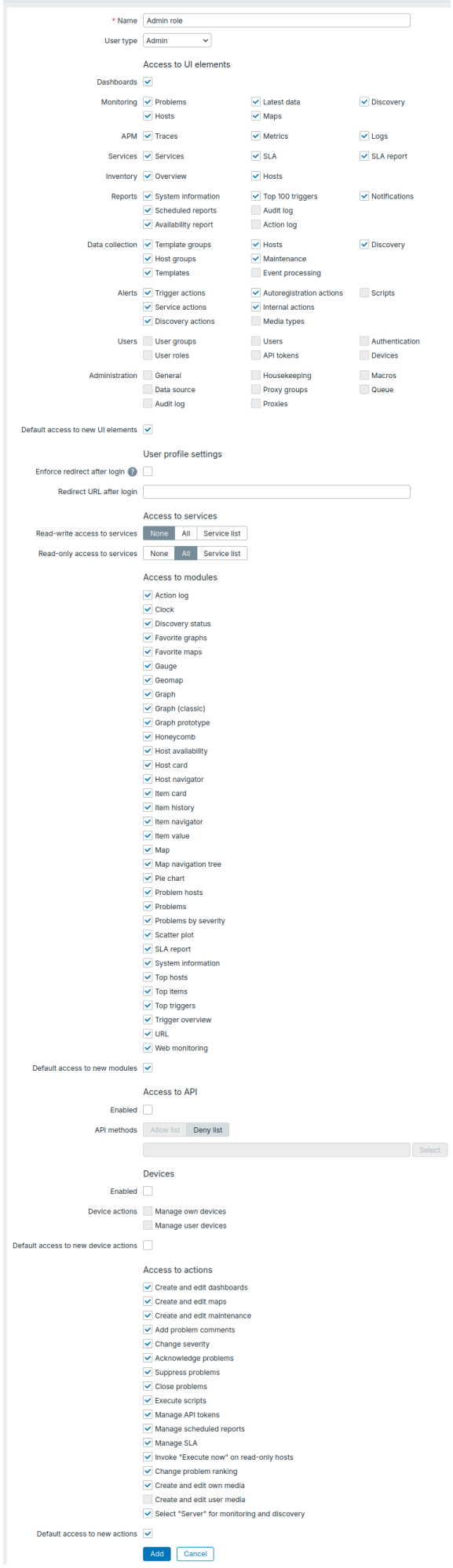
Questi ruoli si basano sui principali tipi di utente in Zabbix. Viene visualizzato l'elenco di tutti gli utenti assegnati al rispettivo ruolo. Gli utenti inclusi nei gruppi disabilitati sono indicati in rosso. Il *Ruolo Guest* è un ruolo di tipo utente con le sole autorizzazioni per visualizzare alcune sezioni del frontend.

Note:
Il *Ruolo Super admin* predefinito non può essere modificato o eliminato, perché in Zabbix deve esistere almeno un utente Super admin con privilegi illimitati. Gli utenti di tipo *Super admin* possono modificare le impostazioni del proprio ruolo, ma non il tipo di utente.

Configurazione

Per creare un nuovo ruolo, fare clic sul pulsante *Crea ruolo utente* nell'angolo superiore destro. Per aggiornare un ruolo esistente, fare clic sul nome del ruolo per aprire il modulo di configurazione.

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

Vengono visualizzate le autorizzazioni disponibili. Per revocare una determinata autorizzazione, deselezionare la casella di controllo corrispondente.

Di seguito sono descritte le autorizzazioni disponibili e quelle predefinite per ciascun ruolo utente preconfigurato in Zabbix.

Permessi predefiniti

Accesso agli elementi dell'interfaccia utente

L'accesso predefinito alle sezioni dei menu dipende dal tipo di utente. Per i dettagli, vedere la pagina Permessi.

Accesso ad altre opzioni

Alcune delle sezioni descritte nella tabella seguente potrebbero essere nascoste:

- *Accesso ai moduli* è nascosto se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` è impostato su `false` nel file di configurazione del frontend.
- *Dispositivi* è nascosto se `EnableMobileDevices` è disabilitato.

Parametro	**Descrizione**	**Ruoli utente predefiniti**			
		Ruolo di super amministratore	*Ruolo di amministratore*	*Ruolo utente*	*Ruolo ospite*
Accesso predefinito ai nuovi elementi dell'interfaccia utente	Questa opzione specifica come saranno accessibili le nuove sezioni dei menu dopo un aggiornamento di Zabbix. Le sezioni dei menu esistenti dei moduli non vengono modificate.	Sì	Sì	Sì	Sì
Impostazioni del profilo utente					
Imponi il reindirizzamento dopo l'accesso	Selezionare la casella di controllo per limitare gli utenti all'URL di reindirizzamento dopo l'accesso. Se non è selezionata, gli utenti possono impostare il proprio URL dopo l'accesso nel profilo; l'URL di reindirizzamento dopo l'accesso viene utilizzato solo come fallback.	Sì	No	No	No
URL di reindirizzamento dopo l'accesso	Specificare l'URL di reindirizzamento dopo l'accesso.				
Accesso ai servizi					
Accesso in lettura-scrittura ai servizi	Selezionare l'accesso in lettura-scrittura ai servizi: **Nessuno** - nessun accesso **Tutti** - accesso in lettura-scrittura a tutti i servizi **Elenco di servizi** - selezionare i servizi per l'accesso in lettura-scrittura L'accesso in lettura-scrittura, se concesso, ha la precedenza sulle impostazioni di accesso in sola lettura e viene ereditato dinamicamente dai servizi figlio.	Tutti	Tutti	Nessuno	Nessuno

Accesso in lettura-scrittura ai servizi con tag	Specificare il nome del tag e, facoltativamente, il valore per concedere inoltre l'accesso in lettura-scrittura ai servizi che corrispondono al tag. Questa opzione è disponibile se nel parametro *Accesso in lettura-scrittura ai servizi* è selezionato *Elenco di servizi*. L'accesso in lettura-scrittura, se concesso, ha la precedenza sulle impostazioni di accesso in sola lettura e viene ereditato dinamicamente dai servizi figlio.				
Accesso in sola lettura ai servizi	Selezionare l'accesso in sola lettura ai servizi: **Nessuno** - nessun accesso **Tutti** - accesso in sola lettura a tutti i servizi **Elenco di servizi** - selezionare i servizi per l'accesso in sola lettura L'accesso in sola lettura non ha la precedenza sull'accesso in lettura-scrittura e viene ereditato dinamicamente dai servizi figlio.			Tutti	Tutti
Accesso in sola lettura ai servizi con tag	Specificare il nome del tag e, facoltativamente, il valore per concedere inoltre l'accesso in sola lettura ai servizi che corrispondono al tag. Questa opzione è disponibile se nel parametro *Accesso in sola lettura ai servizi* è selezionato *Elenco di servizi*. L'accesso in sola lettura non ha la precedenza sull'accesso in lettura-scrittura e viene ereditato dinamicamente dai servizi figlio.				
Accesso ai moduli					
<Nome del modulo>	Consentire o negare l'accesso a un modulo specifico. In questa sezione vengono mostrati solo i moduli abilitati. Non è possibile concedere o limitare l'accesso a un modulo attualmente disabilitato.	Sì	Sì	Sì	Sì
Accesso predefinito ai nuovi moduli	Questa opzione specifica come saranno accessibili i nuovi moduli e widget dopo un aggiornamento di Zabbix. Si applica anche ai moduli e ai widget aggiunti nella sezione *Amministrazione* > *Generale* > *Moduli*.				
Accesso all'API					
Abilitato	Abilitare/disabilitare l'accesso all'API.	Sì	No	No	No

Metodi API	Selezionare *Elenco consentiti* per consentire, oppure *Elenco negati* per negare i metodi API specificati nel campo di ricerca. Non è possibile consentire alcuni metodi API e negarne altri. Nel campo di ricerca, iniziare a digitare il nome del metodo, quindi selezionare il metodo dall'elenco di completamento automatico. È anche possibile fare clic sul pulsante *Seleziona* e selezionare i metodi dall'elenco completo disponibile per questo tipo di utente. Se una determinata azione del blocco Accesso alle azioni non è selezionata, gli utenti non potranno utilizzare i metodi API correlati a tale azione. Sono supportati i caratteri jolly. Esempi: `dashboard.*` (tutti i metodi del servizio API 'dashboard.') `*` (qualsiasi metodo), `*.export` (metodi con nome '.export' di tutti i servizi API). Se non sono specificati metodi, *Elenco consentiti* nega l'accesso a tutti i metodi API, mentre *Elenco negati* consente l'accesso a tutti i metodi API disponibili per il tipo di utente. Per consentire tutti i metodi con *Elenco consentiti*, specificare `*`. Per bloccare completamente l'accesso all'API, deselezionare la casella *Abilitato* invece di lasciare vuoto *Elenco consentiti*. L'elenco dei metodi API non si applica ai metodi che non richiedono autenticazione, come `apiinfo.version` e `user.login`.				
Accesso alle azioni					
Creare e modificare dashboard	Deselezionando questa casella di controllo verranno revocati anche i diritti di utilizzare i metodi API `.create`, `.update` e `.delete` per gli elementi corrispondenti.	Sì	Sì	Sì	No
Creare e modificare mappe					
Creare e modificare la manutenzione				No	
Aggiungere commenti ai problemi	Deselezionando questa casella di controllo verranno revocati anche i diritti di eseguire l'azione corrispondente tramite il metodo API `event.acknowledge`.			Sì	
Modificare la gravità					
Confermare i problemi					
Sopprimere i problemi					
Chiudere i problemi					
Eseguire script	Deselezionando questa casella di controllo verranno revocati anche i diritti di utilizzare il metodo API `script.execute`.				
Gestire i token API	Deselezionando questa casella di controllo verranno revocati anche i diritti di utilizzare tutti i metodi API `token.`.				

Gestire i report pianificati	Deselezionando questa casella di controllo verranno revocati anche i diritti di utilizzare tutti i metodi API `report.*`.			No	
Gestire SLA	Abilitare/disabilitare i diritti di gestione di SLA.				
Invocare "Esegui ora" sui host in sola lettura	Consentire agli utenti di utilizzare l'opzione "Esegui ora" in Ultimi dati per gli item sui host in sola lettura.			Sì	
Modificare la classificazione dei problemi	Consentire agli utenti di modificare la classificazione del problema da causa a sintomo e viceversa.				
Creare e modificare i propri media	Consentire agli utenti di creare e modificare i propri media.				
Creare e modificare i media degli utenti	Consentire agli utenti di creare e modificare i media di altri utenti. Questa opzione è disponibile solo per gli utenti con ruolo di super amministratore.		No	No	
Selezionare "Server" per il monitoraggio e il rilevamento	Deselezionando questa casella di controllo si limita la creazione di host monitorati dal server e delle regole di rilevamento della rete eseguite dal server, nonché il test degli item con il server. L'opzione *Server* sarà disabilitata nella configurazione dell'host per il parametro *Monitorato da*, nella configurazione della regola di rilevamento della rete per il parametro *Rilevato da* e nel test degli item per il parametro *Testa con*.		Sì	No	
Accesso predefinito alle nuove azioni	Questa opzione specifica come saranno accessibili le nuove azioni dopo un aggiornamento di Zabbix.		Sì	Sì	
Dispositivi					
Abilitato	Abilitare/disabilitare l'accesso a *Impostazioni utente* > *Dispositivi* (accessibile a tutti gli utenti) e *Utenti* > *Dispositivi* (accessibile solo ai super amministratori).	Sì	No	No	No
Gestire i propri dispositivi	Consentire agli utenti di gestire i propri dispositivi in *Impostazioni utente* > *Dispositivi*.				
Gestire i dispositivi degli utenti	Consentire agli utenti di gestire i dispositivi appartenenti ad altri utenti in *Utenti* > *Dispositivi*.				
Accesso predefinito alle nuove azioni	Questa opzione specifica come saranno accessibili le nuove azioni dopo un aggiornamento di Zabbix.				

Vedere anche:

- Configurazione di un utente

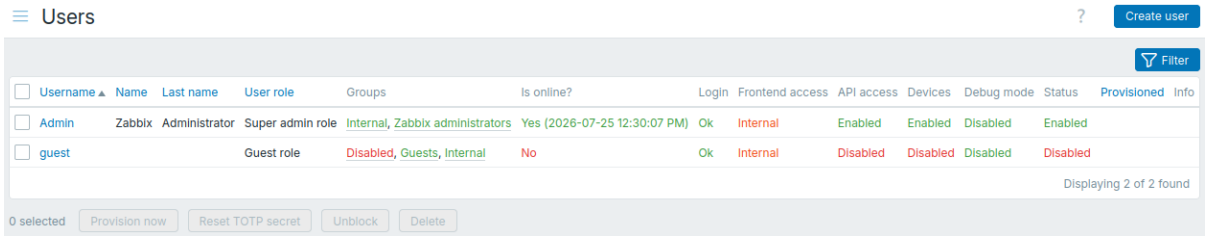
Utenti

Panoramica

Nella sezione *Utenti* > *Utenti* vengono gestiti gli utenti del sistema.

Utenti

Viene visualizzato un elenco degli utenti esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome utente	Nome utente per accedere a Zabbix. Facendo clic sul nome utente si apre il modulo di configurazione dell'utente.
Nome	Nome dell'utente.
Cognome	Cognome dell'utente.
Ruolo utente	Viene visualizzato il ruolo utente.
Gruppi	Vengono elencati i gruppi di cui l'utente è membro. Facendo clic sul nome del gruppo di utenti si apre il modulo di configurazione del gruppo di utenti. I gruppi disabilitati vengono visualizzati in rosso.
È online?	Viene visualizzato lo stato online dell'utente: *Sì* o *No*. Tra parentesi viene visualizzato l'orario dell'ultima attività dell'utente.
Accesso	Viene visualizzato lo stato di accesso dell'utente: *Ok* o *Bloccato*. Un utente può essere bloccato temporaneamente dopo aver superato il numero di tentativi di accesso non riusciti impostato nella sezione *Amministrazione > Generale > Altro* (cinque per impostazione predefinita). Facendo clic su *Bloccato* è possibile sbloccare l'utente.
Accesso al frontend	Viene visualizzato il livello di accesso al frontend: *Predefinito di sistema*, *Interno*, *LDAP* o *Disabilitato*, a seconda di quello impostato per l'intero gruppo di utenti.
Accesso API	Viene visualizzato lo stato di accesso all'API: *Abilitato* o *Disabilitato*, a seconda di quello impostato per il ruolo utente. Se l'accesso all'API è abilitato, passando il mouse su *Abilitato* vengono visualizzati i metodi API del ruolo utente in *Metodi consentiti* o *Metodi negati*, a seconda della modalità dei metodi API. Se non sono specificati metodi, viene visualizzato *Nessuno*.
Dispositivi	Viene visualizzato lo stato di accesso ai dispositivi: *Abilitato* o *Disabilitato*, a seconda di quello impostato per il ruolo utente. Questa colonna viene visualizzata se `EnableMobileDevices` è abilitato.
Modalità debug	Viene visualizzato lo stato della modalità debug: *Abilitata* o *Disabilitata*, a seconda di quello impostato per l'intero gruppo di utenti.
Stato	Viene visualizzato lo stato dell'utente: *Abilitato* o *Disabilitato*, a seconda di quello impostato per l'intero gruppo di utenti.
Provisioning	Viene visualizzata la data dell'ultimo provisioning dell'utente. Utilizzato per gli utenti creati tramite provisioning JIT da LDAP/SAML.
Informazioni	Vengono visualizzate le informazioni sugli errori. Per gli utenti senza gruppi di utenti viene visualizzato un avviso giallo. Per gli utenti senza ruoli e per gli utenti senza ruoli e gruppi di utenti viene visualizzato un avviso rosso.

Per configurare un nuovo utente, fare clic sul pulsante *Crea utente* nell'angolo superiore destro.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

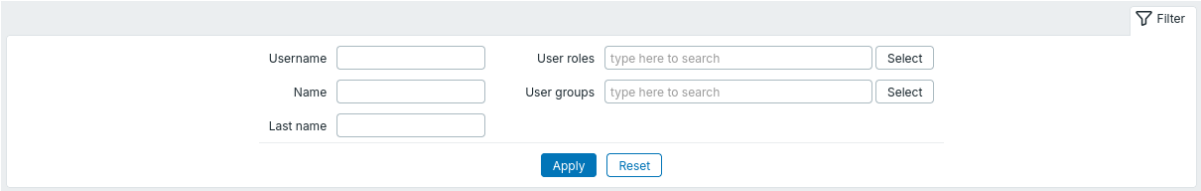
- *Provisioning ora* - aggiorna le informazioni degli utenti da LDAP (questa opzione è abilitata solo se è selezionato un utente LDAP)
- *Reimposta il segreto TOTP* - reimposta i segreti TOTP degli utenti per tutti i metodi TOTP ed elimina la sessione dell'utente (questa opzione è abilitata solo se MFA è abilitata; per gli utenti senza segreti TOTP, la sessione non verrà eliminata)
- *Sblocca* - riabilita l'accesso al sistema per gli utenti bloccati
- *Elimina* - elimina gli utenti

Per utilizzare queste opzioni, selezionare le caselle di controllo accanto agli utenti interessati, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo gli utenti di interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

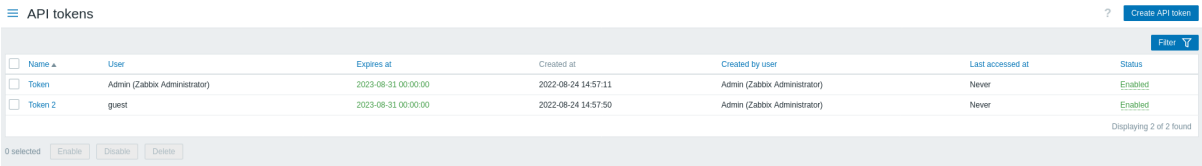
Il link *Filtro* è disponibile sopra l'elenco degli utenti. Facendo clic su di esso, viene visualizzato un filtro che consente di filtrare gli utenti per nome utente, nome, cognome, ruolo utente e gruppo di utenti.



Token API

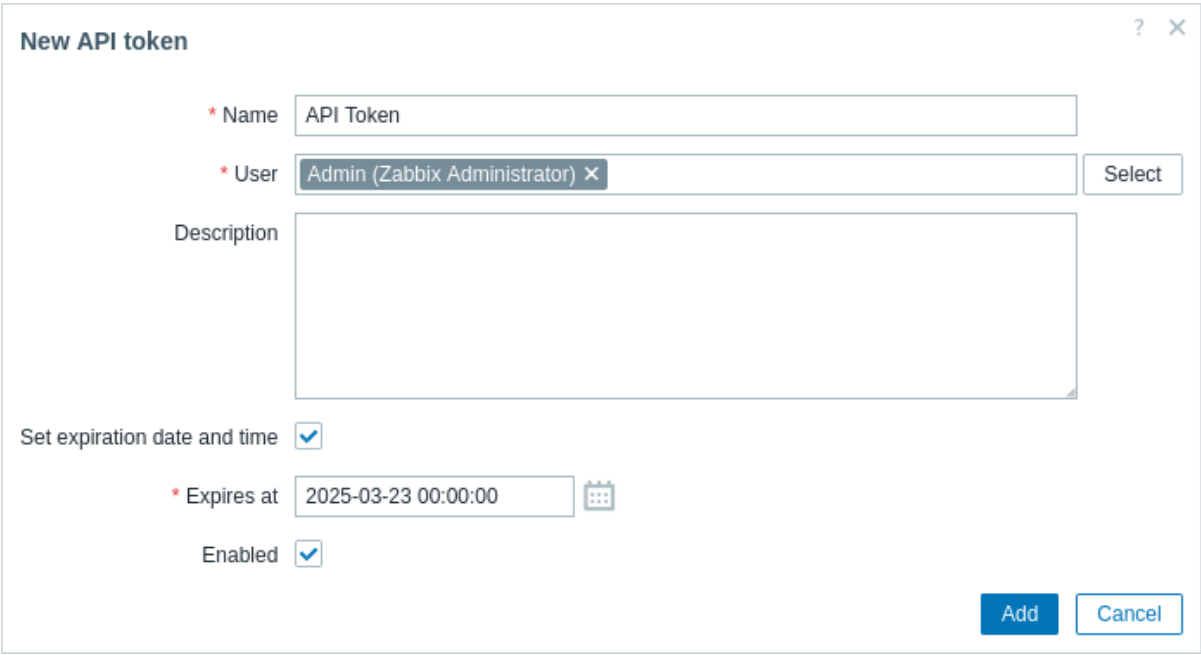
Panoramica

Questa sezione consente di creare e gestire i token API.
 I token API vengono utilizzati per l'autenticazione nell'API di Zabbix.
 Ogni token API viene assegnato a un singolo utente ed eredita le autorizzazioni di tale utente. Le richieste autenticate con un token API vengono autorizzate in base al ruolo e alle autorizzazioni dell'utente associato.



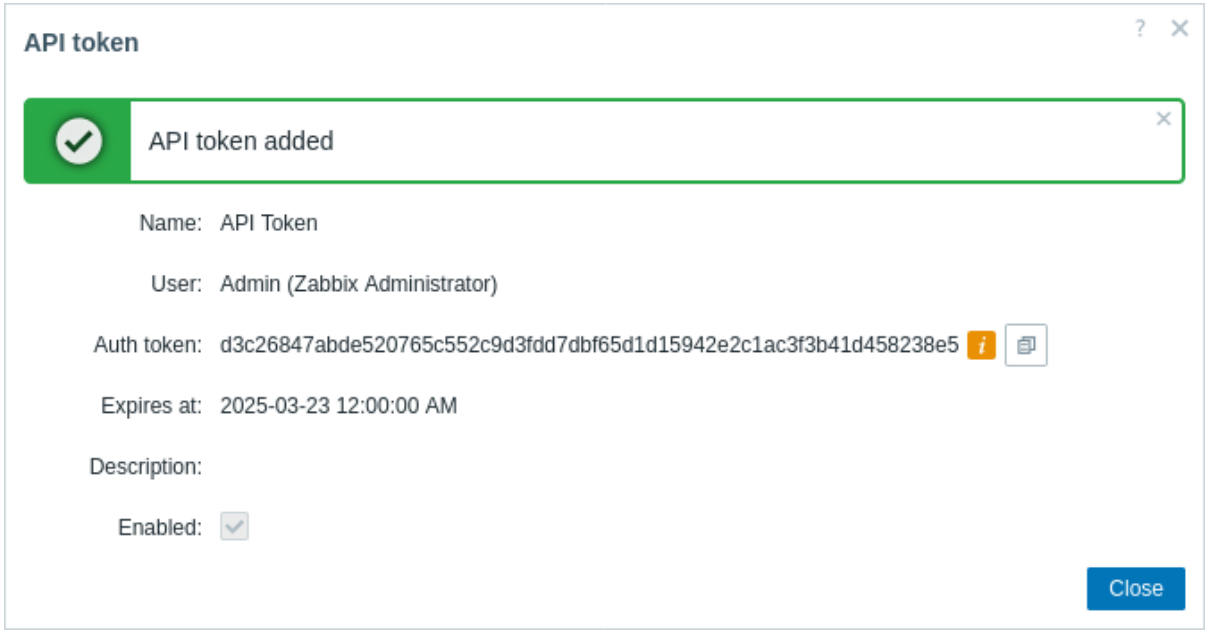
È possibile filtrare i token API per nome, utenti a cui sono assegnati i token, data di scadenza, utenti che hanno creato i token o stato (abilitato/disabilitato). Fare clic sullo stato del token nell'elenco per abilitare/disabilitare rapidamente un token. È inoltre possibile abilitare/disabilitare più token contemporaneamente selezionandoli nell'elenco e facendo clic sui pulsanti *Enable/Disable* sotto l'elenco.

Per creare un nuovo token, fare clic sul pulsante *Create API token* nell'angolo superiore destro, quindi compilare i campi obbligatori nella schermata di configurazione del token:



Parameter	Description
Name	Nome visibile del token.
User	Utente a cui deve essere assegnato il token. Per selezionare rapidamente un utente, iniziare a digitare il nome utente, il nome o il cognome, quindi selezionare l'utente richiesto dall'elenco di completamento automatico. In alternativa, è possibile fare clic sul pulsante *Select* e selezionare un utente dall'elenco completo degli utenti. Un token può essere assegnato a un solo utente.
Description	Descrizione facoltativa del token.
Set expiration date and time	Deselezionare questa casella di controllo se il token non deve avere una data di scadenza.
Expiry date	Fare clic sull'icona del calendario per selezionare la data di scadenza del token oppure immettere manualmente la data nel formato YYYY-MM-DD hh:mm:ss
Enabled	Deselezionare questa casella di controllo se è necessario creare un token in stato disabilitato.

Fare clic su *Add* per creare un token.



Copiare il valore *Auth token* e salvarlo in un luogo sicuro **prima di chiudere la pagina**, quindi fare clic su *Close*. Il token verrà visualizzato nell'elenco.

Warning:
Il valore *Auth token* non può essere visualizzato nuovamente in seguito. È disponibile solo immediatamente dopo la creazione di un token. Se si perde un token salvato, sarà necessario rigenerarlo; questa operazione creerà una nuova stringa di autorizzazione.

Fare clic sul nome del token per modificare il nome, la descrizione, le impostazioni della data di scadenza o lo stato del token. Tenere presente che non è possibile modificare l'utente a cui è assegnato il token. Fare clic sul pulsante *Update* per salvare le modifiche. Se un token è stato perso o esposto, è possibile fare clic sul pulsante *Regenerate* per generare un nuovo valore del token. Verrà visualizzata una finestra di dialogo di conferma che chiederà di confermare l'operazione, poiché dopo averla eseguita il token generato in precedenza diventerà non valido.

Gli utenti senza accesso alla sezione di menu *Administration* possono visualizzare e modificare i dettagli dei token a loro assegnati nella sezione *User profile* > *API tokens* solo se l'autorizzazione *Manage API tokens* è consentita nelle autorizzazioni del loro ruolo utente.

Autenticazione

Panoramica

La sezione *Utenti* > *Autenticazione* consente di specificare il metodo di autenticazione degli utenti per Zabbix e i requisiti per le password interne.

I metodi di autenticazione disponibili sono l'autenticazione interna, HTTP, LDAP, SAML e MFA.

Autenticazione predefinita

Per impostazione predefinita, Zabbix utilizza l'autenticazione **interna** di Zabbix per tutti gli utenti.

È possibile modificare il metodo di autenticazione predefinito impostandolo sul sistema LDAP. A tale scopo, accedere alla scheda *LDAP* e configurare i parametri LDAP, quindi tornare alla scheda *Autenticazione* e impostare il selettore *Autenticazione predefinita* su LDAP.

Si noti che il metodo di autenticazione può essere configurato in modo più dettagliato a livello di gruppo di utenti. Anche se l'autenticazione LDAP è impostata globalmente, alcuni gruppi di utenti possono comunque essere autenticati da Zabbix. Per questi gruppi, l'accesso al frontend deve essere impostato su Interno.

È inoltre possibile abilitare l'autenticazione LDAP solo per gruppi di utenti specifici, se l'autenticazione interna è utilizzata globalmente. In questo caso, è possibile specificare i dettagli dell'autenticazione LDAP e utilizzarli per gruppi di utenti specifici, per i quali l'accesso al frontend deve essere impostato su LDAP. Se un utente appartiene ad almeno un gruppo di utenti con autenticazione LDAP, non potrà utilizzare il metodo di autenticazione interno.

I metodi di autenticazione HTTP, SAML 2.0 e MFA possono essere utilizzati in aggiunta al metodo di autenticazione predefinito.

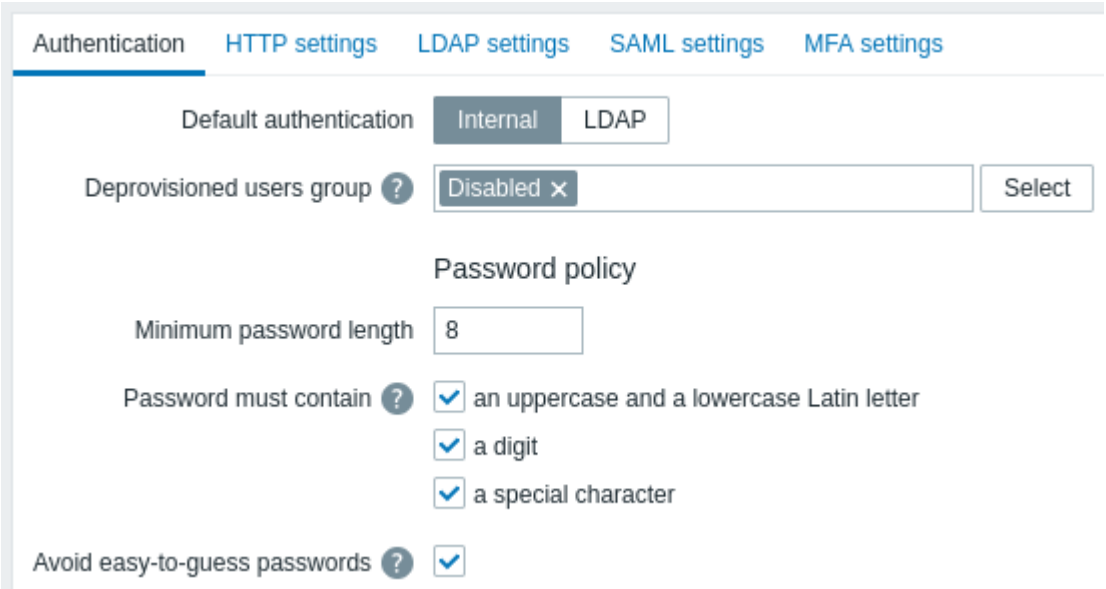
Zabbix supporta il provisioning just-in-time (JIT), che consente di creare account utente in Zabbix la prima volta che un utente esterno esegue l'autenticazione e di effettuare il provisioning di tali account utente. Il provisioning JIT è supportato per LDAP e SAML.

Vedere anche:

- Autenticazione HTTP
- Autenticazione LDAP
- Autenticazione SAML
- Autenticazione MFA

Configurazione

La scheda *Autenticazione* consente di impostare il metodo di autenticazione predefinito, specificare un gruppo per gli utenti de-provisioned e definire i requisiti di complessità delle password per gli utenti Zabbix.



Parametri di configurazione:

Parametro	Descrizione
Autenticazione predefinita	Selezionare il metodo di autenticazione predefinito per Zabbix: *Interna* o *LDAP*.
Gruppo di utenti deprovisioned	Specificare un gruppo di utenti per gli utenti deprovisioned. Questa impostazione è necessaria solo per il provisioning JIT e riguarda gli utenti creati in Zabbix dai sistemi LDAP o SAML, ma che non devono più essere sottoposti a provisioning. È necessario specificare un gruppo di utenti disabilitato.
Lunghezza minima della password	Per impostazione predefinita, la lunghezza minima della password è impostata su 8. Intervallo supportato: 1-70. Le password più lunghe di 72 caratteri verranno troncate.
La password deve contenere	Selezionare una o più caselle di controllo per richiedere l'uso dei caratteri specificati nella password: - una lettera latina maiuscola e una minuscola - una cifra - un carattere speciale Passare il mouse sul punto interrogativo per visualizzare un suggerimento con l'elenco dei caratteri per ciascuna opzione.
Evita le password facili da indovinare	Se selezionata, una password verrà verificata in base ai seguenti requisiti: - non deve contenere il nome, il cognome o il nome utente dell'utente - non deve essere una delle password comuni o specifiche del contesto. L'elenco delle password comuni e specifiche del contesto viene generato automaticamente dall'elenco "Top 100k passwords" di NCSC, dall'elenco "Top 1M passwords" di SecLists e dall'elenco delle password specifiche del contesto di Zabbix. Agli utenti interni non sarà consentito impostare password incluse in questo elenco, poiché tali password sono considerate deboli a causa del loro uso comune.

Le modifiche ai requisiti di complessità delle password non influiranno sulle password degli utenti esistenti, ma se un utente

esistente sceglie di modificare la password, la nuova password dovrà soddisfare i requisiti correnti. Un suggerimento con l'elenco dei requisiti verrà visualizzato accanto al campo *Password* nel profilo utente e nel modulo di configurazione dell'utente, accessibile dal menu *Utenti* > *Utenti*.

HTTP

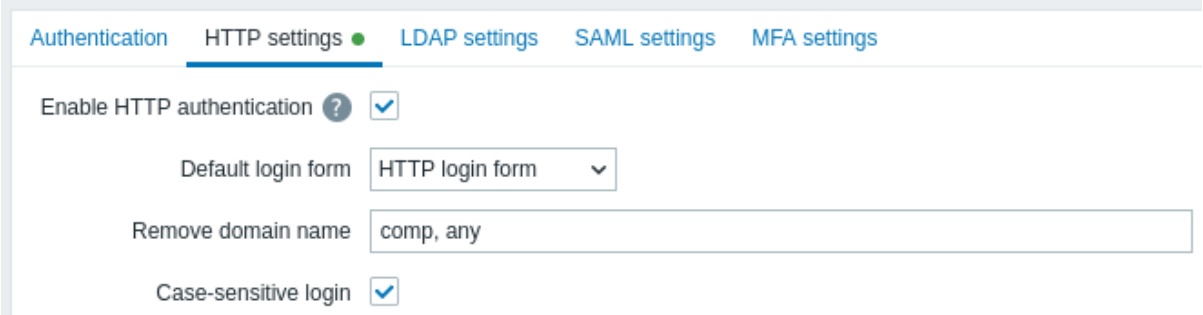
Panoramica

L'autenticazione basata su HTTP o su un web server (ad esempio: BasicAuthentication, NTLM/Kerberos) può essere utilizzata per verificare i nomi utente e le password. Tuttavia, l'utente deve esistere anche in Zabbix; la relativa password Zabbix non verrà utilizzata.

Attention:
Attenzione! Assicurarsi che l'autenticazione del web server sia configurata e funzioni correttamente prima di abilitarla.

Note:
La configurazione dell'autenticazione HTTP può essere disabilitata impostando `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` nel file di configurazione del frontend (`zabbix.conf.php`). In questo caso, la scheda *Impostazioni HTTP* con le opzioni di autenticazione HTTP sarà nascosta.

Configurazione



Parametri di configurazione:

Parametro	Descrizione
Abilita autenticazione HTTP	Selezionare la casella di controllo per abilitare l'autenticazione HTTP. Passando il mouse su verrà visualizzata una casella di suggerimento che avvisa che, in caso di autenticazione tramite web server, tutti gli utenti (anche quelli con accesso al frontend impostato su LDAP/Interno) verranno autenticati dal web server e non da Zabbix.
Modulo di accesso predefinito	Specificare se reindirizzare gli utenti non autenticati a: **Modulo di accesso Zabbix** - pagina di accesso Zabbix standard. **Modulo di accesso HTTP** - pagina di accesso HTTP. Si consiglia di abilitare l'autenticazione basata sul web server solo per la pagina `index_http.php`. Se *Modulo di accesso predefinito* è impostato su "Pagina di accesso HTTP", l'utente accederà automaticamente se il modulo di autenticazione del web server imposta un nome utente valido nella variabile `$_SERVER`. Le chiavi `$_SERVER` supportate sono `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Rimuovi nome dominio	Elenco di nomi di dominio separati da virgole che devono essere rimossi dal nome utente. Ad esempio, `comp,any`: se il nome utente è 'Admin@any' o 'comp\Admin', l'utente accederà come 'Admin'; se il nome utente è 'notacompany\Admin', l'accesso verrà negato.
Accesso con distinzione tra maiuscole e minuscole	Deselezionare la casella di controllo per disabilitare la distinzione tra maiuscole e minuscole per i nomi utente (abilitata per impostazione predefinita). Disabilitando la distinzione tra maiuscole e minuscole è possibile, ad esempio, accedere come "admin" anche se l'utente Zabbix è "Admin" o "ADMIN". Tenere presente che, se la distinzione tra maiuscole e minuscole è disabilitata e sono presenti più utenti Zabbix con nomi utente simili (ad esempio, Admin e admin), l'accesso per tali utenti verrà sempre negato con il seguente messaggio di errore: "Autenticazione non riuscita: le credenziali fornite non sono univoche."

Note:
Per gli utenti interni che non riescono ad accedere utilizzando le credenziali HTTP (con il modulo di accesso HTTP impostato come predefinito), causando l'errore 401, è possibile aggiungere una riga `ErrorDocument 401 /index.php?form=default` alle direttive di autenticazione di base, che reindirizzerà al normale modulo di accesso Zabbix.

LDAP

Panoramica

L'autenticazione LDAP esterna può essere utilizzata per verificare nomi utente e password.

L'autenticazione LDAP di Zabbix funziona almeno con Microsoft Active Directory e OpenLDAP.

Se è configurato solo l'accesso LDAP, l'utente deve comunque esistere anche in Zabbix, tuttavia la sua password di Zabbix non verrà utilizzata. Se l'autenticazione ha esito positivo, Zabbix assocerà un nome utente locale con l'attributo del nome utente restituito da LDAP.

Provisioning degli utenti

È possibile configurare il **provisioning degli utenti** JIT (just-in-time) per gli utenti LDAP. In questo caso, non è necessario che un utente esista già in Zabbix. L'account utente può essere creato quando l'utente accede a Zabbix per la prima volta.

Quando un utente LDAP inserisce il proprio login e la propria password LDAP, Zabbix verifica nel server LDAP *predefinito* se l'utente esiste. Se l'utente esiste e non dispone ancora di un account in Zabbix, viene creato un nuovo utente in Zabbix e l'utente può accedere.

Un utente creato tramite il provisioning JIT viene associato al server LDAP (directory) impostato come predefinito al momento della creazione. La modifica successiva del server LDAP predefinito non modifica né aggiorna il server LDAP collegato agli utenti di cui è già stato eseguito il provisioning.

Attention:
Se il provisioning JIT è abilitato, nella scheda *Authentication* deve essere specificato un gruppo di utenti per gli utenti deprovisioned.

Il provisioning JIT consente inoltre di aggiornare gli account degli utenti di cui è stato eseguito il provisioning in base alle modifiche apportate in LDAP. Ad esempio, se un utente viene spostato da un gruppo LDAP a un altro, verrà spostato anche da un gruppo all'altro in Zabbix; se un utente viene rimosso da un gruppo LDAP, verrà rimosso anche dal gruppo in Zabbix e, se non appartiene ad alcun altro gruppo, verrà aggiunto al gruppo di utenti per gli utenti deprovisioned. Gli account degli utenti di cui è stato eseguito il provisioning vengono aggiornati in base al periodo di provisioning configurato o quando l'utente accede a Zabbix.

Si noti che il provisioning in background viene eseguito dal frontend di Zabbix quando l'utente interagisce con esso o ha almeno una pagina del frontend aperta nel browser. Non esistono processi in background dedicati al provisioning degli utenti.

LDAP supporta tre modalità di binding alla directory per l'autenticazione e le ricerche:

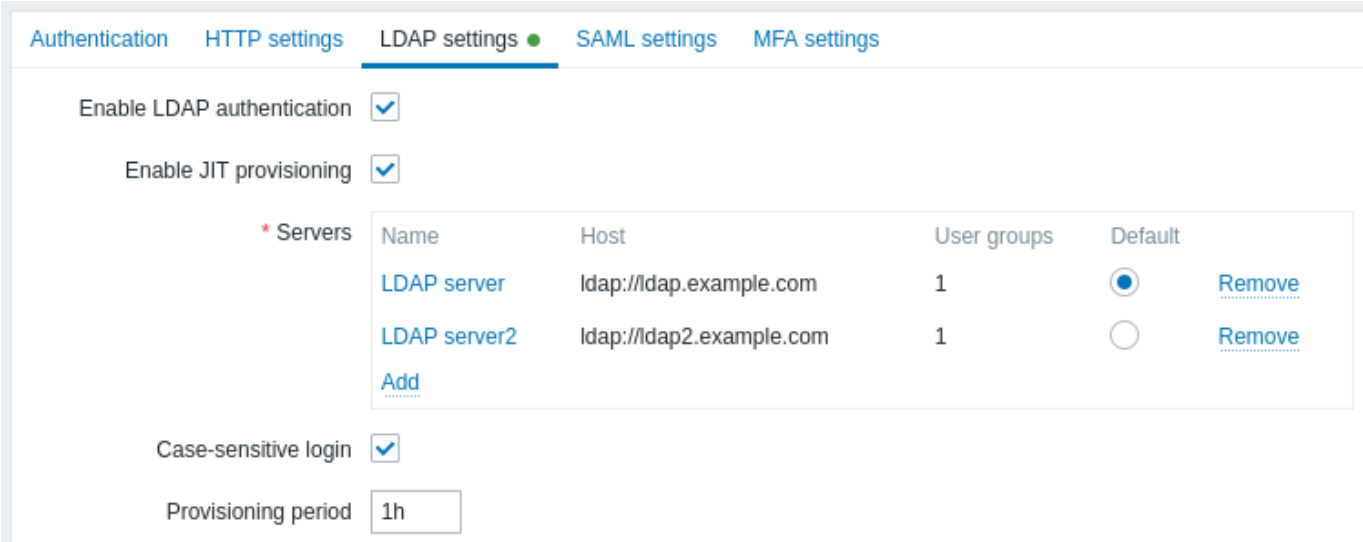
- Binding anonimo: non vengono forniti *Bind DN* / *Bind password* e il server LDAP consente query anonime.
- Utente dedicato per il binding (account di servizio): in *Bind DN* / *Bind password* viene impostato uno specifico account LDAP, utilizzato da Zabbix per le ricerche e il provisioning. Questa è l'opzione consigliata e più flessibile, perché Zabbix può eseguire ricerche e provisioning in background senza le credenziali dell'utente finale.
- Binding diretto dell'utente: Zabbix esegue il binding utilizzando le credenziali inserite dall'utente al momento dell'accesso (nessun *Bind DN* / *Bind password* configurato); questa modalità viene configurata includendo un segnaposto come `uid=%{user}` nel Base DN. In questa modalità Zabbix ha accesso alla password dell'utente solo durante l'accesso interattivo. Di conseguenza, le azioni di provisioning che richiedono l'autenticazione a LDAP al di fuori della sessione di accesso dell'utente (ad esempio, utilizzando il pulsante *Provision now* o eseguendo il provisioning in background quando l'utente non ha effettuato attivamente l'accesso) non possono eseguire l'autenticazione e pertanto non funzioneranno. Il provisioning e gli aggiornamenti per il binding diretto dell'utente vengono eseguiti solo nel momento in cui l'utente accede.

Server multipli

Se necessario, è possibile definire diversi server LDAP. Ad esempio, è possibile utilizzare un server diverso per autenticare un gruppo di utenti differente. Una volta configurati i server LDAP, nella configurazione dei gruppi di utenti diventa possibile selezionare il server LDAP richiesto per il rispettivo gruppo di utenti.

Se un utente appartiene a più gruppi di utenti e a più server LDAP, per l'autenticazione verrà utilizzato il primo server nell'elenco dei server LDAP ordinato per nome in ordine crescente.

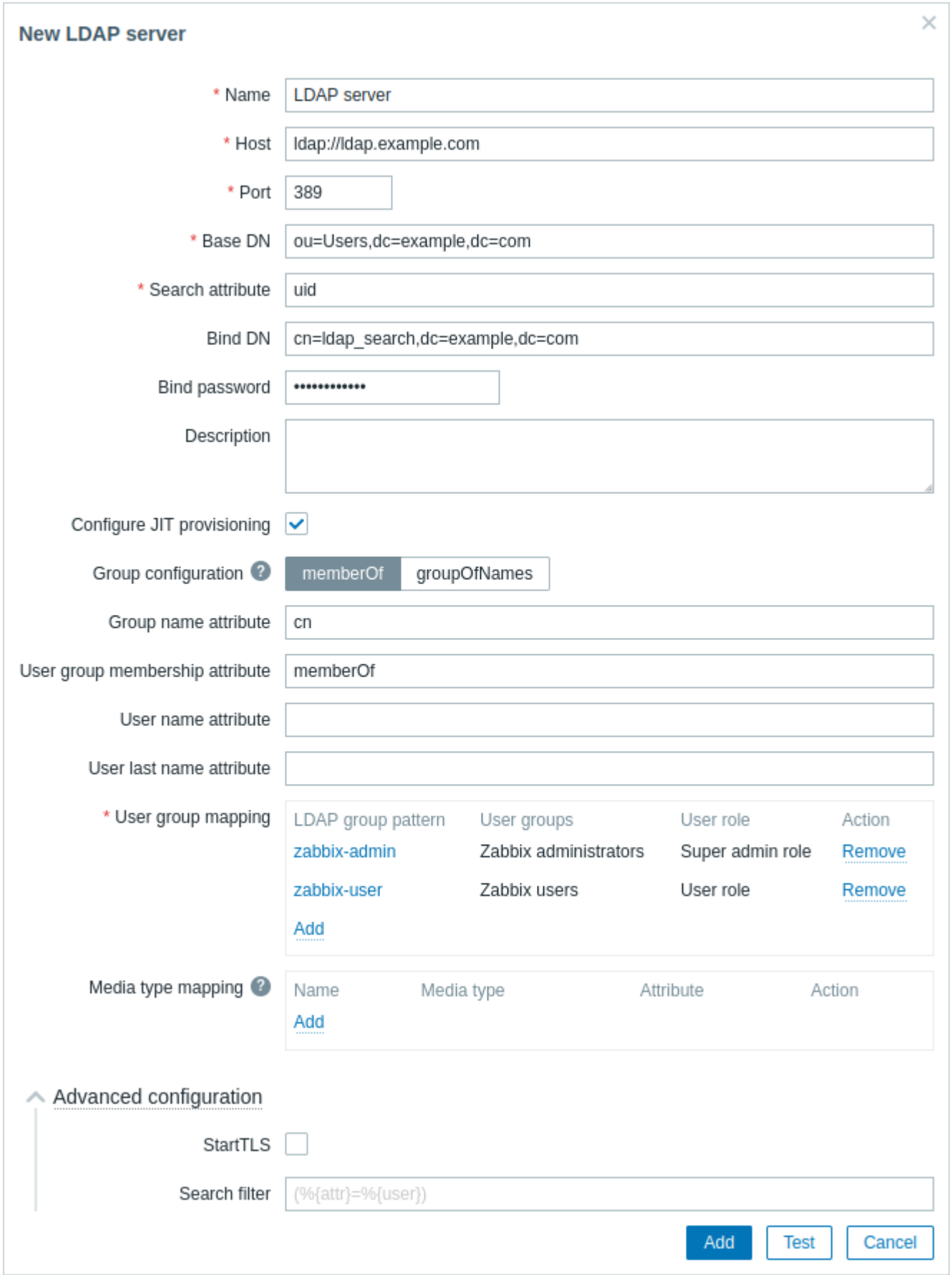
Configurazione



Parametri di configurazione:

Parametro	Descrizione
Abilita autenticazione LDAP	Selezionare la casella di controllo per abilitare l'autenticazione LDAP.
Abilita provisioning JIT	Selezionare la casella di controllo per abilitare il provisioning JIT.
Server	Fare clic su *Aggiungi* per configurare un server LDAP (vedere Configurazione del server LDAP di seguito).
Accesso con distinzione tra maiuscole e minuscole	Deselezionare la casella di controllo per disabilitare la distinzione tra maiuscole e minuscole per i nomi utente (abilitata per impostazione predefinita). La disabilitazione della distinzione tra maiuscole e minuscole consente, ad esempio, di accedere come "admin" anche se l'utente Zabbix è "Admin" o "ADMIN". Tenere presente che, se la distinzione tra maiuscole e minuscole è disabilitata e sono presenti più utenti Zabbix con nomi utente simili (ad esempio, Admin e admin), l'accesso per tali utenti verrà sempre negato con il seguente messaggio di errore: "Autenticazione non riuscita: le credenziali fornite non sono univoche."
Periodo di provisioning	Impostare il periodo di provisioning, ovvero la frequenza con cui l'utente connesso verrà sottoposto a provisioning mentre lavora con il frontend.

Configurazione del server LDAP



Parametri di configurazione del server LDAP:

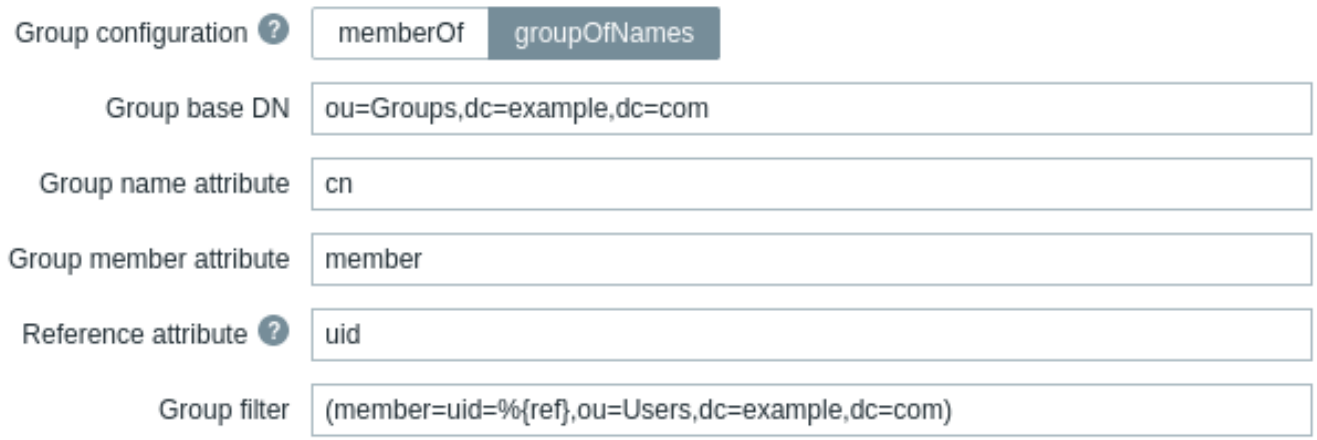
Parameter	Description
Name	Nome del server LDAP nella configurazione di Zabbix.

Parameter	Description
Host	Nome host, IP o URI del server LDAP. Esempi: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Per un server LDAP sicuro, usare il protocollo *ldaps* e il nome host. Esempio: ldaps://ldap.example.com Con OpenLDAP 2.x.x e versioni successive, è possibile usare un URI LDAP completo nel formato ldap://hostname:port oppure ldaps://hostname:port.
Port	Porta del server LDAP. Il valore predefinito è 389. Per una connessione LDAP sicura, il numero di porta è normalmente 636. Non viene usato quando si utilizzano URI LDAP completi.
Base DN	Percorso base degli account utente nel server LDAP: ou=Users,ou=system (per OpenLDAP), DC=company,DC=com (per Microsoft Active Directory) uid=%{user},dc=example,dc=com (per il binding diretto dell'utente, vedere la nota sotto)
Search attribute	Attributo dell'account LDAP usato per la ricerca: uid (per OpenLDAP), sAMAccountName (per Microsoft Active Directory)
Bind DN	Account LDAP per il binding e la ricerca sul server LDAP, esempi: uid=ldap_search,ou=system (per OpenLDAP), CN=ldap_search,OU=user_group,DC=company,DC=com (per Microsoft Active Directory) È supportato anche il binding anonimo. Si noti che il binding anonimo apre potenzialmente la configurazione del dominio a utenti non autorizzati (informazioni su utenti, computer, server, gruppi, servizi, ecc.). Per motivi di sicurezza, disabilitare i bind anonimi sugli host LDAP e usare invece l'accesso autenticato.
Bind password	Password LDAP dell'account per il binding e la ricerca sul server LDAP.
Description	Descrizione del server LDAP.
Configure JIT provisioning	Selezionare questa casella di controllo per mostrare le opzioni relative al provisioning JIT.
Group configuration	Selezionare il metodo di configurazione dei gruppi: **memberOf** - cercando gli utenti e il relativo attributo di appartenenza ai gruppi **groupOfNames** - cercando i gruppi tramite l'attributo member Si noti che memberOf è preferibile perché è più veloce; usare groupOfNames se il server LDAP non supporta `memberOf` o se è richiesto il filtraggio dei gruppi.
Group name attribute	Specificare l'attributo da cui ottenere il nome del gruppo da tutti gli oggetti nell'attributo `memberOf` (vedere il campo *User group membership attribute*) Il nome del gruppo è necessario per la mappatura dei gruppi utente.
User group membership attribute	Specificare l'attributo che contiene informazioni sui gruppi di appartenenza dell'utente (ad esempio, `memberOf`). Per esempio, l'attributo memberOf può contenere informazioni come queste: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Questo campo è disponibile solo per il metodo memberOf.
User name attribute	Specificare l'attributo che contiene il nome dell'utente.
User last name attribute	Specificare l'attributo che contiene il cognome dell'utente.
User group mapping	Associare un pattern di gruppo utente LDAP a un gruppo utente e a un ruolo utente di Zabbix. Questo è necessario per determinare quale gruppo utente/ruolo riceverà l'utente provisioned in Zabbix. Fare clic su *Add* per aggiungere una mappatura. Il campo *LDAP group pattern* supporta i caratteri jolly. Il nome del gruppo deve corrispondere a un gruppo esistente. Se un utente LDAP corrisponde a più gruppi utente di Zabbix, l'utente diventa membro di tutti. Se un utente corrisponde a più ruoli utente di Zabbix, riceverà quello con il livello di autorizzazione più alto tra essi.
Media type mapping	Associare gli attributi media LDAP dell'utente (ad esempio, email) ai media utente di Zabbix per l'invio delle notifiche (il valore dell'attributo viene usato come campo media *Send to*).
Advanced configuration	Fare clic sull'intestazione *Advanced configuration* per visualizzare le opzioni di configurazione avanzate (vedere sotto).
StartTLS	Selezionare la casella di controllo per usare l'operazione StartTLS durante la connessione al server LDAP. La connessione fallirà se il server non supporta StartTLS. StartTLS non può essere usato con server che utilizzano il protocollo *ldaps*.

Parameter	Description
Search filter	Definire una stringa personalizzata durante l'autenticazione di un utente in LDAP. Sono supportati i seguenti segnaposto: `%{attr}` - nome dell'attributo di ricerca (uid, sAMAccountName) `%{user}` - valore del nome utente da autenticare Per esempio, per eseguire una ricerca con distinzione tra maiuscole e minuscole in un ambiente LDAP o Microsoft Active Directory senza distinzione tra maiuscole e minuscole, la stringa può essere definita come segue: `(%{attr}:caseExactMatch:=%{user})`. Se il filtro non viene personalizzato, LDAP userà il valore predefinito: `(%{attr}=%{user})`.

Note:
Per configurare un server LDAP per il **binding diretto dell'utente**, aggiungere un attributo uid=%{user} al parametro *Base DN* (ad esempio, *uid=%{user},dc=example,dc=com*) e lasciare vuoti i parametri *BindDN* e *Bind password*. Durante l'autenticazione, il segnaposto %{user} verrà sostituito dal nome utente inserito al momento dell'accesso.
 Con il binding diretto dell'utente, Zabbix ha accesso alle credenziali dell'utente solo durante l'accesso interattivo. Pertanto, le attività di provisioning che operano senza un accesso interattivo (ad esempio, il pulsante *Provision now* o il metodo API `user.provision`) ignoreranno gli utenti che si autenticano tramite binding diretto dell'utente, perché Zabbix non può eseguire il binding per loro conto. Usare il binding anonimo o un utente di binding dedicato (account di servizio) se è necessario che il provisioning e le ricerche LDAP funzionino dal frontend senza richiedere l'accesso dell'utente finale.

I seguenti campi sono specifici per "groupOfNames" come metodo *Group configuration*:



Parameter	Description
Group base DN	Percorso base dei gruppi nel server LDAP.
Group name attribute	Specificare l'attributo da cui ottenere il nome del gruppo nel percorso base dei gruppi specificato. Il nome del gruppo è necessario per la mappatura dei gruppi utente.
Group member attribute	Specificare l'attributo che contiene informazioni sui membri del gruppo in LDAP (ad esempio, `member`).
Reference attribute	Specificare l'attributo di riferimento per il filtro del gruppo (vedere il campo *Group filter*). Quindi usare `%{ref}` nel filtro del gruppo per ottenere i valori dell'attributo specificato qui.
Group filter	Specificare il filtro per recuperare il gruppo di cui l'utente è membro. Per esempio, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` corrisponderà a "User1" se l'attributo member del gruppo è `uid=User1,ou=Users,dc=example,dc=com` e restituirà il gruppo di cui "User1" è membro.

Warning:
In caso di problemi con i certificati, per far funzionare una connessione LDAP sicura (ldaps) potrebbe essere necessario aggiungere una riga `TLS_REQCERT allow` al file di configurazione `/etc/openldap/ldap.conf`. Ciò può ridurre la sicurezza della connessione al catalogo LDAP.

Note:
Si consiglia di creare un account LDAP separato (*Bind DN*) per eseguire il binding e la ricerca sul server LDAP con privilegi minimi in LDAP, invece di usare account utente reali (usati per accedere al frontend di Zabbix).
 Un approccio di questo tipo offre maggiore sicurezza e non richiede di modificare la *Bind password* quando l'utente cambia la propria password nel server LDAP.
 Nella tabella sopra si tratta dell'account *ldap_search*.

Test dell'accesso

Il pulsante *Test* consente di testare l'accesso dell'utente:

Parametro	Descrizione
Login	Nome utente LDAP da testare (precompilato con il nome utente corrente dal frontend di Zabbix). Questo nome utente deve esistere nel server LDAP. Zabbix non attiverà l'autenticazione LDAP se non è in grado di autenticare l'utente di test.
Password utente	Password dell'utente LDAP da testare.

SAML

Panoramica

L'autenticazione SAML 2.0 authentication può essere utilizzata per accedere a Zabbix.

Se l'accesso SAML è configurato senza provisioning utente JIT, l'utente deve già esistere in Zabbix; la password Zabbix dell'utente non viene utilizzata per l'autenticazione. Dopo l'autenticazione SAML riuscita, Zabbix confronta l'attributo username ricevuto da SAML con un nome utente Zabbix locale esistente.

Provisioning utenti

È possibile configurare il **provisioning** utenti JIT (just-in-time) per gli utenti SAML. In questo caso, non è necessario che un utente esista già in Zabbix. L'account utente può essere creato quando l'utente accede a Zabbix per la prima volta.

Attention:
Se il provisioning JIT è abilitato, nella scheda *Authentication* deve essere specificato un gruppo utenti per gli utenti deprovisionati.

Oltre al provisioning JIT, è anche possibile abilitare e configurare il provisioning SCIM (System for Cross-domain Identity Management), ovvero la gestione *continua* degli account utente per quegli utenti che sono stati creati tramite il provisioning utenti. Il provisioning SCIM richiede un API token di Zabbix (con permessi di Super admin) per l'autenticazione in Zabbix.

Ad esempio, se un utente viene spostato da un gruppo SAML a un altro, anche l'utente verrà spostato da un gruppo a un altro in Zabbix; se un utente viene rimosso da un gruppo SAML, anche l'utente verrà rimosso dal gruppo in Zabbix e, se non appartiene ad alcun altro gruppo, verrà aggiunto al gruppo utenti per gli utenti deprovisionati.

Se SCIM è abilitato e configurato, un utente SAML verrà provisionato nel momento in cui accede a Zabbix e aggiornato continuamente in base alle modifiche in SAML. Gli utenti SAML già esistenti non verranno provisionati e solo gli utenti provisionati verranno aggiornati. Si noti che solo i supporti validi verranno aggiunti a un utente quando l'utente viene provisionato o aggiornato.

Se SCIM non è abilitato, un utente SAML verrà provisionato (e successivamente aggiornato) nel momento in cui accede a Zabbix.

Note:
Se l'autenticazione SAML è abilitata, gli utenti potranno scegliere se accedere localmente o tramite single sign-on SAML. Se viene utilizzato il provisioning JIT, sarà possibile solo il single sign-on.

Configurazione del provider di identità

Per funzionare con Zabbix, un provider di identità SAML (onelogin.com, auth0.com, okta.com, ecc.) deve essere configurato nel seguente modo:

- *Assertion Consumer URL* deve essere impostato su `<path_to_zabbix_ui>/index_sso.php?acs`
- *Single Logout URL* deve essere impostato su `<path_to_zabbix_ui>/index_sso.php?sls`

Esempi di `<path_to_zabbix_ui>`: `https://example.com/zabbix/ui`, `http://another.example.com/zabbix`, `http://<any_public_ip_address>/zabbix`

Attention:
È necessario installare php-openssl per utilizzare l'autenticazione SAML nel frontend.

Per utilizzare l'autenticazione SAML, Zabbix deve essere configurato nel modo seguente:

1. Fornire la chiave privata e i certificati SP. È necessario fornire un certificato IdP (incollando il testo o scegliendo un file) per attivare SAML tramite il frontend. È necessario fornire un certificato SP e una chiave privata SP se sono selezionate le opzioni *Firma* o *Crittografa*.

Se nel file di configurazione del frontend (`zabbix.conf.php`) è impostato `$SSO['CERT_STORAGE'] = 'database'`, è possibile incollare il testo del certificato o caricare il file del certificato nel frontend durante la configurazione SAML: non sono necessari file nel filesystem.

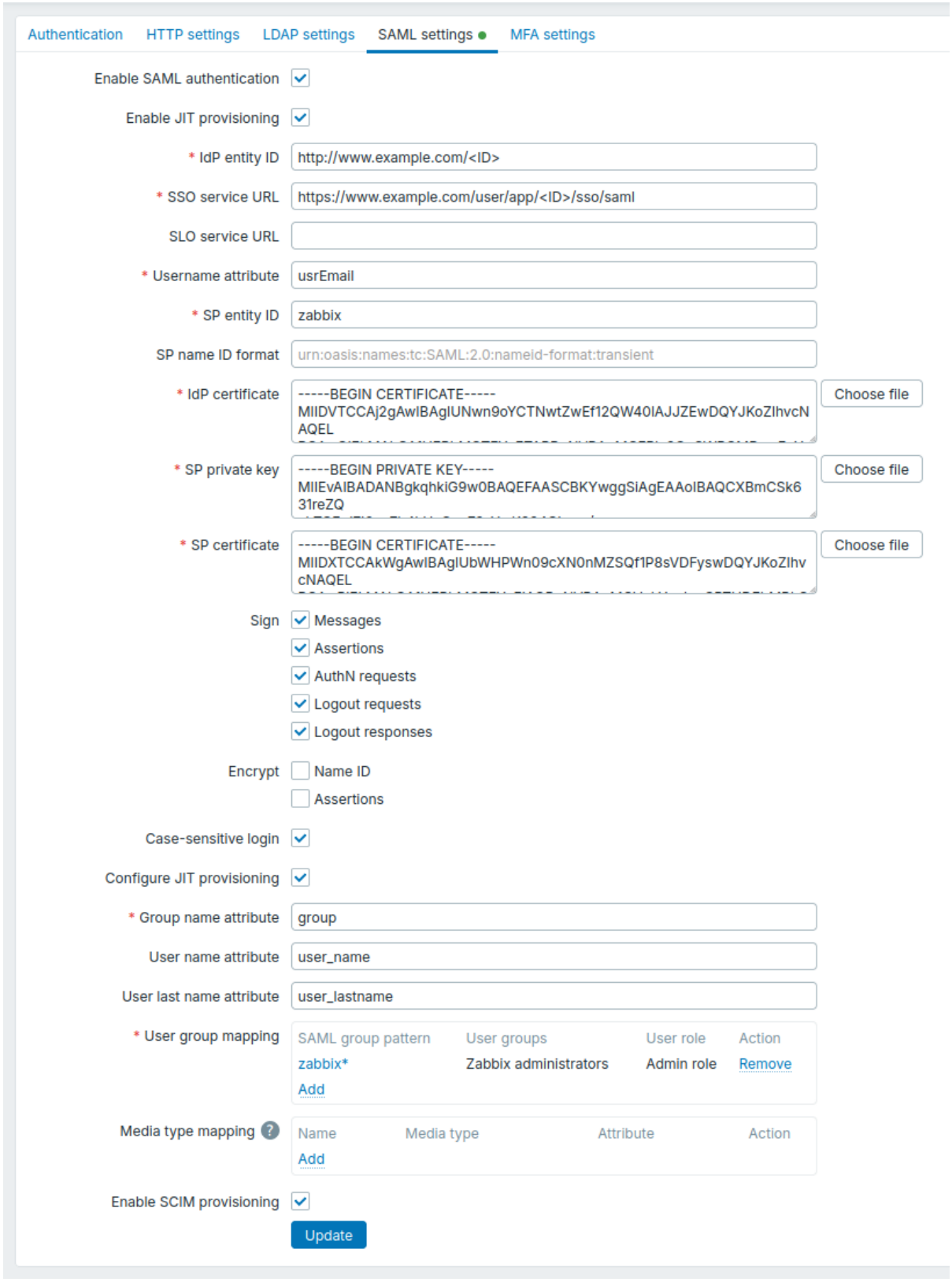
Se nel file di configurazione è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere disponibile nel filesystem (per impostazione predefinita in `ui/conf/certs` o nel percorso configurato nel file di configurazione) e il frontend non memorizzerà i certificati nel database. Se `$SSO['CERT_STORAGE']` non è impostato o è commentato, viene utilizzato lo storage su file.

Per impostazione predefinita, Zabbix cercherà nei seguenti percorsi:

- ui/conf/certs/sp.key - file della chiave privata SP
- ui/conf/certs/sp.crt - file del certificato SP
- ui/conf/certs/idp.crt - file del certificato IDP

Attention:
Il materiale dei certificati può essere importato tramite il frontend incollando il testo o caricando i file. Sebbene in alcuni ambienti siano consentite importazioni non crittografate per comodità, si consiglia vivamente di utilizzare un trasporto crittografato (ad esempio, abilitare HTTPS/TLS per il frontend Zabbix). Quando i certificati/le chiavi SAML sono memorizzati nel database, abilitare TLS/SSL per le connessioni al database, in modo da proteggere il traffico tra l'applicazione e il database.
 Crittografare i backup del database che potrebbero contenere certificati SAML o chiavi private e limitare l'accesso ai file di backup e alle tabelle del database che memorizzano il materiale SAML. Limitare i privilegi dell'utente del database al minimo necessario.

2. Tutte le impostazioni più importanti possono essere configurate nel frontend Zabbix. Tuttavia, è possibile specificare impostazioni aggiuntive nel file di configurazione.



Parametri di configurazione disponibili nel frontend Zabbix:

Parametro	Descrizione
Abilita autenticazione SAML	Selezionare la casella di controllo per abilitare l'autenticazione SAML.
Abilita provisioning JIT	Selezionare la casella di controllo per abilitare il provisioning degli utenti JIT.
ID entità IDP	L'identificatore univoco dell'entità all'interno dell'identity provider SAML.
URL del servizio SSO	L'URL a cui gli utenti verranno reindirizzati durante l'accesso.
URL del servizio SLO	L'URL a cui gli utenti verranno reindirizzati durante la disconnessione. Se lasciato vuoto, il servizio SLO non verrà utilizzato.

Parametro	Descrizione
Attributo del nome utente	Attributo SAML da utilizzare come nome utente durante l'accesso a Zabbix. L'elenco dei valori supportati è determinato dall'identity provider. Esempi: uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
ID entità SP	L'identificatore univoco del service provider (se non corrisponde, l'operazione verrà rifiutata). È possibile specificare un URL o qualsiasi stringa di dati.
Formato dell'ID nome SP	Richiede un formato specifico dell'ID nome nella risposta. Esempi: 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
Certificato IdP	Contenuto del certificato del service provider (SP) per la configurazione del servizio SAML di single sign-on (SSO).
Chiave privata SP	Contenuto della chiave privata del service provider (SP) per la configurazione del servizio SAML di single sign-on (SSO). Fornisce autenticazione sicura e scambio di dati con l'identity provider (IdP).
Certificato SP	Contenuto del certificato del service provider (SP) per la configurazione del servizio SAML di single sign-on (SSO).
Firma	Deselezionare le caselle di controllo per disabilitare le firme SAML per le entità corrispondenti: *Messaggi* *Assertion* *Richieste AuthN* *Richieste di disconnessione* *Risposte di disconnessione*
Crittografa	Selezionare le caselle di controllo per scegliere le entità per le quali abilitare la crittografia SAML: *ID nome* *Assertion*
Accesso con distinzione tra maiuscole e minuscole	Deselezionare la casella di controllo per disabilitare la distinzione tra maiuscole e minuscole per i nomi utente (abilitata per impostazione predefinita). La disabilitazione della distinzione tra maiuscole e minuscole consente, ad esempio, di accedere come "admin" anche se l'utente Zabbix è "Admin" o "ADMIN". Tenere presente che, se la distinzione tra maiuscole e minuscole è disabilitata e sono presenti più utenti Zabbix con nomi utente simili (ad esempio, Admin e admin), l'accesso per tali utenti verrà sempre negato con il seguente messaggio di errore: "Autenticazione non riuscita: le credenziali fornite non sono univoche."
Configura provisioning JIT	Selezionare questa casella di controllo per mostrare le opzioni relative al provisioning degli utenti JIT.
Attributo del nome del gruppo	Specificare l'attributo del nome del gruppo per il provisioning degli utenti JIT.
Attributo del nome utente	Specificare l'attributo del nome utente per il provisioning degli utenti JIT.
Attributo del cognome dell'utente	Specificare l'attributo del cognome dell'utente per il provisioning degli utenti JIT.
Mappatura dei gruppi utente	Associare un modello di gruppo utente SAML a un gruppo utente e a un ruolo utente Zabbix. Questo è necessario per determinare il gruppo/ruolo utente che verrà assegnato all'utente sottoposto a provisioning in Zabbix. Fare clic su *Aggiungi* per aggiungere una mappatura. Il campo *Modello di gruppo SAML* supporta i caratteri jolly. Il nome del gruppo deve corrispondere a un gruppo esistente. Se un utente SAML corrisponde a più gruppi utente Zabbix, diventa membro di tutti questi gruppi. Se un utente corrisponde a più ruoli utente Zabbix, otterrà il livello di autorizzazione più alto tra quelli disponibili.
Mappatura dei tipi di supporto	Associare gli attributi dei supporti SAML dell'utente (ad esempio, l'e-mail) ai supporti utente Zabbix per l'invio delle notifiche.

Parametro	Descrizione
Abilita provisioning SCIM	Selezionare questa casella di controllo per abilitare il provisioning SCIM 2.0.

Vedere gli esempi di configurazione degli identity provider SAML per l'accesso e il provisioning degli utenti in Zabbix con:

- Microsoft Azure AD
- Okta
- Onelogin

Note sul provisioning SCIM

Per il provisioning SCIM, specificare il percorso al frontend Zabbix e aggiungervi `api_scim.php`, sul lato del provider di identità, ovvero:

`https://<path-to-zabbix-ui>/api_scim.php`

Gli attributi utente utilizzati in Zabbix (username, nome utente, cognome utente e attributi dei media) devono essere aggiunti come attributi personalizzati e, se necessario, lo spazio dei nomi esterno deve corrispondere allo schema utente: `urn:ietf:params:scim:schemas:core:2.0:User`.

Impostazioni avanzate

È possibile configurare parametri SAML aggiuntivi nel file di configurazione del frontend di Zabbix (`zabbix.conf.php`):

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

Esempi di configurazione

Esempio 1: Connessione a un'istanza Zabbix con un bilanciatore del carico che utilizza TLS/SSL

Se viene utilizzato un bilanciatore del carico per connettersi a un'istanza Zabbix e il bilanciatore del carico utilizza TLS/SSL mentre Zabbix non lo utilizza, specificare i parametri `baseurl`, `strict` e `use_proxy_headers` come segue:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Esempio 2: Configurazione di algoritmi personalizzati per la firma e il digest SAML

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Esempio 3: Configurazione dell'SSO con autenticazione Kerberos tramite ADFS

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

In questo caso, nel campo *SP name ID* della configurazione SAML, impostare:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

MFA

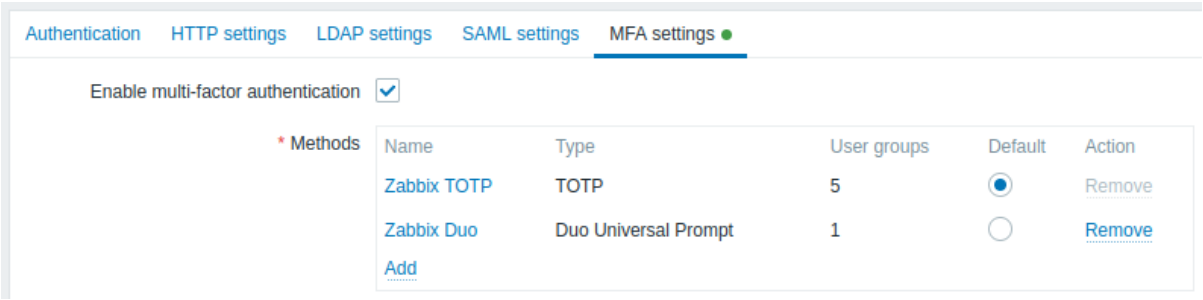
Panoramica

L'autenticazione a più fattori (MFA) può essere utilizzata per accedere a Zabbix, fornendo un ulteriore livello di sicurezza oltre al semplice nome utente e password.

Con la MFA, l'utente deve essere presente in Zabbix, deve fornire le credenziali Zabbix al momento dell'accesso e deve anche dimostrare la propria identità con altri mezzi, solitamente un codice generato da un'app di autenticazione sul cellulare dell'utente.

Sono disponibili diversi metodi di autenticazione a più fattori, consentendo agli utenti di scegliere l'opzione più adatta alle proprie esigenze e preferenze di sicurezza. Questi metodi sono la password monouso a tempo (TOTP) e Duo Universal Prompt.

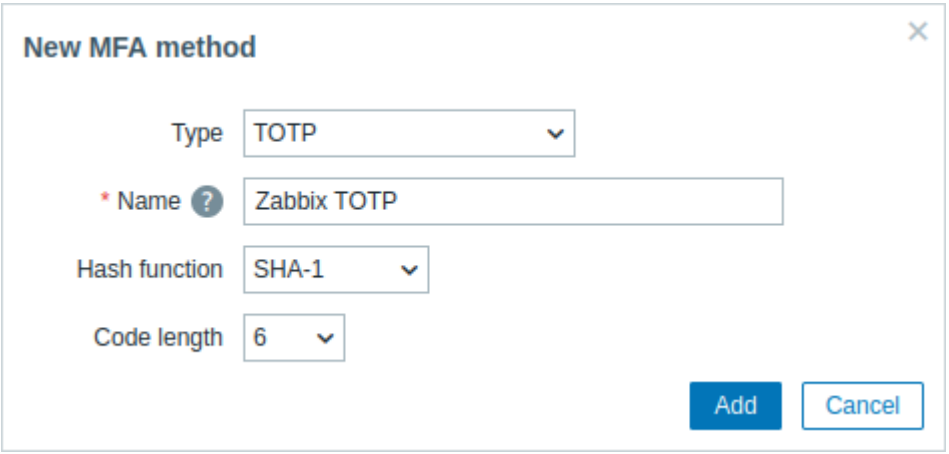
Configurazione



Parametri di configurazione:

Parametro	Descrizione
Abilita autenticazione a più fattori	Selezionare la casella di controllo per abilitare l'autenticazione a più fattori.
Metodi	Fare clic su *Aggiungi* per configurare un metodo MFA (vedere la configurazione del metodo di seguito).

Configurazione del metodo



Parametri di configurazione del metodo:

Parametro	Descrizione
Type	Selezionare il tipo di metodo MFA: **TOTP** - utilizzare un'app di autenticazione per generare password monouso basate sul tempo; **Duo Universal Prompt** - utilizzare il servizio di autenticazione Duo per fornire l'autenticazione a più fattori.
Name	Inserire un nome visualizzato come nome dell'account per tutti gli utenti MFA nelle app di autenticazione (ad esempio, "Zabbix").
Hash function	Selezionare la funzione hash (SHA-1, SHA-256 o SHA-512) per generare i codici TOTP. Questo parametro è disponibile se il tipo di metodo MFA è impostato su "TOTP". Si noti che la scelta di SHA-256 o SHA-512 può limitare significativamente la compatibilità, poiché molte applicazioni attualmente non supportano queste funzioni.

Parametro	Descrizione
Code length	Selezionare la lunghezza del codice di verifica (6 o 8). Questo parametro è disponibile se il tipo di metodo MFA è impostato su ”TOTP”.
API hostname	Inserire il nome host API fornito dal servizio di autenticazione Duo. Questo parametro è disponibile se il tipo di metodo MFA è impostato su ”Duo Universal Prompt”.
Client ID	Inserire l’ID client fornito dal servizio di autenticazione Duo. Questo parametro è disponibile se il tipo di metodo MFA è impostato su ”Duo Universal Prompt”.
Client secret	Inserire il segreto client fornito dal servizio di autenticazione Duo. Questo parametro è disponibile se il tipo di metodo MFA è impostato su ”Duo Universal Prompt”.

Esempi di configurazione

Questa sezione fornisce esempi di configurazione di MFA utilizzando Time-Based One-Time Password (TOTP) e Duo Universal Prompt.

TOTP

Per TOTP, gli utenti devono verificare la propria identità utilizzando un’app di autenticazione (ad esempio, l’app Google Authenticator).

1. Vai alle impostazioni MFA in Zabbix sotto *Users* > *Authentication* e abilita l’autenticazione a più fattori.

2. Aggiungi un nuovo metodo MFA con la seguente configurazione:

- Type: TOTP
- Name: Zabbix TOTP
- Hash function: SHA-1
- Code length: 6

3. Fai clic sul pulsante *Add* e quindi sul pulsante *Update*.

4. Vai a *Users* > *User groups* e crea un nuovo gruppo di utenti con la seguente configurazione:

- Group name: TOTP group
- Users: Admin
- Multi-factor authentication: Default (oppure ”Zabbix TOTP” se non è impostato come predefinito)

5. Disconnettiti da Zabbix e accedi di nuovo usando le tue credenziali. Dopo l’accesso riuscito, ti verrà richiesto di registrarti all’MFA, con la visualizzazione di un codice QR e di una chiave segreta.



6. Scansiona il codice QR oppure inserisci la chiave segreta nell'app Google Authenticator. L'app genererà un codice di verifica che dovrai inserire per completare il processo di accesso.

7. Per gli accessi successivi, recupera il codice di verifica dall'app Google Authenticator e inseriscilo durante l'accesso.

Duo Universal Prompt

Per Duo Universal Prompt, gli utenti devono verificare la propria identità utilizzando l'app autenticatore Duo Mobile.

Attention:
Il metodo MFA Duo Universal Prompt richiede l'installazione dell'estensione php-curl, l'accesso a Zabbix tramite HTTPS e l'autorizzazione per le connessioni in uscita verso i server Duo. Inoltre, se hai abilitato Content Security Policy (CSP) sul web server, assicurati di aggiungere "duo.com" alla direttiva CSP nel file di configurazione del virtual host.

1. Registrati per un account amministratore Duo gratuito su Duo Signup.

2. Apri il pannello di amministrazione Duo, vai su *Applications* > *Protect an Application*, cerca l'applicazione *Web SDK* e fai clic su *Protect*.

3. Prendi nota delle credenziali (Client ID, Client secret, API hostname) necessarie per configurare il metodo MFA in Zabbix.

4. Vai alle impostazioni MFA in Zabbix sotto *Users* > *Authentication* e abilita l'autenticazione a più fattori.

5. Aggiungi un nuovo metodo MFA con la seguente configurazione:

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: (usa l'API hostname di Duo)
- Client ID: (usa il Client ID di Duo)
- Client secret: (usa il Client secret di Duo)

6. Fai clic sul pulsante *Add* e poi sul pulsante *Update*.

7. Vai su *Users* > *User groups* e crea un nuovo gruppo di utenti con la seguente configurazione:

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default (oppure "Zabbix Duo" se non è impostato come predefinito)

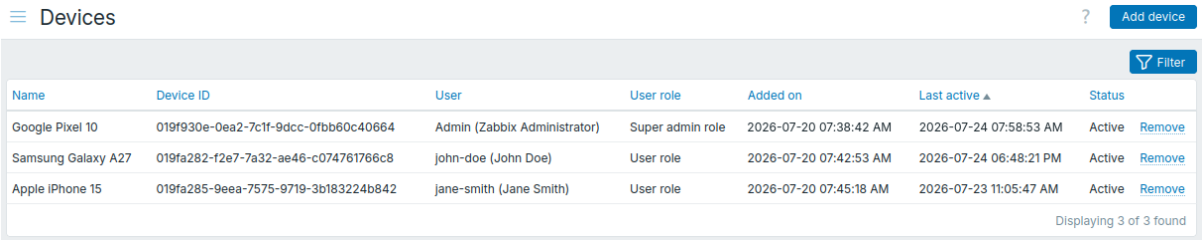
8. Esci da Zabbix e accedi di nuovo usando le tue credenziali. Dopo un accesso riuscito, ti verrà richiesto di registrarti all'MFA e verrai reindirizzato a Duo. Completa la configurazione di Duo e verifica il tuo utente con l'app Duo sul telefono per accedere.

9. Per gli accessi successivi, usa il metodo MFA appropriato fornito dall'app Duo (ad esempio recuperando un codice di verifica, rispondendo alle notifiche push o usando chiavi hardware) e inserisci le informazioni richieste durante l'accesso.

Dispositivi

Questa sezione consente ai Super amministratori di gestire i dispositivi mobili da utilizzare con l'app mobile Zabbix.

Questa sezione viene visualizzata se `EnableMobileDevices` è abilitato.



Per aggiungere un dispositivo:

1. Fare clic su *Aggiungi dispositivo*.
2. Selezionare un utente, quindi fare clic su *Aggiungi*. L'utente deve avere l'accesso ai dispositivi abilitato.
3. Viene visualizzato un codice QR. L'utente deve scansionare questo codice nella propria app mobile Zabbix.

Note:
Quando si aggiunge un dispositivo per un utente, Zabbix aggiunge anche il tipo di media notifica push ai media dell'utente. Quando si rimuove il dispositivo, anche questo media dell'utente viene rimosso. Se sono configurati più tipi di media per le notifiche push, Zabbix aggiunge quello abilitato che viene prima in ordine alfabetico.

I dispositivi aggiunti vengono visualizzati in un elenco, che è possibile filtrare.

È inoltre possibile rimuovere i dispositivi. Quando si rimuove un dispositivo, le relative credenziali di accesso vengono revocate e rimosse da Zabbix; dopodiché il dispositivo non può più accedere a Zabbix.

L'elenco mostra i seguenti dettagli:

Colonna	Descrizione
Nome	Nome del dispositivo.
ID dispositivo	ID del dispositivo.
Utente	Utente per il quale è stato aggiunto il dispositivo, oppure *Utente inaccessibile* se questo utente è stato eliminato.
Ruolo utente	Ruolo utente.
Aggiunto il	Data e ora in cui è stato aggiunto il dispositivo.
Ultima attività	Data e ora dell'ultima attività del dispositivo.
Stato	Stato del dispositivo: **Nuovo** - il dispositivo è stato aggiunto senza scansionare il codice QR e dovrà essere aggiunto nuovamente. **Nuovo (token scaduto)** - il dispositivo è stato aggiunto senza scansionare il codice QR e il token nel codice QR è scaduto; il dispositivo dovrà essere aggiunto nuovamente. **Attivo** - il dispositivo è stato aggiunto, il codice QR è stato scansionato e può utilizzare l'app mobile Zabbix. **Non assegnato** - il dispositivo è stato aggiunto, ma il relativo utente è stato eliminato e il dispositivo non può utilizzare l'app mobile Zabbix.

Gli utenti con autorizzazioni sufficienti possono inoltre gestire i propri dispositivi in *Impostazioni utente* > *Dispositivi*.

Amministrazione

Panoramica

Il menu Administration è dedicato alle funzioni amministrative di Zabbix. Questo menu è disponibile solo per gli utenti del tipo SuperAdmin.

Origine dei dati

Panoramica

La sezione *Amministrazione* > *Origine dati* consente di specificare l'origine dei dati di telemetria, ovvero il database che contiene i dati di telemetria (tracce, metriche e log). I dati di telemetria vengono utilizzati per il monitoraggio delle prestazioni delle applicazioni (APM), che consente di vedere come le richieste fluiscono attraverso le applicazioni e i servizi, misurarne le prestazioni e individuare la causa delle operazioni lente o non riuscite.

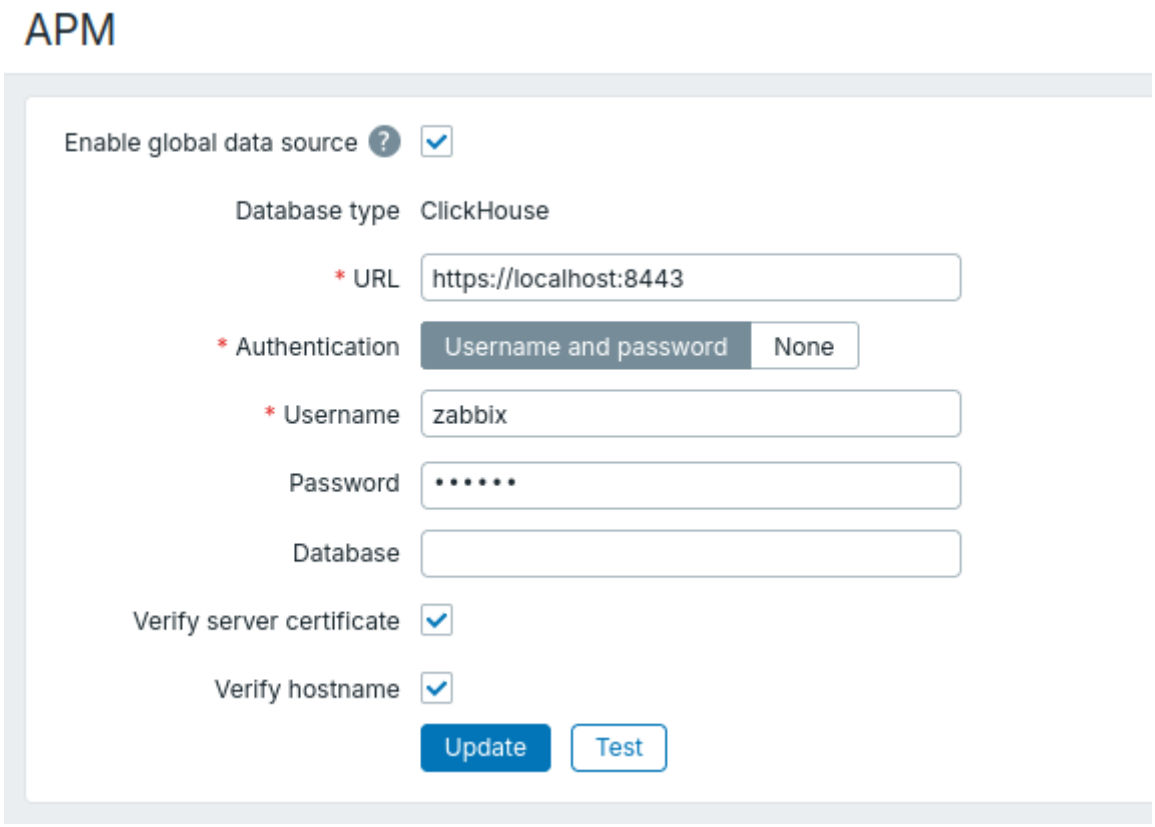
L'origine dei dati di telemetria deve supportare OpenTelemetry.

L'origine dei dati di telemetria specificata in questa sezione è chiamata origine dati globale, perché è condivisa da Zabbix server, proxy e frontend. Tuttavia, è possibile sostituirla per uno qualsiasi di questi componenti specificando un'origine dati di telemetria diversa nella configurazione di server, proxy o frontend.

L'item telemetry query interroga questa origine dati, a meno che nella configurazione del componente non ne venga specificata una diversa.

APM

Per specificare la fonte dati globale, selezionare la casella *Abilita fonte dati globale* e compilare i parametri descritti di seguito. Prima di salvare, è possibile fare clic su *Test* per verificare la connessione a questa fonte dati (è necessaria l'estensione php-curl).



Parametro	Descrizione
Tipo di database	Il tipo di database è impostato su *ClickHouse* e non può essere modificato. Vedere anche: Requisiti e Configurazione di ClickHouse.
URL	Inserire l'URL HTTP o HTTPS utilizzato per connettersi al database.
Autenticazione	Selezionare il metodo di autenticazione utilizzato per accedere al database: *Nome utente e password* oppure *Nessuna*.
Nome utente	Specificare il nome utente utilizzato per accedere al database. Questo parametro è disponibile se *Autenticazione* è impostata su *Nome utente e password*.
Password	Specificare la password utilizzata per accedere al database oppure lasciare il campo vuoto se l'utente del database non dispone di una password. La password viene reimpostata quando si modifica l'URL. Questo parametro è disponibile se *Autenticazione* è impostata su *Nome utente e password*.
Database	Inserire il nome del database che contiene i dati di telemetria. Se lasciato vuoto, Zabbix utilizza il database predefinito configurato in ClickHouse per l'utente.
Verifica certificato del server	Selezionare la casella per verificare il certificato del server. Se selezionata, le opzioni TLS correlate devono essere impostate nella configurazione di server, proxy o frontend. Questo parametro è disponibile se *URL* contiene un URL HTTPS.
Verifica nome host	Selezionare la casella per verificare che il nome host in *URL* corrisponda al nome host nel certificato del server. Se selezionata, le opzioni TLS correlate devono essere impostate nella configurazione di server, proxy o frontend. Questo parametro è disponibile se *URL* contiene un URL HTTPS.

Generale

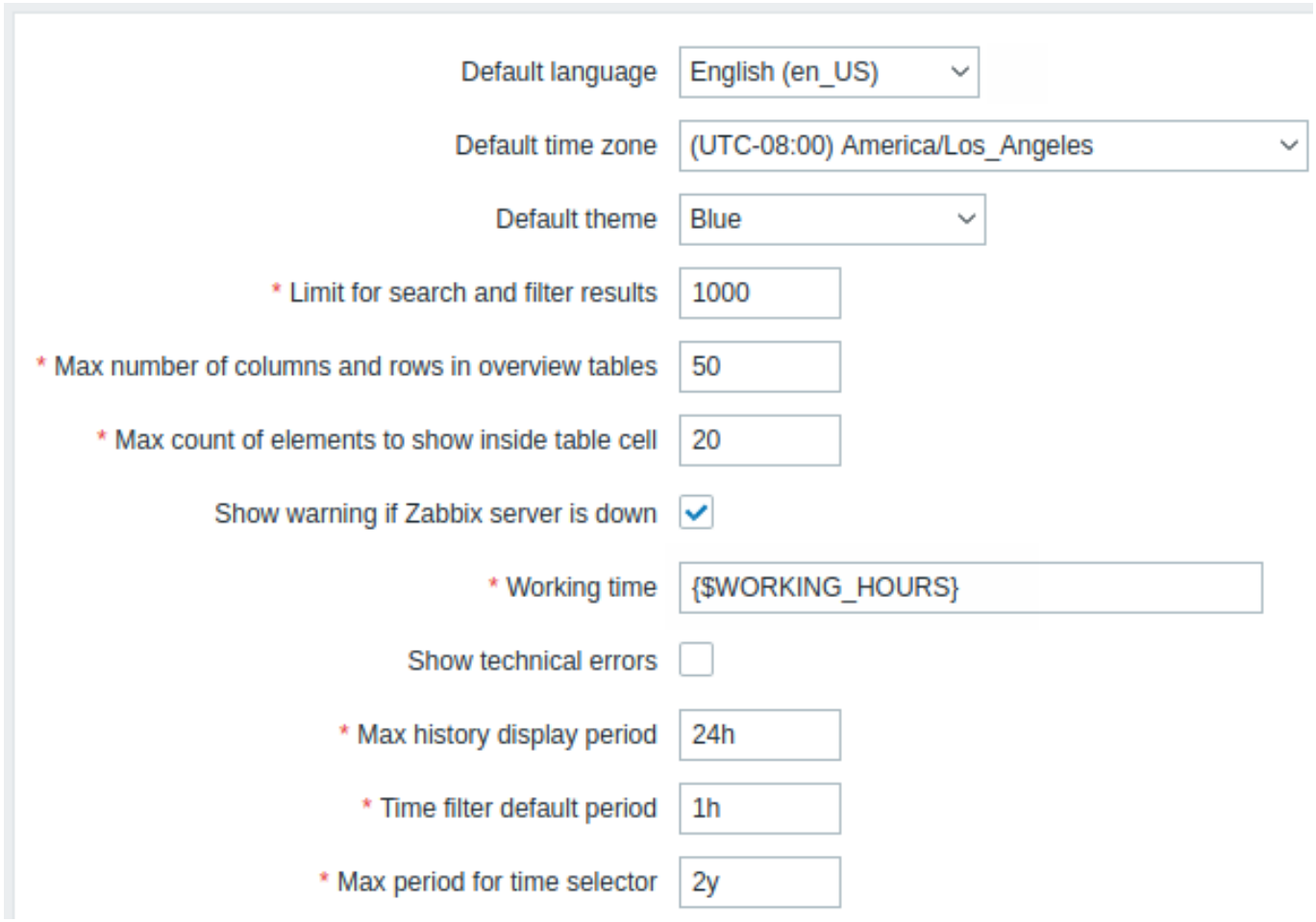
Panoramica

La sezione *Administration* > *General* contiene diverse sottosezioni per impostare i valori predefiniti relativi al frontend e personalizzare Zabbix.

L'elenco delle sottosezioni disponibili viene visualizzato facendo clic su *General* nella sezione del menu *Administration*. È inoltre possibile passare da una sottosezione all'altra utilizzando il menu a discesa del titolo nell'angolo in alto a sinistra.

GUI

Questa sezione consente la personalizzazione di diversi valori predefiniti relativi al frontend.



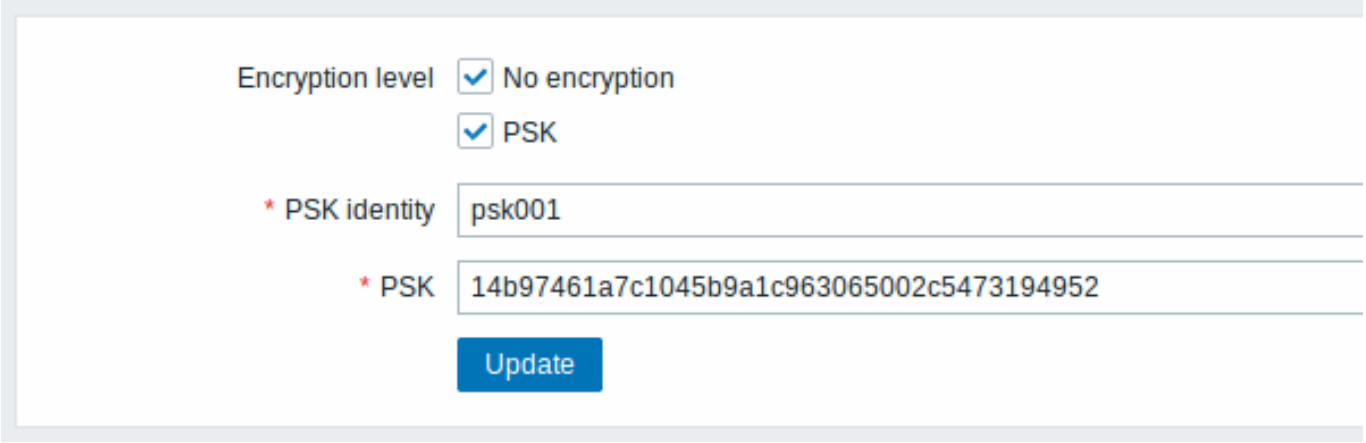
Parametri di configurazione:

Parameter	Description
Default language	Lingua predefinita per gli utenti che non hanno specificato una lingua nei propri profili e per gli utenti guest. Per ulteriori informazioni, vedere Installazione di lingue aggiuntive del frontend.
Default time zone	Fuso orario predefinito per gli utenti che non hanno specificato un fuso orario nei propri profili e per gli utenti guest.
Default theme	Tema predefinito per gli utenti che non hanno specificato un tema nei propri profili e per gli utenti guest.
Limit for search and filter results	Numero massimo di elementi (righe) che verranno visualizzati in un elenco dell'interfaccia web, ad esempio in *Data collection* > *Hosts*. *Nota*: se impostato, ad esempio, su '50', solo i primi 50 elementi verranno visualizzati in tutti gli elenchi del frontend interessati. Se un elenco contiene più di cinquanta elementi, ciò sarà indicato dal segno '+' in *"Displaying 1 to 50 of **50+** found"*. Inoltre, se viene utilizzato il filtro e sono presenti comunque più di 50 corrispondenze, verranno visualizzati solo i primi 50. Si noti che l'aumento del valore di questo parametro può comportare un peggioramento delle prestazioni e un maggiore consumo di memoria sul lato del web server.
Max number of columns and rows in overview tables	Numero massimo di colonne e righe da visualizzare nel widget della dashboard *Trigger overview*. Lo stesso limite si applica sia alle colonne sia alle righe. Se esistono più righe e/o colonne di quelle visualizzate, il sistema mostrerà un avviso nella parte inferiore della tabella: "Not all results are displayed. Please provide more specific search criteria."
Max count of elements to show inside table cell	Per le voci visualizzate in una singola cella della tabella, non verrà mostrato più di quanto configurato qui.
Show warning if Zabbix server is down	Questo parametro abilita la visualizzazione di un messaggio di avviso in una finestra del browser se non è possibile raggiungere il server Zabbix (possibilmente non operativo). Il messaggio rimane visibile anche se l'utente scorre la pagina verso il basso. Quando vi si passa sopra con il puntatore, il messaggio viene temporaneamente nascosto per mostrare il contenuto sottostante.

Parameter	Description
Working time	Questo parametro a livello di sistema definisce l'orario di lavoro. Nei grafici, l'orario di lavoro è visualizzato con sfondo bianco e il tempo non lavorativo con sfondo grigio. Vedere la pagina Specifica del periodo di tempo per la descrizione del formato dell'ora. Sono supportate le macro utente.
Show technical errors	Se selezionato, tutti gli utenti registrati potranno visualizzare gli errori tecnici (PHP/SQL). Se non selezionato, le informazioni sono disponibili solo per gli Zabbix Super Admin e per gli utenti appartenenti ai gruppi di utenti con debug mode abilitata.
Max history display period	Periodo massimo per il quale visualizzare i dati storici in *Monitoring* > *Latest data*, le informazioni della web scenario dell'host in *Monitoring* > *Hosts* e nel widget della dashboard *Top items*. Intervallo consentito: 24 ore (predefinito) - 1 settimana. Sono supportati i suffissi temporali, ad esempio 1w (una settimana), 36h (36 ore).
Time filter default period	Periodo di tempo da utilizzare per impostazione predefinita nei grafici e nelle dashboard. Intervallo consentito: 1 minuto - 10 anni (predefinito: 1 ora). Sono supportati i suffissi temporali, ad esempio 10m (dieci minuti), 5w (cinque settimane). Nota: quando un utente modifica il periodo di tempo durante la visualizzazione di un grafico, questo periodo viene memorizzato come preferenza dell'utente, sostituendo il valore predefinito globale o una precedente selezione dell'utente.
Max period for time selector	Periodo massimo disponibile per grafici e dashboard. Gli utenti non potranno visualizzare dati più vecchi. Intervallo consentito: 1 anno - 10 anni (predefinito: 2 anni). Sono supportati i suffissi temporali, ad esempio 1y (un anno), 365w (365 settimane).

Autoregistrazione

In questa sezione è possibile configurare il livello di crittografia per l'autoregistrazione dell'agent attivo.



I parametri contrassegnati da un asterisco sono obbligatori.

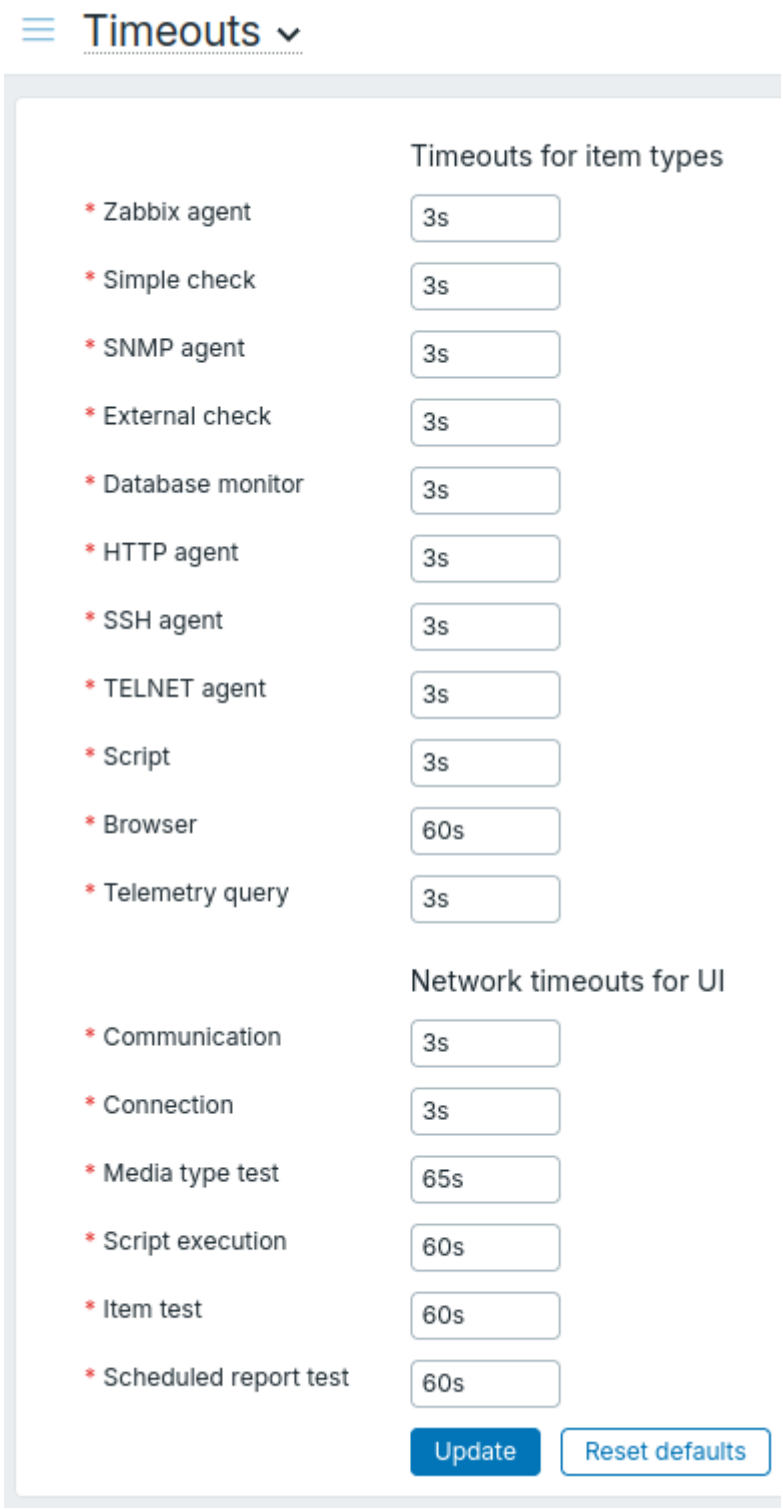
Parametri di configurazione:

Parametro	Descrizione
Livello di crittografia	Selezionare una o entrambe le opzioni per il livello di crittografia: **Nessuna crittografia** - sono consentite connessioni non crittografate **PSK** - sono consentite connessioni crittografate TLS con una chiave precondivisa
Identità PSK	Immettere la stringa di identità della chiave precondivisa. Questo campo è disponibile solo se 'PSK' è selezionato come *Livello di crittografia*. Non inserire informazioni sensibili nell'identità PSK, poiché viene trasmessa non crittografata sulla rete per informare il destinatario su quale PSK utilizzare.
PSK	Immettere la chiave precondivisa (un numero pari di caratteri esadecimali). Lunghezza massima: 512 cifre esadecimali (PSK da 256 byte) se Zabbix utilizza la libreria GnuTLS o OpenSSL, 64 cifre esadecimali (PSK da 32 byte) se Zabbix utilizza la libreria mbed TLS (PolarSSL). Esempio: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Questo campo è disponibile solo se 'PSK' è selezionato come *Livello di crittografia*.

Vedi anche: Autoregistrazione sicura

Timeout

In questa sezione è possibile impostare i timeout globali per i tipi di item e i timeout di rete. Tutti i campi di questo modulo sono obbligatori.

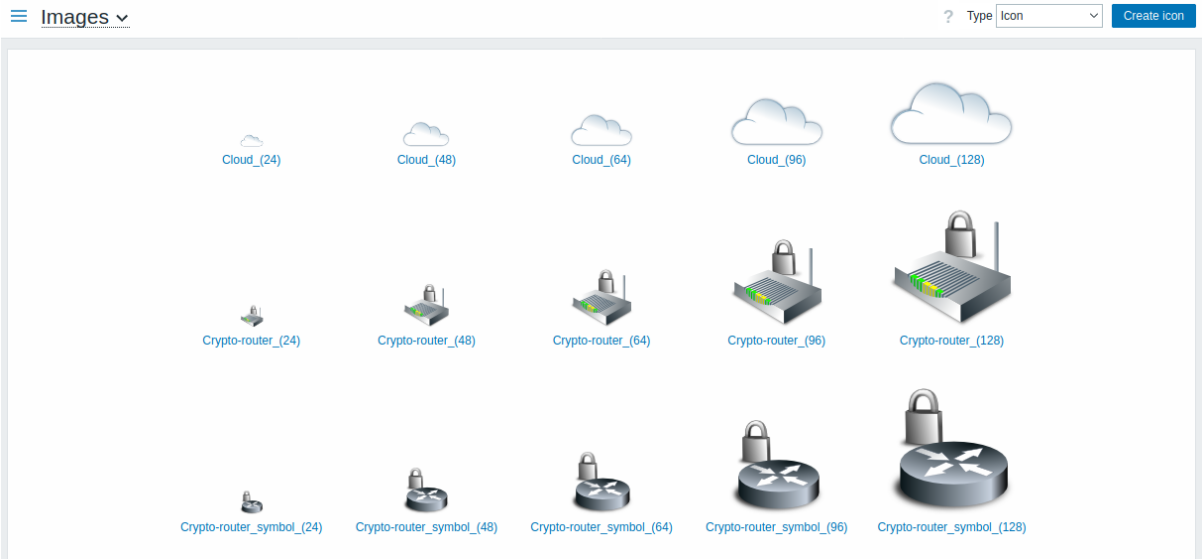


Parametro	Descrizione
Timeout per i tipi di item	Quanti secondi Zabbix deve attendere prima di interrompere il controllo (in base al relativo tipo). Per ulteriori dettagli e limitazioni, vedere item timeout. Intervallo consentito: 1 - 600s (predefinito: 3s; predefinito per gli item browser: 60s). Sono supportati i suffissi di tempo, ad esempio 30s, 1m, e le macro utente. Si noti che i timeout a livello di proxy, se configurati, hanno la precedenza su queste impostazioni globali. Inoltre, se sono configurati timeout per specifici item, questi hanno la precedenza sia sulle impostazioni globali sia su quelle del proxy.
Timeout di rete per UI	

Parametro		Descrizione
	Comunicazione	Quanti secondi attendere prima di chiudere un socket inattivo (se in precedenza è stata stabilita una connessione con Zabbix server, ma frontend non riesce a completare l'operazione di lettura/invio dei dati entro questo intervallo, la connessione verrà interrotta). Intervallo consentito: 1 - 300s (predefinito: 3s).
	Connessione	Quanti secondi attendere prima di interrompere un tentativo di connessione a Zabbix server. Intervallo consentito: 1 - 30s (predefinito: 3s).
	Test del tipo di media	Quanti secondi attendere una risposta durante il test di un tipo di media. Intervallo consentito: 1 - 300s (predefinito: 65s).
	Esecuzione script	Quanti secondi attendere una risposta durante l'esecuzione di uno script. Intervallo consentito: 1 - 300s (predefinito: 60s). Questo timeout riguarda l'intera catena di esecuzione dello script, che può avere una durata variabile. Ad esempio, se lo script viene eseguito sull'agent, si tratta di un roundtrip attraverso server (eventualmente anche proxy) verso l'agent e ritorno. Per gli script eseguiti sull'agent, si applica il timeout di server o proxy. Per gli script eseguiti solo su un agent attivo, è probabile che il timeout predefinito di server/proxy debba essere aumentato. Il timeout di server/proxy deve essere superiore alla frequenza di aggiornamento del controllo attivo, altrimenti il timeout di server/proxy potrebbe essere superato prima ancora che l'agent attivo riceva lo script. Vedere anche: Script timeout.
	Test item	Quanti secondi attendere i dati restituiti durante il test di un item. Intervallo consentito: 1 - 600s (predefinito: 60s).
	Test report pianificato	Quanti secondi attendere i dati restituiti durante il test di un report pianificato. Intervallo consentito: 1 - 300s (predefinito: 60s).

Immagini

La sezione Immagini mostra tutte le immagini disponibili in Zabbix.
Le immagini sono memorizzate nel database.

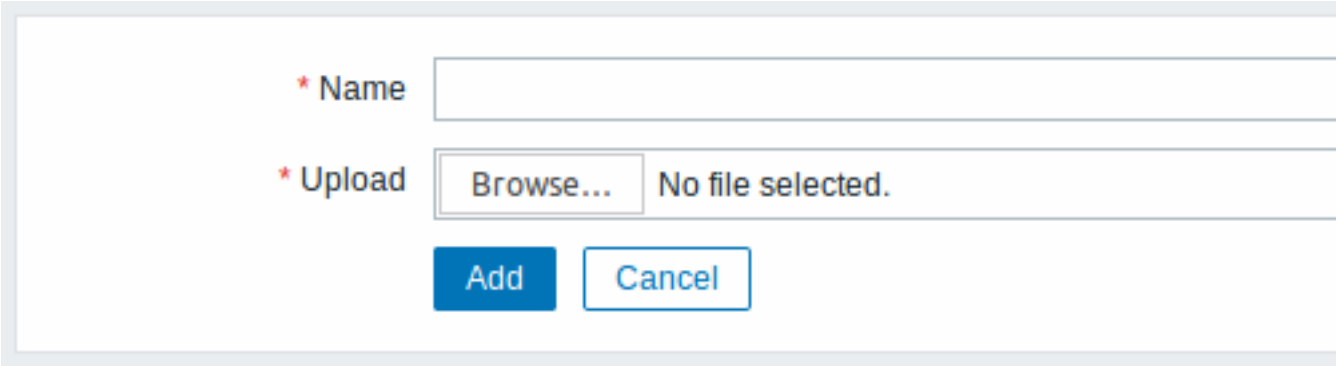


Il menu a discesa *Tipo* consente di passare dalle immagini icona alle immagini di sfondo e viceversa:

- Le icone vengono utilizzate per visualizzare gli elementi delle mappe di rete.
- Gli sfondi vengono utilizzati come immagini di sfondo delle mappe di rete.

Aggiunta di un'immagine

È possibile aggiungere una propria immagine facendo clic sul pulsante *Crea icona* o *Crea sfondo* nell'angolo superiore destro.



Attributi dell'immagine:

Parametro	Descrizione
Nome	Nome univoco dell'immagine.
Carica	Selezionare dal sistema locale il file (PNG, JPEG, GIF o WebP) da caricare in Zabbix. *Nota*: potrebbe essere possibile caricare altri formati, che verranno convertiti in PNG durante il caricamento. Per l'elaborazione delle immagini viene utilizzata la libreria GD; pertanto, i formati supportati dipendono dalla versione della libreria utilizzata (Zabbix richiede la versione 2.0.28 o superiore).

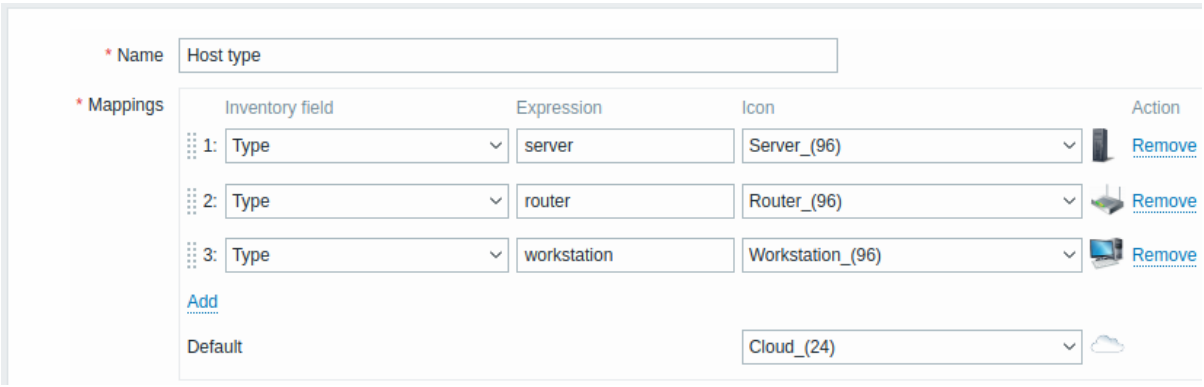
Note:
La dimensione massima del file da caricare è limitata dal valore di `ZBX_MAX_IMAGE_SIZE`, pari a 1024x1024 byte o 1 MB. Il caricamento di un'immagine potrebbe non riuscire se la dimensione dell'immagine è prossima a 1 MB e il parametro di configurazione MySQL `max_allowed_packet` è impostato sul valore predefinito di 1 MB. In questo caso, aumentare il valore del parametro `max_allowed_packet`.

Mappatura delle icone

Questa sezione consente di creare la mappatura di determinati host con determinate icone. Le informazioni del campo inventario dell'host vengono utilizzate per creare la mappatura.

Le mappature possono quindi essere utilizzate nella configurazione della mappa di rete per assegnare automaticamente le icone appropriate agli host corrispondenti.

Per creare una nuova mappa di icone, fare clic su *Create icon map* nell'angolo in alto a destra.



Parametri di configurazione:

Parameter	Description
Name	Nome univoco della mappa di icone.
Mappings	Un elenco di mappature. L'ordine delle mappature determina quale avrà la priorità. È possibile spostare le mappature su e giù nell'elenco con il trascinamento.
Inventory field	Campo dell'inventario dell'host che verrà esaminato per cercare una corrispondenza.
Expression	Espressione regolare che descrive la corrispondenza.
Icon	Icona da usare se viene trovata una corrispondenza per l'espressione.
Default	Icona predefinita da usare.

Espressioni regolari

Questa sezione consente di creare espressioni regolari personalizzate che possono essere utilizzate in diversi punti del frontend. Per i dettagli, vedere la sezione Espressioni regolari.

Opzioni di visualizzazione dei trigger

Questa sezione consente di personalizzare il modo in cui lo stato dei trigger viene visualizzato nel frontend e i nomi e i colori della severità del trigger.

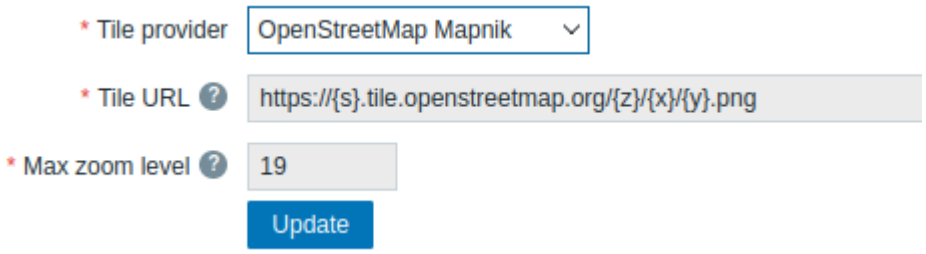


Parameter	Description
Use custom event status colors	Selezionando questo parametro si attiva la personalizzazione dei colori per i problemi riconosciuti/non riconosciuti.
Unacknowledged PROBLEM events, *Acknowledged PROBLEM events*, *Unacknowledged RESOLVED events*, *Acknowledged RESOLVED events*	Immettere un nuovo codice colore oppure fare clic sul colore per selezionarne uno nuovo dalla tavolozza disponibile. Se la casella di controllo *blinking* è selezionata, i trigger lampeggeranno per un certo periodo dopo il cambio di stato, per essere più visibili.
Display OK triggers for	Periodo di tempo per la visualizzazione dei trigger OK. Intervallo consentito: 0 - 24 ore. Sono supportati i suffissi di tempo, ad esempio 5m, 2h, 1d.
On status change triggers blink for	Durata del lampeggiamento del trigger. Intervallo consentito: 0 - 24 ore. Sono supportati i suffissi di tempo, ad esempio 5m, 2h, 1d.
Not classified, *Information*, *Warning*, *Average*, *High*, *Disaster*	Nomi e/o colori personalizzati delle severità da visualizzare al posto di quelli predefiniti del sistema. Immettere un nuovo codice colore oppure fare clic sul colore per selezionarne uno nuovo dalla tavolozza disponibile. Si noti che i nomi personalizzati delle severità inseriti qui verranno utilizzati in tutte le impostazioni locali. Se è necessario tradurli in altre lingue per determinati utenti, vedere la pagina Personalizzazione delle severità dei trigger.

Mappe geografiche

Questa sezione consente di selezionare il provider del servizio di tile della mappa geografica e di configurare le impostazioni del provider per il widget della dashboard Geomap dashboard widget. Per fornire la visualizzazione tramite le mappe geografiche,

Zabbix utilizza la libreria open-source JavaScript Leaflet per mappe interattive. Si noti che Zabbix non ha alcun controllo sulla qualità delle immagini fornite dai provider di tile di terze parti, inclusi i provider di tile predefiniti.



Parameter	Description
Tile provider	Selezionare uno dei provider di servizi di tile disponibili oppure selezionare *Other* per aggiungere un altro provider di tile o tile self-hosted (vedere Using a custom tile service provider).
Tile URL	Il modello di URL (fino a 2048 caratteri) per caricare e visualizzare il layer di tile sulle mappe geografiche. Questo campo è modificabile solo se *Tile provider* è impostato su *Other*. Sono supportati i seguenti segnaposto: *{s}* rappresenta uno dei sottodomini disponibili; *{z}* rappresenta il parametro del livello di zoom nell'URL; *{x}* e *{y}* rappresentano le coordinate del tile; *{r}* può essere usato per aggiungere "@2x" all'URL e caricare tile retina. Esempio: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	Testo di attribuzione del provider di tile da visualizzare in una piccola casella di testo sulla mappa. Questo campo è visibile solo se *Tile provider* è impostato su *Other*.
Max zoom level	Livello massimo di zoom della mappa. Questo campo è modificabile solo se *Tile provider* è impostato su *Other*.

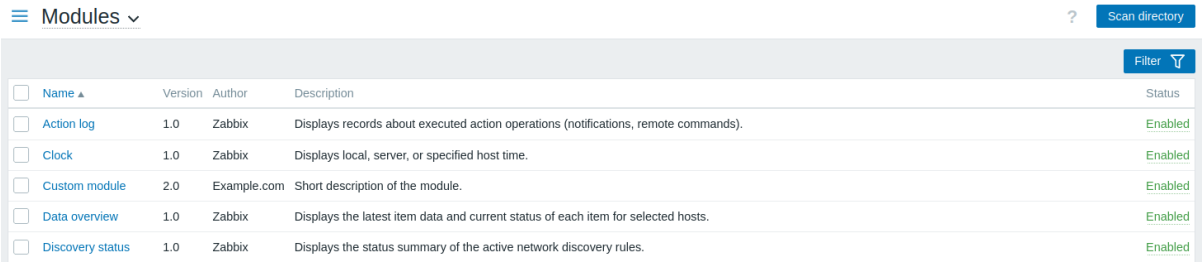
Utilizzo di un provider di servizi di tile personalizzato

Il widget Geomap è in grado di caricare immagini raster di tile da un servizio provider di tile personalizzato ospitato in locale o di terze parti. Per usare un servizio provider di tile di terze parti personalizzato oppure una cartella o un server di tile ospitato in locale, selezionare *Other* nel campo *Tile provider* e specificare l'URL personalizzato nel campo *Tile URL* usando i segnaposto appropriati.

Moduli

Questa sezione consente di amministrare i moduli frontend personalizzati e integrati.

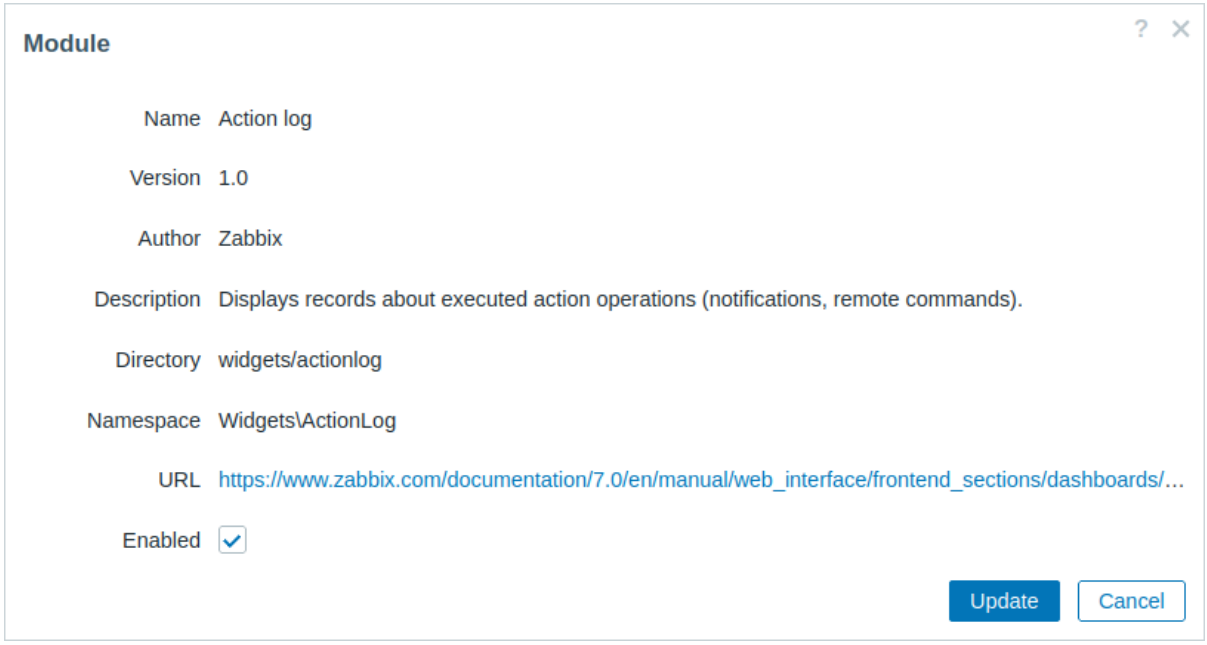
Note:
Se il flag delle funzionalità `$ZBX_FEATURE_FLAGS['modules_config_enabled']` è impostato su `false` nel file di configurazione del frontend, l'accesso a questa sezione è disabilitato.



Fare clic su *Scansiona directory* per registrare o annullare la registrazione dei moduli personalizzati. I moduli registrati verranno visualizzati nell'elenco; i moduli per i quali è stata annullata la registrazione verranno rimossi dall'elenco.

Fare clic sullo stato del modulo nell'elenco per abilitarlo o disabilitarlo. È inoltre possibile abilitare o disabilitare più moduli contemporaneamente selezionandoli nell'elenco e facendo clic sui pulsanti *Abilita/Disabilita* sotto l'elenco.

Fare clic sul nome del modulo nell'elenco per visualizzarne i dettagli in una finestra pop-up.

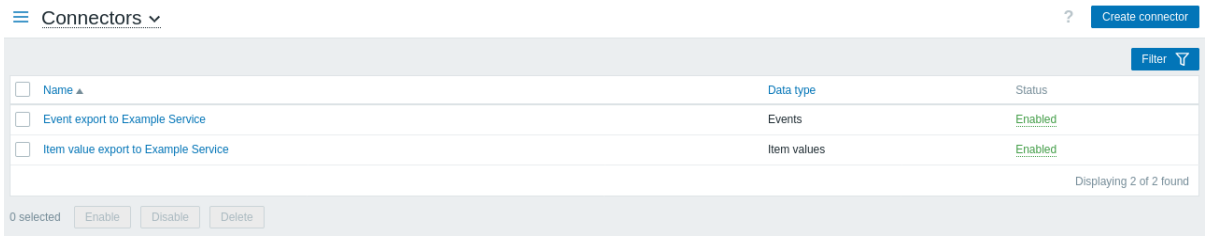


Lo stato del modulo può essere aggiornato anche nella finestra pop-up dei dettagli del modulo; a tale scopo, selezionare o deselezionare la casella di controllo *Abilitato*, quindi fare clic su *Aggiorna*.

È possibile filtrare i moduli per nome o stato (abilitato/disabilitato).

Connettori

Questa sezione consente di configurare i connettori per lo streaming dei dati di Zabbix verso sistemi esterni tramite HTTP.



Fare clic su *Crea connettore* per configurare un nuovo connettore.

È possibile filtrare i connettori per nome o stato (abilitato/disabilitato). Fare clic sullo stato del connettore nell'elenco per abilitarlo/disabilitarlo. È inoltre possibile abilitare/disabilitare più connettori contemporaneamente selezionandoli nell'elenco e facendo clic sui pulsanti *Abilita/Disabilita* sotto l'elenco.

Altro

Questa sezione consente di configurare vari altri parametri del frontend.



Parametro	Descrizione
URL del frontend	URL (fino a 2048 caratteri) dell'interfaccia web di Zabbix. Questo parametro è utilizzato dal servizio web di Zabbix per la comunicazione con il frontend e deve essere specificato per abilitare i report pianificati.
Gruppo per gli host rilevati	Gli host rilevati da network discovery e agent autoregistration verranno automaticamente inseriti nel gruppo host selezionato qui.
Modalità predefinita dell'inventario host	Modalità predefinita per l'inventario host. Verrà applicata ogni volta che un nuovo host o prototipo di host viene creato dal server o dal frontend, a meno che non venga sovrascritta durante il rilevamento/la registrazione automatica dell'host tramite l'operazione *Set host inventory mode*.

Parametro	Descrizione
Gruppo utenti per il messaggio di database non disponibile	Gruppo utenti per l'invio del messaggio di allarme di database non disponibile. Zabbix server dipende dalla disponibilità del database backend. Non può funzionare senza un database. Se il database non è disponibile, gli utenti selezionati possono essere avvisati da Zabbix. Le notifiche verranno inviate al gruppo utenti impostato qui utilizzando le voci media utente abilitate. Le notifiche vengono trasmesse utilizzando i seguenti tipi di supporto (quando abilitati): email; SMS; script di avviso personalizzati. Anche se una voce media webhook è configurata e abilitata, non attiverà notifiche. Zabbix server non si fermerà; attenderà che il database torni disponibile per continuare l'elaborazione. La notifica è composta dal seguente contenuto: `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> non viene aggiunto al messaggio se è definito come valore vuoto e <DB Port> non viene aggiunto se è il valore predefinito ("0"). L'alert manager (uno speciale processo di Zabbix server) tenta di stabilire una nuova connessione al database ogni 10 secondi. Se il database è ancora non disponibile, l'alert manager ripete l'invio degli avvisi, ma non più frequentemente di una volta ogni 15 minuti.
Registra trap SNMP senza corrispondenza	Registra il trap SNMP se non sono state trovate interfacce SNMP corrispondenti.

Autorizzazione

Parametro	Descrizione
Tentativi di accesso *Intervallo di blocco dell'accesso*	Numero di tentativi di accesso non riusciti prima che la possibilità di accedere venga bloccata. Periodo di tempo durante il quale l'accesso sarà vietato quando viene superato il limite di *Tentativi di accesso*. Intervallo consentito: 30 - 3600 secondi. Sono supportati i suffissi di tempo, ad es. 90s, 5m, 1h.

Archiviazione dei segreti

Parameter	Description
Vault provider	Selezionare il software di gestione dei segreti per archiviare i valori delle user macro - *HashiCorp Vault* (predefinito) oppure *CyberArk Vault*.
Resolve secret vault macros by	Risolvi le macro del vault dei segreti tramite: **Zabbix server** - i segreti vengono recuperati da Zabbix server e inoltrati ai proxy quando necessario (predefinito); **Zabbix server and proxy** - i segreti vengono recuperati sia da Zabbix server sia dai proxy, consentendo loro di risolvere le macro in modo indipendente.

Vedi anche: Storage of secrets.

Sicurezza

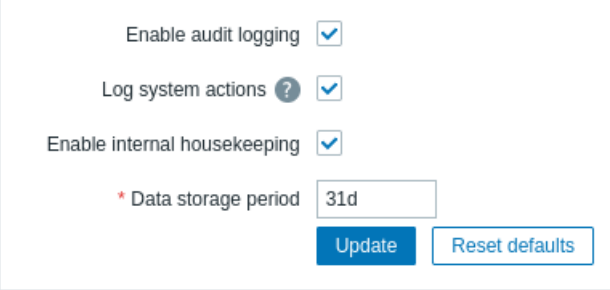
Parameter	Description
Convalida schemi URI	Deselezionare questa casella di controllo per disabilitare la convalida degli schemi URI (abilitata per impostazione predefinita). Se selezionata, è possibile specificare un elenco separato da virgole degli schemi URI consentiti (predefinito: http,https,ftp,file,mailto,tel,ssh). Si applica a tutti i campi nel frontend in cui vengono utilizzati URI (ad esempio, gli URL degli elementi della mappa).

Parameter	Description
Usa l'header HTTP X-Frame-Options	Deselezionare questa casella di controllo per disabilitare l'header HTTP X-Frame-options (scelta non consigliata). Se selezionata, è possibile specificare il valore dell'header HTTP X-Frame-options. Valori supportati: **SAMEORIGIN** (predefinito) oppure **'self'** (deve essere racchiuso tra apici singoli) - la pagina può essere visualizzata in un frame solo sulla stessa origine della pagina stessa; **DENY** oppure **'none'** (deve essere racchiuso tra apici singoli) - la pagina non può essere visualizzata in un frame, indipendentemente dal sito che tenta di farlo; **una stringa di nomi host separati da spazi**; aggiungendo **'self'** (deve essere racchiuso tra apici singoli) all'elenco si consente di visualizzare la pagina in un frame sulla stessa origine della pagina stessa. Si noti che l'uso di **'self'** o **'none'** senza apici singoli farà sì che vengano considerati come nomi host.
Usa il sandboxing iframe	Deselezionare questa casella di controllo per disabilitare l'inserimento del contenuto dell'URL recuperato in sandbox (scelta non consigliata). Se selezionata, è possibile specificare le eccezioni del sandboxing iframe; le restrizioni non specificate continueranno comunque a essere applicate. Se questo campo è vuoto, si applicano tutte le restrizioni dell'attributo sandbox. Per ulteriori informazioni, vedere la descrizione dell'attributo `sandbox`.

Registro di audit

Panoramica

Questa sezione consente di configurare le impostazioni del log di audit.



Sono disponibili i seguenti parametri:

Parametro	Descrizione
Abilita registrazione audit	Abilita (predefinito) o disabilita la registrazione audit.
Registra azioni di sistema	Abilita (predefinito) o disabilita la registrazione audit delle attività di low-level discovery, network discovery e autoregistrazione eseguite dal server (utente System).
Abilita housekeeping interno	Abilita (predefinito) o disabilita l'housekeeping interno per i record del log di audit.
Periodo di conservazione dei dati	Numero di giorni per cui i record del log di audit devono essere conservati prima di essere rimossi dall'housekeeper. Obbligatorio se l'housekeeping è abilitato. Predefinito: 31 giorni.

I parametri di compressione del log di audit sono disponibili nella sezione *Administration > Housekeeping*, nel blocco *History, trends and audit log compression*, che diventa visibile quando viene utilizzato TimescaleDB.

Manutenzione periodica

Panoramica

La sezione *Administration > Housekeeping* consente di configurare il housekeeping interno di Zabbix, che rimuove automaticamente dal database i dati obsoleti o eliminati dall'utente. Questo impedisce al database di crescere indefinitamente, evitando un uso inutile dello spazio su disco e un peggioramento delle prestazioni del database.

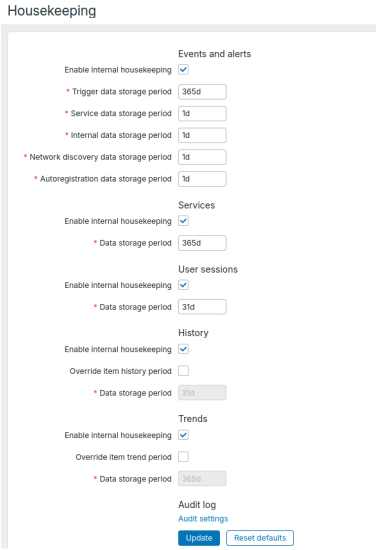
Il housekeeping può essere abilitato e configurato per i seguenti tipi di dati:

- Eventi e alert provenienti da trigger, servizi, dati interni, discovery di rete e autoregistrazione
- Servizi
- Sessioni utente
- History e trends per gli item

Note:
Il housekeeping del log di audit è configurato separatamente.

Configurazione

Sono disponibili i seguenti parametri:



Parameter	Description
Enable internal housekeeping	Abilita o disabilita la manutenzione interna (abilitata per impostazione predefinita). Se abilitata, la procedura di housekeeping rimuove automaticamente dal database i dati che superano il *Data storage period*.
Data storage period	Specifica per quanto tempo conservare i dati prima che vengano rimossi dall'housekeeper. Obbligatorio se la manutenzione interna è abilitata. Intervallo: da 1 giorno (1 ora per la history) a 25 anni; oppure "0". Sono supportati i suffissi temporali (ad esempio, 1d, 1w). Per *Events and alerts*, il periodo di conservazione dei dati è impostato separatamente per trigger, servizi, dati interni, network discovery e autoregistrazione. L'housekeeper rimuove solo gli eventi non associati a problemi. Ad esempio, un evento di problema/ripristino più vecchio del *Data storage period* non verrà rimosso se è associato a un problema non risolto. Quando l'housekeeper rimuove entità obsolete, elimina prima i problemi e poi gli eventi. Per *History* e *Trends*, i periodi di conservazione dei dati determinano anche per quanto tempo i dati restano visibili in *Monitoring > Latest data*, anche se la manutenzione interna è disabilitata. Si noti che quando un item o un trigger viene eliminato, vengono eliminati anche i relativi problemi e gli eventi associati.
Override item history period	Se abilitata, il periodo di conservazione di history/trend specificato nella configurazione dell'item viene sostituito dall'impostazione *Data storage period* (tranne per gli item con l'opzione *Do not store* abilitata). Questa opzione può essere utilizzata anche quando la manutenzione interna è disabilitata e viene usato un housekeeper esterno.
Override item trend period	

Se si utilizza **ClickHouse**, diventa disponibile la sezione *ClickHouse data storage period*. Mostra il periodo di conservazione dei

dati per ciascun tipo di valore dell'item. Per modificare il periodo, vedere Configurazione del periodo di conservazione dei dati in ClickHouse. Si noti che l'housekeeper non elimina i dati da ClickHouse.

Se si utilizza **Elasticsearch**, diventa disponibile la sezione *Elasticsearch data storage period*. Mostra i tipi di valore degli item salvati in Elasticsearch, con il periodo di conservazione dei dati visualizzato come *Unknown*. Per controllare per quanto tempo i dati vengono conservati, vedere Index Lifecycle Management. Si noti che l'housekeeper non elimina i dati da Elasticsearch.

Se si utilizza **TimescaleDB**, diventa disponibile la sezione *History, trends and audit log compression*.

Note:
Per TimescaleDB, abilitare *Override item history period*, *Override item trend period* e *Enable internal housekeeping* per history e trends per sfruttare appieno il partizionamento automatico. Se queste opzioni sono disabilitate, i dati conservati nelle tabelle di history e trends continueranno a essere partizionati, ma l'housekeeper non eliminerà le partizioni obsolete e verranno visualizzati avvisi di configurazione. Quando l'eliminazione delle partizioni obsolete è abilitata, Zabbix server e frontend non terranno più traccia degli item eliminati e la history di tali item verrà cancellata quando viene eliminata una partizione obsoleta.

L'housekeeper viene eseguito automaticamente ogni ora per impostazione predefinita. È possibile modificare la frequenza di esecuzione impostando il parametro di configurazione di Zabbix server `HousekeepingFrequency`.

Il pulsante *Reset defaults* consente di ripristinare eventuali modifiche apportate.

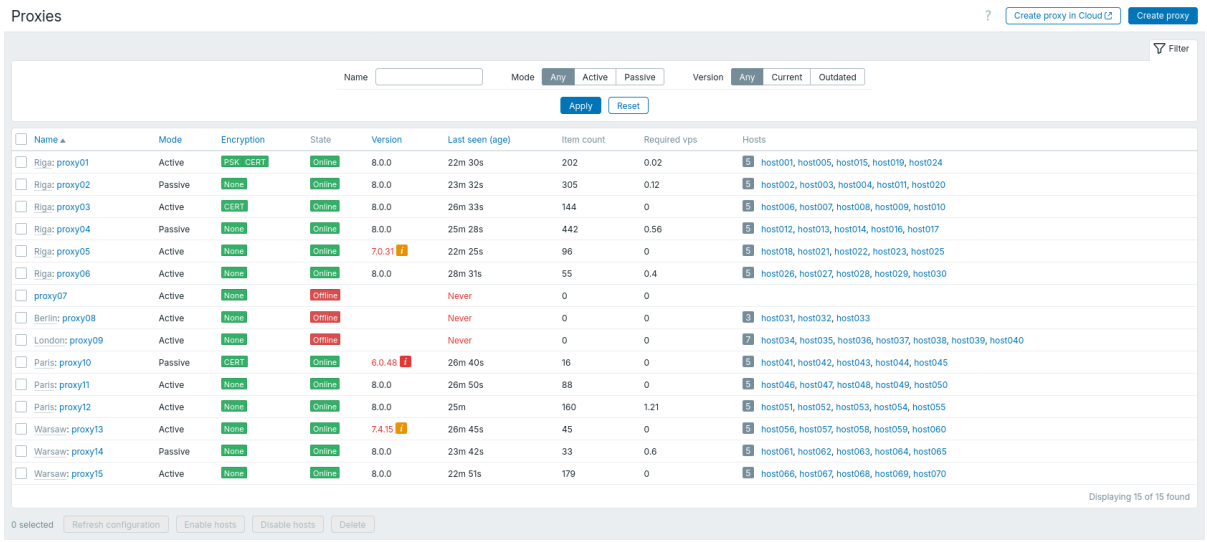
Proxy

Panoramica

Nella sezione *Amministrazione* > *Proxy* è possibile configurare i proxy per il monitoraggio distribuito nel frontend di Zabbix.

Proxy

Viene visualizzato un elenco dei proxy esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del proxy. Facendo clic sul nome del proxy si apre il modulo di configurazione del proxy. Se il proxy appartiene a un gruppo di proxy, il nome del gruppo viene visualizzato prima del nome del proxy come collegamento grigio. Facendo clic sul nome del gruppo si apre il modulo di configurazione del gruppo di proxy.
Modalità	Modalità del proxy: *Attiva* o *Passiva*.
Crittografia	Stato della crittografia per le connessioni dal proxy: **Nessuna** - nessuna crittografia; **PSK** - utilizzo di una chiave precondivisa; **Cert** - utilizzo di un certificato.

Colonna	Descrizione
Stato	Stato del proxy: **Sconosciuto** - il proxy è stato creato mentre il server Zabbix era inattivo oppure il server non ha ancora aggiornato lo stato; **Online** - il proxy ha comunicato con il server Zabbix (il proxy passivo ha risposto alla richiesta del server; il proxy attivo ha inviato una richiesta) durante il periodo di failover; **Offline** - il proxy non ha comunicato con il server Zabbix durante il periodo di failover.
Versione	Versione del proxy (numero di versione a tre cifre). Se il proxy è obsoleto o non supportato, il numero di versione viene evidenziato (in rosso) e viene visualizzata un'icona di stato informativa (gialla o rossa). Passare il mouse sull'icona per visualizzare i dettagli.
Ultima visualizzazione (età)	L'ora in cui il proxy è stato rilevato l'ultima volta dal server.
Numero di item	Il numero di item abilitati sugli host abilitati assegnati al proxy.
vps richiesti	Prestazioni richieste al proxy (il numero di valori che devono essere raccolti al secondo).
Host	Numero di host abilitati assegnati al proxy e elenco degli host monitorati dal proxy. Facendo clic sul nome dell'host si apre il modulo di configurazione dell'host.

Per configurare un nuovo proxy, fare clic sul pulsante *Crea proxy* nell'angolo superiore destro. Per distribuire un nuovo proxy in Zabbix Cloud, fare clic sul pulsante *Crea proxy in Cloud* accanto ad esso. Zabbix Cloud si apre in una nuova scheda del browser, dove è possibile creare un nodo proxy. Dopo aver creato il nodo proxy, configurare il proxy utilizzando il pulsante *Crea proxy*. Per ulteriori informazioni, vedere Configurazione di un proxy Zabbix Cloud.

Opzioni di modifica massiva

I pulsanti sotto l'elenco offrono alcune opzioni di modifica massiva:

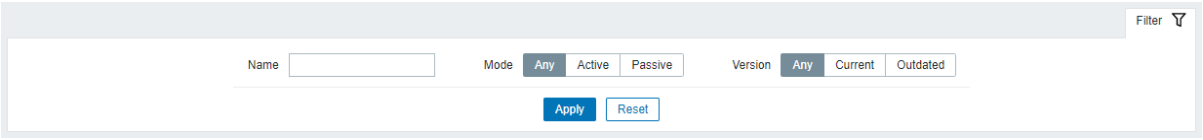
- *Aggiorna configurazione* - aggiorna la configurazione dei proxy;
- *Abilita host* - cambia lo stato degli host monitorati dal proxy in *Monitorato*;
- *Disabilita host* - cambia lo stato degli host monitorati dal proxy in *Non monitorato*;
- *Elimina* - elimina i proxy.

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi proxy, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo i proxy di interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Il link *Filtro* è disponibile sopra l'elenco dei proxy. Facendo clic su di esso, viene visualizzato un filtro che consente di filtrare i proxy per nome, modalità e versione. Si noti che l'opzione di filtro *Obsoleto* visualizza sia i proxy obsoleti (parzialmente supportati) sia quelli non supportati.



Gruppi di proxy

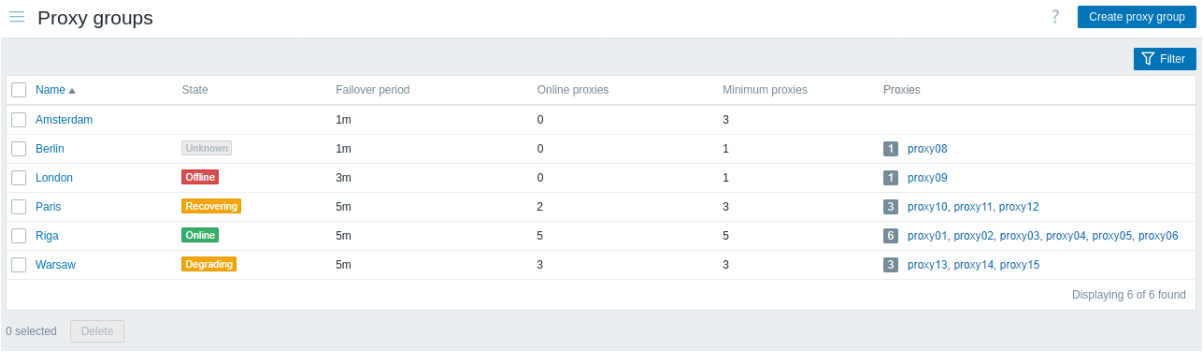
Panoramica

In *Administration* > *Proxy groups* è possibile configurare i gruppi di proxy.

I gruppi di proxy vengono utilizzati nel bilanciamento del carico dei proxy con distribuzione automatica degli host tra i proxy e alta disponibilità tra i proxy.

Gruppi di proxy

Viene visualizzato un elenco dei gruppi di proxy esistenti con i relativi dettagli.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del gruppo di proxy. Facendo clic sul nome del gruppo di proxy si apre il modulo di configurazione del gruppo di proxy.
Stato	Stato del gruppo di proxy (visualizzato se il gruppo contiene almeno un proxy): **Sconosciuto** - il gruppo è stato creato mentre Zabbix server era inattivo oppure server non ha ancora aggiornato lo stato; **Online** - il numero minimo di proxy ha comunicato con Zabbix server durante il periodo di failover; **In peggioramento** - il gruppo passerà a breve offline perché il numero di proxy online è sceso al di sotto della soglia *Proxy minimi*; **Offline** - un numero di proxy inferiore al minimo ha comunicato con Zabbix server durante il periodo di failover; **Ripristino** - il gruppo tornerà online a breve perché il numero di proxy online ha raggiunto la soglia *Proxy minimi*. Lo stato del gruppo di proxy può essere monitorato con gli item zabbix[proxy group,<name>,state] e zabbix[proxy group,discovery].
Periodo di failover	Periodo durante il quale un proxy nel gruppo di proxy deve comunicare con Zabbix server per essere considerato online.
Proxy online	Numero di proxy online (visualizzato in rosso se inferiore al minimo del gruppo).
Proxy minimi	Numero minimo di proxy online richiesto per mantenere online il gruppo di proxy.
Proxy	Numero di proxy nel gruppo e elenco dei proxy nel gruppo, con collegamenti al modulo di configurazione del proxy. Il numero massimo di proxy elencati è limitato dal valore *Numero massimo di elementi da visualizzare all'interno della cella della tabella*.

Per configurare un nuovo gruppo di proxy, fare clic sul pulsante *Crea gruppi di proxy* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

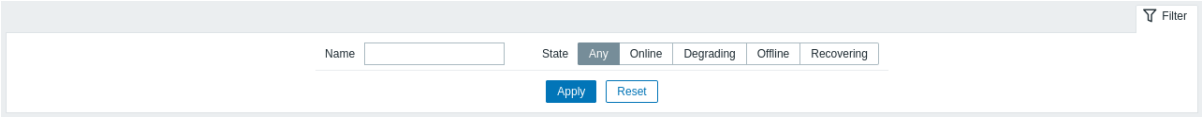
- *Elimina* - elimina i gruppi di proxy.

Per usare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi gruppi di proxy, quindi fai clic sul pulsante desiderato.

Uso del filtro

È possibile usare il filtro per visualizzare solo i gruppi di proxy di interesse. Per migliorare le prestazioni di ricerca, i dati vengono cercati con i macro non risolti.

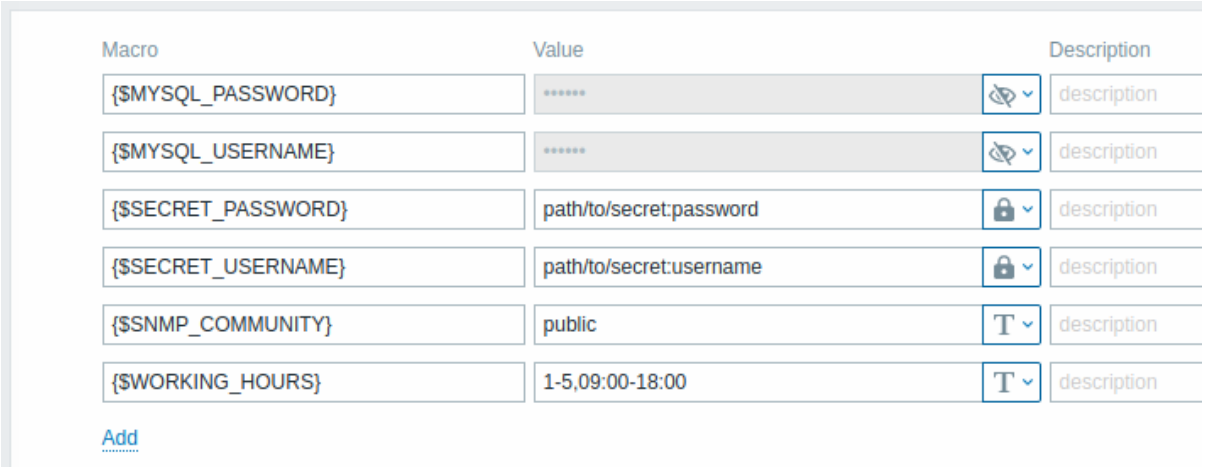
Il collegamento *Filter* è disponibile sopra l'elenco dei gruppi di proxy. Se vi si fa clic, diventa disponibile un filtro che consente di filtrare i gruppi di proxy per nome e stato.



Macro

Panoramica

Questa sezione consente di definire macro utente a livello di sistema come coppie nome-valore. Si noti che i valori delle macro possono essere mantenuti come testo normale, testo segreto o segreto Vault. È inoltre possibile aggiungere una descrizione.



Coda

Panoramica

Nella sezione *Amministrazione* > *Coda* vengono visualizzati gli item in attesa di essere aggiornati.

Idealmente, quando si apre questa sezione, tutto dovrebbe essere "verde", ovvero non dovrebbero esserci item nella coda. Se tutti gli item vengono aggiornati senza ritardi, non ce ne sono in attesa. Tuttavia, a causa di prestazioni insufficienti del server, alcuni item potrebbero subire ritardi e le relative informazioni vengono visualizzate in questa sezione. Per ulteriori informazioni, vedere la sezione Coda.

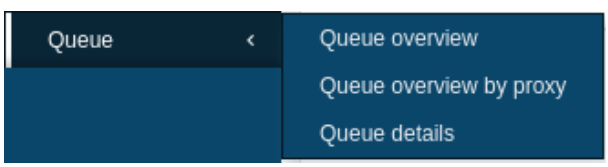
Note:
Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; per ulteriori informazioni, vedere Calcolo delle code durante la manutenzione.

Note:
La coda è disponibile solo se il server Zabbix è in esecuzione. Gli item non vengono conteggiati nella coda se l'interfaccia dell'item diventa non disponibile a causa di problemi di connessione o se l'agent non funziona correttamente.

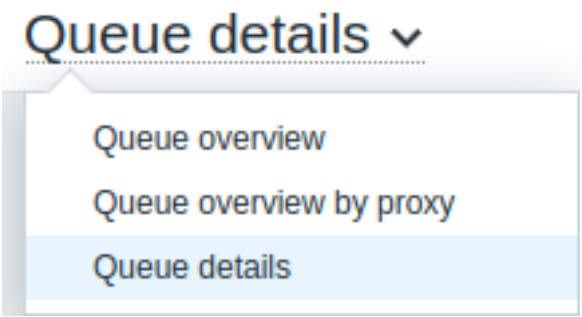
La sezione *Amministrazione* > *Coda* contiene le seguenti pagine:

- Panoramica della coda - visualizza la coda per tipo di item;
- Panoramica della coda per proxy - visualizza la coda per proxy;
- Dettagli della coda - visualizza un elenco degli item in ritardo.

L'elenco delle pagine disponibili viene visualizzato facendo clic su *Coda* nella sezione del menu *Amministrazione*. È inoltre possibile passare da una pagina all'altra utilizzando il menu a discesa del titolo nell'angolo superiore sinistro.



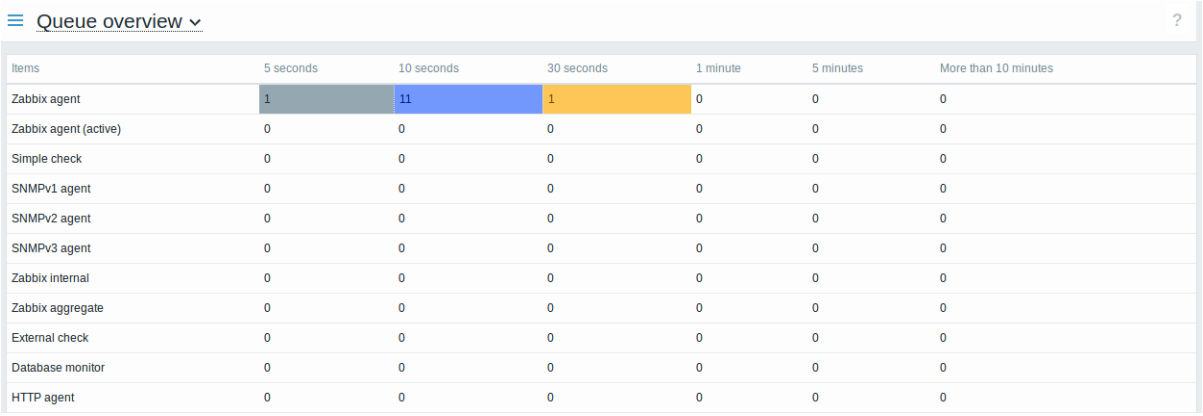
Menu di terzo livello.



Menu a discesa del titolo.

Panoramica per tipo di articolo

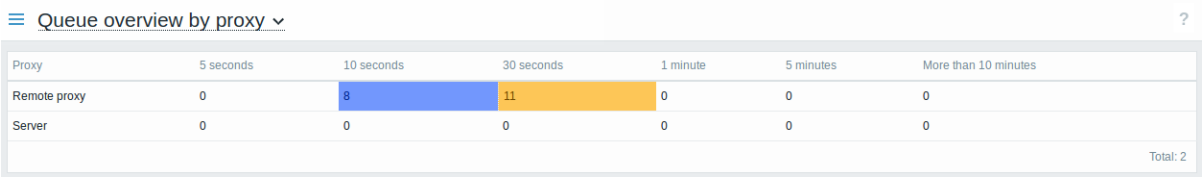
In questa schermata è facile individuare se il problema è correlato a uno o diversi tipi di articoli.



Ogni riga contiene un tipo di elemento. Ogni colonna mostra il numero di attesa elementi - in attesa di 5-10 secondi/10-30 secondi/30-60 secondi/1-5 minuti/5-10 minuti o più di 10 minuti rispettivamente.

Panoramica per procura

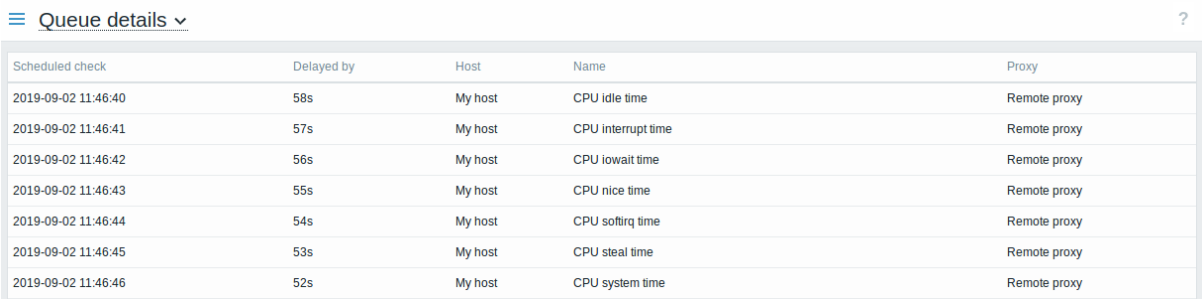
In questa schermata è facile individuare se il problema è correlato a uno dei proxy o il server.



Ogni riga contiene un proxy, con il server per ultimo nell'elenco. ogni colonna mostra il numero di elementi in attesa - in attesa di 5-10 secondi/10-30 secondi/30-60 secondi/1-5 minuti/5-10 minuti o più 10 minuti rispettivamente.

Elenco degli elementi in attesa

In questa schermata viene elencato ogni elemento in attesa.



Dati visualizzati:

Colonna	Descrizione
Assegno programmato	Viene visualizzata l'ora in cui l'assegno era scaduto.
Ritardato di	Viene visualizzata la durata del ritardo.
Host	Viene visualizzato l'host dell'elemento.
Nome	Viene visualizzato il nome dell'elemento in attesa.
Proxy	Viene visualizzato il nome del proxy, se l'host è monitorato dal proxy.

Possibili messaggi di errore

Potresti riscontrare una situazione in cui non vengono visualizzati dati e compare il seguente messaggio di errore:



In questo caso, il messaggio di errore è il seguente:

```
Cannot display item queue. Permission denied
```

Questo accade quando i parametri di configurazione PHP nel file di configurazione del frontend - `$ZBX_SERVER` oppure sia `$ZBX_SERVER` che `$ZBX_SERVER_PORT` - puntano a un server Zabbix esistente che utilizza un database diverso.

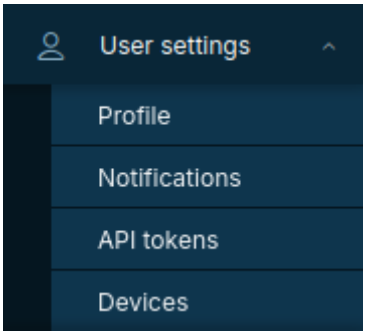
Impostazioni utente

Panoramica

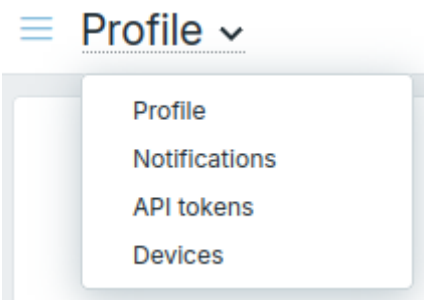
A seconda del proprio ruolo utente, la sezione *Impostazioni utente* può contenere le seguenti pagine:

- *Profilo* o *Profilo utente* - per personalizzare alcune funzionalità del frontend di Zabbix;
- *Notifiche* - per personalizzare le notifiche dell'utente corrente;
- *Token API* - per gestire i token API assegnati all'utente corrente;
- *Dispositivi* - per gestire i dispositivi mobili appartenenti all'utente corrente.

L'elenco delle pagine disponibili viene visualizzato facendo clic sull'icona utente nella parte inferiore del menu di Zabbix (non disponibile per l'utente *guest*). È inoltre possibile passare da una pagina all'altra utilizzando il menu a discesa del titolo nell'angolo superiore sinistro.



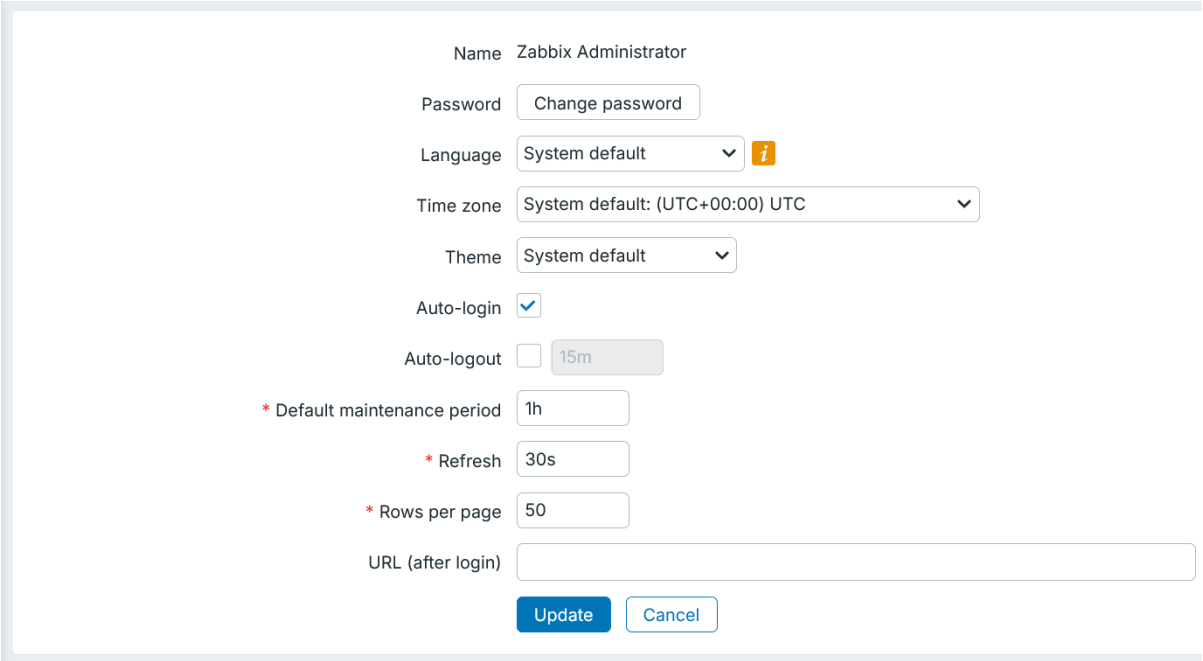
Menu di terzo livello.



Menu a discesa del titolo.

Profilo utente

La sezione *Profilo utente* offre opzioni per impostare la lingua personalizzata dell'interfaccia, il tema dei colori, il numero di righe visualizzate negli elenchi e altro. Le modifiche apportate qui verranno applicate solo all'utente corrente.



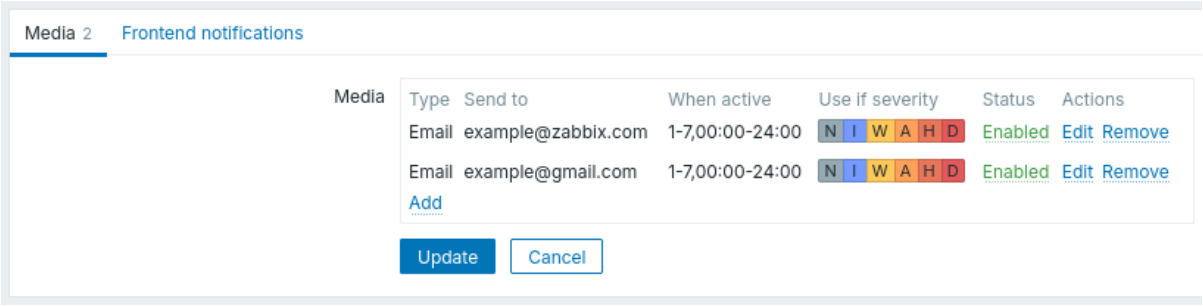
Parametro	Descrizione
Nome	Vengono visualizzati il nome e il cognome dell'utente. Se il nome e il cognome non sono disponibili, viene visualizzato il nome utente.
Password	Fare clic sul pulsante *Cambia password* per aprire tre campi: *Vecchia password*, *Nuova password*, *Nuova password (di nuovo)*. Dopo una modifica della password riuscita, l'utente verrà disconnesso da tutte le sessioni attive. La password può essere modificata solo dagli utenti che utilizzano l'autenticazione interna di Zabbix.

Parametro	Descrizione
Lingua	Selezionare la lingua dell'interfaccia desiderata oppure selezionare **Predefinita di sistema** per utilizzare le impostazioni di sistema predefinite. Selezionando *Inglese (en_US)* verrà inoltre abilitato il formato statunitense di ora e data nel frontend. Per ulteriori informazioni, vedere Installazione di lingue aggiuntive per il frontend.
Fuso orario	Selezionare il fuso orario per sostituire il fuso orario globale a livello utente oppure selezionare **Predefinito di sistema** per utilizzare le impostazioni globali del fuso orario.
Tema	Selezionare un tema di colori specifico per il proprio profilo: **Predefinito di sistema** - utilizza le impostazioni di sistema predefinite; **Blu** - tema blu standard; **Blu (classico)** - tema blu standard con caratteri legacy; **Scuro** - tema scuro alternativo; **Blu scuro** - tema blu scuro; **Scuro (classico)** - tema scuro alternativo con caratteri legacy; **Chiaro ad alto contrasto** - tema chiaro ad alto contrasto; **Scuro ad alto contrasto** - tema scuro ad alto contrasto.
Accesso automatico	Selezionare questa casella per consentire a Zabbix di ricordare l'utente e di effettuare automaticamente l'accesso per 30 giorni. Quando si effettua l'accesso con *Ricordami per 30 giorni* abilitato: - Il timer di disconnessione automatica viene reimpostato (la sessione rimane attiva fino al termine del periodo di 30 giorni). - L'accesso automatico viene abilitato per gli accessi successivi entro 30 giorni. Quando si effettua l'accesso senza *Ricordami per 30 giorni*: - Il parametro di accesso automatico viene cancellato. - La disconnessione automatica segue normalmente il timeout configurato. A questo scopo vengono utilizzati i cookie del browser.
Disconnessione automatica	Selezionando questa casella, l'utente verrà disconnesso automaticamente dopo il numero di secondi impostato (minimo 90 secondi, massimo 1 giorno). Sono supportati i suffissi temporali, ad esempio: 90s, 5m, 2h, 1d. Questa opzione non funzionerà nei seguenti casi: * Quando le pagine del menu *Monitoraggio* eseguono aggiornamenti delle informazioni in background. Se vengono lasciate aperte pagine che aggiornano i dati a intervalli specifici (dashboard, grafici, dati recenti ecc.), la durata della sessione viene estesa, disabilitando di fatto la funzione di disconnessione automatica. * Se si effettua l'accesso con l'opzione *Ricordami per 30 giorni* selezionata; in questo caso il timer di disconnessione automatica viene ignorato e la sessione rimane attiva per l'intero periodo. * Quando l'autenticazione viene eseguita tramite un Identity Provider (IdP) SSO, la disconnessione automatica termina solo la sessione Zabbix. La sessione SSO con l'IdP rimane attiva, quindi l'utente potrebbe riuscire ad accedere nuovamente senza inserire una password. *Disconnessione automatica* può anche accettare "0", il che significa che la funzione di disconnessione automatica viene disabilitata dopo l'aggiornamento delle impostazioni del profilo.
Periodo di manutenzione predefinito	Impostare il periodo di manutenzione predefinito per la manutenzione ad hoc.
Aggiornamento	Impostare la frequenza con cui verranno aggiornate le informazioni nelle pagine del menu *Monitoraggio* (minimo 0 secondi, massimo 1 ora). Sono supportati i suffissi temporali, ad esempio: 30s, 90s, 1m, 1h.
Righe per pagina	Impostare il numero di righe visualizzate per pagina negli elenchi. Un numero inferiore di righe (e quindi di record da visualizzare) comporta tempi di caricamento più rapidi.
URL (dopo l'accesso)	Impostare un URL specifico da visualizzare dopo l'accesso. Al posto delle *Dashboard* predefinite, può essere, ad esempio, l'URL di *Monitoraggio* > *Trigger*. Questo campo è disabilitato se l'URL dopo l'accesso è limitato a una pagina specifica per il proprio ruolo utente da un utente *Super amministratore*. Se il campo è disabilitato o l'utente non specifica alcun valore, sotto il campo viene visualizzato il valore predefinito, ad esempio `zabbix.php?action=dashboard.view`

Notifiche

La sezione *Notifiche* fornisce opzioni per personalizzare le notifiche dell'utente corrente.

La scheda *Media* consente di specificare i dettagli dei supporti di notifica per l'utente, come i tipi di supporto e gli indirizzi da utilizzare e quando utilizzarli per recapitare le notifiche.



Se il tipo di supporto è stato disabilitato:

- viene visualizzata un'icona informativa gialla dopo il nome;
- nella colonna *Stato* viene visualizzato "Disabled".

Note:
Le autorizzazioni dell'utente per modificare i propri dettagli dei supporti di notifica possono essere concesse/revocate in base al relativo ruolo utente (vedere l'opzione *Create and edit own media*). Anche le autorizzazioni del Super admin per modificare i dettagli dei supporti di notifica di altri utenti possono essere concesse/revocate in base al relativo ruolo utente (vedere l'opzione *Create and edit user media*)

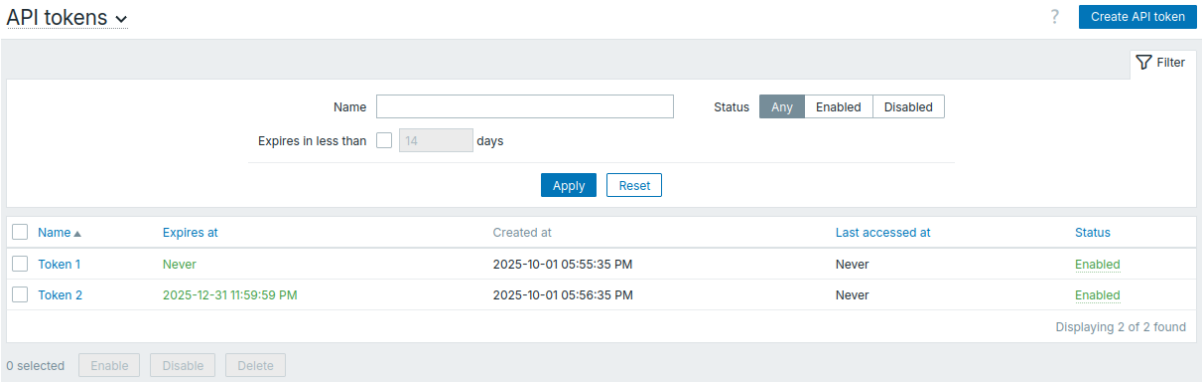
Si noti che per gli utenti sottoposti a provisioning:

- i supporti di notifica dell'utente sottoposto a provisioning non possono essere eliminati;
- i supporti di notifica dell'utente sottoposto a provisioning possono essere disabilitati/abilitati;
- i campi di tali supporti di notifica dell'utente sottoposto a provisioning, come *When active*, *Use if severity* ed *Enabled*, possono essere modificati manualmente;
- per gli utenti sottoposti a provisioning possono essere aggiunti manualmente ulteriori supporti di notifica (ad esempio, indirizzi email aggiuntivi);
- i supporti di notifica aggiunti manualmente possono essere eliminati.

La scheda *Frontend notifications* consente di impostare le notifiche globali.

Token API

La sezione *Token API* consente di visualizzare i token assegnati all'utente, modificare i dettagli del token e creare nuovi token. Questa sezione è disponibile per un utente solo se l'azione *Manage API tokens* è consentita nelle impostazioni del ruolo utente.



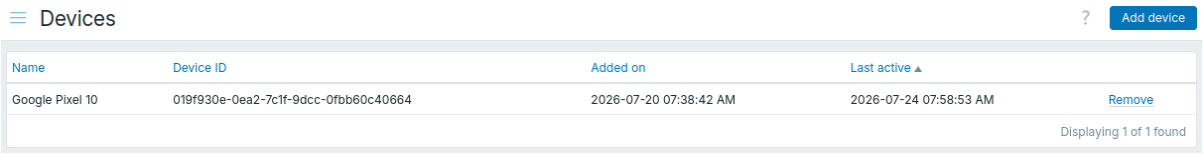
È possibile filtrare i token API per nome, data di scadenza o stato (*Enabled/Disabled*). Fai clic sullo stato del token nell'elenco per abilitarlo/disabilitarlo rapidamente. Puoi anche abilitare/disabilitare più token contemporaneamente selezionandoli nell'elenco e facendo clic sui pulsanti *Enable/Disable* sotto l'elenco.

Attention:
Gli utenti non possono visualizzare il valore del *Auth token* dei token assegnati loro in Zabbix. Il valore del *Auth token* viene visualizzato solo una volta, immediatamente dopo la creazione di un token. Se il token è stato perso, deve essere rigenerato.

Dispositivi

La sezione *Dispositivi* consente di aggiungere o rimuovere i dispositivi mobili da utilizzare con l'app mobile Zabbix.

Questa sezione viene visualizzata se `EnableMobileDevices` è abilitato.



Per aggiungere un dispositivo:

1. Fare clic su *Aggiungi dispositivo*.
2. Scansionare il codice QR nell'app mobile Zabbix.

Note:
Quando si aggiunge un dispositivo, Zabbix aggiunge anche il tipo di supporto notifiche push ai supporti dell'utente. Quando si rimuove il dispositivo, viene rimosso anche tale supporto dell'utente. Se sono configurati più tipi di supporto per le notifiche push, Zabbix aggiunge quello abilitato che viene prima in ordine alfabetico.

I dispositivi aggiunti vengono visualizzati in un elenco. L'elenco mostra il nome e l'ID dei dispositivi, nonché le date e gli orari in cui sono stati aggiunti e in cui sono stati attivi l'ultima volta.

È inoltre possibile rimuovere i dispositivi. Quando si rimuove un dispositivo, le relative credenziali di accesso vengono revocate e rimosse da Zabbix; dopodiché il dispositivo non può più accedere a Zabbix.

I super amministratori possono anche gestire i dispositivi degli altri utenti in *Utenti* > *Dispositivi*.

Notifiche globali

Panoramica

Le notifiche globali consentono di visualizzare i problemi in tempo reale direttamente sulla schermata corrente nel frontend di Zabbix.

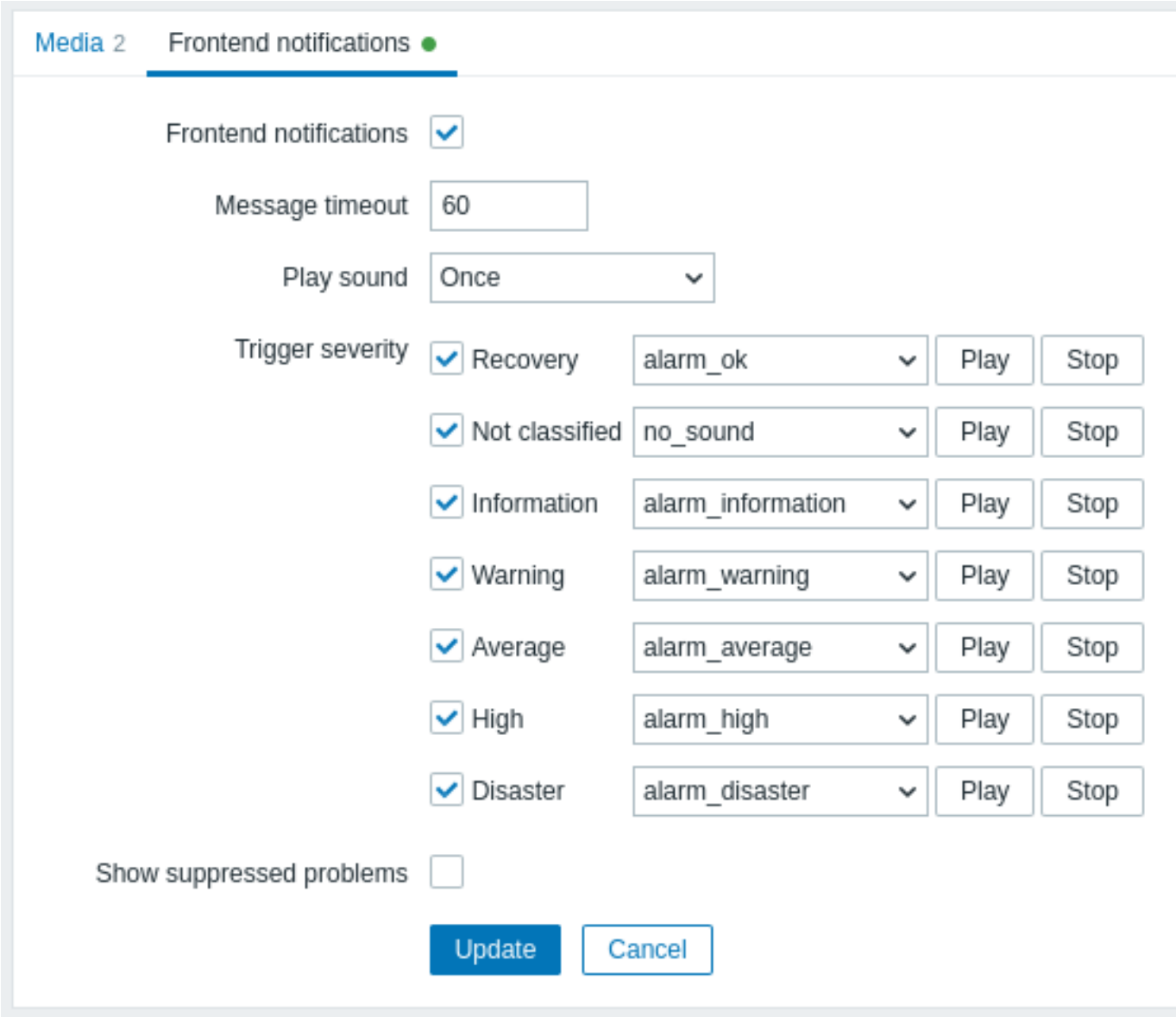
Senza le notifiche globali, quando si lavora al di fuori delle sezioni *Problemi* o *Dashboard*, non si riceverebbe alcuna informazione sui problemi correnti. Le notifiche globali garantiscono che queste informazioni vengano visualizzate, indipendentemente dalla posizione corrente all'interno del frontend di Zabbix.

Le notifiche globali includono sia la visualizzazione di un messaggio sia la riproduzione di un suono.

Attention:
La riproduzione automatica dei suoni potrebbe essere disabilitata (per impostazione predefinita) nelle versioni recenti dei browser. In questi casi, è necessario abilitare manualmente questa impostazione.

Configurazione

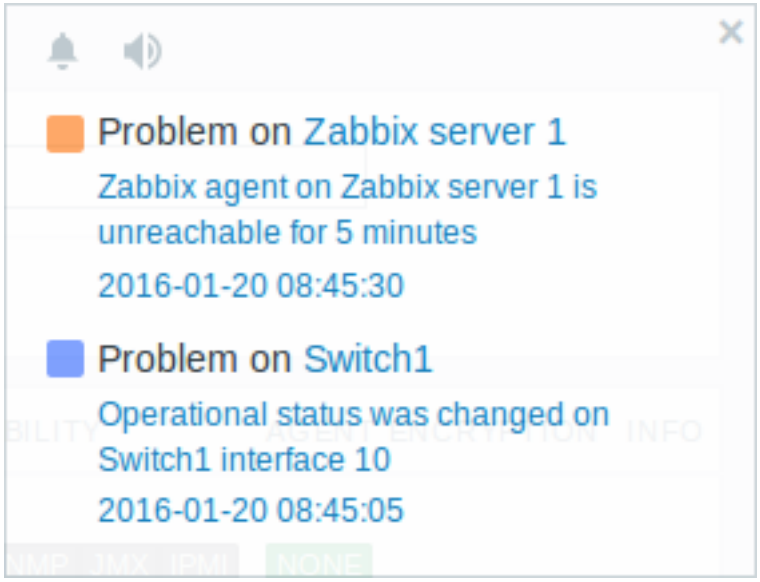
Le notifiche globali possono essere abilitate per ciascun utente nella scheda **Frontend notifications** della sezione Notifications.



Parametro	Descrizione
Frontend notifications	Selezionare la casella per abilitare le notifiche globali.
Message timeout	Impostare la durata per cui il messaggio verrà visualizzato. Per impostazione predefinita, i messaggi rimangono sullo schermo per 60 secondi. Sono supportati i suffissi di tempo, ad esempio: 30s, 5m, 2h, 1d.
Play sound	Impostare la durata per cui il suono verrà riprodotto. **Once** - il suono viene riprodotto una volta per intero; **10 seconds** - il suono viene ripetuto per 10 secondi; **Message timeout** - il suono viene ripetuto finché il messaggio è visibile.
Trigger severity	Impostare le severità dei trigger per cui verranno attivate le notifiche globali e i suoni. È inoltre possibile selezionare suoni appropriati per le varie severità. Se non viene selezionata alcuna severità, non verrà visualizzato alcun messaggio. Inoltre, i messaggi di ripristino verranno visualizzati solo per le severità selezionate. Ad esempio, se *Recovery* e *Disaster* sono selezionati, le notifiche globali verranno visualizzate per i problemi e i ripristini dei trigger con severità *Disaster*.
Show suppressed problems	Selezionare la casella per visualizzare le notifiche relative ai problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione del host.

Messaggi globali visualizzati

Quando arrivano dei messaggi, vengono visualizzati in una sezione mobile sul lato destro. Puoi riposizionare liberamente questa sezione trascinando l'intestazione della sezione.



Per questa sezione sono disponibili diversi controlli:

- Pulsante **Snooze** che silenzia il suono di allarme attualmente attivo;
- Pulsante **Mute/Unmute** che alterna tra la riproduzione e la non riproduzione dei suoni di allarme.

Suono nei browser

Panoramica

Il suono viene utilizzato nelle notifiche globali.

Affinché i suoni vengano riprodotti nel frontend di Zabbix, le *Notifiche frontend* devono essere abilitate nella scheda *Notifiche frontend* del profilo utente, con tutte le severità dei trigger selezionate. Inoltre, i suoni devono essere abilitati nella finestra pop-up delle notifiche globali.

Se, per qualsiasi motivo, l'audio non può essere riprodotto sul dispositivo, il pulsante nella finestra pop-up delle notifiche globali rimarrà permanentemente nello stato "mute", accompagnato dal messaggio "Cannot support notification audio for this device" passando il cursore sul pulsante .

I suoni, inclusi i clip audio predefiniti, sono supportati solo nel formato MP3.

I suoni del frontend di Zabbix sono stati testati con successo nelle versioni recenti dei browser Firefox e Opera su Linux, e nei browser Chrome, Firefox, Microsoft Edge e Opera su Windows.

Attention:
La riproduzione automatica dei suoni potrebbe essere disabilitata (per impostazione predefinita) nelle versioni recenti dei browser. In questi casi, è necessario abilitare manualmente questa impostazione.

Ricerca globale

È possibile cercare nel frontend di Zabbix host, gruppi di host, template e gruppi di template.

La casella di input della ricerca si trova sotto il logo di Zabbix nel menu. La ricerca può essere avviata premendo *Enter* o facendo clic sull'icona di ricerca .



Se esiste un host che contiene la stringa inserita in qualsiasi parte del nome, verrà visualizzato un menu a discesa che elenca tutti questi host (con la parte corrispondente evidenziata in arancione). Il menu a discesa elencherà anche un host se il nome visibile di quell'host corrisponde al nome tecnico inserito come stringa di ricerca; l'host corrispondente verrà elencato, ma senza alcuna evidenziazione.

Attributi ricercabili

Gli host possono essere cercati in base alle seguenti proprietà:

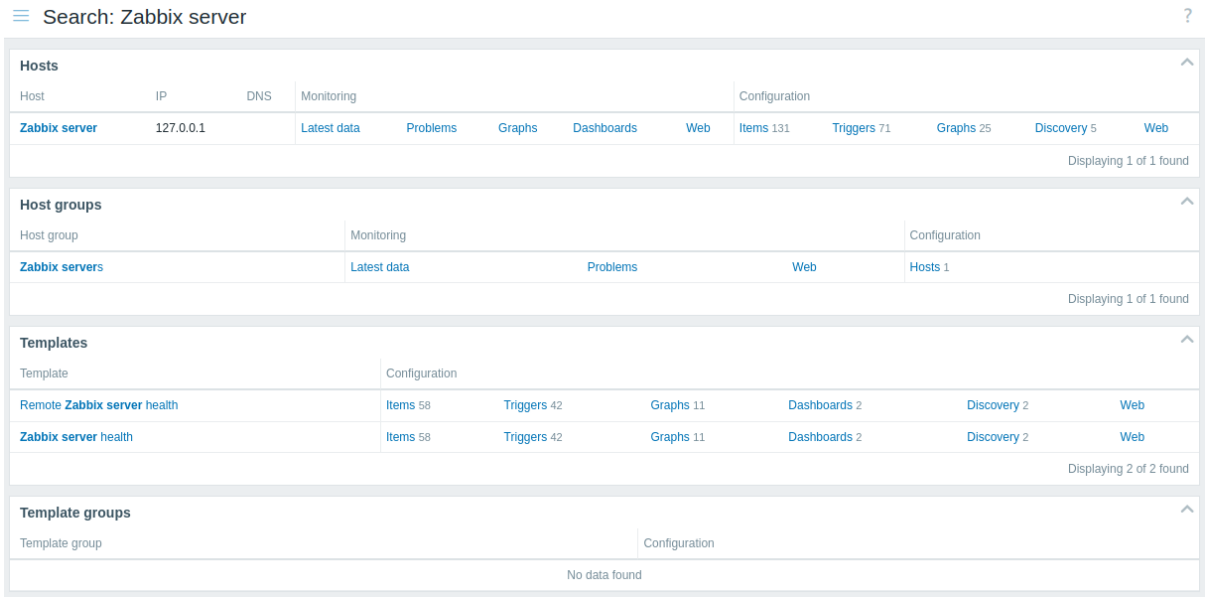
- Nome host
- Nome visibile
- Indirizzo IP
- Nome DNS

I template possono essere cercati per nome o nome visibile. Se si esegue una ricerca per un nome diverso dal nome visibile (di un template/host), nei risultati della ricerca esso viene visualizzato sotto il nome visibile tra parentesi.

I gruppi di host e template possono essere cercati per nome. Specificando un gruppo padre vengono selezionati implicitamente tutti i gruppi nidificati.

Risultati della ricerca

I risultati della ricerca sono costituiti da quattro blocchi separati per host, gruppi di host, template e gruppi di template.



È possibile comprimere/espandere ciascun blocco singolarmente. Il numero di voci è visualizzato nella parte inferiore di ciascun blocco, ad esempio *Visualizzazione di 13 su 13 trovati*. Se non sono presenti voci, il numero di voci non viene visualizzato. Il numero totale di voci visualizzate all'interno di un blocco è limitato a 100.

Ogni voce fornisce collegamenti ai dati di monitoraggio e di configurazione. Vedere l'elenco completo dei collegamenti.

Per tutti i dati di configurazione (come item, trigger, grafici), il numero di entità trovate è visualizzato accanto al nome dell'entità, in grigio. **Nota**: se il numero di entità è zero, non viene visualizzato alcun numero.

Gli host abilitati sono visualizzati in blu, gli host disabilitati in rosso.

Link disponibili

Per ogni voce sono disponibili i seguenti link:

- Host
  - Monitoraggio
    - Dati più recenti
    - Problemi
    - Grafici
    - Dashboard host
    - Scenari web
  - Configurazione
    - Item
    - Trigger
    - Grafici
    - Regole di individuazione
    - Scenari web
- Gruppi di host
  - Monitoraggio
    - Dati più recenti
    - Problemi
    - Scenari web
  - Configurazione
    - Host
- Template
  - Configurazione
    - Item
    - Trigger
    - Grafici
    - Dashboard template
    - Regole di individuazione
    - Scenari web
- Gruppi di template
  - Configurazione
    - Template

Modalità di manutenzione del frontend

Panoramica

È possibile disabilitare temporaneamente il frontend di Zabbix per limitarne l'accesso. Questo è utile per proteggere il database di Zabbix dalle modifiche avviate dagli utenti, preservandone l'integrità.

Mentre il frontend di Zabbix è in modalità di manutenzione, è possibile arrestare in sicurezza il database ed eseguire attività di manutenzione.

Gli utenti provenienti dagli indirizzi IP definiti potranno interagire normalmente con il frontend durante la modalità di manutenzione.

Configurazione

Per abilitare la modalità di manutenzione, aprire il file `maintenance.inc.php` (situato in `/conf` della directory dei documenti HTML di Zabbix sul web server) e rimuovere il commento dalle righe seguenti:

```
// Modalità di manutenzione.
define('ZBX_DENY_GUI_ACCESS', 1);

// Array di indirizzi IP autorizzati a connettersi al frontend (facoltativo).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Messaggio visualizzato nella schermata di avviso (facoltativo).
$ZBX_GUI_ACCESS_MESSAGE = 'Stiamo aggiornando il database MySQL fino alle 15:00. Restate sintonizzati...';
```

Note:
Nella maggior parte dei casi, il file `maintenance.inc.php` si trova in `/conf` della directory dei documenti HTML di Zabbix sul web server.
Tuttavia, alcuni sistemi operativi e web server potrebbero utilizzare un percorso diverso.
Ad esempio, il percorso per:

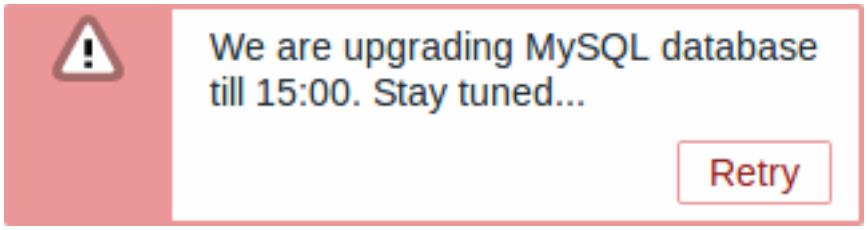
- SUSE e RedHat è `/etc/zabbix/web/maintenance.inc.php`.
- I sistemi basati su Debian è `/usr/share/zabbix/conf/`.

Vedere anche Copia dei file PHP.

Parameter	Details
ZBX_DENY_GUI_ACCESS	Se definito con un valore qualsiasi, la modalità di manutenzione verrà abilitata. Per disabilitare la modalità di manutenzione, commentare o eliminare la definizione.
ZBX_GUI_ACCESS_IP_RANGE	Array di indirizzi IP autorizzati a connettersi al frontend (facoltativo). Ad esempio: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Un messaggio per informare gli utenti della manutenzione (facoltativo). Se non definito, verrà utilizzato il messaggio predefinito *'Zabbix is under maintenance'*.

Visualizzazione

Gli utenti vedranno la seguente schermata quando tenteranno di accedere al frontend di Zabbix mentre è in modalità di manutenzione. La schermata viene aggiornata ogni 30 secondi per tornare allo stato normale senza intervento dell'utente al termine della manutenzione.



Gli indirizzi IP definiti in *ZBX_GUI_ACCESS_IP_RANGE* potranno accedere al frontend come di consueto.

Parametri della pagina

Panoramica

La maggior parte delle pagine dell'interfaccia web di Zabbix supporta vari parametri HTTP GET che controllano ciò che verrà visualizzato. Possono essere passati specificando coppie parametro=valore dopo l'URL, separate dall'URL da un punto interrogativo (?) e tra loro da e commerciali (&).

Monitoring > Problems

Sono supportati i seguenti parametri:

Parameter	Description	Example
show	Opzione del filtro *Show*. Valori possibili: 1 - problemi recenti; 2 - tutti; 3 - in stato di problema.	show=1
name	Opzione del filtro *Problem*: stringa libera.	name=Zabbix agent

Parameter	Description	Example
severities	Opzione del filtro *Severity*: array delle severità selezionate nel formato `severities[*]=*` (sostituire `*` con il livello di severità). Valori possibili: 0 - non classificato; 1 - informazione; 2 - warning; 3 - medio; 4 - alto; 5 - disastro.	severities[3]=3
inventory	Opzione del filtro *Host inventory*: array di campi dell'inventario `[field],[value]`	inventory[0][field]=type&invento
evaltype	Opzione del filtro *Tags*: metodo di valutazione del tag. Valori possibili: 0 - and/or; 2 - or.	evaltype=0
tags	Opzione del filtro *Tags*: array di tag definiti `[tag], [operator], [value]` Valori possibili di operator: 0 - contiene; 1 - uguale a; 2 - non contiene; 3 - diverso da; 4 - esiste; 5 - non esiste.	tags0=target&tags[0][operator]=
show_tags	Opzione del filtro *Show tags*. Valori possibili: 0 - nessuno; 1 - uno; 2 - due; 3 - tre.	show_tags=3
tag_name_format	Opzione del filtro *Tag name*. Valori possibili: 0 - nome completo; 1 - nome abbreviato; 2 - nessuno.	tag_name_format=1
tag_priority	Opzione del filtro *Tag display priority*: stringa separata da virgole della priorità di visualizzazione dei tag	tag_priority=customer, target
show_suppressed	Opzione del filtro *Show suppressed problems*. Valori possibili: 1 - mostra; 2 - non mostrare.	show_suppressed=1
acknowledgement_status	Opzione del filtro *Acknowledgement status*. Valori possibili: 0 - tutti; 1 - non riconosciuti; 2 - riconosciuti.	acknowledgement_status=0
acknowledged_by_me	Opzione del filtro *By me*; supportata solo con `acknowledgement_status=2`. Valori possibili: 0 - disabilitato; 1 - abilitato.	acknowledged_by_me=1
compact_view	Opzione del filtro *Compact view*. Valori possibili: 0 - disabilitato; 1 - abilitato.	compact_view=1

Parameter	Description	Example
highlight_row	Opzione del filtro *Highlight whole row*. Valori possibili: 0 - disabilitato; 1 - abilitato.	highlight_row=1
filter_name	Opzione delle proprietà del filtro *Name*: stringa libera	filter_name=Databases
filter_show_counter	Opzione delle proprietà del filtro *Show number of records*. Valori possibili: 0 - disabilitato; 1 - abilitato.	filter_show_counter=1
filter_custom_time	Opzione delle proprietà del filtro *Set custom time period*. Valori possibili: 0 - disabilitato; 1 - abilitato.	filter_custom_time=1
sort	Colonna per l'ordinamento. Valori possibili: clock - ordina per la colonna *Time*; host - ordina per la colonna *Host*; severity - ordina per la colonna *Severity*; name - ordina per la colonna *Problem*.	sort=clock
sortorder	Ordine di ordinamento dei risultati. Valori possibili: DESC - decrescente; ASC - crescente.	sortorder=DESC
age_state	Opzione del filtro *Age less than*; supportata solo con `show`=3. Valori possibili: 0 - disabilita il parametro age; 1 - abilita il parametro age.	age_state=1
age	Opzione del filtro *Age less than*: intero, numero di giorni; supportata solo con `age_state`=1 e `show`=3.	age=7
groupids	Opzione del filtro *Host groups*: array di ID dei gruppi host.	groupids[]=4
hostids	Opzione del filtro *Hosts*: array di ID degli host	hostids[]=10084
triggerids	Opzione del filtro *Triggers*: array di ID dei trigger	triggerids[]=22382
show_timeline	Opzione del filtro *Show timeline*; non supportata con `compact_view`=1. Valori possibili: 0 - non mostrare; 1 - mostrare.	show_timeline=1
details	Opzione del filtro *Show details*. Valori possibili: 0 - non mostrare; 1 - mostrare.	details=1
from	Inizio dell'intervallo di date, può essere relativo (ad esempio, now-1m); supportato solo con `filter_custom_time`=1	from=now-2h
to	Fine dell'intervallo di date, può essere relativa (ad esempio, now-1m); supportata solo con `filter_custom_time`=1	to=now

Vedi anche: opzioni del filtro della pagina Problems.

Modalità kiosk

La modalità kiosk nelle pagine frontend supportate può essere attivata utilizzando i parametri URL. Ad esempio, nei dashboard:

- `/zabbix.php?action=dashboard.view&kiosk=1` - attiva la modalità kiosk
- `/zabbix.php?action=dashboard.view&kiosk=0` - attiva la modalità normale

Presentazione

È possibile attivare una presentazione nella dashboard:

- `/zabbix.php?action=dashboard.view&slideshow=on` - attiva la presentazione

Definizioni

Panoramica

Sebbene molte cose nel frontend possano essere configurate utilizzando il frontend stesso, alcune personalizzazioni sono attualmente possibili solo modificando un file di definizioni.

Questo file è `defines.inc.php` situato in /include della directory dei documenti HTML di Zabbix.

Parametri

Parametri in questo file che potrebbero essere di interesse per gli utenti:

- ZBX_MIN_PERIOD

Periodo minimo del grafico, in secondi. Per impostazione predefinita, un minuto.

- GRAPH_YAXIS_SIDE_DEFAULT

Posizione predefinita dell'asse Y nei grafici semplici e valore predefinito della casella a discesa quando si aggiungono item ai grafici personalizzati. Valori possibili: 0 - sinistra, 1 - destra.

Predefinito: 0

- ZBX_SESSION_NAME

Stringa usata come nome del cookie di sessione del frontend di Zabbix.

Predefinito: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Timeout TTL in secondi usato per invalidare la cache dei dati della Vault response. Impostare 0 per disabilitare la memorizzazione nella cache della risposta di Vault.

Predefinito: 60

- SUBFILTER_VALUES_PER_GROUP

Numero di valori del subfilter per gruppo (ad esempio, nel subfilter di latest data).

Predefinito: 1000

- ZBX_MAX_WIDGET_LINES

Numero massimo di righe del widget da visualizzare.

Predefinito: 1000

Lingue aggiuntive del frontend

Panoramica Per utilizzare una lingua diversa dall'inglese nell'interfaccia web di Zabbix, la relativa locale deve essere installata sul server web. Inoltre, per far funzionare le traduzioni è necessaria l'estensione PHP gettext.

Installazione delle impostazioni locali Per elencare tutte le lingue installate, eseguire:

```
locale -a
```

Se alcune lingue necessarie non sono elencate, aprire il file `/etc/locale.gen` e decommentare le impostazioni locali richieste. Poiché Zabbix utilizza la codifica UTF-8, è necessario selezionare impostazioni locali con charset UTF-8.

Ora eseguire:

```
locale-gen
```

Riavviare il server web. Si noti che, se PHP-FPM è installato e utilizzato dal server web, deve essere riavviato insieme al server web.

Le impostazioni locali dovrebbero ora essere installate. Potrebbe essere necessario ricaricare la pagina del frontend di Zabbix nel browser usando `Ctrl + F5` affinché le nuove lingue vengano visualizzate.

Installazione di Zabbix Se si installa Zabbix direttamente dal repository git di Zabbix, i file di traduzione devono essere generati manualmente. Per generare i file di traduzione, eseguire:

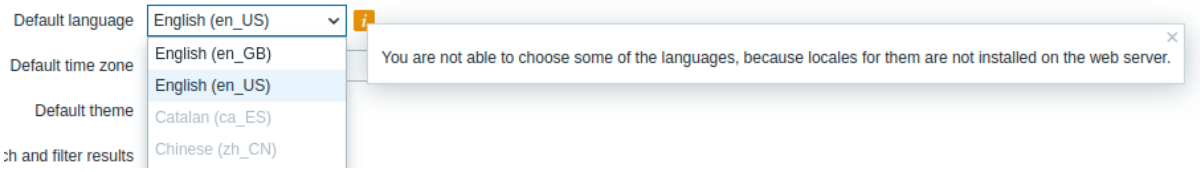
```
make gettext
locale/make_mo.sh
```

Questo passaggio non è necessario quando si installa Zabbix da pacchetti o da file `tar.gz` sorgente.

Selezione di una lingua Esistono diversi modi per selezionare una lingua nell'interfaccia web di Zabbix:

- Durante l'installazione dell'interfaccia web - nel wizard di installazione del frontend. La lingua selezionata verrà impostata come predefinita del sistema.
- Dopo l'installazione, la lingua predefinita del sistema può essere modificata in *Administration* > *General* > *GUI*.
- La lingua per un determinato utente può essere modificata nel profilo utente.

Se una locale per una lingua non è installata sulla macchina, questa lingua verrà visualizzata in grigio nel selettore della lingua di Zabbix. Se almeno una locale manca, accanto al selettore della lingua viene visualizzata un'icona arancione. Facendo clic su questa icona verrà visualizzato il seguente messaggio: "Non è possibile scegliere alcune lingue, perché le relative locale non sono installate sul web server."



Creare il proprio tema

Panoramica

Per impostazione predefinita, Zabbix fornisce una serie di temi frontend predefiniti. Per creare un tema personalizzato, seguire la procedura passo passo fornita qui.

Passaggio 1: Creare il tema

Per creare il proprio tema:

1. Creare un file CSS da zero oppure copiare un tema esistente (ad esempio `blue-theme.css`, `dark-theme.css` o un altro) dalla directory `assets/styles/` e modificarlo.
2. Salvare il tema (ad esempio `custom-theme.css`) nella directory `assets/styles/`.

Passaggio 2: Abilitare il tema nell'elenco dei temi

Per rendere il tema personalizzato disponibile nel frontend, è necessario aggiungerlo all'elenco dei temi restituito dal metodo `APP::getThemes()`. A tale scopo, eseguire l'override del metodo `ZBase::getThemes()` all'interno della classe APP come segue:

1. Aprire il file `include/classes/core/APP.php` e individuare la classe APP:

```
class APP extends ZBase {

}
```

2. All'interno della classe, prima della parentesi graffa di chiusura finale }, aggiungere il metodo seguente:

```
public static function getThemes() {
        return array_merge(parent::getThemes(), [
            'custom-theme' => _('Custom theme')
        ]);
}
```

dove:

- `custom-theme` - identificatore interno del tema, che deve corrispondere al nome del file CSS senza l'estensione `.css`;
- `Custom theme` - nome visualizzato che verrà mostrato nel frontend.

Per aggiungere più temi, elencarli nell'array, separando ogni voce con una virgola. La virgola dopo l'ultima voce è facoltativa.

Ad esempio:

```
      public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme'),
              'anothertheme' => _('Another theme'),
              'onemoretheme' => _('One more theme')
          ]);
      }
```

Passaggio 3: Modificare i colori dei grafici

Per modificare i colori dei grafici, è necessario aggiungere un record per il nuovo tema alla tabella del database `graph_theme`.

Esempio di aggiunta di un record a un database MySQL/MariaDB:

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

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

Passaggio 4: Utilizzare il nuovo tema

Nel frontend di Zabbix, è possibile impostare questo tema come tema predefinito oppure selezionarlo in un profilo utente.

Modalità di debug

Panoramica

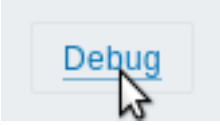
La modalità di debug può essere utilizzata per diagnosticare problemi di prestazioni con le pagine di frontend.

Configurazione

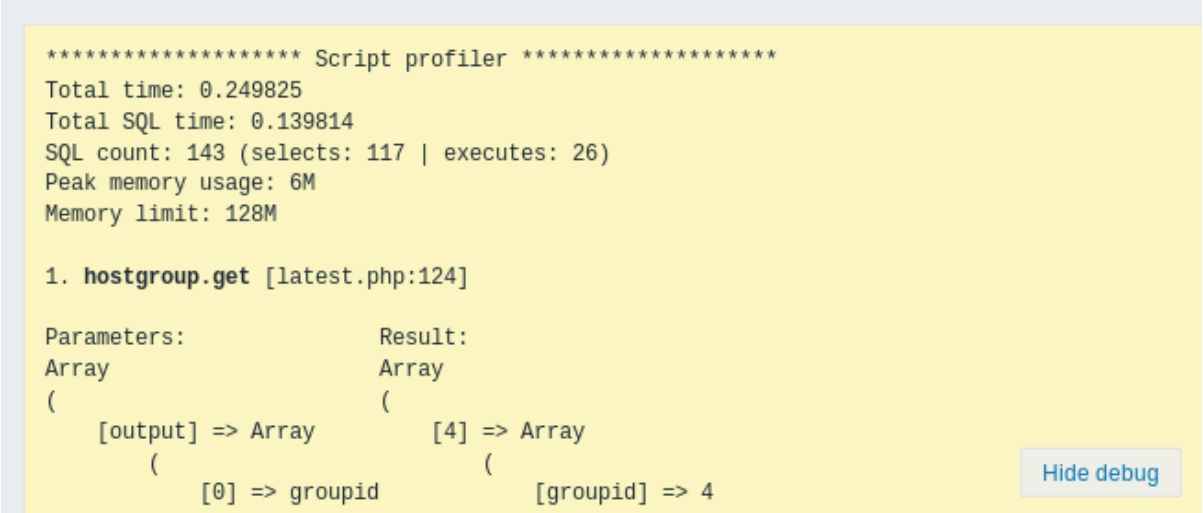
La modalità di debug può essere attivata per singoli utenti che appartengono a un gruppo di utenti:

- quando si configura un gruppo di utenti;
- quando si visualizzano i gruppi di utenti configurati.

Quando *Debug mode* è abilitata per un gruppo di utenti, i relativi utenti vedranno un pulsante *Debug* nell'angolo in basso a destra della finestra del browser:



Facendo clic sul pulsante *Debug* si apre una nuova finestra sotto il contenuto della pagina, che contiene le statistiche SQL della pagina, insieme a un elenco di chiamate API e singole istruzioni SQL:



In caso di problemi di prestazioni della pagina, questa finestra può essere utilizzata per individuare la causa principale del problema.

Warning:
L'attivazione della *Debug mode* influisce negativamente sulle prestazioni del frontend.

Cookie utilizzati da Zabbix

Panoramica

Questa pagina fornisce un elenco dei cookie utilizzati da Zabbix.

Nome	Descrizione	Valori	Scadenza/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSIONID	Dati sessione del frontend di Zabbix, memorizzati come JSON codificato in base64		Sessione (scade al termine della sessione di navigazione)	+	+ (solo se HTTPS è abilitato su un server web)

Nome	Descrizione	Valori	Scadenza/Max-Age	HttpOnly[1]	Secure[2]
tab	Numero della scheda attiva; questo cookie viene utilizzato solo nelle pagine con più schede (ad es. pagina di configurazione di *Host*, *Trigger* o *Action*) e viene creato quando un utente passa da una scheda primaria a un'altra scheda (come la scheda *Tags* o *Dependencies*). 0 viene utilizzato per la scheda primaria.	Esempio: 1	Sessione (scade al termine della sessione di navigazione)	-	-
browserwarning_ignore	Indica se un avviso relativo all'utilizzo di un browser obsoleto debba essere ignorato.	yes	Sessione (scade al termine della sessione di navigazione)	-	-
system-message-ok	Un messaggio da mostrare non appena la pagina viene ricaricata.	Messaggio in testo semplice	Sessione (scade al termine della sessione di navigazione) o non appena la pagina viene ricaricata	+	-
system-message-error	Un messaggio di errore da mostrare non appena la pagina viene ricaricata.	Messaggio in testo semplice	Sessione (scade al termine della sessione di navigazione) o non appena la pagina viene ricaricata	+	-

Note:
L'imposizione del flag 'HttpOnly' sui cookie di Zabbix tramite una direttiva del server web non è supportata.

Note

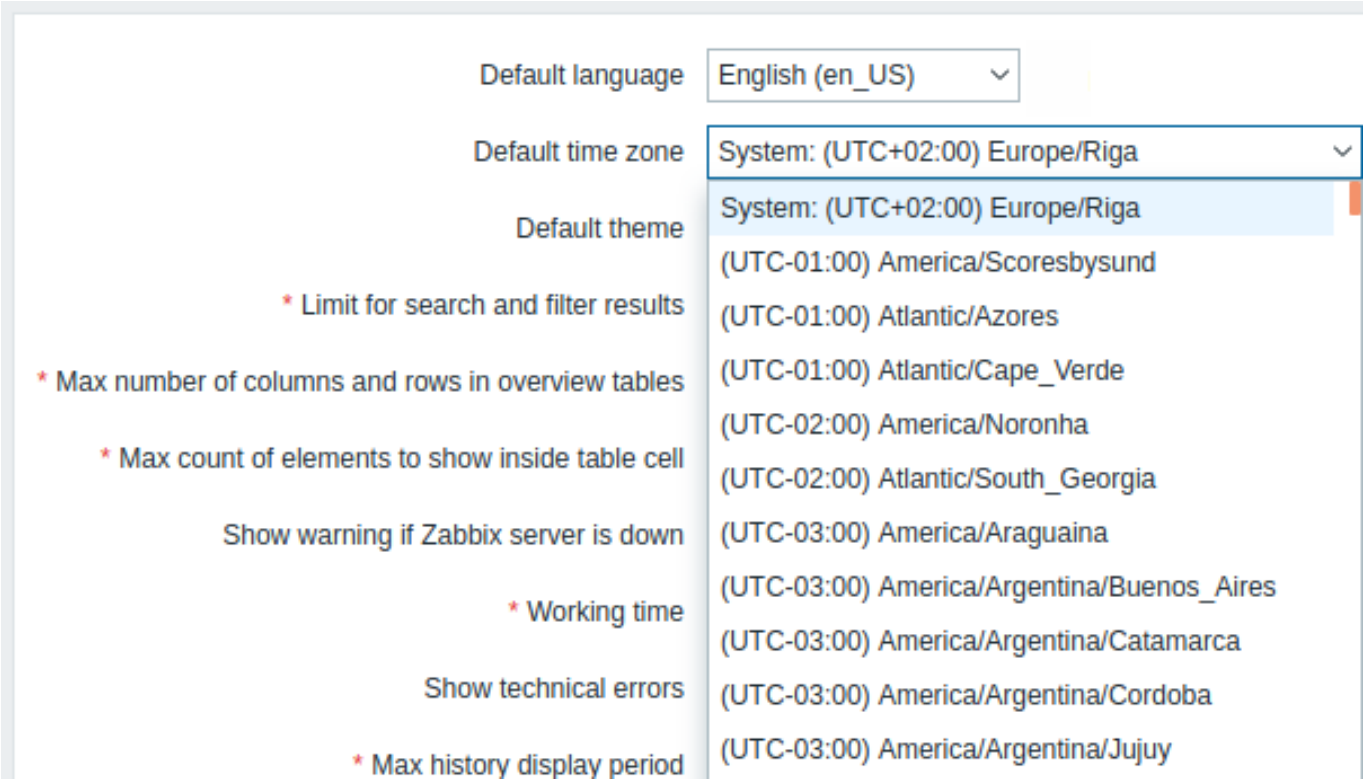
[1] Quando `HttpOnly` è impostato su 'true', il cookie sarà accessibile solo tramite il protocollo HTTP. Ciò significa che il cookie non sarà accessibile dai linguaggi di scripting, come JavaScript. Questa impostazione può contribuire efficacemente a ridurre il furto di identità tramite attacchi XSS (sebbene non sia supportata da tutti i browser).

[2] `Secure` indica che il cookie deve essere trasmesso solo tramite una connessione HTTPS sicura dal client. Quando è impostato su 'true', il cookie verrà impostato solo se esiste una connessione sicura.

Fusi orari

Panoramica

Il fuso orario del frontend può essere impostato globalmente nel frontend e modificato per i singoli utenti.



Se è selezionato *Sistema*, per il frontend verrà utilizzato il fuso orario del web server (incluso il valore di `date.timezone` in `php.ini`, se impostato), mentre il server Zabbix utilizzerà il fuso orario della macchina su cui è in esecuzione.

Note:
Il server Zabbix utilizzerà il fuso orario globale/utente specificato solo durante l'espansione delle macro nelle notifiche (ad esempio, `{EVENT.TIME}` può espandersi in un fuso orario diverso per ciascun utente) e per il limite temporale relativo all'invio delle notifiche (vedere l'impostazione "Quando attivo" nella configurazione dei media utente).

La scelta del fuso orario non influisce sul formato di data/ora del frontend. È invece possibile modificare la lingua dell'interfaccia (durante l'installazione o nelle impostazioni utente): selezionando *Inglese (en_US)* verrà attivato anche il formato statunitense di data/ora nel frontend.

Configurazione

Il fuso orario globale:

- Può essere impostato manualmente durante l'installazione del frontend.
- Può essere modificato in *Administration* > *General* > *GUI*.

Il fuso orario a livello utente:

- Può essere impostato durante la configurazione/aggiornamento di un utente.
- Può essere impostato da ciascun utente nel proprio profilo utente.

Vedi anche: Allineamento dei fusi orari quando si utilizzano intervalli di pianificazione.

Reimpostazione della password

Panoramica Questa sezione descrive i passaggi per reimpostare le password degli utenti in Zabbix.

Passaggi Rivolgersi all'amministratore di Zabbix se si è dimenticata la password di Zabbix e non si riesce ad accedere.

Un utente Super amministratore può modificare le password di tutti gli utenti nel modulo di configurazione dell'utente.

Se un Super amministratore ha dimenticato la propria password e non riesce ad accedere, è necessario eseguire la seguente query SQL per applicare la password predefinita all'utente Super amministratore (sostituire 'Admin' con il nome utente appropriato del Super amministratore):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Dopo aver eseguito questa query, la password dell'utente sarà impostata su *zabbix*. Assicurarsi di modificare la password predefinita al primo accesso.

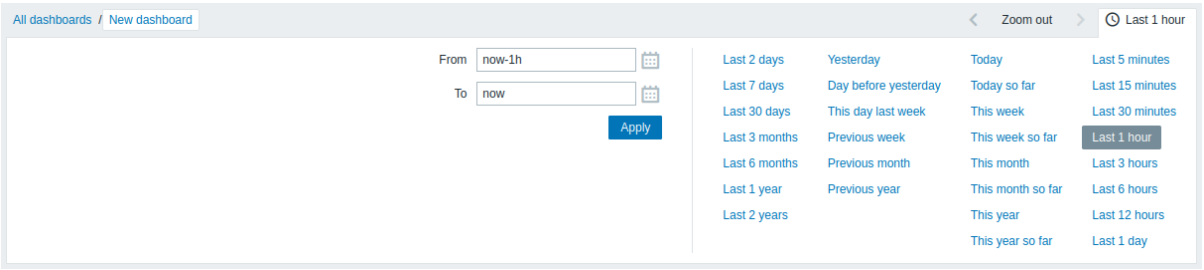
Periodo di tempo e selettori host

Panoramica

Questa pagina descrive il periodo di tempo e i selettori di host disponibili nei dashboard. Questi selettori consentono di modificare dinamicamente i dati visualizzati in tutti i widget del dashboard compatibili, senza modificare le impostazioni dei singoli widget.

Selettore del periodo di tempo

Il selettore del periodo di tempo consente di selezionare i periodi utilizzati più frequentemente con un solo clic del mouse. È possibile espanderlo o comprimerlo facendo clic sul pulsante del selettore del periodo di tempo nell'angolo superiore destro della dashboard.



Le opzioni come *Oggi*, *Questa settimana*, ecc. visualizzano l'intero periodo, comprese le ore/i giorni futuri. Le opzioni come *Oggi finora*, *Questa settimana finora*, ecc. visualizzano solo le ore trascorse.

Dopo aver selezionato un periodo, è possibile spostarlo avanti e indietro nel tempo facendo clic sui pulsanti freccia . Il pulsante *Riduci zoom* consente di ridurre il periodo del 50% in entrambe le direzioni.

Note:
Per i grafici (ad eccezione di quelli con il periodo di tempo *Personalizzato*), è anche possibile selezionare il periodo di tempo visualizzato evidenziando un'area del grafico con il pulsante sinistro del mouse. Dopo aver rilasciato il pulsante sinistro del mouse, il grafico verrà ingrandito sull'area evidenziata. È anche possibile ridurre lo zoom facendo doppio clic nel grafico.

I campi *Da/A* visualizzano il periodo selezionato utilizzando la sintassi del tempo assoluto (nel formato `Y-m-d H:i:s`) oppure la sintassi del tempo relativo. Un periodo di tempo relativo può contenere una o più operazioni matematiche (- o +), ad esempio `now-1d` o `now-1d-2h+5m`.

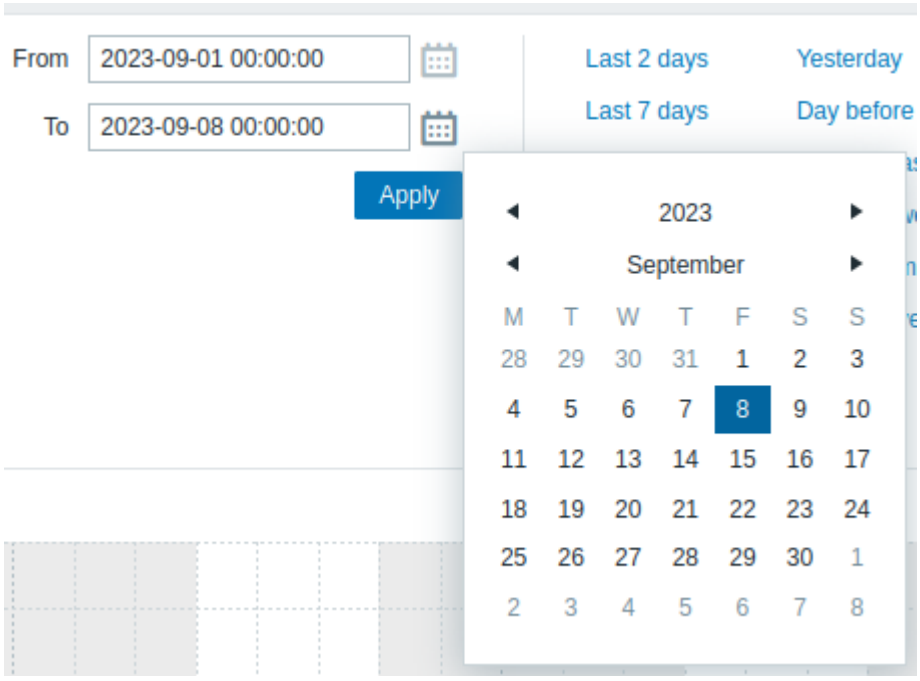
Sono supportate le seguenti abbreviazioni per il tempo relativo:

- `now`
- `s` (secondi)
- `m` (minuti)
- `h` (ore)
- `d` (giorni)
- `w` (settimane)
- `M` (mesi)
- `y` (anni)

Nel selettore del periodo di tempo è supportata la precisione (ad esempio, `/M` in `now-1d/M`). Dettagli della precisione:

Precisione	Da	A
`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`	Lunedì della settimana 00:00:00	Domenica della settimana 23:59:59
`M`	Primo giorno del mese 00:00:00	Ultimo giorno del mese 23:59:59
`y`	1° gennaio dell'anno 00:00:00	31 dicembre dell'anno 23:59:59

È anche possibile selezionare un periodo di tempo utilizzando il selettore della data. Per aprirlo, fare clic sull'icona del calendario accanto ai campi *Da/A*.



Note:
Nel selettore della data è possibile spostarsi tra anno/mese/data utilizzando `Tab`, `Shift+Tab` e i tasti freccia della tastiera. Premere `Enter` per confermare la selezione.

Esempi:

Da	A	Periodo selezionato
`now/d`	`now/d`	00:00 - 23:59 oggi
`now/d`	`now/d+1d`	00:00 di oggi - 23:59 di domani
`now/w`	`now/w`	Lunedì 00:00:00 - Domenica 23:59:59 di questa settimana
`now-1y/w`	`now-1y/w`	La settimana da lunedì 00:00:00 a domenica 23:59:59 di un anno fa

Attention:
L'uso di "now/M+1M" per il parametro *A* può aggiungere 31 giorni, con il risultato che la data può slittare di 1-3 giorni a seconda del numero di giorni del mese. Ad esempio, se usato a gennaio, il risultato potrebbe essere 02 marzo invece del previsto 28 febbraio. Per evitare questo problema, utilizzare "now/M-3d+1M/M", che tiene conto correttamente della lunghezza del mese. Analogamente, se si configura *Da* per andare indietro, utilizzare "now/M+3d-1M/M".

Selettore host

Il selettore host consente di selezionare un host per visualizzarne i dati nei widget della dashboard compatibili.

Diventa disponibile dopo aver modificato e salvato una dashboard, se almeno un widget è configurato con il parametro *Override host* impostato per utilizzare la dashboard come origine dati.



Parametri di configurazione: frontend Zabbix

Panoramica I parametri di configurazione consentono di personalizzare il comportamento del frontend di Zabbix.

Specificare i valori dei parametri nel file di configurazione del frontend (`zabbix.conf.php`).

Questa pagina è strutturata come segue:

- Panoramica dei parametri (per visualizzare tutti i dettagli di un parametro, fare clic sul suo nome)
- Dettagli dei parametri

Parametro	Descrizione
$DB['TYPE']	Tipo di database.
$DB['SERVER']	Nome host o indirizzo IP del database.
$DB['PORT']	Porta del database.
$DB['DATABASE']	Nome del database utilizzato dal frontend di Zabbix per connettersi al database configurato.
$DB['USER']	Nome utente utilizzato dal frontend di Zabbix per connettersi al database.
$DB['PASSWORD']	Password utilizzata dal frontend di Zabbix per connettersi al database.
$DB['SCHEMA']	Nome dello schema del database utilizzato per PostgreSQL.
$DB['ENCRYPTION']	Abilita o disabilita la crittografia TLS per le connessioni tra il frontend di Zabbix e il database.
$DB['KEY_FILE']	Percorso del file della chiave privata client utilizzato per la connessione TLS al database.
$DB['CERT_FILE']	Percorso del file del certificato client utilizzato per la connessione TLS al database.
$DB['CA_FILE']	Percorso del file del certificato CA utilizzato per la connessione TLS al database.
$DB['VERIFY_HOST']	Abilita o disabilita la verifica del nome host del server del database nel certificato TLS.
$DB['CIPHER_LIST']	Elenco dei cifrari TLS consentiti per la connessione al database.
$DB['VAULT']	Specifica il gestore dei segreti Vault utilizzato per archiviare le credenziali del database.
$DB['VAULT_URL']	URL del gestore dei segreti Vault utilizzato per recuperare le credenziali del database.
$DB['VAULT_PREFIX']	Prefisso per il percorso o la query Vault utilizzato per recuperare le credenziali del database.
$DB['VAULT_DB_PATH']	Percorso delle credenziali del database nel gestore dei segreti Vault.
$DB['VAULT_TOKEN']	Token di autenticazione utilizzato per accedere al gestore dei segreti Vault.
$DB['VAULT_APP_ROLE_ID']	ID del ruolo AppRole utilizzato per autenticarsi a HashiCorp Vault.
$DB['VAULT_APP_SECRET_ID']	ID segreto AppRole utilizzato per autenticarsi a HashiCorp Vault.
$DB['VAULT_CERT_FILE']	Percorso del file del certificato client utilizzato per autenticarsi al gestore dei segreti Vault.
$DB['VAULT_KEY_FILE']	Percorso del file della chiave privata client utilizzato per autenticarsi al gestore dei segreti Vault.
$DB['VAULT_CACHE']	Abilita o disabilita la memorizzazione nella cache delle credenziali recuperate dal gestore dei segreti Vault.
$ZBX_SERVER	Specifica un valore per sostituire il nome host o l'indirizzo IP di Zabbix.
$ZBX_SERVER_PORT	Specifica un valore per sostituire la porta del server Zabbix.
$ZBX_SERVER_NAME	Nome del server Zabbix.
$IMAGE_FORMAT_DEFAULT	Formato immagine predefinito.
$HISTORY_PROVIDERS[]	Configurazione dei provider di archiviazione della cronologia per Elasticsearch o ClickHouse.
$SSO['SETTINGS']	Impostazioni di configurazione per l'accesso SAML.
$SSO['CERT_STORAGE']	Specifica dove vengono archiviati i certificati SAML.
$SSO['SP_KEY']	Percorso del file della chiave privata del provider di servizi (SP) SAML quando la chiave privata è archiviata nel filesystem.
$SSO['SP_CERT']	Percorso del file del certificato del provider di servizi (SP) SAML quando il certificato è archiviato nel filesystem.
$SSO['IDP_CERT']	Percorso del file del certificato del provider di identità SAML quando il certificato è archiviato nel filesystem.
$ZBX_FEATURE_FLAGS['banners_enabled']	Abilita o disabilita il supporto per i banner.
$ZBX_FEATURE_FLAGS['http_auth_enabled']	Abilita o disabilita l'autenticazione HTTP degli utenti.
$ZBX_FEATURE_FLAGS['modules_configurable']	Abilita o disabilita l'accesso ai moduli.
$ZBX_FEATURE_FLAGS['media_type_templates']	Abilita o disabilita la modifica di specifici tipi di media preconfigurati.
$ZBX_SERVER_TLS['ACTIVE']	Specifica se TLS è abilitato per le connessioni al server Zabbix.
$ZBX_SERVER_TLS['CA_FILE']	Percorso del file del certificato dell'autorità di certificazione (CA) utilizzato per verificare il certificato del server.
$ZBX_SERVER_TLS['KEY_FILE']	Percorso della chiave privata client corrispondente al certificato client.
$ZBX_SERVER_TLS['CERT_FILE']	Percorso del file del certificato client quando è richiesta l'autenticazione TLS reciproca.
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	Nome distinto (DN) dell'emittente da confrontare con il certificato del server.
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	Nome distinto (DN) del soggetto da confrontare con il certificato del server.
$TELEMETRY_PROVIDERS[]	Configurazione di una fonte di dati di telemetria.

Dettagli dei parametri

$DB['TYPE']

Tipo di database.

Valori: `MYSQL` - MySQL, `POSTGRESQL` - PostgreSQL.

$DB['SERVER']

Nome host o indirizzo IP del database.

Se `$DB['SERVER']` non è definito, viene utilizzato il socket Unix predefinito. Per MySQL, impostando `$DB['SERVER']` su `localhost` si ottiene una connessione tramite il socket Unix predefinito.
 Per PostgreSQL, `$DB['SERVER']` può contenere

più voci `host:port` separate da virgole. Se per un host nell'elenco separato da virgole non è definita una porta, viene utilizzato come porta predefinita il valore definito in `$DB['PORT']`. Se in `$DB['SERVER']` viene specificato un percorso del socket (ad esempio, `/var/run/pgbouncer`), viene utilizzato tale socket Unix.

Esempi:

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

Porta del database.

Se non definito, viene utilizzato il valore 0.

$DB['DATABASE']

Nome del database utilizzato dal frontend di Zabbix per connettersi al database configurato.

Esempio:

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Nome utente utilizzato dal frontend Zabbix per connettersi al database.

$DB['PASSWORD']

Password utilizzata dal frontend Zabbix per connettersi al database.

$DB['SCHEMA']

Nome dello schema del database utilizzato per PostgreSQL.

$DB['ENCRYPTION']

Abilita o disabilita la crittografia per le connessioni tra il frontend Zabbix e il database.

Valori: `true` - abilitata, `false` - disabilitata.

$DB['KEY_FILE']

Percorso al file della chiave privata del client utilizzato per la connessione TLS al database.

Esempio:

```
// Per MySQL:
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

Percorso al file del certificato client utilizzato per la connessione TLS al database.

Esempio:

```
// Per MySQL:
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

Percorso al file del certificato CA utilizzato per la connessione TLS al database.

Esempi:

```
// Per MySQL:
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
// Per PostgreSQL:
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

Abilita o disabilita la verifica del nome host del server del database nel certificato TLS.

Valori: `true` - abilitata, `false` - disabilitata.

$DB['CIPHER_LIST']

Elenco dei cifrari TLS consentiti per la connessione al database.

Esempi:

```
// Per MySQL:
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

Specificare il gestore dei segreti Vault utilizzato per archiviare le credenziali del database.

Valori: `CyberArk` o `HashiCorp`.

Esempi:

```
// Per CyberArk:
$DB['VAULT'] = 'CyberArk';
// Per HashiCorp:
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

URL del gestore dei segreti Vault utilizzato per recuperare le credenziali del database.

Esempi:

```
// Per CyberArk:
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
// Per HashiCorp:
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

Prefisso per il percorso o la query di Vault utilizzato per recuperare le credenziali del database.

Esempio:

```
// Per HashiCorp:
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Percorso delle credenziali del database nel gestore dei segreti Vault.

Esempi:

```
// Per CyberArk:
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
// Per HashiCorp:
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Token di autenticazione utilizzato per accedere al gestore dei segreti Vault.

Esempio:

```
// Per HashiCorp:
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

ID del ruolo VaultAppSecretID utilizzato per l'autenticazione a HashiCorp Vault.

Esempio:

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

VaultAppSecretID utilizzato per l'autenticazione a HashiCorp Vault.

Esempio:

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Percorso al file del certificato client utilizzato per l'autenticazione al gestore dei segreti Vault.

Esempio:

```
// Per CyberArk:
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Percorso del file della chiave privata del client utilizzato per l'autenticazione al gestore dei segreti Vault.

Esempio:

```
// Per CyberArk:
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Abilita o disabilita la memorizzazione nella cache delle credenziali recuperate dal gestore dei segreti Vault.

Valori: `true` - abilitata, `false` - disabilitata.

$ZBX_SERVER

Specificare un valore per sovrascrivere il nome host/l'indirizzo IP di Zabbix.
 Se `$ZBX_SERVER` viene specificato senza `$ZBX_SERVER_PORT`, viene utilizzata la porta predefinita (10051).

$ZBX_SERVER_PORT

Specificare un valore per sovrascrivere la porta del server Zabbix.
 Questo parametro viene utilizzato insieme a `$ZBX_SERVER`.

$ZBX_SERVER_NAME

Nome del server Zabbix.

$IMAGE_FORMAT_DEFAULT

Formato immagine predefinito.

Predefinito: `IMAGE_FORMAT_PNG`
 Valori: - `IMAGE_FORMAT_PNG` - formato PNG - `IMAGE_FORMAT_JPEG` - formato JPEG - `IMAGE_FORMAT_GIF` - formato GIF

$HISTORY_PROVIDERS[]

Configurazione dei provider di archiviazione della cronologia per Elasticsearch o ClickHouse.

Parametri di configurazione supportati: - `types` - array dei tipi di dati da archiviare nello storage esterno; - `provider` - tipo di provider della cronologia: `elasticsearch` o `clickhouse`; - `url` - URL del provider della cronologia; - `db` - nome del database (utilizzato per ClickHouse); - `username` - utente del database (utilizzato per ClickHouse); - `password` - password del database (utilizzata per ClickHouse).

Esempi:

```
// Per ClickHouse:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
// Per Elasticsearch:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

Impostazioni di configurazione per l'accesso SAML. L'array `$SSO['SETTINGS']` deve seguire la struttura prevista dalla libreria *SAML PHP Toolkit* (fornita con Zabbix).

In `$SSO['SETTINGS']` è possibile specificare solo le seguenti opzioni:

- `strict`

- baseurl
- compress
- contactPerson
- organization
- sp (solo le opzioni specificate in questo elenco)
  - attributeConsumingService
  - x509certNew
- idp (solo le opzioni specificate in questo elenco)
  - singleLogoutService (una sola opzione)
    * responseUrl
  - certFingerprint
  - certFingerprintAlgorithm
  - x509certMulti
- security (solo le opzioni specificate in questo elenco)
  - signMetadata
  - wantNameId
  - requestedAuthnContext
  - requestedAuthnContextComparison
  - wantXMLValidation
  - relaxDestinationValidation
  - destinationStrictlyMatches
  - rejectUnsolicitedResponsesWithInResponseTo
  - signatureAlgorithm
  - digestAlgorithm
  - lowercaseUrlencoding

Tutte le altre opzioni vengono recuperate dal database e non possono essere sovrascritte. L'opzione debug verrà ignorata.

Se la UI di Zabbix si trova dietro un proxy o un bilanciatore del carico, è possibile utilizzare l'opzione personalizzata use_proxy_headers:

- false (predefinito) - ignora le intestazioni HTTP X-Forwarded-*;
- true - utilizza le intestazioni HTTP X-Forwarded-* per creare l'URL di base.

Per una descrizione completa delle opzioni di configurazione disponibili, consultare la documentazione ufficiale della libreria.

Esempio:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384',
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

Specificare dove vengono archiviati i certificati SAML.

Predefinito: file
 Valori: - database - la chiave privata e i certificati vengono archiviati nel database; - file - la chiave privata e i certificati vengono archiviati nel filesystem.

Esempio:

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

Percorso al file della chiave privata del service provider (SP) SAML quando la chiave privata è archiviata nel filesystem.

Esempio:

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

Percorso al file del certificato del service provider (SP) SAML quando il certificato è archiviato nel filesystem.

Esempio:

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

Percorso al file del certificato del provider di identità SAML quando il certificato è archiviato nel file system.

Esempio:

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

Abilita o disabilita il supporto per i banner.

Predefinito: `true`
 Valori: `true` - abilitato, `false` - disabilitato.

$ZBX_FEATURE_FLAGS['http_auth_enabled']

Abilita o disabilita l'autenticazione HTTP degli utenti.

Predefinito: `true`
 Valori: `true` - abilitata, `false` - disabilitata.

$ZBX_FEATURE_FLAGS['modules_config_enabled']

Abilita o disabilita l'accesso ai moduli.

Predefinito: `true`
 Valori: `true` - abilitato, `false` - disabilitato.

$ZBX_FEATURE_FLAGS['media_type_denylist']

Disabilita la modifica di specifici tipi di supporto preconfigurati.

Valori: `email`, `script`, `sms`, `webhook`, `push`.

Esempio:

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Specifica se una connessione TLS tra il frontend Zabbix e il server Zabbix è abilitata.

Predefinito: `false`
 Valori: `true` - abilitata, `false` - disabilitata.

$ZBX_SERVER_TLS['CA_FILE']

Percorso al file del certificato dell'Autorità di certificazione (CA) utilizzato per verificare il certificato del server.

Esempio:

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

Percorso del file della chiave privata del client corrispondente al certificato client utilizzato per l'autenticazione TLS reciproca.

Esempio:

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

Percorso al file del certificato client quando è richiesta l'autenticazione TLS reciproca.

Esempio:

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

Nome distinto (DN) dell'emittente da confrontare con il certificato del server.

Esempio:

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

Nome distinto (DN) del soggetto da confrontare con il certificato del server.

Esempio:

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

Configurazione di una fonte di dati di telemetria.

Parametri di configurazione supportati per ClickHouse: - `url` - specifica l'URL HTTP o HTTPS utilizzato per connettersi alla fonte di dati di telemetria. - `db` - specifica il nome del database. - `username` - specifica il nome utente utilizzato per accedere al database. - `password` - specifica la password utilizzata per accedere al database. - `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_ca_file` - imposta queste opzioni per configurare una connessione sicura (TLS/SSL) al database. - `vault_path` - specifica il percorso di un segreto Vault contenente i valori `username` e `password`. Non utilizzare insieme alle opzioni `username` e `password`.

Esempio:

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

Panoramica La Zabbix API consente di recuperare e modificare programmaticamente la configurazione di Zabbix e fornisce accesso ai dati storici. È ampiamente utilizzata per:

- Creare nuove applicazioni che lavorano con Zabbix.
- Integrare Zabbix in software di terze parti.
- Automatizzare attività di routine.

La Zabbix API è una API basata su HTTP ed è distribuita come parte del frontend. Utilizza il protocollo JSON-RPC 2.0, il che significa due cose:

- La API è composta da un insieme di metodi separati.
- Le richieste e le risposte tra i client e la API sono codificate usando il formato JSON.

Per ulteriori informazioni sul protocollo e su JSON, vedere la specifica JSON-RPC 2.0 e la homepage del formato JSON.

Per ulteriori informazioni sull'integrazione delle funzionalità di Zabbix nelle vostre applicazioni Python, vedere Python library for Zabbix.

> **Note:**
> L'accesso degli utenti in Zabbix, sia per la configurazione sia per i dati storici, dipende dal tipo di utente, dal ruolo utente assegnato e dai gruppi utente.

Struttura L'API è composta da una serie di metodi raggruppati nominalmente in API separate. Ciascuno dei metodi esegue un'attività specifica. Ad esempio, il metodo `host.create` appartiene all'*host* API ed è usato per creare nuovi host. Storicamente, le API sono talvolta indicate come "classi".

La maggior parte delle API contiene almeno quattro metodi: `get`, `create`, `update` e `delete` per recuperare, creare, aggiornare ed eliminare i dati rispettivamente, ma alcune API possono fornire un insieme di metodi completamente diverso.

Esecuzione delle richieste Una volta configurato il frontend, puoi usare richieste HTTP remote per chiamare l'API. Per farlo, devi inviare richieste HTTP POST al file `api_jsonrpc.php` situato nella directory del frontend.
 Ad esempio, se il tuo frontend Zabbix è installato in `https://example.com/zabbix`, una richiesta HTTP per chiamare il metodo `apiinfo.version` potrebbe essere simile a questa:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'
```

La richiesta deve avere l'intestazione `Content-Type` impostata su uno di questi valori: `application/json-rpc`, `application/json` o `application/jsonrequest`.

L'oggetto della richiesta deve contenere le seguenti proprietà:

- `jsonrpc` - la versione del protocollo JSON-RPC utilizzata dall'API (Zabbix API implementa la versione 2.0 di JSON-RPC);
- `method` - il metodo API chiamato;
- `params` - i parametri che verranno passati al metodo API;
- `id` - un identificatore arbitrario della richiesta (se omesso, l'API considera la richiesta come una notifica).

Se la richiesta è corretta, la risposta restituita dall'API dovrebbe essere simile a questa:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

L'oggetto della risposta, a sua volta, contiene le seguenti proprietà:

- `jsonrpc` - la versione del protocollo JSON-RPC;
- `result` - i dati restituiti dal metodo;
- `id` - un identificatore della richiesta corrispondente.

Autenticazione Per accedere a qualsiasi dato in Zabbix, è necessario:

- Usare un token API esistente creato nel frontend di Zabbix, sezione *Users* > *API tokens*, oppure creato tramite la Token API.
- Usare un token di autenticazione ottenuto con il metodo `user.login`.

Ad esempio, se si volesse ottenere un nuovo token di autenticazione effettuando l'accesso come utente standard *Admin*, una richiesta JSON avrebbe il seguente aspetto:

```
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}
```

Se le credenziali sono state fornite correttamente, la risposta restituita dall'API dovrebbe contenere il token di autenticazione dell'utente:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Metodi di autorizzazione Tramite l'header "Authorization"

Tutte le richieste API richiedono un'autenticazione o un token API. Puoi fornire le credenziali utilizzando l'header Authorization nella richiesta:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
Se riscontri problemi di autenticazione, consulta inoltro dell'header Authorization.

Zabbix API accetta gli header in modo case-insensitive (ad esempio, `authorization`, `Authorization` e `AUTHORIZATION` vengono trattati allo stesso modo).

L'header Authorization è supportato nelle richieste cross-origin (CORS).

Tramite cookie di Zabbix

Un cookie *"zbx_session"* viene utilizzato per autorizzare una richiesta API dall'interfaccia utente di Zabbix eseguita tramite JavaScript (da un componente o da un widget personalizzato).

Flusso di lavoro di esempio La sezione seguente illustra in modo più dettagliato diversi esempi di utilizzo.

Recupero degli host Ora disponi di un token di autenticazione utente valido (rappresentato come variabile nei seguenti esempi) che può essere utilizzato per accedere ai dati in Zabbix. Ad esempio, puoi usare il metodo `host.get` per recuperare gli ID, i nomi degli host e le interfacce di tutti gli host configurati:

Richiesta:

```
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` è un file che contiene una query JSON. In alternativa a un file, puoi passare la query nell'argomento `--data`.

`data.json`

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

L'oggetto di risposta conterrà i dati richiesti sugli host:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
Per motivi di prestazioni, è sempre consigliato elencare le proprietà dell'oggetto che desideri recuperare. In questo modo eviterai di recuperare tutto.

Creazione di un nuovo item Ora, crea un nuovo item sull'host "Zabbix server" utilizzando i dati ottenuti dalla precedente richiesta `host.get`. Questo può essere fatto usando il metodo `item.create`:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

Una risposta riuscita conterrà l'ID del nuovo item creato, che può essere usato per fare riferimento all'item nelle richieste successive:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
Il metodo `item.create`, così come altri *metodi di creazione*, può anche accettare array di oggetti e creare più item con una sola chiamata API.

Creazione di più trigger Pertanto, se i *metodi di creazione* accettano array, puoi aggiungere più trigger, ad esempio questo:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

La risposta corretta conterrà gli ID dei trigger appena creati:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Aggiornamento di un item Abilita un item impostando il suo stato su 0:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

La risposta corretta conterrà l'ID dell'item aggiornato:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
Il metodo `item.update`, così come altri *metodi di aggiornamento*, può anche accettare array di oggetti e aggiornare più item con una sola chiamata API.

Aggiornamento di più trigger Abilita più trigger impostando il loro stato su 0:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.update","params":[{"triggerid":"13938","status":0},{"triggeri
```

La risposta corretta conterrà gli ID dei trigger aggiornati:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Gestione degli errori Fino a questo momento, tutto ciò che hai provato ha funzionato correttamente. Ma cosa succederebbe se provassi a effettuare una chiamata errata all'API? Prova a creare un altro host chiamando `host.create` ma omettendo il parametro obbligatorio `groups`:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

La risposta conterrà quindi un messaggio di errore:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Se si è verificato un errore, al posto della proprietà `result`, l'oggetto di risposta conterrà la proprietà `error` con i seguenti dati:

- `code` - un codice di errore;
- `message` - un breve riepilogo dell'errore;
- `data` - un messaggio di errore più dettagliato.

Gli errori possono verificarsi in vari casi, ad esempio quando si usano valori di input non corretti, si verifica un timeout della sessione oppure si tenta di accedere a oggetti inesistenti. La tua applicazione dovrebbe essere in grado di gestire in modo corretto questo tipo di errori.

Versioni API Per semplificare il versioning dell'API, a partire da Zabbix 2.0.4, la versione dell'API corrisponde alla versione di Zabbix stesso. È possibile usare il metodo `apiinfo.version` per scoprire la versione dell'API con cui si sta lavorando. Questo può essere utile per adattare la propria applicazione all'uso di funzionalità specifiche della versione.

Zabbix garantisce la compatibilità retroattiva delle funzionalità all'interno di una major version. Quando vengono introdotte modifiche non compatibili tra major release, Zabbix in genere lascia le vecchie funzionalità come deprecate nella release successiva e le rimuove solo nella release dopo quella. Occasionalmente, Zabbix può rimuovere funzionalità tra major release senza fornire alcuna compatibilità retroattiva. È importante non fare mai affidamento su funzionalità deprecate e migrare il prima possibile verso alternative più recenti.

> **Note:**
> È possibile seguire tutte le modifiche apportate all'API nel API changelog.

Ulteriori letture Ora hai conoscenze sufficienti per iniziare a lavorare con le API di Zabbix, tuttavia non fermarti qui. Per ulteriori letture, ti consigliamo di dare un'occhiata all'elenco delle API disponibili.

Riferimento ai metodi

Questa sezione fornisce una panoramica delle funzioni fornite dall'API di Zabbix e ti aiuterà a orientarti tra le classi e i metodi disponibili.

Monitoraggio L'API di Zabbix consente di accedere allo storico e ad altri dati raccolti durante il monitoraggio.

Dashboard

Gestisci le dashboard e crea report pianificati basati su di esse.

API della dashboard | API del template della dashboard | API del report

Cluster ad alta disponibilità

Recupera un elenco dei nodi server e il loro stato.

API del cluster ad alta disponibilità

Storico

Recupera i valori dello storico raccolti dai processi di monitoraggio di Zabbix per presentazione o ulteriori elaborazioni.

API dello storico

Trend

Recupera i valori dei trend calcolati dal server Zabbix per la presentazione o per ulteriori elaborazioni.

API del trend

Eventi

Recupera gli eventi generati da trigger, discovery di rete e altri sistemi Zabbix per una gestione più flessibile della situazione o per l'integrazione con strumenti di terze parti.

API dell'evento

Problemi

Recupera i problemi secondo i parametri dati.

API del problema

Mappe

Configura le mappe per creare rappresentazioni dinamiche dettagliate della tua infrastruttura IT.

API della mappa

Attività

Interagisci con il task manager del server Zabbix, creando attività e recuperando risposte.

API dell'attività

APM L'API di Zabbix consente di accedere ai dati di telemetria raccolti per il monitoraggio delle prestazioni delle applicazioni (APM).

Tracce

Recupera le tracce e gli span che le compongono.

API delle tracce APM | API degli span APM

Metriche

Recupera i punti dati delle metriche.

API delle metriche APM

Log

Recupera i record dei log.

API dei log APM

Servizi L'API Zabbix consente di accedere ai dati raccolti durante il monitoraggio dei servizi.

Service Level Agreement

Definire i Service Level Objectives (SLO), recuperare informazioni dettagliate sugli Indicatori dei Service Level Indicators (SLI) sulle prestazioni del servizio.

API SLA

Servizi

Gestisci i servizi per il monitoraggio dei livelli di servizio e recupera informazioni dettagliate sugli SLA di qualsiasi servizio.

Service API

Raccolta dati L'API di Zabbix consente di gestire la configurazione del sistema di monitoraggio.

Host e gruppi di host

Gestisci i gruppi di host, gli host e tutto ciò che li riguarda, incluse le interfacce degli host, le macro degli host e i periodi di manutenzione.

API Host | API dei gruppi di host | API delle macro utente | API delle mappe dei valori | API della manutenzione

Elementi

Definisci gli elementi da monitorare.

API dell'elemento

trigger

Configura i trigger per ricevere notifiche sui problemi nel sistema. Gestisci le dipendenze dei trigger.

Trigger API

Grafici

Modifica i grafici o separa gli elementi dei grafici per una migliore presentazione dei dati raccolti.

API del grafico | API dell'elemento del grafico

Template e gruppi di template

Gestisci i template e collegali agli host o ad altri template.

API del template | API del gruppo di template | API della mappa dei valori

Discovery di basso-livello

Configura le regole di discovery di basso-livello, nonché i prototipi di elementi, trigger e grafici per monitorare le entità dinamiche.

API della regola LLD | API del prototipo di elemento | API del prototipo di trigger | API del prototipo di grafico | API del prototipo di host

Complex event processing

Create complex event processing rules.

Complex event processing API

Correlazione degli eventi

Crea regole di correlazione degli eventi personalizzate.

API della correlazione

Discovery della rete

Gestisci le regole di discovery a livello di rete per trovare e monitorare automaticamente i nuovi host. Ottieni l'accesso completo alle informazioni sui servizi e gli host discoverati.

API della regola di discovery | API di controllo della discovery | API dell'host discoverato | API del servizio discoverato

Esportazione e importazione

Esporta e importa i dati di configurazione di Zabbix per backup di configurazione, migrazioni o aggiornamenti di configurazione su larga scala.

API della configurazione

Monitoraggio web

Configura scenari web per monitorare le tue applicazioni e servizi web.

API dello scenario web

Avvisi L'API Zabbix ti consente di gestire le azioni e gli avvisi del tuo sistema di monitoraggio.

Azioni e avvisi

Definisci azioni e operazioni per notificare agli utenti determinati eventi o eseguire automaticamente comandi remoti. Accedi alle informazioni sugli avvisi generati e sui relativi destinatari.

API dell'zione | API dell'avviso

Tipi di supporto

Configura i tipi di supporto e le diverse modalità di ricezione degli avvisi da parte degli utenti.

API del tipo di supporto

Script

Configura ed esegui script per aiutarti nelle tue attività quotidiane.

API dello script

Utenti L'API Zabbix ti consente di gestire gli utenti del tuo sistema di monitoraggio.

Utenti e gruppi di utenti

Aggiungi gli utenti che avranno accesso a Zabbix, assegnali ai gruppi di utenti e concedi i permessi. Crea ruoli per una gestione granulare dei diritti degli utenti.

API dell'utente | API del gruppo di utenti | API della directory dell'utente | API del ruolo dell'utente

Token API

Gestisci i token di autorizzazione.

API del token

Autenticazione

Modifica le opzioni di configurazione dell'autenticazione.

API dell'autenticazione

Dispositivi

Gestisci i dispositivi mobili da utilizzare con l'app mobile Zabbix.

API dei dispositivi

Amministrazione Con l'API Zabbix puoi modificare le impostazioni di amministrazione del tuo sistema di monitoraggio.

Generale

Modifica alcune opzioni di configurazione globali.

API della registrazione automatica | API della mappa di icone | API dell'immagine | API delle impostazioni | API dell'espressione regolare | API del componente | API del connettore

Registro di audit

Tieni traccia delle modifiche alla configurazione apportate da ciascun utente.

API del registro di audit

Housekeeping

Configura l'housekeeping.

API dell'housekeeping

Proxy e gruppi di proxy

Gestisci i proxy utilizzati nella tua configurazione di monitoraggio distribuito.

API del proxy | API del gruppo di proxy

Macro

Gestisci le macro.

API della macro dell'utente

Informazioni su Zabbix Recupera la versione dell'API di Zabbix o l'ora dell'ultimo aggiornamento dal server Zabbix.

API info API | Server info API

Alert

Questa classe è progettata per lavorare con gli avvisi.

Riferimenti oggetto:

- Alert

Metodi disponibili:

- alert.get - recupera gli avvisi

Oggetto di un avviso

I seguenti oggetti sono direttamente correlati all'API `alert`.

Avviso

> **Note:**
> Gli avvisi vengono creati da Zabbix server e non possono essere modificati tramite l'API.

L'oggetto alert contiene informazioni sull'esecuzione corretta o meno di determinate operazioni di azione. Ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
alertid	ID	ID dell'avviso.
actionid	ID	ID dell'azione che ha generato l'avviso.
alerttype	integer	Tipo di avviso. Valori possibili: 0 - messaggio; 1 - comando remoto.
clock	timestamp	Ora in cui l'avviso è stato generato.
error	string	Testo dell'errore se si verificano problemi durante l'invio di un messaggio o l'esecuzione di un comando.
esc_step	integer	Passo di escalation dell'azione durante il quale è stato generato l'avviso.
eventid	ID	ID dell'evento che ha attivato l'azione.
mediatypeid	ID	ID del tipo di supporto utilizzato per inviare il messaggio.
message	text	Testo del messaggio. Comportamento della proprietà: - *supportato* se `alerttype` è impostato su "message"
retries	integer	Numero di volte in cui Zabbix ha tentato di inviare il messaggio.
sendto	string	Indirizzo, nome utente o altro identificatore del destinatario. Comportamento della proprietà: - *supportato* se `alerttype` è impostato su "message"

Proprietà	Tipo	Descrizione
status	integer	Stato che indica se l'operazione dell'azione è stata eseguita correttamente. Valori possibili se `alerttype` è impostato su "message": 0 - messaggio non inviato; 1 - messaggio inviato; 2 - non riuscito dopo un certo numero di tentativi; 3 - il nuovo avviso non è ancora stato elaborato dal gestore degli avvisi. Valori possibili se `alerttype` è impostato su "remote command": 0 - comando non eseguito; 1 - comando eseguito; 2 - è stato tentato l'avvio del comando su Zabbix agent, ma non era disponibile.
subject	string	Oggetto del messaggio. Comportamento della proprietà: - *supportato* se `alerttype` è impostato su "message"
userid	ID	ID dell'utente a cui è stato inviato il messaggio.
p_eventid	ID	ID dell'evento del problema che ha generato l'avviso.
acknowledgeid	ID	ID della conferma che ha generato l'avviso.

alert.get

Descrizione

`integer/array alert.get(object parameters)`

Il metodo consente di recuperare gli alert in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
alertids	ID/array	Restituisce solo gli avvisi con gli ID specificati.
actionids	ID/array	Restituisce solo gli avvisi generati dalle azioni specificate.
eventids	ID/array	Restituisce solo gli avvisi generati dagli eventi specificati.
groupids	ID/array	Restituisce solo gli avvisi generati da oggetti dei gruppi host specificati.
hostids	ID/array	Restituisce solo gli avvisi generati da oggetti degli host specificati.
mediatypeids	ID/array	Restituisce solo gli avvisi di messaggio che hanno utilizzato i tipi di supporto specificati.
objectids	ID/array	Restituisce solo gli avvisi generati dagli oggetti specificati
userids	ID/array	Restituisce solo gli avvisi di messaggio inviati agli utenti specificati.
eventobject	integer	Restituisce solo gli avvisi generati da eventi relativi a oggetti del tipo specificato. Vedere `object` dell'evento per un elenco dei tipi di oggetto supportati. Predefinito: 0 - trigger.
eventsource	integer	Restituisce solo gli avvisi generati da eventi del tipo specificato. Vedere `source` dell'evento per un elenco dei tipi di evento supportati. Predefinito: 0 - eventi trigger.

Parametro	Tipo	Descrizione
time_from	timestamp	Restituisce solo gli avvisi generati dopo l'ora specificata.
time_till	timestamp	Restituisce solo gli avvisi generati prima dell'ora specificata.
selectHosts	query	Restituisce una proprietà `hosts` con i dati degli host che hanno attivato l'operazione dell'azione.
selectMediatypes	query	Restituisce una proprietà `mediatypes` con un array dei tipi di supporto utilizzati per l'avviso di messaggio. Vedere `mediatype.get` per le limitazioni in base al tipo di utente.
selectUsers	query	Restituisce una proprietà `users` con un array degli utenti a cui era indirizzato il messaggio. Vedere `user.get` per le limitazioni in base al tipo di utente.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` ha stato usato.

Esempi

Recuperare gli avvisi per ID azione

Recupera tutti gli avvisi generati dall'azione "3".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
```

```
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent sul server Linux non è raggiungibile da 5 minuti: ",
            "message": "Trigger: Zabbix agent sul server Linux non è raggiungibile da 5 minuti: \nStato de
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```


Vedi anche

- Host
- Tipo di supporto
- Utente

Sorgente

CAlert::get() in *ui/include/classes/api/services/CAlert.php*.

Autenticazione

Questa classe è progettata per lavorare con le impostazioni di autenticazione.

Riferimenti oggetto:

- Autenticazione

Metodi disponibili:

- authentication.get - recupera le impostazioni di autenticazione
- authentication.update - aggiorna le impostazioni di autenticazione

Oggetto di autenticazione

I seguenti oggetti sono direttamente correlati all'API `authentication`.

Autenticazione

L'oggetto di autenticazione ha le seguenti proprietà.

Property	Type	Description
authentication_type	integer	Autenticazione predefinita. Valori possibili: 0 - *(predefinita)* Interna; 1 - LDAP.
http_auth_enabled	integer	Autenticazione HTTP. Valori possibili: 0 - *(predefinita)* Disabilitata; 1 - Abilitata. Comportamento della proprietà: - *supportata* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` è impostato su `true` nel file di configurazione del frontend.

Property	Type	Description
http_login_form	integer	Modulo di accesso predefinito. Valori possibili: 0 - *(predefinito)* Modulo di accesso Zabbix; 1 - Modulo di accesso HTTP. Comportamento della proprietà: - *supportata* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` è impostato su `true` nel file di configurazione del frontend.
http_strip_domains	string	Nome del dominio da rimuovere. Comportamento della proprietà: - *supportata* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` è impostato su `true` nel file di configurazione del frontend.
http_case_sensitive	integer	Accesso HTTP con distinzione tra maiuscole e minuscole. Valori possibili: 0 - Disattivato; 1 - *(predefinito)* Attivato. Comportamento della proprietà: - *supportata* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` è impostato su `true` nel file di configurazione del frontend.
ldap_auth_enabled	integer	Autenticazione LDAP. Valori possibili: 0 - *(predefinita)* Disabilitata; 1 - Abilitata.
ldap_case_sensitive	integer	Accesso LDAP con distinzione tra maiuscole e minuscole. Valori possibili: 0 - Disattivato; 1 - *(predefinito)* Attivato.
ldap_userdirectoryid	ID	ID della directory utenti predefinita per l'autenticazione LDAP. Utilizzato per i gruppi utenti con `gui_access` impostato su LDAP o Predefinito di sistema. Comportamento della proprietà: - *obbligatoria* se `ldap_auth_enabled` è impostato su "Abilitata"
saml_auth_enabled	integer	Autenticazione SAML. Valori possibili: 0 - *(predefinita)* Disabilitata; 1 - Abilitata.
saml_case_sensitive	integer	Accesso SAML con distinzione tra maiuscole e minuscole. Valori possibili: 0 - Disattivato; 1 - *(predefinito)* Attivato.
passwd_min_length	integer	Requisito di lunghezza minima della password. L'intervallo dei valori possibili va da 1 a 70. Predefinito: 8.

Property	Type	Description
passwd_check_rules	integer	Regole di verifica della password. Valori bitmap possibili: 0 - Verifica la lunghezza della password; 1 - Verifica se la password utilizza lettere latine maiuscole e minuscole; 2 - Verifica se la password utilizza cifre; 4 - Verifica se la password utilizza caratteri speciali; 8 - *(predefinito)* Verifica che la password non sia inclusa nell'elenco delle password comunemente utilizzate e non contenga derivazioni della parola "Zabbix" o del nome, cognome o nome utente dell'utente. Questo è un campo con maschera di bit; è accettabile qualsiasi somma dei valori bitmap possibili (ad esempio, 15 per verificare tutte le regole).
ldap_jit_status	integer	Stato del provisioning LDAP. Valori possibili: 0 - Disabilitato per gli IdP LDAP configurati; 1 - Abilitato per gli IdP LDAP configurati.
saml_jit_status	integer	Stato del provisioning SAML. Valori possibili: 0 - Disabilitato per gli IdP SAML configurati; 1 - Abilitato per gli IdP SAML configurati.
jit_provision_interval	string	Intervallo di tempo tra le richieste di provisioning JIT per l'utente connesso. Accetta un numero di secondi (ad esempio, 3600) o un valore con un suffisso temporale (ad esempio, `3600s`, `60m`, `1h`, `1d`, `1M`, `1y`), incluso il supporto per mesi e anni. Valore minimo: `1h`. Predefinito: `1h`.
disabled_usrgrpid	ID	Disponibile solo per il provisioning LDAP. ID del gruppo utenti a cui assegnare l'utente a cui è stato rimosso il provisioning. Il gruppo utenti deve essere disabilitato e non può essere abilitato o eliminato quando è configurato. Comportamento della proprietà: - *obbligatoria* se `ldap_jit_status` è impostato su "Abilitato per gli IdP LDAP configurati" oppure `saml_jit_status` è impostato su "Abilitato per gli IdP SAML configurati"
mfa_status	integer	Autenticazione a più fattori. Valori possibili: 0 - Disabilitata (per tutti i metodi MFA configurati); 1 - Abilitata (per tutti i metodi MFA configurati).
mfaid	ID	Metodo MFA predefinito per i gruppi utenti con MFA abilitata. Comportamento della proprietà: - *obbligatoria* se `mfa_status` è impostato su "Abilitata"

authentication.get

Descrizione

`object authentication.get(object parameters)`

Il metodo consente di recuperare l'oggetto di autenticazione in base ai parametri specificati.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta un solo parametro.

Parametro	Tipo	Descrizione
output	query	Questo parametro è descritto nel commento di riferimento.

Valori restituiti

`(object)` Restituisce l'oggetto di autenticazione.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Sorgente

CAuthentication::get() in *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

Descrizione

`object authentication.update(object authentication)`

Questo metodo consente di aggiornare le impostazioni di autenticazione esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object)` Proprietà di autenticazione da aggiornare.

Valori restituiti

`(array)` Restituisce un array con i nomi dei parametri aggiornati.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

Sorgente

CAuthentication::update() in *ui/include/classes/api/services/CAuthentication.php*.

Azione

Questa classe è progettata per funzionare con le azioni.

Riferimenti agli oggetti:

- Azione
- Operazione dell'azione
  - Messaggio dell'operazione dell'azione
  - Condizione dell'operazione dell'azione
- Operazione di ripristino dell'azione
- Operazione di aggiornamento dell'azione
- Filtro dell'azione
  - Condizione del filtro dell'azione

Metodi disponibili:

- action.create - crea nuove azioni
- action.delete - elimina azioni
- action.get - recupera azioni
- action.update - aggiorna azioni

Oggetto azione

I seguenti oggetti sono direttamente correlati all'API `action`.

Azione

L'oggetto azione dispone delle seguenti proprietà.

Property	Type	Description
actionid	ID	ID dell'azione. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
esc_period	string	Durata predefinita del passaggio dell'operazione. Deve essere di almeno 60 secondi. Accetta un numero di secondi (ad esempio, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h) o una macro utente. Comportamento della proprietà: - *supportata* se `eventsource` è impostato su "evento creato da un trigger", "evento interno" o "evento creato durante l'aggiornamento dello stato del servizio"
eventsource	integer	Tipo di eventi gestiti dall'azione. Per un elenco dei tipi di evento supportati, fare riferimento alla proprietà `source` dell'evento. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
name	string	Nome dell'azione. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
status	integer	Indica se l'azione è abilitata o disabilitata. Valori possibili: 0 - *(predefinito)* abilitata; 1 - disabilitata.
pause_symptoms	integer	Indica se mettere in pausa l'escalation quando l'evento è un evento sintomo. Valori possibili: 0 - Non mettere in pausa l'escalation per i problemi sintomo; 1 - *(predefinito)* Mettere in pausa l'escalation per i problemi sintomo. Comportamento della proprietà: - *supportata* se `eventsource` è impostato su "evento creato da un trigger"
pause_suppressed	integer	Indica se mettere in pausa l'escalation durante i periodi di manutenzione. Valori possibili: 0 - Non mettere in pausa l'escalation; 1 - *(predefinito)* Mettere in pausa l'escalation. Comportamento della proprietà: - *supportata* se `eventsource` è impostato su "evento creato da un trigger"

Property	Type	Description
notify_if_canceled	integer	Indica se inviare una notifica quando l'escalation viene annullata. Valori possibili: 0 - Non inviare una notifica quando l'escalation viene annullata; 1 - *(predefinito)* Inviare una notifica quando l'escalation viene annullata. Comportamento della proprietà: - *supportata* se `eventsource` è impostato su "evento creato da un trigger"

Operazione dell'azione

L'oggetto operazione dell'azione definisce un'operazione che verrà eseguita quando viene eseguita un'azione. Presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
operationtype	integer	Tipo di operazione. Valori possibili: 0 - invia messaggio; 1 - script globale; 2 - aggiungi host; 3 - rimuovi host; 4 - aggiungi al gruppo di host; 5 - rimuovi dal gruppo di host; 6 - collega template; 7 - scollega template; 8 - abilita host; 9 - disabilita host; 10 - imposta la modalità di inventario dell'host; 13 - aggiungi tag agli host; 14 - rimuovi tag dagli host. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger" o "evento creato durante l'aggiornamento dello stato di un servizio": 0 - "invia messaggio"; 1 - "script globale". Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento interno": 0 - "invia messaggio". Comportamento della proprietà: - *obbligatoria*
esc_period	string	Durata di un passaggio di escalation in secondi. Deve essere superiore a 60 secondi. Accetta un numero di secondi (ad esempio, 60, 300), un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`) o una macro utente. Se impostato su 0 o 0s, verrà utilizzato il periodo di escalation predefinito dell'azione. Predefinito: 0s. Comportamento della proprietà: - *supportata* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", "evento interno" o "evento creato durante l'aggiornamento dello stato di un servizio"

Proprietà	Tipo	Descrizione
esc_step_from	integer	Passaggio da cui iniziare l'escalation. Predefinito: 1. Comportamento della proprietà: - *supportata* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", "evento interno" o "evento creato durante l'aggiornamento dello stato di un servizio"
esc_step_to	integer	Passaggio in cui terminare l'escalation. Predefinito: 1. Comportamento della proprietà: - *supportata* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", "evento interno" o "evento creato durante l'aggiornamento dello stato di un servizio"
evaltype	integer	Metodo di valutazione delle condizioni dell'operazione. Valori possibili: 0 - *(predefinito)* And/Or; 1 - And; 2 - Or.
opcommand	object	Script globale da eseguire. Lo script globale deve avere la proprietà `scriptid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "script globale"
opcommand_grp	array	Gruppi di host sui quali eseguire gli script globali. I gruppi di host devono avere la proprietà `groupid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "script globale" e `opcommand_hst` non è impostato
opcommand_hst	array	Host sul quale eseguire gli script globali. Gli host devono avere la proprietà `hostid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "script globale" e `opcommand_grp` non è impostato
opconditions	array	Condizioni dell'operazione utilizzate per le azioni dei trigger. L'oggetto condizione dell'operazione è descritto dettagliatamente di seguito.
opgroup	array	Gruppi di host ai quali aggiungere gli host. I gruppi di host devono avere la proprietà `groupid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "aggiungi al gruppo di host" o "rimuovi dal gruppo di host"
opmessage	object	Oggetto contenente i dati relativi al messaggio inviato dall'operazione. L'oggetto messaggio dell'operazione è descritto dettagliatamente di seguito. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "invia messaggio"

Proprietà	Tipo	Descrizione
opmessage_grp	array	Gruppi di utenti ai quali inviare i messaggi. I gruppi di utenti devono avere la proprietà `usrgrpid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su ”invia messaggio” e `opmessage_usr` non è impostato
opmessage_usr	array	Utenti ai quali inviare i messaggi. Gli utenti devono avere la proprietà `userid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su ”invia messaggio” e `opmessage_grp` non è impostato
optemplate	array	Template da collegare agli host. I template devono avere la proprietà `templateid` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su ”collega template” o ”scollega template”
opinventory	object	Modalità di inventario da impostare sull’host. L’inventario deve avere la proprietà `inventory_mode` definita. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su ”imposta la modalità di inventario dell’host”
optag	array	Tag degli host da aggiungere o rimuovere. I tag devono avere la proprietà `tag` definita. La proprietà `value` è facoltativa. Comportamento della proprietà: - *supportata* se `operationtype` è impostato su ”aggiungi tag agli host” o ”rimuovi tag dagli host”.

Messaggio dell’operazione dell’azione

L’oggetto del messaggio dell’operazione contiene i dati relativi al messaggio che verrà inviato dall’operazione. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
default_msg	integer	Indica se utilizzare il testo e l’oggetto predefiniti del messaggio dell’azione. Valori possibili: 0 - utilizzare i dati dell’operazione; 1 - *(predefinito)* utilizzare i dati del tipo di supporto.
mediatypeid	ID	ID del tipo di supporto che verrà utilizzato per inviare il messaggio. Comportamento della proprietà: - *supportata* se `operationtype` di Action operation object, Action recovery operation object o Action update operation object è impostato su ”send message”, oppure se `operationtype` di Action update operation object è impostato su ”notify all involved”

Proprietà	Type	Descrizione
message	string	Testo del messaggio dell'operazione. Comportamento della proprietà: - *supportata* se `default_msg` è impostato su "use the data from the operation"
subject	string	Oggetto del messaggio dell'operazione. Comportamento della proprietà: - *supportata* se `default_msg` è impostato su "use the data from the operation"

Condizione di operazione dell'azione

L'oggetto condizione di operazione dell'azione definisce una condizione che deve essere soddisfatta per eseguire l'operazione corrente. Ha le seguenti proprietà.

Property	Type	Description
conditiontype	integer	Tipo di condizione. Valori possibili: 14 - evento riconosciuto. Comportamento della proprietà: - *obbligatorio*
value	string	Valore con cui confrontare. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione. Valori possibili: 0 - *(predefinito)* =

I seguenti operatori e valori sono supportati per ciascun tipo di condizione di operazione.

Condition	Condition name	Supported operators	Expected value
14	Evento riconosciuto	=	Se l'evento è riconosciuto. Valori possibili: 0 - non riconosciuto; 1 - riconosciuto.

Operazione di ripristino dell'azione

L'oggetto operazione di ripristino dell'azione definisce un'operazione che verrà eseguita quando un problema viene risolto. Le operazioni di ripristino sono possibili **solo** per azioni di trigger, interne e di servizio. Ha le seguenti proprietà.

Property	Type	Description
operationtype	integer	Tipo di operazione. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger" o "evento creato all'aggiornamento dello stato del servizio": 0 - invia messaggio; 1 - script globale; 11 - notifica tutti i coinvolti. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento interno": 0 - invia messaggio; 11 - notifica tutti i coinvolti. Comportamento della proprietà: - *obbligatoria*
opcommand	object	Script globale da eseguire. Lo script globale deve avere definita la proprietà `scriptid`. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "script globale"
opcommand_grp	array	Gruppi di host su cui eseguire script globali. I gruppi di host devono avere definita la proprietà `groupid`. Comportamento della proprietà: - *obbligatoria* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", `operationtype` è impostato su "script globale" e `opcommand_hst` non è impostato
opcommand_hst	array	Host su cui eseguire script globali. Gli host devono avere definita la proprietà `hostid`. Comportamento della proprietà: - *obbligatoria* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", `operationtype` è impostato su "script globale" e `opcommand_grp` non è impostato
opmessage	object	Oggetto contenente i dati relativi al messaggio inviato dall'operazione di ripristino. L'oggetto messaggio dell'operazione è descritto in dettaglio sopra. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "invia messaggio"
opmessage_grp	array	Gruppi di utenti a cui inviare messaggi. I gruppi di utenti devono avere definita la proprietà `usrgrpid`. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "invia messaggio" e `opmessage_usr` non è impostato
opmessage_usr	array	Utenti a cui inviare messaggi. Gli utenti devono avere definita la proprietà `userid`. Comportamento della proprietà: - *obbligatoria* se `operationtype` è impostato su "invia messaggio" e `opmessage_grp` non è impostato

Operazione di aggiornamento dell'azione

L'oggetto operazione di aggiornamento dell'azione definisce un'operazione che verrà eseguita quando un problema viene aggiornato (con un commento, confermato, con gravità modificata o chiuso manualmente). Le operazioni di aggiornamento sono possibili **solo** per le azioni di trigger e di servizio. Ha le seguenti proprietà.

Property	Type	Description
operationtype	integer	Tipo di operazione. Valori possibili: 0 - invia messaggio; 1 - script globale; 12 - notifica tutti i soggetti coinvolti. Property behavior: - *required*
opcommand	object	Script globale da eseguire. Lo script globale deve avere definita la proprietà `scriptid`. Property behavior: - *required* se `operationtype` è impostato su "script globale"
opcommand_grp	array	Gruppi di host sui quali eseguire gli script globali. I gruppi di host devono avere definita la proprietà `groupid`. Property behavior: - *required* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", `operationtype` è impostato su "script globale" e `opcommand_hst` non è impostato
opcommand_hst	array	Host sui quali eseguire gli script globali. Gli host devono avere definita la proprietà `hostid`. Property behavior: - *required* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", `operationtype` è impostato su "script globale" e `opcommand_grp` non è impostato
opmessage	object	Oggetto contenente i dati relativi al messaggio inviato dall'operazione di aggiornamento. L'oggetto messaggio dell'operazione è descritto in dettaglio sopra.
opmessage_grp	array	Gruppi di utenti a cui inviare messaggi. I gruppi di utenti devono avere definita la proprietà `usrgrpid`. Property behavior: - *required* se `operationtype` è impostato su "invia messaggio" e `opmessage_usr` non è impostato
opmessage_usr	array	Utenti a cui inviare messaggi. Gli utenti devono avere definita la proprietà `userid`. Property behavior: - *required* se `operationtype` è impostato su "invia messaggio" e `opmessage_grp` non è impostato

Filtro dell'azione

L'oggetto filtro dell'azione definisce un insieme di condizioni che devono essere soddisfatte per eseguire le operazioni dell'azione configurata. Ha le seguenti proprietà.

Property	Type	Description
conditions	array	Insieme di condizioni del filtro da utilizzare per filtrare i risultati. Le condizioni saranno ordinate in base al loro posizionamento nella formula. Property behavior: - *required*
evaltype	integer	Metodo di valutazione delle condizioni del filtro. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Property behavior: - *required*
eval_formula	string	Espressione generata che verrà utilizzata per valutare le condizioni del filtro. L'espressione contiene ID che fanno riferimento a specifiche condizioni del filtro tramite il relativo `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Property behavior: - *read-only*
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a specifiche condizioni del filtro tramite il relativo `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni del filtro: nessuna condizione può rimanere inutilizzata o essere omessa. Property behavior: - *required* se `evaltype` è impostato su "custom expression"

Condizione del filtro dell'azione

L'oggetto della condizione del filtro dell'azione definisce una condizione specifica che deve essere verificata prima di eseguire le operazioni dell'azione.

Proprietà	Tipo	Descrizione
conditiontype	integer	Tipo di condizione. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger": 0 - gruppo di host; 1 - host; 2 - trigger; 3 - nome dell'evento; 4 - gravità del trigger; 6 - periodo di tempo; 13 - template dell'host; 16 - il problema è soppresso; 25 - tag dell'evento; 26 - valore del tag dell'evento. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da una regola di discovery": 7 - IP dell'host; 8 - tipo di servizio rilevato; 9 - porta del servizio rilevato; 10 - stato della discovery; 11 - durata di uptime o downtime; 12 - valore ricevuto; 18 - regola di discovery; 19 - controllo di discovery; 20 - proxy; 21 - oggetto rilevato; 29 - gruppo di proxy. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato dall'autoregistrazione dell'agent attivo": 20 - proxy; 22 - nome dell'host; 24 - metadati dell'host; 29 - gruppo di proxy. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento interno": 0 - gruppo di host; 1 - host; 13 - template dell'host; 23 - tipo di evento; 25 - tag dell'evento; 26 - valore del tag dell'evento. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato durante l'aggiornamento dello stato del servizio": 25 - tag dell'evento; 26 - valore del tag dell'evento; 27 - servizio; 28 - nome del servizio. Comportamento della proprietà: - *obbligatoria*
value	string	Valore con cui effettuare il confronto. Comportamento della proprietà: - *obbligatoria*

Proprietà	Tipo	Descrizione
value2	string	Valore secondario con cui effettuare il confronto. Comportamento della proprietà: - *obbligatoria* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", `conditiontype` è impostato su un valore possibile per le azioni dei trigger e il tipo di condizione (vedere di seguito) è "26" - *obbligatoria* se `eventsource` dell'oggetto Action è impostato su "evento interno", `conditiontype` è impostato su un valore possibile per le azioni interne e il tipo di condizione (vedere di seguito) è "26" - *obbligatoria* se `eventsource` dell'oggetto Action è impostato su "evento creato durante l'aggiornamento dello stato del servizio", `conditiontype` è impostato su un valore possibile per le azioni dei servizi e il tipo di condizione (vedere di seguito) è "26"
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente durante la modifica delle condizioni del filtro, ma verrà generato nuovamente quando queste verranno richieste in seguito.
operator	integer	Operatore della condizione. Valori possibili: 0 - *(predefinito)* uguale; 1 - diverso; 2 - contiene; 3 - non contiene; 4 - incluso in; 5 - maggiore o uguale a; 6 - minore o uguale a; 7 - non incluso in; 8 - corrisponde; 9 - non corrisponde; 10 - Sì; 11 - No.

Note:
Per comprendere meglio come utilizzare i filtri con vari tipi di espressioni, vedere gli esempi nelle pagine dei metodi action.get e action.create.

Gli operatori e i valori seguenti sono supportati per ogni tipo di condizione.

Condizione	Nome della condizione	Operatori supportati	Valore previsto
0	Gruppo di host	uguale, diverso	ID del gruppo di host.
1	Host	uguale, diverso	ID dell'host.
2	Trigger	uguale, diverso	ID del trigger.
3	Nome dell'evento	contiene, non contiene	Nome dell'evento.
4	Gravità del trigger	uguale, diverso, maggiore o uguale a, minore o uguale a	Gravità del trigger. Consultare la proprietà `severity` del trigger per un elenco delle gravità del trigger supportate.
5	Valore del trigger	uguale	Valore del trigger. Consultare la proprietà `value` del trigger per un elenco dei valori del trigger supportati.
6	Periodo di tempo	incluso in, non incluso in	Momento in cui l'evento è stato attivato, espresso come periodo di tempo.

Condizione	Nome della condizione	Operatori supportati	Valore previsto
7	IP dell'host	uguale, diverso	Uno o più intervalli di IP da verificare, separati da virgole. Consultare la sezione configurazione della network discovery per ulteriori informazioni sui formati supportati degli intervalli di IP.
8	Tipo di servizio rilevato	uguale, diverso	Tipo di servizio rilevato. Il tipo di servizio corrisponde al tipo di controllo di discovery utilizzato per rilevare il servizio. Consultare la proprietà `type` del controllo di discovery per un elenco dei tipi supportati.
9	Porta del servizio rilevato	uguale, diverso	Uno o più intervalli di porte, separati da virgole.
10	Stato della discovery	uguale	Stato di un oggetto rilevato. Valori possibili: 0 - host o servizio attivo; 1 - host o servizio non attivo; 2 - host o servizio rilevato; 3 - host o servizio non più rilevato.
11	Durata di uptime o downtime	maggiore o uguale a, minore o uguale a	Tempo che indica da quanti secondi l'oggetto rilevato si trova nello stato corrente.
12	Valori ricevuti	uguale, diverso, maggiore o uguale a, minore o uguale a, contiene, non contiene	Valore restituito durante l'esecuzione di un controllo di discovery tramite Zabbix agent, SNMPv1, SNMPv2 o SNMPv3.
13	Template dell'host	uguale, diverso	ID del template collegato.
16	Il problema è soppresso	Sì, No	Non è richiesto alcun valore: l'operatore "Sì" indica che il problema deve essere soppresso, mentre "No" indica che non deve essere soppresso.
18	Regola di discovery	uguale, diverso	ID della regola di discovery.
19	Controllo di discovery	uguale, diverso	ID del controllo di discovery.
20	Proxy	uguale, diverso	ID del proxy.
21	Oggetto rilevato	uguale	Tipo di oggetto che ha attivato l'evento di discovery. Valori possibili: 1 - host rilevato; 2 - servizio rilevato.
22	Nome dell'host	contiene, non contiene, corrisponde, non corrisponde	Nome dell'host. L'uso di un'espressione regolare è supportato per gli operatori *corrisponde* e *non corrisponde* nelle condizioni di autoregistrazione.
23	Tipo di evento	uguale	Evento interno specifico. Valori possibili: 0 - item nello stato "non supportato"; 1 - item nello stato "normale"; 2 - regola LLD nello stato "non supportato"; 3 - regola LLD nello stato "normale"; 4 - trigger nello stato "sconosciuto"; 5 - trigger nello stato "normale".
24	Metadati dell'host	contiene, non contiene, corrisponde, non corrisponde	Metadati dell'host autoregistrato. L'uso di un'espressione regolare è supportato per gli operatori *corrisponde* e *non corrisponde*.

Condizione	Nome della condizione	Operatori supportati	Valore previsto
25	Tag	uguale, diverso, contiene, non contiene	Tag dell'evento.
26	Valore del tag	uguale, diverso, contiene, non contiene	Valore del tag dell'evento.
27	Servizio	uguale, diverso	ID del servizio.
28	Nome del servizio	uguale, diverso	Nome del servizio.
29	Gruppo di proxy	uguale, diverso	ID del gruppo di proxy.

action.create

Descrizione

`object action.create(object/array actions)`

Questo metodo consente di creare nuove azioni.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Azioni da creare.

Oltre alle proprietà standard dell'azione, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
filter	object	Oggetto filtro dell'azione per l'azione.
operations	array	Operazioni dell'azione da creare per l'azione.
recovery_operations	array	Operazioni di ripristino dell'azione da creare per l'azione.
update_operations	array	Operazioni di aggiornamento dell'azione da creare per l'azione.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle azioni create sotto la proprietà `actionids`. L'ordine degli ID restituiti corrisponde all'ordine delle azioni passate.

Esempi

Creare un'azione trigger

Creare un'azione trigger che verrà avviata quando un trigger (con la parola "memory" nel nome) del host "10084" passa allo stato PROBLEM. L'azione avrà 4 operazioni configurate. La prima operazione, eseguita immediatamente, invierà un messaggio a tutti gli utenti del gruppo utenti "7" tramite il tipo di supporto "1". Se l'evento non viene risolto entro 30 minuti, verrà eseguita la seconda operazione script "5" (script con ambito "Action operation") su tutti gli host del gruppo "2". Se l'evento viene risolto, un'operazione di ripristino notificherà tutti gli utenti che hanno ricevuto messaggi relativi al problema. Se l'evento viene aggiornato, un'operazione di conferma/aggiornamento notificherà (con un oggetto e un messaggio personalizzati) tutti gli utenti che hanno ricevuto messaggi relativi al problema.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

Creare un'azione di discovery

Creare un'azione di discovery che collegherà il template "10001" agli host rilevati.

Richiesta:

```
{
    "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
```

```
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Utilizzo di un filtro di espressione personalizzato

Creare un'azione trigger che utilizzi un'espressione personalizzata - "A and (B or C)" - per valutare le condizioni dell'azione. Quando un trigger con gravità superiore o uguale a "Warning" dal host "10084" o dal host "10106" passa allo stato PROBLEM, l'azione invierà un messaggio a tutti gli utenti del gruppo utenti "7" tramite il tipo di supporto "1". Gli ID formula "A", "B" e "C" sono stati scelti arbitrariamente.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Creare una regola di autoregistrazione dell'agent

Creare un'azione di autoregistrazione che aggiunga un host al gruppo host "2" quando il nome dell'host contiene "SRV" oppure i metadati contengono "AlmaLinux".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

Creare una regola di autoregistrazione dell'agent con tag host

Creare un'azione di autoregistrazione che aggiunga un host al gruppo host "2" e aggiunga due tag host.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

Vedi anche

- Filtro azione
- Operazione dell'azione
- Script

Sorgente

CAction::create() in *ui/include/classes/api/services/CAction.php*.

action.delete

Descrizione

`object action.delete(array actionIds)`

Questo metodo consente di eliminare le azioni.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.
 L'utente non potrà eliminare un'azione che contiene un proxy o un gruppo di proxy non accessibile nelle condizioni del filtro. Per ulteriori informazioni, vedere configurazione del gruppo utenti (*Elenco di accesso ai proxy*).

Parametri

`(array)` ID delle azioni da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle azioni eliminate nella proprietà `actionids`.

Esempi

Eliminare molteplici azioni

Elimina due azioni.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "action.delete",
"params": [
"17",
"18"
],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"actionids": [
"17",
"18"
]
},
"id": 1
}
```

Sorgente

CAction::delete() in *ui/include/classes/api/services/CAction.php*.

action.get

Descrizione

`integer/array action.get(object parameters)`

Il metodo consente di recuperare le azioni in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
actionids	ID/array	Restituisce solo le azioni con gli ID specificati.
groupids	ID/array	Restituisce solo le azioni che utilizzano i gruppi di host specificati nelle condizioni dell'azione.
hostids	ID/array	Restituisce solo le azioni che utilizzano gli host specificati nelle condizioni dell'azione.
triggerids	ID/array	Restituisce solo le azioni che utilizzano i trigger specificati nelle condizioni dell'azione.
mediatypeids	ID/array	Restituisce solo le azioni che utilizzano i tipi di supporto specificati per inviare messaggi.
usrgrpids	ID/array	Restituisce solo le azioni configurate per inviare messaggi ai gruppi di utenti specificati.
userids	ID/array	Restituisce solo le azioni configurate per inviare messaggi agli utenti specificati.
scriptids	ID/array	Restituisce solo le azioni configurate per eseguire gli script specificati.
selectFilter	query	Restituisce una proprietà `filter` con il filtro delle condizioni dell'azione.
selectOperations	query	Restituisce una proprietà `operations` con le operazioni dell'azione.
selectRecoveryOperations	query	Restituisce una proprietà `recovery_operations` con le operazioni di ripristino dell'azione.
selectUpdateOperations	query	Restituisce una proprietà `update_operations` con le operazioni di aggiornamento dell'azione.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `actionid`, `name`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(intero/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recuperare le azioni dei trigger

Recupera tutte le azioni dei trigger configurate insieme alle condizioni e alle operazioni dell'azione.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
```

```
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Segnala i problemi agli amministratori di Zabbix",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ]
                }
            ],
            "recovery_operations": [
                {
                    "operationid": "7",
                    "actionid": "3",
                    "operationtype": "11",
                    "evaltype": "0",
                    "opconditions": [],
```

```
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                        "message": "Trigger: {TRIGGER.NAME}\r\nStato del trigger: {TRIGGER.STATUS}\r\nGrav
                        "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": "Aggiornato: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} ha aggiornato il problema il {EVENT.UPDATE.DATE} alle
                        "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
}
```


Recuperare le azioni di discovery

Recupera tutte le azioni di discovery configurate insieme alle condizioni e alle operazioni dell'azione. Il filtro utilizza il tipo di valutazione "and", quindi la proprietà `formula` è vuota e `eval_formula` viene generata automaticamente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Filtro azione
- Operazione azione

Sorgente

CAction::get() in *ui/include/classes/api/services/CAction.php*.

action.update

Descrizione

`object action.update(object/array actions)`

Questo metodo consente di aggiornare le azioni esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dell'azione da aggiornare.

La proprietà `actionid` deve essere definita per ogni azione, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard dell'azione, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
filter	object	Oggetto filtro dell'azione per sostituire il filtro corrente.
operations	array	Operazioni dell'azione per sostituire le operazioni esistenti.

Parametro	Tipo	Descrizione
recovery_operations	array	Operazioni di ripristino dell'azione per sostituire le operazioni di ripristino esistenti. Comportamento del parametro: - *supportato* se `eventsource` dell'oggetto azione è impostato su "evento creato da un trigger", "evento interno" o "evento creato all'aggiornamento dello stato del servizio"
update_operations	array	Operazioni di aggiornamento dell'azione per sostituire le operazioni di aggiornamento esistenti. Comportamento del parametro: - *supportato* se `eventsource` dell'oggetto azione è impostato su "evento creato da un trigger" o "evento creato all'aggiornamento dello stato del servizio"

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle azioni aggiornate nella proprietà `actionids`.

Esempi

Disabilitare un'azione

Disabilita un'azione, ovvero imposta il suo stato a "1".

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "action.update",
"params": {
"actionid": "2",
"status": "1"
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"actionids": [
"2"
]
},
"id": 1
}
```

Vedi anche

- Filtro azione
- Operazione azione

Sorgente

CAction::update() in *ui/include/classes/api/services/CAction.php*.

Complex event processing

This class is designed to work with complex event processing.

Object references:

- CEP rule
- CEP rule filter
    - CEP rule filter condition

- CEP rule time window
- CEP rule operation
- CEP rule operation filter
- CEP rule operation filter condition

Available methods:

- ceprule.create - create new CEP rules
- ceprule.delete - delete CEP rules
- ceprule.get - retrieve CEP rules
- ceprule.update - update CEP rules
- ceprule.resettimewindows - reset time windows for CEP rule

CEP rule object

The following objects are directly related to the `ceprule` API.

CEP rule

The CEP rule object has the following properties.

Property	Type	Description
cep_ruleid	ID	ID of the CEP rule. Property behavior: - *read-only* - *required* for update operations
name	string	Name of the CEP rule. Property behavior: - *required* for create operations
window_type	integer	Time window type: 0 - (default) None 1 - Simple (multiple); 2 - Cause and symptom grouping; 3 - Simple (once); 4 - Event pattern match.
stop	integer	Possible values: 0 - (default) Continue with the next CEP rule according to sort order; 1 - Stop after executing this rule.
sortorder	integer	Execution order of the CEP rule.
description	string	Description of the CEP rule.
status	integer	Whether the CEP rule is enabled or disabled. Possible values: 0 - *(default)* enabled; 1 - disabled.
error	string	Error text if the CEP rule processing failed on rule level. Property behavior: - *read-only*

CEP rule filter

The CEP rule filter object defines a set of conditions that must be met for events to be processed by CEP rule. It has the following properties.

Property	Type	Description
conditions	array	Set of filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	User-defined expression to be used for evaluating conditions of filters with a custom expression. The expression must contain IDs that reference specific filter conditions by its `formulaid`. The IDs used in the expression must exactly match the ones defined in the filter conditions: no condition can remain unused or omitted. Property behavior: - *required* if `evaltype` is set to "custom expression"

CEP rule filter condition

The CEP rule filter condition object defines a specific condition that must be checked before processing events by the CEP rule.

Property	Type	Description
type	integer	Type of condition. Possible values: 3 - event name; 4 - severity; 6 - time period; 25 - tag name; 26 - tag value; 36 - host visible name; 37 - host group name. Property behavior: - *required*

Property	Type	Description
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 4 - in; 5 - is greater than or equals; 6 - is less than or equals; 7 - not in. Property behavior: - *required*.
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”event name”
severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
time_period	string	Time period for a specific operation type. Property behavior: - *required* if `type` is set to ”time period”
tag	string	Tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”tag name” or ”tag value”
tag_value	string	Tag value for a specific operation type. Property behavior: - *required* if `type` is set to ”tag value”
event_severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
host_name	string	Host name for a specific operation type. Property behavior: - *required* if `type` is set to ”host visible name”
host_group_name	string	Host group name for a specific operation type. Property behavior: - *required* if `type` is set to ”host group name”
formulaid	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required* if CEP rule filter `evaltype` is set to ”custom expression”

CEP rule time window

The CEP rule time window object defines the time window of the CEP rule. It has the following properties.

Property	Type	Description
duration	string	Duration of the CEP rule window (365 days maximum). Accepts seconds and time units with suffix. Default: `10m`
capacity	integer	Capacity of the CEP rule window. Supports user macros. Default: 0 (unlimited)
group_by_host_group	integer	Group by flag for host group. Possible values: 0 - no (default); 1 - yes.
group_by_host	integer	Group by flag for host. Possible values: 0 - no (default); 1 - yes.
group_by_tags	integer	Group by flag for tag. Possible values: 0 - no (default); 1 - yes.
tags	string	Tag names to use for grouping. Property behavior: - *required* if `group_by_tag` is set to ”1”
event_count_tag	string	Tag name for displaying the count of affected events. Property behavior: - *supported* if CEP rule `window_type` is set to ”2”; not allowed otherwise
script	string	Script to execute. Property behavior: - *supported* if CEP rule `window_type` is set to ”4”; not allowed otherwise

CEP rule operation

The CEP rule operation object defines an operation that will be performed when a CEP rule is executed. It has the following properties.

Property	Type	Description
sortorder	integer	Execution order (ascending) of the operation. Property behavior: - *required*
execute_when	integer	Type of condition for operation. Possible values: 0 - event occurred; 1 - event added; 2 - event evicted; 3 - window closed; 4 - event pattern matched; 5 - event closed. Property behavior: - *required*

Property	Type	Description
scope	integer	Scope of the operation. Possible values: 0 - current event (default); 1 - other events; 2 - all events. Possible values when `execute_when` is ”Window closed” or ”Pattern matched”: 2 - all events. Possible values when `execute_when` is ”Event ocurred”: 0 - current event.
filter	object	Filter object.
scope_filter	object	Scope filter object. scope_filter is editable only if the `scope` field contains ”Other events” or ”All events” value. Otherwise scope_filter accepts only object with default values.
type	integer	CEP operation type. Possible values: 1 - set event name; 2 - close event; 3 - discard event; 4 - set event severity; 5 - increase event severity; 6 - decrease event severity; 7 - suppress event; 8 - unsuppress event; 9 - copy first event as new; 10 - copy last event as new; 11 - add tag; 12 - set tag; 13 - set tag value; 14 - increase tag value; 15 - decrease tag value; 16 - rename tag; 17 - remove tag; 18 - close window. Property behavior: - *required* - *supported*: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17 - supported if ”Event occurred” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event added” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event evicted” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 - supported if ”Window closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event pattern matched” (see `execute_when`)
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event name”

Property	Type	Description
tag	string	Tag name for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
new_tag	string	New tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”Rename tag”
tag_value	string	Tag value for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
severity	integer	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event severity”
suppress_duration	string	Duration of the event suppression. Accepts seconds and time unit with suffix. Property behavior: - *required* if `type` is set to ”Suppress event”

CEP rule operation filter

The CEP rule operation filter object defines a set of conditions that must be met to perform the configured CEP operations. It has the following properties.

Property	Type	Description
conditions	array	Set of operation filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*

Property	Type	Description
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation filter condition

The CEP rule operation filter condition object defines a specific condition that must be checked before running the CEP rule operations.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if "Event occurred" (see `execute_when`)
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to "Tag name" 0, 1, 2, 3, 5, 6 - supported values if `type` set to "Tag value" 10, 11 - supported values if `type` set to "Problem opened", "First", "Last", "Symptom", "Cloned", "Suppressed".
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to "Tag name" or "Tag value"
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to "Tag value"

CEP rule operation scope filter

The CEP rule operation scope filter object defines a set of conditions that must be met to apply the configured CEP operations to events. It has the following properties.

Property	Type	Description
conditions	array	Set of operation scope filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation scope filter condition

The CEP rule operation scope filter condition object defines a specific condition that must be checked before applying the CEP rule operations to events.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if ”Event occurred” (see `execute_when`)

Property	Type	Description
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to ”Tag name” 0, 1, 2, 3, 5, 6 - supported values if `type` set to ”Tag value” 10, 11 - supported values if `type` set to ”Problem opened”, ”First”, ”Last”, ”Symptom”, ”Cloned”, ”Suppressed”.
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to ”Tag name” or ”Tag value”
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to ”Tag value”

ceprule.create

Description

`object ceprule.create(object/array CEP rules)`

This method allows you to create new CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rules to create.

Additionally to the standard CEP properties, the method accepts the following parameters.

Parameter	Type	Description
operations	array	CEP rule operations to create for the CEP rule. Parameter behavior: - *required*
filter	object	CEP rule filter object for the CEP rule. Parameter behavior: - *required*

Return values

`(object)` Returns an object containing the IDs of the created CEP rules under the `cep_ruleids` property. The order of the returned IDs matches the order of the passed CEP rules.

Examples

Create a new CEP rule

Create a CEP rule using evaluation method `AND/OR` with one condition and one window. By default, the rule will be enabled.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.create",
    "params": {
      "name": "Group events by cause",
      "stop": "1",
      "sortorder": "0",
      "description": "",
      "status": "0",
      "window_type": "2",
      "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": ""
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "",
        "script": ""
      },
      "operations": []
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "cep_ruleids": [
      "1"
    ]
  },
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::create() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.delete

Description

`object ceprule.delete(array cep_ruleids)`

This method allows you to delete CEP rules.

> **Note:**
> This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(array)` IDs of the CEP rules to delete.

Return values

`(object)` Returns an object containing the IDs of the deleted CEP rules under the `cep_ruleids` property.

Example

Delete multiple CEP rules

Delete two CEP rules.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Source

CCepRule::delete() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.get

Description

`integer/array ceprule.get(object parameters)`

The method allows you to retrieve CEP rules according to the given parameters.

Note:
This method is available to users of any type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` Parameters defining the desired output.

The method supports the following parameters.

Parameter	Type	Description
cep_ruleids	ID/array	Return only CEP rules with the given IDs.
selectFilter	query	Return a `filter` property with the CEP rule conditions.
selectOperations	query	Return an `operations` property with the CEP rule operations.
selectWindow	query	Return a `[window]`(/manual/api/reference/ceprule/object#cep-rule-time-window) property with the CEP rule time window.
sortfield	string/array	Sort the result by the given properties. Possible values: `cep_ruleid`, `name`, `stop`, `sortorder`, `status`.
countOutput	boolean	These parameters are described in the reference commentary.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Return values

`(integer/array)` Returns either:

- an array of objects;
- the count of retrieved objects, if the `countOutput` parameter has been used.

Examples

Retrieve CEP rules

Retrieve enabled CEP rules, including conditions, windows, and operations.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.get",
    "params": {
        "output": "extend",
        "filter": {"status": 0},
        "selectFilter": "extend",
        "selectWindow": "extend",
        "selectOperations": "extend"
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "cep_ruleid": "2",
```

```
      "name": "Group events by cause",
      "stop": "0",
      "sortorder": "1",
      "description": "",
      "status": "0",
      "window_type": "2",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": "A"
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "symptoms",
        "script": ""
      },
      "operations": []
    },
    {
      "cep_ruleid": "10",
      "name": "Suppress consecutive events",
      "stop": "1",
      "sortorder": "2",
      "description": "",
      "status": "0",
      "window_type": "3",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
```

```
          "host_group_name": "",
          "time_period": "",
          "formulaid": "A"
        }
      ]
    },
    "window": {
      "duration": "3m",
      "capacity": "0",
      "group_by_host_group": "0",
      "group_by_host": "1",
      "group_by_tags": "1",
      "tags": [
        "cause"
      ],
      "event_count_tag": "",
      "script": ""
    },
    "operations": [
      {
        "sortorder": "1",
        "execute_when": "1",
        "type": "7",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "6m",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
        "filter": {
          "evaltype": "0",
          "eval_formula": "A",
          "formula": "",
          "conditions": [
            {
              "type": "33",
              "operator": "10",
              "tag": "",
              "tag_value": "",
              "formulaid": "A"
            }
          ]
        },
        "scope_filter": {
          "evaltype": "0",
          "eval_formula": "",
          "formula": "",
          "conditions": []
        }
      },
      {
        "sortorder": "2",
        "execute_when": "2",
        "type": "8",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "0",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
```

```
          "filter": {
            "evaltype": "0",
            "eval_formula": "A",
            "formula": "",
            "conditions": [
              {
                "type": "35",
                "operator": "10",
                "tag": "",
                "tag_value": "",
                "formulaid": "A"
              }
            ]
          },
          "scope_filter": {
            "evaltype": "0",
            "eval_formula": "",
            "formula": "",
            "conditions": []
          }
        }
      ]
    }
  ],
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::get() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.resettimewindows

Description

`object ceprule.resettimewindows(object/array cep_ruleid)`

This method allows you to reset time windows for an existing CEP rule.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` CEP rule object must have a `cep_ruleid` property defined.

Return values

`(object)` Returns an object containing the ID of the CEP rule whose time windows have been reset under the `cep_ruleid` property.

Examples

Reset time windows for CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.resettimewindows",
    "params": {
        "cep_ruleid": "1"
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "cep_ruleid": "1"
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::resettimewindows() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.update

Description

`object ceprule.update(object/array CEP rules)`

This method allows you to update existing CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rule properties to be updated.

The `cep_ruleid` property must be defined for each CEP rule, all other properties are optional. Only the passed properties will be updated, all others will remain unchanged.

Additionally to the standard CEP rule properties, the method accepts the following parameters.

Parameter	Type	Description
filter	object	CEP rule filter object to replace the current filter.
window	array	CEP rule window to be updated. Property behavior: - *required* if `window_type` is changed to ”Cause and symptom grouping” and ”group_by_host_group”, ”group_by_host”, ”group_by_tags” are all set to NO; - *required* if `window_type` is changed to ”Event pattern match”; - *supported* if `window_type` is ”Simple (multiple)” or ”Simple (once)”
operations	array	CEP rule operations to replace existing operations. Property behavior: - *required* if `window_type` is changed to ”None” (0), ”Simple (multiple)” (1), ”Simple (once)” (3) or ”Event pattern match” (4).

Return values

`(object)` Returns an object containing the IDs of the updated CEP rules under the `cep_ruleids` property.

Examples

Disable CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "1",
        "status": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

Replace condition

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "3",
        "filter": {
        "evaltype": "0",
        "formula": "",
            "conditions": [
                {
                    "type": "3",
                    "operator": "2",
                    "event_name": "Inaccessible port",
                    "severity": "0"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::update() in *ui/include/classes/api/services/CCepRule.php*.

Configurazione

Questa classe è progettata per esportare e importare i dati di configurazione di Zabbix.

Metodi disponibili:

- configuration.export - esporta i dati di configurazione
- configuration.import - importa i dati di configurazione
- configuration.importcompare - confronta il file di importazione con gli elementi di sistema correnti

configuration.export

Descrizione

`string configuration.export(object parameters)`

Questo metodo consente di esportare i dati di configurazione come stringa serializzata.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Parametri

`(object)` Parametri che definiscono gli oggetti da esportare e il formato da utilizzare.

Parameter	Type	Description
format	string	Formato in cui i dati devono essere esportati. Valori possibili: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - array PHP non elaborato. Comportamento del parametro: - *required*
prettyprint	boolean	Rende l'output più leggibile per l'utente aggiungendo l'indentazione. Valori possibili: `true` - aggiunge l'indentazione; `false` - *(default)* non aggiunge l'indentazione.
options	object	Oggetti da esportare. L'oggetto `options` ha i seguenti parametri: `host_groups` - `(array)` ID dei gruppi host da esportare; `hosts` - `(array)` ID degli host da esportare; `images` - `(array)` ID delle immagini da esportare; `maps` - `(array)` ID delle mappe da esportare; `mediaTypes` - `(array)` ID dei tipi di media da esportare; `template_groups` - `(array)` ID dei gruppi template da esportare; `templates` - `(array)` ID dei template da esportare; `dashboards` - `(array)` ID delle dashboard da esportare; `global_regexes` - `(array)` ID delle espressioni regolari globali da esportare. Gli utenti di tipo *Admin* e *User* possono esportare solo gli oggetti per i quali dispongono dell'autorizzazione di *sola lettura* o *lettura e scrittura* permission, oltre alle immagini, ma non ai tipi di media. Comportamento del parametro: - *required*.

Valori restituiti

`(string)` Restituisce una stringa serializzata contenente i dati di configurazione richiesti.

Esempi

Esportazione di un template

Esporta la configurazione del template "10571" come stringa XML.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```

Sorgente

CConfiguration::export() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

Descrizione

`boolean configuration.import(object parameters)`

Questo metodo consente di importare dati di configurazione da una stringa serializzata.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti i dati da importare e le regole su come i dati devono essere gestiti.

Parameter	Type	Description
format	string	Formato della stringa serializzata. Valori possibili: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *required*
source	string	Stringa serializzata contenente i dati di configurazione. Parameter behavior: - *required*

Parameter	Type	Description
rules	object	Regole su come devono essere importati i nuovi oggetti e quelli esistenti. Gli utenti di tipo *Admin* possono importare solo gli oggetti per i quali dispongono dell'autorizzazione *read-write* permission, così come le mappe. Ad esempio, un host e le sue entità (items, triggers, graph, ecc.) possono essere importati solo se il gruppo utente dell'utente ha l'autorizzazione al gruppo host a cui apparterrà l'host importato. Le immagini e i tipi di media non possono essere importati dagli utenti di tipo *Admin*. Il parametro `rules` è descritto in dettaglio nella tabella seguente. Parameter behavior: - *required*
returnMissingObjects	boolean	Regole per stabilire se l'oggetto di risposta debba contenere informazioni sugli oggetti mancanti; predefinito: `false`.

Attention:
Se non vengono fornite regole, la configurazione non verrà aggiornata.

L'oggetto `rules` supporta i seguenti parametri.

Parameter	Type	Description
dashboards	object	Regole su come importare i dashboard. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi dashboard; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true` e un particolare dashboard importabile è presente nel sistema, verrà aggiornato completamente. Le pagine e i widget verranno riscritti integralmente con quelli nuovi; predefinito: `false`.
discoveryRules	object	Regole su come importare le regole LLD. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove regole LLD; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le regole LLD esistenti verranno aggiornate; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, le regole LLD non presenti nei dati importati verranno eliminate dal database; predefinito: `false`.
global_regexes	object	Regole su come importare le espressioni regolari globali. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove espressioni regolari globali; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le espressioni regolari globali esistenti verranno aggiornate, con le relative sottoespressioni completamente sovrascritte; predefinito: `false`.

Parameter	Type	Description
graphs	object	Regole su come importare i graph. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi graph; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i graph esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i graph non presenti nei dati importati verranno eliminati dal database; predefinito: `false`.
host_groups	object	Regole su come importare i gruppi host. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi gruppi host; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i gruppi host esistenti verranno aggiornati; predefinito: `false`.
hosts	object	Regole su come importare gli host. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi host; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, gli host esistenti verranno aggiornati; predefinito: `false`.
httptests	object	Regole su come importare gli scenari web. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi scenari web; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, gli scenari web esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, gli scenari web non presenti nei dati importati verranno eliminati dal database; predefinito: `false`.
images	object	Regole su come importare le immagini. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove immagini; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le immagini esistenti verranno aggiornate; predefinito: `false`.
items	object	Regole su come importare gli item. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi item; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, gli item esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, gli item non presenti nei dati importati verranno eliminati dal database; predefinito: `false`.
maps	object	Regole su come importare le mappe. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove mappe; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le mappe esistenti verranno aggiornate; predefinito: `false`.

Parameter	Type	Description
mediaTypes	object	Regole su come importare i tipi di media. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi tipi di media; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i tipi di media esistenti verranno aggiornati; predefinito: `false`.
templateDashboards	object	Regole su come importare i dashboard dei template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi dashboard dei template; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i dashboard dei template esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i dashboard dei template non presenti nei dati importati verranno eliminati dal database; predefinito: `false`.
template_groups	object	Regole su come importare i gruppi template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi gruppi template; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i gruppi template esistenti verranno aggiornati; predefinito: `false`.
templateLinkage	object	Regole su come importare i collegamenti dei template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i template che non sono collegati all'host o al template in fase di importazione, ma sono presenti nei dati importati, verranno collegati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i template collegati all'host o al template in fase di importazione, ma non presenti nei dati importati, verranno scollegati senza rimuovere le entità (items, triggers, ecc.) ereditate dai template scollegati; predefinito: `false`.
templates	object	Regole su come importare i template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi template; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i template esistenti verranno aggiornati; predefinito: `false`.
triggers	object	Regole su come importare i trigger. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi trigger; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i trigger esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i trigger non presenti nei dati importati verranno eliminati dal database; predefinito: `false`.
valueMaps	object	Regole su come importare i value map di host o template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi value map; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i value map esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i value map non presenti nei dati importati verranno eliminati dal database; predefinito: `false`.

Valori restituiti

(boolean) Restituisce true se l'importazione ha avuto successo.

Esempi

Importazione di un template

Importa la configurazione del template contenuta nella stringa XML. Se nella stringa XML mancano item o trigger, questi verranno eliminati dal database, mentre tutto il resto rimarrà invariato.

Richiesta:


```
{
    "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
}
```


Risposta:


```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```


Sorgente

CConfiguration::import() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

Descrizione

`array configuration.importcompare(object parameters)`

Questo metodo consente di confrontare il file di importazione con gli elementi correnti del sistema e mostra cosa verrà modificato se questo file di importazione verrà importato.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Parametri

`(object)` Parametri contenenti i possibili dati da importare e le regole su come i dati devono essere gestiti.

Parameter	Type	Description
format	string	Formato della stringa serializzata. Valori possibili: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *required*
source	string	Stringa serializzata contenente i dati di configurazione. Parameter behavior: - *required*
rules	object	Regole su come confrontare gli oggetti nuovi ed esistenti. Gli utenti di tipo *Admin* e *User* possono confrontare i nuovi oggetti solo con quelli esistenti per i quali dispongono di autorizzazione *read-only* o *read-write* permission. Il parametro `rules` è descritto in dettaglio nella tabella seguente. Parameter behavior: - *required*
returnMissingObjects	boolean	Regole per stabilire se l'oggetto di risposta debba contenere informazioni sugli oggetti mancanti; default: `false`.

Attention:
Se non vengono fornite regole, non ci sarà nulla da aggiornare e il risultato sarà vuoto.

Note:
Il confronto verrà eseguito solo per i gruppi host e i template. I trigger e i grafici verranno confrontati solo per i template importati; tutti gli altri saranno considerati "nuovi".

L'oggetto `rules` supporta i seguenti parametri.

Parameter	Type	Description
dashboards	object	Regole su come importare i dashboard. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi dashboard verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true` e un particolare dashboard importabile è presente nel sistema, verrà aggiornato completamente. Le pagine e i widget verranno completamente sovrascritti con quelli nuovi; default: `false`.
discoveryRules	object	Regole su come importare le regole LLD. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, le nuove regole LLD verranno create; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le regole LLD esistenti verranno aggiornate; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, le regole LLD non presenti nei dati importati verranno eliminate dal database; default: `false`.

Parameter	Type	Description
global_regexes	object	Regole su come importare le espressioni regolari globali. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, le nuove espressioni regolari globali verranno create; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le espressioni regolari globali esistenti verranno aggiornate, con le relative sottoespressioni completamente sovrascritte; default: `false`.
graphs	object	Regole su come importare i grafici. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi grafici verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i grafici esistenti verranno aggiornati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i grafici non presenti nei dati importati verranno eliminati dal database; default: `false`.
host_groups	object	Regole su come importare i gruppi host. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi gruppi host verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i gruppi host esistenti verranno aggiornati; default: `false`.
httptests	object	Regole su come importare gli scenari web. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi scenari web verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, gli scenari web esistenti verranno aggiornati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, gli scenari web non presenti nei dati importati verranno eliminati dal database; default: `false`.
images	object	Regole su come importare le immagini. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, le nuove immagini verranno create; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le immagini esistenti verranno aggiornate; default: `false`. Questo parametro non farà alcuna differenza nell'output. È consentito solo per coerenza con `configuration.import`.
items	object	Regole su come importare gli item. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi item verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, gli item esistenti verranno aggiornati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, gli item non presenti nei dati importati verranno eliminati dal database; default: `false`.

Parameter	Type	Description
maps	object	Regole su come importare le mappe. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, le nuove mappe verranno create; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le mappe esistenti verranno aggiornate; default: `false`. Questo parametro non farà alcuna differenza nell'output. È consentito solo per coerenza con `configuration.import`.
mediaTypes	object	Regole su come importare i tipi di media. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi tipi di media verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i tipi di media esistenti verranno aggiornati; default: `false`. Questo parametro non farà alcuna differenza nell'output. È consentito solo per coerenza con `configuration.import`.
templateDashboards	object	Regole su come importare i dashboard del template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi dashboard del template verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i dashboard del template esistenti verranno aggiornati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i dashboard del template non presenti nei dati importati verranno eliminati dal database; default: `false`.
template_groups	object	Regole su come importare i gruppi template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi gruppi template verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i gruppi template esistenti verranno aggiornati; default: `false`.
templateLinkage	object	Regole su come importare i collegamenti dei template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i template che non sono collegati all'host o al template in fase di importazione, ma che sono presenti nei dati importati, verranno collegati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i template collegati all'host o al template in fase di importazione, ma non presenti nei dati importati, verranno scollegati senza rimuovere le entità (item, trigger, ecc.) ereditate dai template scollegati; default: `false`.
templates	object	Regole su come importare i template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi template verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i template esistenti verranno aggiornati; default: `false`.

Parameter	Type	Description
triggers	object	Regole su come importare i trigger. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi trigger verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i trigger esistenti verranno aggiornati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i trigger non presenti nei dati importati verranno eliminati dal database; default: `false`.
valueMaps	object	Regole su come importare i value map di host o template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, i nuovi value map verranno creati; default: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i value map esistenti verranno aggiornati; default: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i value map non presenti nei dati importati verranno eliminati dal database; default: `false`.

Valori restituiti

`(array)` Restituisce un array con le modifiche nella configurazione che verranno apportate.

Esempi

Confronto dell'importazione di un template

Confronta il template contenuto nella stringa XML con gli elementi correnti del sistema e mostra quali modifiche verranno apportate se questo template verrà importato.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "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
}
```

Sorgente

CConfiguration::importcompare() in *ui/include/classes/api/services/CConfiguration.php*.

Connettore

Questa classe è progettata per lavorare con i connettori.

Riferimenti agli oggetti:

- Connettore
- Filtro tag

Metodi disponibili:

- connector.create - crea nuovi connettori
- connector.delete - elimina connettori
- connector.get - recupera connettori
- connector.update - aggiorna connettori

Oggetto connettore

I seguenti oggetti sono direttamente correlati all'API `connector`.

Connettore

L'oggetto connettore ha le seguenti proprietà.

Property	Type	Description
connectorid	ID	ID del connettore. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del connettore. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
url	string	URL dell'endpoint, ovvero l'URL del destinatario. Sono supportate le macro utente. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
protocol	integer	Protocollo di comunicazione. Valori possibili: 0 - *(predefinito)* Zabbix Streaming Protocol v1.0.
data_type	integer	Tipo di dati. Valori possibili: 0 - *(predefinito)* Valori degli item; 1 - Eventi.

Property	Type	Description
item_value_type	integer	Somma dei tipi di valore degli item da inviare. Valori possibili: 1 - Numerico (virgola mobile); 2 - Carattere; 4 - Log; 8 - Numerico (senza segno); 16 - Testo; 32 - Binario; 64 - JSON. Predefinito: 31 - Tutti i tipi di item (eccetto binario e JSON). Comportamento della proprietà: - *supportata* se `data_type` è impostato su "Valori degli item".
max_records	integer	Numero massimo di eventi o item che possono essere inviati in un singolo messaggio. Valori possibili: 0-2147483647 (valore massimo di un intero con segno a 32 bit). Predefinito: 0 - Illimitato.
max_senders	integer	Numero di processi sender da eseguire per questo connettore. Valori possibili: 1-100. Predefinito: 1.
max_attempts	integer	Numero di tentativi. Valori possibili: 1-5. Predefinito: 1.
attempt_interval	string	Intervallo tra i tentativi successivi. Accetta i secondi. Valori possibili: 0s-10s. Predefinito: 5s. Comportamento della proprietà: - *supportata* se `max_attempts` è maggiore di 1.
timeout	string	Timeout. Sono supportati i suffissi temporali (ad esempio, `30s`, `1m`). Sono supportate le macro utente. Valori possibili: 1s-60s. Predefinito: 5s.
http_proxy	string	Stringa di connessione al proxy HTTP(S) nel formato *[protocol]://[username[:password]@]proxy.example.com[:port]*. Sono supportate le macro utente.
authtype	integer	Metodo di autenticazione HTTP. Valori possibili: 0 - *(predefinito)* Nessuno; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.

Property	Type	Description
username	string	Nome utente. Sono supportate le macro utente. Comportamento della proprietà: - *supportata* se `authtype` è impostato su ”Basic”, ”NTLM”, ”Kerberos” o ”Digest”
password	string	Password. Sono supportate le macro utente. Comportamento della proprietà: - *supportata* se `authtype` è impostato su ”Basic”, ”NTLM”, ”Kerberos” o ”Digest”
token	string	Token Bearer. Sono supportate le macro utente. Comportamento della proprietà: - *obbligatorio* se `authtype` è impostato su ”Bearer”
verify_peer	integer	Indica se verificare che il certificato dell’host sia autentico. Valori possibili: 0 - Non verificare; 1 - *(predefinito)* Verificare.
verify_host	integer	Indica se verificare che il nome dell’host della connessione corrisponda a quello nel certificato dell’host. Valori possibili: 0 - Non verificare; 1 - *(predefinito)* Verificare.
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Sono supportate le macro utente.
ssl_key_file	string	Percorso del file della chiave SSL privata. Sono supportate le macro utente.
ssl_key_password	string	Password del file della chiave SSL. Sono supportate le macro utente.
description	text	Descrizione del connettore.
status	integer	Indica se il connettore è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.
tags_evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.

Filtro tag

Il filtro tag consente di esportare solo i valori degli item o gli eventi corrispondenti. Se non impostato, verrà esportato tutto. L’oggetto filtro tag ha le seguenti proprietà.

Property	Type	Description
tag	string	Nome del tag. Property behavior: - *required*

Property	Type	Description
operator	integer	Operatore della condizione operator. Valori possibili: 0 - *(default)* Uguale; 1 - Non uguale; 2 - Contiene; 3 - Non contiene; 12 - Esiste; 13 - Non esiste.
value	string	Valore del tag.

connector.create

Descrizione

`object connector.create(object/array connectors)`

Questo metodo consente di creare nuovi oggetti connector.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Oggetti connector da creare.

Oltre alle proprietà standard del connector, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
tags	array	Filtro tag del connector.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei connettori creati nella proprietà `connectorids`. L'ordine degli ID restituiti corrisponde all'ordine dei connettori passati.

Esempi

Creazione di un connettore

Crea un connettore per esportare gli eventi trigger con un filtro tag. L'autenticazione HTTP verrà eseguita tramite token Bearer.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

Sorgente

CConnector::create() in *ui/include/classes/api/services/CConnector.php*.

connector.delete

Descrizione

`object connector.delete(array connectorids)`

Questo metodo consente di eliminare le voci del connector.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei connettori da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei connettori eliminati nella proprietà `connectorids`.

Esempi

Eliminare più connettori

Elimina due voci di connettore.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
```

```
    "id": 1
}
```

Sorgente

CConnector::delete() in *ui/include/classes/api/services/CConnector.php*.

connector.get

Descrizione

`integer/array connector.get(object parameters)`

Il metodo consente di recuperare gli oggetti connector in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
connectorids	ID/array	Restituisce solo i connettori con gli ID specificati.
selectTags	query	Restituisce una proprietà `tags` con il filtro tag del connettore. Supporta `count`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `connectorid`, `name`, `data_type`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero di tutti i connettori

Recupera tutti i dati su tutti i connettori e le loro proprietà.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
```

```
        "preservekeys": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Esportazione dei valori degli item",
            "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": "Esportazione degli eventi",
            "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
}
```

Fonte

CConnector:get() in *ui/include/classes/api/services/CConnector.php*.

connector.update

Descrizione

`object connector.update(object/array connectors)`

Questo metodo consente di aggiornare i connector esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà del connector da aggiornare.

La proprietà `connectorid` deve essere definita per ciascun connector, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del connector, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
tags	array	Filtro tag del connector per sostituire il filtro tag corrente.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei connettori aggiornati nella proprietà `connectorids`.

Esempi

Modifica del tipo di autenticazione HTTP

Modifica il tipo di autenticazione HTTP in Bearer per il connettore con ID "3".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

Aggiornamento del filtro tag

Modifica il filtro tag per il connettore con ID ”5”.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
    "id": 1
}
```

Sorgente

CConnector::update() in *ui/include/classes/api/services/CConnector.php*.

Controllo di individuazione

Questa classe è progettata per funzionare con i controlli di individuazione.

Riferimenti agli oggetti:

- Controllo di individuazione

Metodi disponibili:

- dcheck.get - recupera i controlli di individuazione

Oggetto controllo della discovery

I seguenti oggetti sono direttamente correlati all'API `dcheck`.

Controllo di individuazione

L'oggetto controllo di individuazione definisce un controllo specifico eseguito da una regola di individuazione di rete. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
dcheckid	ID	ID del controllo di individuazione.
druleid	ID	ID della regola di individuazione a cui appartiene il controllo.
key_	string	Chiave item (se `type` è impostato su "Zabbix agent") oppure OID SNMP (se `type` è impostato su "SNMPv1 agent", "SNMPv2 agent" o "SNMPv3 agent"). Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Zabbix agent", "SNMPv1 agent", "SNMPv2 agent" o "SNMPv3 agent"
ports	string	Uno o più intervalli di porte da controllare, separati da virgole. Predefinito: 0. Comportamento della proprietà: - *supportata* se `type` è impostato su "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) o "Telnet" (15)
snmp_community	string	Community SNMP. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SNMPv1 agent" o "SNMPv2 agent"
snmpv3_authpassphrase	string	Passphrase di autenticazione. Comportamento della proprietà: - *supportata* se `type` è impostato su "SNMPv3 agent" e `snmpv3_securitylevel` è impostato su "authNoPriv" o "authPriv"
snmpv3_authprotocol	integer	Protocollo di autenticazione. Valori possibili: 0 - *(predefinito)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Comportamento della proprietà: - *supportata* se `type` è impostato su "SNMPv3 agent" e `snmpv3_securitylevel` è impostato su "authNoPriv" o "authPriv"
snmpv3_contextname	string	Nome del contesto SNMPv3. Comportamento della proprietà: - *supportata* se `type` è impostato su "SNMPv3 agent"
snmpv3_privpassphrase	string	Passphrase di privacy. Comportamento della proprietà: - *supportata* se `type` è impostato su "SNMPv3 agent" e `snmpv3_securitylevel` è impostato su "authPriv"

Proprietà	Type	Descrizione
snmpv3_privprotocol	integer	Protocollo di privacy. Valori possibili: 0 - *(predefinito)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SNMPv3 agent” e `snmpv3_securitylevel` è impostato su ”authPriv”
snmpv3_securitylevel	string	Livello di sicurezza. Valori possibili: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SNMPv3 agent”
snmpv3_securityname	string	Nome di sicurezza. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SNMPv3 agent”
type	integer	Tipo di controllo. Valori possibili: 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. Comportamento della proprietà: - *obbligatoria*
uniq	integer	Indica se utilizzare questo controllo come criterio di unicità del dispositivo. Per una regola di individuazione può essere configurato un solo controllo univoco. Valori possibili: 0 - *(predefinito)* non utilizzare questo controllo come criterio di unicità; 1 - utilizzare questo controllo come criterio di unicità. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” o ”SNMPv3 agent”

Proprietà	Type	Descrizione
host_source	integer	Origine del nome host. Valori possibili: 1 - *(predefinito)* DNS; 2 - IP; 3 - valore di individuazione di questo controllo.
name_source	integer	Origine del nome visibile. Valori possibili: 0 - *(predefinito)* non specificato; 1 - DNS; 2 - IP; 3 - valore di individuazione di questo controllo.
allow_redirect	integer	Consente la situazione in cui la destinazione sottoposta a ICMP ping risponde da un indirizzo IP diverso. Valori possibili: 0 - *(predefinito)* tratta le risposte reindirizzate come se il host di destinazione fosse non disponibile (errore); 1 - tratta le risposte reindirizzate come se il host di destinazione fosse disponibile (successo). Comportamento della proprietà: - *supportata* se `type` è impostato su "ICMP ping"

dcheck.get

Descrizione

`integer/array dcheck.get(object parameters)`

Il metodo consente di recuperare i controlli di discovery in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
dcheckids	ID/array	Restituisce solo i controlli di discovery con gli ID specificati.
druleids	ID/array	Restituisce solo i controlli di discovery che appartengono alle regole di discovery specificate.
dserviceids	ID/array	Restituisce solo i controlli di discovery che hanno rilevato i servizi individuati specificati.
selectDRules	query	Restituisce le regole di discovery correlate ai controlli di discovery.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `dcheckid`, `druleid`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

Parametro	Tipo	Descrizione
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare i controlli di individuazione per una regola di individuazione

Recupera tutti i controlli di individuazione utilizzati dalla regola di individuazione "6".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Fonte

CDCheck::get() in *ui/include/classes/api/services/CDCheck.php*.

Correlazione

Questa classe è progettata per funzionare con le correlazioni.

Riferimenti agli oggetti:

- Correlazione
- Operazione di correlazione
- Filtro di correlazione
  - Condizione del filtro di correlazione

Metodi disponibili:

- correlation.create - crea nuove correlazioni
- correlation.delete - elimina correlazioni
- correlation.get - recupera correlazioni
- correlation.update - aggiorna correlazioni

Oggetto Correlazione

I seguenti oggetti sono direttamente correlati all'API `correlation`.

Correlazione

L'oggetto correlazione ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
correlationid	ID	ID della correlazione. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome della correlazione. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
description	string	Descrizione della correlazione.
status	integer	Indica se la correlazione è abilitata o disabilitata. Valori possibili: 0 - *(predefinito)* abilitata; 1 - disabilitata.

Operazione di correlazione

L'oggetto operazione di correlazione definisce un'operazione che verrà eseguita quando una correlazione viene eseguita. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
type	integer	Tipo di operazione. Valori possibili: 0 - chiudi eventi precedenti; 1 - chiudi nuovo evento. Comportamento della proprietà: - *obbligatorio*

Filtro di correlazione

L'oggetto filtro di correlazione definisce un insieme di condizioni che devono essere soddisfatte per eseguire le operazioni di correlazione configurate. Ha le seguenti proprietà.

Property	Type	Description
conditions	array	Insieme di condizioni del filtro da usare per filtrare i risultati. Le condizioni saranno ordinate nell'ordine in cui compaiono nella formula. Property behavior: - *required*
evaltype	integer	Metodo di valutazione delle condizioni del filtro. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Property behavior: - *required*
eval_formula	string	Espressione generata che verrà usata per valutare le condizioni del filtro. L'espressione contiene ID che fanno riferimento a specifiche condizioni del filtro tramite il relativo `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con espressione personalizzata. Property behavior: - *read-only*
formula	string	Espressione definita dall'utente da usare per valutare le condizioni dei filtri con espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a specifiche condizioni del filtro tramite il relativo `formulaid`. Gli ID usati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni del filtro: nessuna condizione può rimanere inutilizzata o omessa. Property behavior: - *required* se `evaltype` è impostato su "custom expression"

Condizione del filtro di correlazione

L'oggetto condizione del filtro di correlazione definisce una condizione specifica che deve essere verificata prima di eseguire le operazioni di correlazione.

Property	Type	Description
type	integer	Tipo di condizione. Valori possibili: 0 - tag del vecchio evento; 1 - tag del nuovo evento; 2 - gruppo host del nuovo evento; 3 - coppia di tag evento; 4 - valore del tag del vecchio evento; 5 - valore del tag del nuovo evento. Property behavior: - *obbligatorio*
tag	string	Tag evento (vecchio o nuovo). Property behavior: - *obbligatorio* se `type` è impostato su "tag del vecchio evento", "tag del nuovo evento", "valore del tag del vecchio evento" o "valore del tag del nuovo evento"

Property	Type	Description
groupid	ID	ID del gruppo host. Property behavior: - *obbligatorio* se `type` è impostato su ”gruppo host del nuovo evento”
oldtag	string	Tag del vecchio evento. Property behavior: - *obbligatorio* se `type` è impostato su ”coppia di tag evento”
newtag	string	Tag del vecchio evento. Property behavior: - *obbligatorio* se `type` è impostato su ”coppia di tag evento”
value	string	Valore del tag evento (vecchio o nuovo). Property behavior: - *obbligatorio* se `type` è impostato su ”valore del tag del vecchio evento” o ”valore del tag del nuovo evento”
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un’espressione personalizzata. Può contenere solo lettere maiuscole. L’ID deve essere definito dall’utente durante la modifica delle condizioni del filtro, ma verrà generato nuovamente quando verranno richieste in seguito.
operator	integer	Operatore della condizione. Property behavior: - *obbligatorio* se `type` è impostato su ”gruppo host del nuovo evento”, ”valore del tag del vecchio evento” o ”valore del tag del nuovo evento”

Note:
Per comprendere meglio come utilizzare i filtri con vari tipi di espressioni, vedere gli esempi nelle pagine dei metodi correlation.get e correlation.create.

Per ciascun tipo di condizione sono supportati i seguenti operatori e valori.

Condition	Condition name	Supported operators	Expected value
2	Gruppo host	=, <>	ID del gruppo host.
4	Valore del tag del vecchio evento	=, <>, like, not like	string
5	Valore del tag del nuovo evento	=, <>, like, not like	string

correlation.create

Descrizione

`object correlation.create(object/array correlations)`

Questo metodo consente di creare nuove correlazioni.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Correlazioni da creare.

Oltre alle proprietà standard della correlazione, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
operations	array	Operazioni di correlazione da creare per la correlazione. Comportamento del parametro: - *obbligatorio*
filter	object	Oggetto filtro di correlazione per la correlazione. Comportamento del parametro: - *obbligatorio*

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle correlazioni create nella proprietà `correlationids`. L'ordine degli ID restituiti corrisponde all'ordine delle correlazioni passate.

Esempi

Creare una nuova correlazione dei tag evento

Creare una correlazione utilizzando il metodo di valutazione `AND/OR` con una condizione e un'operazione. Per impostazione predefinita, la correlazione sarà abilitata.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Utilizzo di un filtro di espressione personalizzato

Creare una correlazione che utilizzerà una condizione di filtro personalizzata. Gli ID formula "A" o "B" sono stati scelti arbitrariamente. Il tipo di condizione sarà "Host group" con operatore "<>".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

Vedi anche

- Filtro di correlazione
- Operazione di correlazione

Sorgente

CCorrelation::create() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

Descrizione

`object correlation.delete(array correlationids)`

Questo metodo consente di eliminare le correlazioni.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle correlazioni da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle correlazioni eliminate nella proprietà `correlationids`.

Esempio

Cancella multiple correlazioni

Cancella due correlazioni.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Sorgente

CCorrelation::delete() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

Descrizione

`integer/array correlation.get(object parameters)`

Il metodo consente di recuperare le correlazioni in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
correlationids	ID/array	Restituisce solo le correlazioni con gli ID specificati.
selectFilter	query	Restituisce una proprietà `filter` con le condizioni della correlazione.
selectOperations	query	Restituisce una proprietà `operations` con le operazioni della correlazione.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `correlationid`, `name`, `status`.

Parametro	Type	Descrizione
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare le correlazioni

Recupera tutte le correlazioni configurate insieme alle condizioni e alle operazioni di correlazione. Il filtro utilizza il tipo di valutazione "and/or", quindi la proprietà `formula` è vuota e `eval_formula` viene generata automaticamente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Filtro di correlazione
- Operazione di correlazione

Sorgente

CCorrelation::get() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

Descrizione

`object correlation.update(object/array correlations)`

Questo metodo consente di aggiornare correlazioni esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà della correlazione da aggiornare.

La proprietà `correlationid` deve essere definita per ogni correlazione, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

In aggiunta alle proprietà standard della correlazione, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
filter	object	Oggetto filtro di correlazione che sostituisce il filtro corrente.
operations	array	Operazioni di correlazione che sostituiscono le operazioni esistenti.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle correlazioni aggiornate nella proprietà `correlationids`.

Esempi

Disabilitare la correlazione

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
```

```
            "1"
        ]
    },
    "id": 1
}
```

Sostituire le condizioni, ma mantenere il metodo di valutazione

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Vedi anche

- Filtro di correlazione
- Operazione di correlazione

Sorgente

CCorrelation::update() in *ui/include/classes/api/services/CCorrelation.php*.

Dashboard

Questa classe è progettata per funzionare con le dashboard.

Riferimenti agli oggetti:

- Dashboard
- Pagina della dashboard
  - Widget della dashboard
    - Campo del widget della dashboard
- Gruppo utenti della dashboard
- Utente della dashboard

Metodi disponibili:

- dashboard.create - crea nuove dashboard
- dashboard.delete - elimina dashboard
- dashboard.get - recupera dashboard
- dashboard.update - aggiorna dashboard

Oggetto dashboard

I seguenti oggetti sono direttamente correlati all'API `dashboard`.

Dashboard

L'oggetto dashboard ha le seguenti proprietà.

Property	Type	Description
dashboardid	ID	ID del dashboard. Property behavior: - *sola lettura* - *obbligatorio* per le operazioni di aggiornamento
name	string	Nome del dashboard. Property behavior: - *obbligatorio* per le operazioni di creazione
userid	ID	ID dell'utente proprietario del dashboard.
private	integer	Tipo di condivisione del dashboard. Valori possibili: 0 - dashboard pubblico; 1 - *(predefinito)* dashboard privato.
display_period	integer	Intervallo predefinito della presentazione automatica (in secondi). Valori possibili: 10, 30, 60, 120, 600, 1800, 3600. Predefinito: 30.
auto_start	integer	Avvio automatico della presentazione automatica. Valori possibili: 0 - non avviare automaticamente la presentazione automatica; 1 - *(predefinito)* avviare automaticamente la presentazione automatica.

Pagina dashboard

L'oggetto della pagina dashboard ha le seguenti proprietà.

Property	Type	Description
dashboard_pageid	ID	ID della pagina dashboard. Property behavior: - *read-only*
name	string	Nome della pagina dashboard. Default: stringa vuota.
display_period	integer	Intervallo della presentazione della dashboard (in secondi). Valori possibili: 0, 10, 30, 60, 120, 600, 1800, 3600. Default: 0 (verrà usato l'intervallo predefinito della presentazione).
widgets	array	Array degli oggetti dashboard widget.

Widget della dashboard

L'oggetto widget della dashboard ha le seguenti proprietà.

Property	Type	Description
widgetid	ID	ID del widget della dashboard. Property behavior: - *sola lettura*
type	string	Tipo del widget della dashboard. Valori possibili: actionlog - Registro azioni; clock - Orologio; discovery - Stato del rilevamento; favgraphs - Grafici preferiti; favmaps - Mappe preferite; gauge - Indicatore; geomap - Geomappa; graph - Grafico (classico); graphprototype - Prototipo di grafico; honeycomb - Nido d’ape; hostavail - Disponibilità host; hostcard - Scheda host; hostnavigator - Navigatore host; itemcard - Scheda item; itemhistory - Cronologia item; itemnavigator - Navigatore item; item - Valore item; map - Mappa; navtree - Albero di navigazione della mappa; piechart - Grafico a torta; problemhosts - Host con problemi; problems - Problemi; problemsbysv - Problemi per gravità; scatterplot - Grafico a dispersione; slareport - Report SLA; svggraph - Grafico; systeminfo - Informazioni di sistema; tophosts - Top host; topitems - Top item; toptriggers - Top trigger; trigover - Panoramica trigger; url - URL; web - Monitoraggio web. Property behavior: - *obbligatorio*
name	string	Nome personalizzato del widget.
x	integer	Posizione orizzontale dal lato sinistro della dashboard. I valori possibili vanno da 0 a 71.
y	integer	Posizione verticale dalla parte superiore della dashboard. I valori possibili vanno da 0 a 63.
width	integer	Larghezza del widget. I valori possibili vanno da 1 a 72.
height	integer	Altezza del widget. I valori possibili vanno da 1 a 64.
view_mode	integer	Modalità di visualizzazione del widget. Valori possibili: 0 - *(predefinita)* visualizzazione predefinita del widget; 1 - con intestazione nascosta;

Property	Type	Description
fields	array	Array di oggetti campo del widget della dashboard. Property behavior: - vedere i singoli widget in Campi del widget della dashboard

Campo del widget della dashboard

L'oggetto campo del widget della dashboard ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
type	integer	Tipo del campo del widget. Valori possibili: 0 - Intero; 1 - Stringa; 2 - Gruppo di host; 3 - Host; 4 - Item; 5 - Prototipo di item; 6 - Grafico; 7 - Prototipo di grafico; 8 - Mappa; 9 - Servizio; 10 - SLA; 11 - Utente; 12 - Azione; 13 - Tipo di supporto. Comportamento della proprietà: - *obbligatorio*
name	string	Nome del campo del widget. Valori possibili: vedere Campi del widget della dashboard. Comportamento della proprietà: - *obbligatorio*
value	mixed	Valore del campo del widget in base al tipo. Valori possibili: vedere Campi del widget della dashboard. Comportamento della proprietà: - *obbligatorio*

Gruppo utenti della dashboard

Elenco delle autorizzazioni della dashboard in base ai gruppi utenti. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
usrgrpid	ID	ID del gruppo utenti. Comportamento della proprietà: - *obbligatoria*

Proprietà	Type	Descrizione
permission	integer	Tipo di livello di autorizzazione. Valori possibili: 2 - sola lettura; 3 - lettura-scrittura. Comportamento della proprietà: - *obbligatoria*

Utente dashboard

Elenco dei permessi della dashboard in base agli utenti. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
userid	ID	ID dell'utente. Comportamento della proprietà: - *obbligatoria*
permission	integer	Tipo di livello di permesso. Valori possibili: 2 - sola lettura; 3 - lettura-scrittura. Comportamento della proprietà: - *obbligatoria*

dashboard.create

Descrizione

`object dashboard.create(object/array dashboards)`

Questo metodo consente di creare nuovi dashboard.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Dashboard da creare.

Oltre alle proprietà standard della dashboard, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
pages	array	Pagine della dashboard da creare per la dashboard. Le pagine della dashboard saranno ordinate nello stesso ordine in cui sono specificate. Comportamento dei parametri: - *obbligatorio*
users	array	Condivisioni utente della dashboard da creare sulla dashboard.
userGroups	array	Condivisioni gruppo utenti della dashboard da creare sulla dashboard.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle dashboard create nella proprietà `dashboardids`. L'ordine degli ID restituiti corrisponde all'ordine delle dashboard passate.

Esempi

Creazione di una dashboard

Creare una dashboard denominata ”My dashboard” con un widget Problemi con tag e utilizzando due tipi di condivisione (gruppo utenti e utente) in una singola pagina della dashboard.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Vedi anche

- Pagina dashboard
- Widget dashboard
- Campo del widget dashboard
- Utente dashboard
- Gruppo utenti dashboard

Sorgente

CDashboard::create() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Descrizione

`object dashboard.delete(array dashboardids)`

Questo metodo consente di eliminare i dashboard.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle dashboard da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle dashboard eliminate nella proprietà `dashboardids`.

Esempi

Eliminare molteplici dashboard

Elimina due dashboard.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "dashboard.delete",
"params": [
"2",
"3"
],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"dashboardids": [
"2",
"3"
]
},
```

```
"id": 1
}
```

Fonte

CDashboard::delete() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

Descrizione

`integer/array dashboard.get(object parameters)`

Il metodo consente di recuperare i dashboard in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
dashboardids	ID/array	Restituisce solo le dashboard con gli ID specificati.
selectPages	query	Restituisce una proprietà `pages` con le pagine della dashboard, ordinate correttamente.
selectUsers	query	Restituisce una proprietà `users` con gli utenti con cui la dashboard è condivisa.
selectUserGroups	query	Restituisce una proprietà `userGroups` con i gruppi di utenti con cui la dashboard è condivisa.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `dashboardid`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(intero/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero di una dashboard tramite ID

Recupera tutti i dati delle dashboard "1" e "2".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Pagina dashboard
- Widget dashboard
- Campo del widget dashboard
- Utente dashboard
- Gruppo utenti dashboard

Sorgente

CDashboard::get() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

Descrizione

`object dashboard.update(object/array dashboards)`

Questo metodo consente di aggiornare dashboard esistenti.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà della dashboard da aggiornare.

La proprietà `dashboardid` deve essere definita per ogni dashboard, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard della dashboard, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
pages	array	Pagine della dashboard per sostituire le pagine della dashboard esistenti. Le pagine della dashboard vengono aggiornate tramite la proprietà `dashboard_pageid`. Le nuove pagine della dashboard verranno create per gli oggetti senza la proprietà `dashboard_pageid` e le pagine della dashboard esistenti verranno eliminate se non riutilizzate. Le pagine della dashboard saranno ordinate nello stesso ordine specificato. Verranno aggiornate solo le proprietà specificate delle pagine della dashboard.
users	array	Condivisioni utente della dashboard per sostituire gli elementi esistenti.
userGroups	array	Condivisioni gruppo utenti della dashboard per sostituire gli elementi esistenti.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle dashboard aggiornate nella proprietà `dashboardids`.

Esempi

Rinominare una dashboard

Rinominare una dashboard in ”SQL server status”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
```

```
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Aggiornamento delle pagine della dashboard

Rinominare la prima pagina della dashboard, sostituire i widget nella seconda pagina della dashboard e aggiungere una nuova pagina come terza. Eliminare tutte le altre pagine della dashboard.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Pagina rinominata"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Cambia proprietario della dashboard

Disponibile solo per admin e super admin.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "dashboard.update",
"params": {
"dashboardid": "2",
"userid": "1"
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"dashboardids": [
"2"
]
},
"id": 1
}
```

Vedi anche

- Pagina dashboard
- Widget dashboard
- Campo del widget dashboard
- Utente dashboard
- Gruppo utenti dashboard

Sorgente

CDashboard::update() in *ui/include/classes/api/services/CDashboard.php*.

Campi del widget della dashboard

Questa pagina contiene i link di navigazione per i parametri dei widget della dashboard e i possibili valori delle proprietà per i rispettivi oggetti dashboard widget field.

Per vedere i parametri e i valori delle proprietà per ciascun widget, utilizzare la barra laterale per accedere alla pagina del widget corrispondente.

1 Registro delle azioni

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Registro delle azioni* nei metodi `dashboard.create` e `dashboard.update`.

> **Attention:**
> Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Registro delle azioni*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Action log*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Destinatari	11	userids.0	ID User. Nota: per configurare più utenti, creare un oggetto campo del widget dashboard per ogni utente, utilizzando un numero incrementato nel nome della proprietà.
Actions	12	actionids.0	ID Action. Nota: per configurare più actions, creare un oggetto campo del widget dashboard per ogni action, utilizzando un numero incrementato nel nome della proprietà.
Tipi di supporto	13	mediatypeids.0	ID Media type. Nota: per configurare più tipi di supporto, creare un oggetto campo del widget dashboard per ogni tipo di supporto, utilizzando un numero incrementato nel nome della proprietà.
Stato	0	statuses.0	0 - In corso; 1 - Inviato/Eseguito; 2 - Non riuscito. Nota: per configurare più valori, creare un oggetto campo del widget dashboard per ogni valore, utilizzando un numero incrementato nel nome della proprietà.
Stringa di ricerca	1	message	Qualsiasi valore stringa.
Intervallo di tempo	1	time_period._referenceDASHBOARD._timeperiod - imposta il selettore dell'intervallo di tempo del dashboard come origine dati; ABCDE._timeperiod - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati. Predefinito: DASHBOARD._timeperiod In alternativa, è possibile impostare l'intervallo di tempo solo nei parametri *Da* e *A*.	
Da	1	time_period.from	Stringa temporale valida in formato assoluto (YYYY-MM-DD hh:mm:ss) o nella sintassi temporale relativa (now, now/d, now/w-1w, ecc.). Comportamento del parametro: - *supportato* se *Intervallo di tempo* non è impostato - *obbligatorio* se time_period.to è impostato
A	1	time_period.to	Stringa temporale valida in formato assoluto (YYYY-MM-DD hh:mm:ss) o nella sintassi temporale relativa (now, now/d, now/w-1w, ecc.). Comportamento del parametro: - *supportato* se *Intervallo di tempo* non è impostato - *obbligatorio* se time_period.from è impostato

Parametro	type	name	value
Ordina voci per	0	sort_triggers	3 - Ora (crescente); 4 - *(predefinito)* Ora (decrescente); 5 - Tipo (crescente); 6 - Tipo (decrescente); 7 - Stato (crescente); 8 - Stato (decrescente); 11 - Destinatario (crescente); 12 - Destinatario (decrescente).
Mostra righe	0	show_lines	I valori possibili sono compresi tra 1 e 100. Predefinito: 25.

Esempi

Gli esempi seguenti mirano a descrivere solo la configurazione degli oggetti campo del widget del dashboard per il widget *Registro azioni*. Per ulteriori informazioni sulla configurazione di un dashboard, vedere `dashboard.create`.

Configurazione di un widget *Registro azioni*

Configurare un widget *Registro azioni* che visualizzi 10 voci con i dettagli delle operazioni delle azioni, ordinate per ora (in ordine crescente). Inoltre, visualizzare i dettagli solo per quelle operazioni di azione che hanno tentato di inviare un'email all'utente "1", ma senza successo.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

2 Orologio

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Orologio* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Orologio*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Clock*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.
Tipo di ora	0	time_type	0 - *(predefinito)* Ora locale; 1 - Ora del server; 2 - Ora del host.
Tipo di orologio	0	clock_type	0 - *(predefinito)* Analogico; 1 - Digitale.

I seguenti parametri sono supportati se *Tipo di ora* è impostato su ”Ora del host”.

Parametro	type	name	value
Item	4	itemid.0	ID item. Comportamento del parametro: - *obbligatorio*

I seguenti parametri sono supportati se *Tipo di orologio* è impostato su ”Digitale”.

Parametro	type	name	value
Mostra	0	show.0	1 - Data; 2 - *(predefinito)* Ora; 3 - Fuso orario. Nota: per configurare più valori, creare un oggetto campo widget della dashboard per ciascun valore con un numero incrementato nel nome della proprietà.

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati se *Clock type* è impostato su ”Digital”.

Parametro	type	name	value
Colore di sfondo	1	bg_color	Codice colore esadecimale (ad es. FF0000). Predefinito: "" (vuoto).

Data

I seguenti parametri di configurazione avanzata sono supportati se *Clock type* è impostato su ”Digital” e *Show* è impostato su ”Date”.

Parametro	type	name	value
Grassetto	0	date_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	date_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).

Ora

I seguenti parametri di configurazione avanzata sono supportati se *Clock type* è impostato su ”Digital” e *Show* è impostato su ”Time”.

Parameter	type	name	value
Grassetto	0	time_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	time_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).
Secondi	0	time_sec	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Formato	0	time_format	0 - *(predefinito)* 24 ore; 1 - 12 ore.

Fuso orario

I seguenti parametri di configurazione avanzata sono supportati se *Clock type* è impostato su ”Digital” e *Show* è impostato su ”Time zone”.

Parametro	type	name	value
Grassetto	0	tzone_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	tzone_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).
Fuso orario	1	tzone_timezone	Stringa di fuso orario valida (ad esempio Europe/Riga, system, UTC, ecc.). Per l’elenco completo dei fusi orari supportati, fare riferimento alla documentazione PHP. Predefinito: local. Comportamento del parametro: - *supportato* se *Time type* è impostato su ”Local time” o ”Server time”
Formato	0	tzone_format	0 - *(predefinito)* Breve; 1 - Completo. Comportamento del parametro: - *supportato* se *Time type* è impostato su ”Local time” o ”Server time”

Esempi

Gli esempi seguenti mirano a descrivere solo la configurazione degli oggetti campo del widget del dashboard per il widget *Orologio*. Per ulteriori informazioni sulla configurazione di un dashboard, vedere dashboard.create.

Configurazione di un widget *Clock*

Configurare un widget *Clock* che visualizza data, ora e fuso orario locali in un orologio digitale personalizzato.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

3 Stato della discovery

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Stato del rilevamento* nei metodi `dashboard.create` e `dashboard.update`.

Parametri

I seguenti parametri sono supportati per il widget *Stato della discovery*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget dashboard per il widget *Stato della discovery*. Per ulteriori informazioni sulla configurazione di una dashboard, consultare `dashboard.create`.

Configurazione del widget *Stato del discovery*

Configurare un widget *Stato del discovery* con l'intervallo di aggiornamento impostato su 15 minuti.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

4 Grafici preferiti

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget del dashboard consentono di configurare il widget *Grafici preferiti* nei metodi `dashboard.create` e `dashboard.update`.

Parametri

I seguenti parametri sono supportati per il widget *Grafici preferiti*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere esclusivamente la configurazione degli oggetti campo del widget dashboard per il widget *Grafici preferiti*. Per ulteriori informazioni sulla configurazione di una dashboard, consultare `dashboard.create`.

Configurazione di un widget *Grafici preferiti*

Configurare un widget *Grafici preferiti* con l'intervallo di aggiornamento impostato su 10 minuti.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

5 Mappe preferite

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Mappe preferite* nei metodi `dashboard.create` e `dashboard.update`.

Parametri

I seguenti parametri sono supportati per il widget *Mappe preferite*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere esclusivamente la configurazione degli oggetti campo del widget dashboard per il widget *Mappe preferite*. Per ulteriori informazioni sulla configurazione di una dashboard, consultare `dashboard.create`.

Configurazione di un widget *Mappe preferite*

Configurare un widget *Mappe preferite* con l'intervallo di aggiornamento impostato su 10 minuti.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

6 Misuratore

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Gauge* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Gauge*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Gauge*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Item	4	itemid.0	ID item. Comportamento del parametro: - *obbligatorio* se *Item (Widget)* non è impostato
Item (Widget)	1	itemid._reference	Invece dell'ID item: `ABCDE._itemid` - imposta un widget compatibile (con il relativo parametro *Reference* impostato su "ABCDE") come origine dati per gli item. Comportamento del parametro: - *obbligatorio* se *Item* non è impostato
Min	1	min	Qualsiasi valore numerico. I suffissi (ad es. "1d", "2w", "4K", "8G") sono supportati. Predefinito: "0".
Max	1	max	Qualsiasi valore numerico. I suffissi (ad es. "1d", "2w", "4K", "8G") sono supportati. Predefinito: "100".
Arco del valore	1	value_arc_color	Codice colore esadecimale (ad es. `FF0000`). Predefinito: `""` (vuoto).
Sfondo dell'arco	1	empty_color	Codice colore esadecimale (ad es. `FF0000`). Predefinito: `""` (vuoto).
Sfondo	1	bg_color	Codice colore esadecimale (ad es. `FF0000`). Predefinito: `""` (vuoto).
Mostra	0	show.0	1 - Descrizione; 2 - Valore; 3 - Lancetta; 4 - Scala; 5 - Arco del valore. Nota: per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementato nel nome della proprietà. Predefinito: 1, 2, 4, 5. I valori "Lancetta" e "Scala" non sono supportati se entrambe le condizioni seguenti sono vere: - non è impostato alcun oggetto campo widget dashboard per *Mostra* con il valore "Arco del valore"; - il parametro di configurazione avanzata *Show arc* è impostato su "Disabled". I parametri di configurazione avanzata per le opzioni di *Mostra* non sono supportati se non è impostato alcun oggetto campo widget dashboard con i rispettivi valori.

Parametro	type	name	value
Sostituisci host	1	override_hostid._reference	ABCDE._hostid - imposta un widget compatibile (con il relativo parametro *Reference* impostato su "ABCDE") come origine dati per gli host; DASHBOARD._hostid - imposta il selettore host della dashboard come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati per il widget *Gauge*.

Note:
Il numero nel nome della proprietà *Thresholds* (ad esempio, thresholds.0.color) fa riferimento alla posizione della soglia in un elenco ordinato in ordine crescente. Tuttavia, se le soglie sono configurate in un ordine diverso, i valori verranno ordinati in ordine crescente dopo l'aggiornamento della configurazione del widget nel frontend di Zabbix (ad esempio, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametro	tipo	nome	valore
Angolo	0	angle	Valori possibili: 180 *(predefinito)* o 270.
Descrizione			
Descrizione	1	description	Qualsiasi valore stringa, incluse le macro. Macro supportate: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macro utente. Predefinito: {ITEM.NAME}.
Dimensione	0	desc_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 15.
Posizione verticale	0	desc_v_pos	0 - In alto; 1 - In basso *(predefinito)*.
Grassetto	0	desc_bold	0 - Disabilitato *(predefinito)*; 1 - Abilitato.
Colore	1	desc_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto).
Valore			
Posizioni decimali	0	decimal_places	I valori possibili sono compresi tra 1 e 10. Predefinito: 2.
Dimensione	0	value_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 25.
Grassetto	0	value_bold	0 - Disabilitato *(predefinito)*; 1 - Abilitato.
Colore	1	value_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto).
Unità			
Unità (casella di controllo)	0	units_show	0 - Disabilitato; 1 - Abilitato *(predefinito)*.
Unità (valore)	1	units	Qualsiasi valore stringa. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su "Abilitato"

Parametro	tipo	nome	valore
Dimensione	0	units_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 25. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su ”Abilitato”
Grassetto	0	units_bold	0 - Disabilitato *(predefinito)*; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su ”Abilitato”
Posizione	0	units_pos	0 - Prima del valore; 1 - Sopra il valore; 2 - Dopo il valore *(predefinito)*; 3 - Sotto il valore. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su ”Abilitato” Questo parametro viene ignorato se impostato su una delle seguenti unità relative al tempo: unixtime, uptime, s.
Colore	1	units_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto).
Arco del valore			
Dimensione dell’arco	0	value_arc_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 20.
Lancetta			
Colore	1	needle_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto). Comportamento del parametro: - *supportato* se è impostato un oggetto campo del widget dashboard per *Mostra* con il valore ”Arco del valore”, oppure se *Mostra arco* è impostato su ”Abilitato”
Scala			
Mostra unità	0	scale_show_units	0 - Disabilitato; 1 - Abilitato *(predefinito)*. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su ”Abilitato” e se è impostato un oggetto campo del widget dashboard per *Mostra* con il valore ”Arco del valore”, oppure *Mostra arco* è impostato su ”Abilitato”
Dimensione	0	scale_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 15. Comportamento del parametro: - *supportato* se è impostato un oggetto campo del widget dashboard per *Mostra* con il valore ”Arco del valore”, oppure se *Mostra arco* è impostato su ”Abilitato”
Posizioni decimali	0	scale_decimal_places	I valori possibili sono compresi tra 1 e 10. Predefinito: 0. Comportamento del parametro: - *supportato* se è impostato un oggetto campo del widget dashboard per *Mostra* con il valore ”Arco del valore”, oppure se *Mostra arco* è impostato su ”Abilitato”

Parametro	tipo	nome	valore
Soglie			
Colore	1	thresholds.0.color	Codice colore esadecimale (ad esempio, FF0000).
Soglia	1	thresholds.0.threshold	Qualsiasi valore numerico. Sono supportati i suffissi (ad esempio, ”1d”, ”2w”, ”4K”, ”8G”).
Mostra etichette	0	th_show_labels	0 - Disabilitato *(predefinito)*; 1 - Abilitato. Comportamento del parametro: - *supportato* se sono impostate le *Soglie* e se è impostato un oggetto campo del widget dashboard per *Mostra* con il valore ”Arco del valore”, oppure *Mostra arco* è impostato su ”Abilitato”
Mostra arco	0	th_show_arc	0 - Disabilitato *(predefinito)*; 1 - Abilitato. Comportamento del parametro: - *supportato* se sono impostate le *Soglie*
Dimensione dell’arco	0	th_arc_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 5. Comportamento del parametro: - *supportato* se *Mostra arco* è impostato su ”Abilitato”

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Gauge*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Gauge*

Configurare un widget *Gauge* che visualizza il valore dell’item per l’item ”44474” (Interfaccia enp0s3: Bit inviati). Inoltre, perfezionare visivamente il widget con più opzioni avanzate, incluse le soglie.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

7 Geomap

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Geomap* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Geomap*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Geomap*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.

Parametro	tipo	nome	valore
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: per configurare più gruppi di host, creare un oggetto campo del widget dashboard per ogni gruppo di host, utilizzando un numero incrementato nel nome della proprietà. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Gruppi di host (widget)	1	groupids._reference	Invece dell'ID del gruppo di host: `ABCDE._hostgroupids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi di host. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Host	3	hostids.0	ID dell'host. Nota: per configurare più host, creare un oggetto campo del widget dashboard per ogni host, utilizzando un numero incrementato nel nome della proprietà. Per più host, il parametro *Gruppi di host* deve essere non configurato oppure configurato con almeno un gruppo di host a cui appartengono gli host configurati. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Host (widget/dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - impostare il selettore di host della dashboard come origine dati per gli host; `ABCDE._hostids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Tag			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Nome del tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* quando si configura *Tag* Questo parametro non è supportato quando si configura il widget in una dashboard di template.

Parametro	tipo	nome	valore
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* quando si configura *Tag* Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Valore del tag	1	tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* quando si configura *Tag* Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Vista iniziale	1	default_view	*Latitudine*, *longitudine*, *livello di zoom* separati da virgole (*facoltativo*, i valori possibili sono compresi tra 0 e 30). Esempio: `40.6892494,-74.0466891,10`.
Raggruppamento			
Modalità di raggruppamento	0	clustering_mode	0 - *(predefinita)* Automatica; 1 - Livello di zoom.
Livello di zoom del raggruppamento	0	clustering_zoom_level	I valori possibili sono compresi tra 0 e 30. Predefinito: 0. Comportamento del parametro: - *supportato* se *Modalità di raggruppamento* è impostata su "Livello di zoom"
Riferimento	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco nella dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo esclusivamente di descrivere la configurazione degli oggetti campo del widget della dashboard per il widget *Geomap*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Geomap*

Configurare un widget *Geomap* che visualizza gli host dei gruppi host "2" e "22" in base alla seguente configurazione dei tag: il tag con nome "component" contiene il valore "node" e il tag con nome "location" è uguale al valore "New York". Inoltre, impostare la vista iniziale della mappa sulle coordinate "40.6892494" (latitudine), "-74.0466891" (longitudine) con il livello di zoom "10".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

8 Grafico

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Graph* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Graph*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Graph*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Riferimento	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Set di dati

I seguenti parametri sono supportati per la configurazione di un *set di dati*.

Note:
Il primo numero nel nome della proprietà (ad esempio, ds.0.hosts.0, ds.0.items.0) rappresenta il set di dati specifico, mentre il secondo numero, se presente, rappresenta l'host o l'item configurato.

Parametro	type	name	value
Tipo di set di dati	0	ds.0.dataset_type	0 - Elenco di item; 1 - * (predefinito)* Pattern di item.
Item	4	ds.0.itemids.0	ID dell'item. Quando si configura il widget in una dashboard di template, devono essere impostati solo gli item configurati nel template. Nota: per configurare più item, creare un oggetto campo del widget dashboard per ogni item, incrementando il numero nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco di item" e *Item (widget)* non è impostato
Item (widget)	1	ds.0.itemids.0._reference	Riferimento dell'ID dell'item: `ABCDE._itemid` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli item. Nota: per configurare più widget, creare un oggetto campo del widget dashboard per ogni widget, incrementando il numero nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco di item" e *Item* non è impostato
Colore	1	ds.0.color.0	Codice colore esadecimale (ad esempio, `FF0000`). Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco di item"

Parametro	type	name	value
Pattern di host	1	ds.0.hosts.0	Nome o pattern dell'host (ad esempio, "Zabbix*"). Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Pattern di item"
Pattern di item	1	ds.0.items.0	Questo parametro non è supportato quando si configura il widget in una dashboard di template. Nome o pattern dell'item (ad esempio, "*: Numero di *valori elaborati al secondo"). Quando si configura il widget in una dashboard di template, devono essere impostati solo i pattern degli item configurati nel template. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Pattern di item"
Colore	1	ds.0.color	Codice colore esadecimale (ad esempio, FF0000). Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su "Pattern di item" e *Tavolozza dei colori* non è impostata
Tavolozza dei colori	0	ds.0.color_palette	Indice della tavolozza dei colori. I valori possibili vanno da 0 a 11. Predefinito: 0. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su "Pattern di item" e *Colore* non è impostato
Disegno	0	ds.0.type	0 - * (predefinito)* Linea; 1 - Punti; 2 - Gradini; 3 - Barre.
In pila	0	ds.0.stacked	0 - * (predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Disegno* è impostato su "Linea", "Gradini" o "Barre"
Larghezza	0	ds.0.width	I valori possibili vanno da 1 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Disegno* è impostato su "Linea" o "Gradini"
Dimensione dei punti	0	ds.0.pointsize	I valori possibili vanno da 1 a 10. Predefinito: 3. Comportamento del parametro: - *supportato* se *Disegno* è impostato su "Punti"
Trasparenza	0	ds.0.transparency	I valori possibili vanno da 1 a 10. Predefinito: 5.
Riempimento	0	ds.0.fill	I valori possibili vanno da 1 a 10. Predefinito: 3. Comportamento del parametro: - *supportato* se *Disegno* è impostato su "Linea" o "Gradini"

Parametro	type	name	value
Dati mancanti	0	ds.0.missingdatafunc	0 - * (predefinito)* Nessuno; 1 - Connessi; 2 - Considera come 0; 3 - Ultimo noto. Comportamento del parametro: - *supportato* se *Disegno* è impostato su ”Linea” o ”Gradini”
Sovrascrivi host	1	ds.0.override_hostid	ABCDE._hostid - impostare un widget compatibile (con il parametro ”Reference” impostato su ”ABCDE”) come origine dati per gli host; DASHBOARD._hostid - impostare il selettore di host della dashboard come origine dati per gli host. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Asse Y	0	ds.0.axisy	0 - * (predefinito)* Sinistra; 1 - Destra.
Inverti valori	0	ds.0.invert_values	0 - (predefinito) Disabilitato; 1 - Abilitato.
Spostamento temporale	1	ds.0.timeshift	Stringa temporale valida (ad esempio, 3600, 1h, ecc.). È possibile utilizzare i suffissi temporali. Sono consentiti anche valori negativi. Predefinito: "" (vuoto).
Funzione di aggregazione	0	ds.0.aggregate_function	0 - * (predefinito)* non utilizzata; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Intervallo di aggregazione	1	ds.0.aggregate_interval	Stringa temporale valida (ad esempio, 3600, 1h, ecc.). È possibile utilizzare i suffissi temporali. Predefinito: 1h.
Aggrega	0	ds.0.aggregate_grouping	0 - * (predefinito)* Ogni item; 1 - Set di dati. Comportamento del parametro: - *supportato* se *Funzione di aggregazione* è impostata su ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” o ”last”
Approssimazione	0	ds.0.approximation	1 - min; 2 - * (predefinito)* avg; 4 - max; 7 - tutti.
Etichetta del set di dati	1	ds.0.data_set_label	Qualsiasi valore stringa. Predefinito: "" (vuoto).
Tag degli item			
Tipo di valutazione	0	ds.0.item_tags_evaltype	0 - * (predefinito)* And/Or; 2 - Or.

Parametro	type	name	value
Nome del tag	1	ds.0.item_tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* quando si configurano i *Tag degli item*
Operatore	0	ds.0.item_tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* quando si configurano i *Tag degli item*
Valore del tag	1	ds.0.item_tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* quando si configurano i *Tag degli item*

Opzioni di visualizzazione

I seguenti parametri sono supportati per la configurazione delle *Opzioni di visualizzazione*.

Parametro	type	name	value
Selezione dei dati cronologici	0	source	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Tendenze.
trigger semplici	0	simple_triggers	0 - *(predefinito)* Disabilitati; 1 - Abilitati.
Orario di lavoro	0	working_time	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Nomi degli host nelle etichette *Linea del percentile (sinistra)*	0	show_hostnames	0 - *(predefinito)* Automatico; 1 - Mostra; 2 - Nascondi. Questo parametro non è supportato se si configura il widget in una dashboard di template.

Parametro	type	name	value
Stato	0	percentile_left	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su "Sinistra"
Valore	0	percentile_left_value	I valori possibili sono compresi tra 1 e 100. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su "Sinistra"
Linea del per-centile (de-stra)			
Stato	0	percentile_right	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su "Destra"
Valore	0	percentile_right_value	I valori possibili sono compresi tra 1 e 100. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su "Destra"

Periodo di tempo

I seguenti parametri sono supportati per la configurazione di *Periodo di tempo*.

Parametro	type	name	value
Periodo di tempo	1	time_period._reference	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il suo parametro *Reference* impostato su "`ABCDE`") come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.
Da	1	time_period.from	Stringa temporale valida nella sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
A	1	time_period.to	Stringa temporale valida nella sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.from` è impostato

Assi

I seguenti parametri sono supportati per la configurazione di *Assi*.

Parametro	type	name	value
Y sinistro	0	lefty	0 - Disabilitato; 1 - *(predefinito)* Abilitato. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su ”Sinistra”
Y destro	0	righty	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su ”Destra”
Scala	0	lefty_scale righty_scale	0 - *(predefinito)* Lineare; 1 - Logaritmica.
Min	1	lefty_min righty_min	Qualsiasi valore numerico. Predefinito: "" (vuoto).
Max	1	lefty_max righty_max	Qualsiasi valore numerico. Predefinito: "" (vuoto).
Unità (tipo)	0	lefty_units righty_units	0 - *(predefinito)* Automatico; 1 - Statico.
Unità (valore)	1	lefty_static_units righty_static_units	Qualsiasi valore stringa. Predefinito: "" (vuoto).
Asse X	0	xaxis	0 - Disabilitato; 1 - *(predefinito)* Abilitato.

Legenda

I parametri seguenti sono supportati per la configurazione della *Legenda*.

Parametro	tipo	nome	valore
Mostra legenda	0	legend	0 - Disabilitata; 1 - *(predefinita)* Abilitata.
Visualizza min/media/max	0	legend_statistic	0 - *(predefinita)* Disabilitata; 1 - Abilitata. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitata”
Mostra funzione di aggregazione	0	legend_aggregation	0 - *(predefinita)* Disabilitata; 1 - Abilitata. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitata”
Righe	0	legend_lines_mode	0 - *(predefinita)* Fisse; 1 - Variabili. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitata”

Parametro	tipo	nome	valore
Numero di righe/ Numero massimo di righe	0	legend_lines	I valori possibili vanno da 1 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata"
Numero di colonne	0	legend_columns	I valori possibili vanno da 1 a 4. Predefinito: 4. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata" e *Visualizza min/media/max* è impostato su "Disabilitata"

Problemi

I seguenti parametri sono supportati per la configurazione di *Problemi*.

Parametro	type	name	value
Mostra problemi	0	show_problems	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Solo item selezionati	0	graph_item_problems	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Host dei problemi	1	problemhosts.0	Nome Host. Nota: Il numero nel nome della proprietà fa riferimento all'host configurato. Per configurare più host, creare un oggetto campo widget dashboard per ciascun host con un numero incrementato nel nome della proprietà. Questo parametro non è supportato se si configura il widget su una dashboard template.
Gravità	0	severities.0	0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: Per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementato nel nome della proprietà.
Problema	1	problem_name	Nome evento del problema (senza distinzione tra maiuscole e minuscole, nome completo o parte di esso).
Tag del problema			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.

Parametro	type	name	value
Nome tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag del problema*
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag del problema*
Valore tag	1	tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag del problema*

Override

I seguenti parametri sono supportati per la configurazione degli *Override*.

Note:
Il primo numero nel nome della proprietà (ad esempio, or.0.hosts.0, or.0.items.0) rappresenta il set di dati specifico, mentre il secondo numero, se presente, rappresenta l'host o l'item configurato.

Parametro	type	name	value
Pattern degli host	1	or.0.hosts.0	Nome o pattern dell'host (ad esempio, `Zabbix*`). Questo parametro non è supportato quando si configura il widget in una dashboard di template. Comportamento del parametro: - *obbligatorio* quando si configurano gli *Override*
Pattern degli item	1	or.0.items.0	Nome o pattern dell'item (ad esempio, `*: Number of processed *values per second`). Quando si configura il widget in una dashboard di template, devono essere impostati solo i pattern degli item configurati nel template. Comportamento del parametro: - *obbligatorio* quando si configurano gli *Override*
Colore di base	1	or.0.color	Codice colore esadecimale (ad esempio, FF0000).
Larghezza	0	or.0.width	I valori possibili vanno da 1 a 10.
Disegno	0	or.0.type	0 - Linea; 1 - Punti; 2 - Gradini; 3 - Barre.
Trasparenza	0	or.0.transparency	I valori possibili vanno da 1 a 10.
Riempimento	0	or.0.fill	I valori possibili vanno da 1 a 10.
Dimensione dei punti	0	or.0.pointsize	I valori possibili vanno da 1 a 10.

Parametro	type	name	value
Dati mancanti	0	or.0.missingdatafunc	0 - Nessuno; 1 - Connessi; 2 - Considera come 0; 3 - Ultimo valore noto.
Asse Y	0	or.0.axisy	0 - Sinistra; 1 - Destra.
Inverti valori	0	or.0.invert_values	0 - (predefinito) Disabilitato; 1 - Abilitato.
Spostamento temporale	1	or.0.timeshift	Stringa temporale valida (ad esempio, 3600, 1h, ecc.). È possibile utilizzare i suffissi temporali. Sono consentiti valori negativi.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Graph*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Graph*

Configurare un widget *Graph* nel modo seguente:

- 2 set di dati per un totale di 9 item su 1 host.
- Il primo set di dati è di tipo ”Item list” ed è composto da 3 item rappresentati da linee con colore diverso, ma con la stessa larghezza, trasparenza e riempimento.
- Il secondo set di dati è di tipo ”Item patterns”, è composto da 6 item, ha un’aggregazione configurata ed è rappresentato da una linea con colore, larghezza, trasparenza e riempimento personalizzati.
- Il secondo set di dati ha anche un’etichetta personalizzata del set di dati.
- I dati nel grafico vengono visualizzati per un periodo di tempo corrispondente alle ultime 3 ore.
- I problemi nel grafico vengono visualizzati solo per gli item configurati.
- Il grafico ha due assi Y, di cui l’asse Y destro visualizza valori solo per il secondo set di dati.
- La legenda del grafico visualizza gli item configurati in 4 righe, nonché i valori minimo, massimo e medio dei set di dati.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

9 Grafico (classico)

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Graph (classic)* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Graph (classic)*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Graph (classic)*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Sorgente	0	source_type	0 - *(predefinito)* Graph; 1 - Grafico semplice.
Graph	6	graphid.0	ID Graph. Comportamento del parametro: - *obbligatorio* se *Sorgente* è impostato su "Graph"
Graph (Widget)	1	graphid._reference	Invece dell'ID Graph: `ABCDE._graphid` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come sorgente dati per i grafici. Comportamento del parametro: - *obbligatorio* se *Sorgente* è impostato su "Simple graph" e *Graph* non è impostato
Item	4	itemid.0	ID Item. Comportamento del parametro: - *obbligatorio* se *Sorgente* è impostato su "Simple graph" e *Item (Widget)* non è impostato
Item (Widget)	1	itemid._reference	Invece dell'ID Item: `ABCDE._itemid` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come sorgente dati per gli item. Comportamento del parametro: - *obbligatorio* se *Sorgente* è impostato su "Simple graph" e *Item* non è impostato
Periodo di tempo	1	time_period._reference	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come sorgente dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come sorgente dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *From* e *To*.
From	1	time_period.from	Stringa temporale valida in sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
To	1	time_period.to	Stringa temporale valida in sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.from` è impostato
Mostra legenda	0	show_legend	0 - Disabilitato; 1 - *(predefinito)* Abilitato.

Parametro	type	name	value
Sostituisci host	1	override_hostid._reference	`ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come sorgente dati per gli host; `DASHBOARD._hostid` - imposta il selettore host della dashboard come sorgente dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

I seguenti esempi hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Graph (classic)*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Graph (classic)*

Configurare un widget *Graph (classic)* che visualizza un grafico semplice per l'item "42269".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

10 Prototipo di grafico

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Graph prototype* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Graph prototype*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Graph prototype*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Origine	0	source_type	2 - *(predefinito)* Graph prototype; 3 - Simple graph prototype.

Parametro	type	name	value
Graph prototype	7	graphid.0	ID del Graph prototype. Comportamento del parametro: - *obbligatorio* se *Origine* è impostato su ”Graph prototype”
Item prototype	5	itemid.0	ID dell’Item prototype. Comportamento del parametro: - *obbligatorio* se *Origine* è impostato su ”Simple graph prototype”
Periodo di tempo	1	time_period._referenda	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dei dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro *Reference* impostato su ”`ABCDE`”) come origine dei dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.
Da	1	time_period.from	Stringa temporale valida in formato assoluto (`YYYY-MM-DD hh:mm:ss`) o nella sintassi temporale relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
A	1	time_period.to	Stringa temporale valida in formato assoluto (`YYYY-MM-DD hh:mm:ss`) o nella sintassi temporale relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.from` è impostato
Mostra leggenda	0	show_legend	0 - Disabilitata; 1 - *(predefinita)* Abilitata.
Sovrascrivi host	1	override_hostid._referenda	`ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su ”`ABCDE`”) come origine dei dati per gli host; `DASHBOARD._hostid` - imposta il selettore degli host della dashboard come origine dei dati per gli host. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Colonne	0	columns	I valori possibili sono compresi tra 1 e 24. Predefinito: 2.
Righe	0	rows	I valori possibili sono compresi tra 1 e 16. Predefinito: 1.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco nella dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Graph prototype*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Prototipo di grafico*

Configurare un widget *Prototipo di grafico* che visualizza una griglia di 3 grafici (3 colonne, 1 riga) creati da un prototipo di item

(ID: ”42316”) tramite low-level discovery.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

11 Nido d'ape

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Honeycomb* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Honeycomb*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Honeycomb*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi host	2	groupids.0	ID del gruppo host. Nota: Per configurare più gruppi host, creare un oggetto campo widget dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID del gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Host	3	hostids.0	ID dell'host. Nota: Per configurare più host, creare un oggetto campo widget dashboard per ciascun host con un numero incrementale nel nome della proprietà. Per più host, il parametro *Gruppi host* non deve essere configurato oppure deve essere configurato con almeno un gruppo host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host (Widget/Dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Tag host			
Tipo di valutazione	0	evaltype_host	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget in una dashboard template.
Nome tag	1	host_tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Operatore	0	host_tags.0.operator	0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Valore tag	1	host_tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Pattern item	1	items.0	Nome o pattern dell'item. Nota: Per configurare più pattern item, creare un oggetto campo widget dashboard per ciascun pattern item con un numero incrementale nel nome della proprietà. Comportamento del parametro: - *obbligatorio*
Tag item			
Tipo di valutazione	0	evaltype_item	0 - *(predefinito)* And/Or; 2 - Or.
Nome tag	1	item_tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag item*
Operatore	0	item_tags.0.operator	0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag item*
Valore tag	1	item_tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag item*
Mostra host in manuten-zione	0	maintenance	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Mostra	0	show.0	1 - Etichetta primaria; 2 - Etichetta secondaria. Nota: Per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementale nel nome della proprietà. Predefinito: 1, 2.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati per il widget *Honeycomb*.

Note:
Il numero nel nome della proprietà *Thresholds* (ad esempio, thresholds.0.color) fa riferimento alla posizione della soglia in un elenco ordinato in ordine crescente. Tuttavia, se le soglie sono configurate in un ordine diverso, i valori verranno ordinati in ordine crescente dopo l'aggiornamento della configurazione del widget nel frontend di Zabbix (ad esempio, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametro	type	name	value
Etichetta primaria			
Tipo	0	primary_label_type	0 - *(predefinito)* Testo; 1 - Valore.
Testo	1	primary_label	Qualsiasi valore stringa, incluse le macro. Macro supportate: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, macro utente. Predefinito: `{HOST.NAME}` Comportamento del parametro: - *supportato* se *Tipo* è impostato su "Testo"
Posizioni decimali	0	primary_label_decimal_places	I valori possibili vanno da 0 a 6. Predefinito: 2. Comportamento del parametro: - *supportato* se *Tipo* è impostato su "Valore"
Dimensione (tipo)	0	primary_label_size_type	0 - *(predefinito)* Automatica; 1 - Personalizzata.
Dimensione	0	primary_label_size	I valori possibili vanno da 1 a 100. Predefinito: 20. Comportamento del parametro: - *supportato* se *Dimensione* (tipo) è impostata su "Personalizzata"
Grassetto	0	primary_label_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	primary_label_color	Codice colore esadecimale (ad esempio, `FF0000`). Predefinito: basato su `theme` dell'oggetto Settings e dell'oggetto User: `1F2C33` per "blue-theme", "blue-classic-theme" o "hc-light"; `EEEEEE` per "dark-theme", "dark-classic-theme" o "hc-dark"; `E5F1FF` per "dark-blue-theme".
Unità (casella di controllo)	0	primary_label_units_show	0 - Disabilitate; 1 - *(predefinite)* Abilitate. Comportamento del parametro: - *supportato* se *Tipo* è impostato su "Valore"
Unità (valore)	1	primary_label_units	Qualsiasi valore stringa. "" (vuoto) Comportamento del parametro: - *supportato* se *Tipo* è impostato su "Valore" e *Unità* (casella di controllo) è impostata su "Abilitate"

Parametro	type	name	value
Posizione	0	primary_label_units_pos	0 - Prima del valore; 1 - *(predefinita)* Dopo il valore. Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Valore” e *Unità* (casella di controllo) è impostata su ”Abilitate” Questo parametro viene ignorato se impostato su una delle seguenti unità relative al tempo: unixtime, uptime, s.
Etichetta secondaria			
Tipo	0	secondary_label_type	0 - Testo; 1 - *(predefinito)* Valore.
Testo	1	secondary_label	Qualsiasi valore stringa, incluse le macro. Macro supportate: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, macro utente. Predefinito: `{{ITEM.LASTVALUE}.fmtnum(2)}` Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Testo”
Posizioni decimali	0	secondary_label_decimal_places	I valori possibili vanno da 0 a 6. Predefinito: 2. Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Valore”
Dimensione (tipo)	0	secondary_label_size_type	0 - *(predefinita)* Automatica; 1 - Personalizzata.
Dimensione	0	secondary_label_size	I valori possibili vanno da 1 a 100. Predefinito: 30. Comportamento del parametro: - *supportato* se *Dimensione* (tipo) è impostata su ”Personalizzata”
Grassetto	0	secondary_label_bold	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Colore	1	secondary_label_color	Codice colore esadecimale (ad esempio, `FF0000`). Predefinito: basato su `theme` dell’oggetto Settings e dell’oggetto User: `1F2C33` per ”blue-theme”, ”blue-classic-theme” o ”hc-light”; `EEEEEE` per ”dark-theme”, ”dark-classic-theme” o ”hc-dark”; `E5F1FF` per ”dark-blue-theme”.
Unità (casella di controllo)	0	secondary_label_units_show	0 - Disabilitate; 1 - *(predefinite)* Abilitate. Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Valore”
Unità (valore)	1	secondary_label_units	Qualsiasi valore stringa. "" (vuoto) Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Valore” e *Unità* (casella di controllo) è impostata su ”Abilitate”

Parametro	type	name	value
Posizione	0	secondary_label_position	0 - Prima del valore; 1 - *(predefinita)* Dopo il valore. Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Valore” e *Unità* (casella di controllo) è impostata su ”Abilitate” Questo parametro viene ignorato se impostato su una delle seguenti unità relative al tempo: unixtime, uptime, s.
Colore di sfondo			
Colore di sfondo	1	bg_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: basato su theme dell’oggetto Settings e dell’oggetto User: D9E7ED per ”blue-theme” e ”blue-classic-theme”; 3D5059 per ”dark-theme” e ”dark-classic-theme”; 004FA8 per ”dark-blue-theme”; AAD7E9 per ”hc-light”; 335463 per ”hc-dark”.
Soglie			
Interpolazione del colore	0	interpolation	0 - Disabilitata; 1 - *(predefinita)* Abilitata.
Colore	1	thresholds.0.color	Codice colore esadecimale (ad esempio, FF0000).
Soglia	1	thresholds.0.threshold	Qualsiasi valore numerico. Sono supportati i suffissi (ad esempio, ”1d”, ”2w”, ”4K”, ”8G”).

Esempi

Gli esempi seguenti hanno lo scopo esclusivo di descrivere la configurazione degli oggetti campo del widget della dashboard per il widget *Honeycomb*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere dashboard.create.

Configurazione di un widget *Honeycomb*

Configurare un widget *Honeycomb* che visualizza l’utilizzo dei processi del server Zabbix. Inoltre, modificare l’etichetta primaria delle celle honeycomb e ottimizzare visivamente il widget con le soglie.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

12 Disponibilità dell'host

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Disponibilità host* nei metodi `dashboard.create` e `dashboard.update`.

Parametri

I seguenti parametri sono supportati per il widget *Disponibilità host*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.
Gruppi host	2	groupids.0	ID del gruppo host. Nota: per configurare più gruppi host, creare un oggetto campo widget dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID del gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Tipo di interfaccia	0	interface_type.0	0 - Nessuno; 1 - Zabbix agent (controlli passivi); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix agent (controlli attivi). Predefinito: 1, 2, 3, 4, 5 (tutti abilitati). Nota: per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementale nel nome della proprietà.
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale.
Mostra host in manutenzione	0	maintenance	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Mostra solo i totali	0	only_totals	0 - *(predefinito)* Disabilitato; 1 - Abilitato.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Host availability*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Disponibilità host*

Configurare un widget *Disponibilità host* che visualizzi informazioni sulla disponibilità (in un layout verticale) per gli host nel gruppo host "4" con interfacce "Zabbix agent" e "SNMP" configurate.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

13 Scheda dell'host

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Host card* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Host card*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Host card*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Host	3	hostid.0	ID host. Comportamento del parametro: - *obbligatorio* se *Host (Widget/Dashboard)* non è impostato Questo parametro non è supportato se si configura il widget in una dashboard template.
Host (Widget/Dashboard)	1	hostid._reference	Invece dell'ID host: `DASHBOARD.hostid` - imposta il selettore host della dashboard come origine dati per l'host; `ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per l'host. Comportamento del parametro: - *obbligatorio* se *Host* non è impostato Questo parametro non è supportato se si configura il widget in una dashboard template.
Mostra problemi soppressi	0	show_suppressed	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Mostra	0	sections.0	0 - Gruppi host; 1 - Descrizione; 2 - Monitoraggio; 3 - Disponibilità; 4 - Monitorato da; 5 - Template; 6 - Inventario; 7 - Tag. Nota: Il numero nel nome della proprietà fa riferimento all'ordine della sezione nell'elenco delle sezioni. Per configurare più sezioni, creare un oggetto campo widget della dashboard per ciascuna sezione con un numero incrementale nel nome della proprietà.

I seguenti parametri sono supportati se *Mostra* è impostato su "Inventario".

Parametro	type	name	value
Campi inventario	0	inventory.0	ID Inventory. Nota: Per configurare più campi inventario, creare un oggetto campo widget della dashboard per ciascun campo inventario con un numero incrementale nel nome della proprietà.

Esempi

Gli esempi seguenti hanno lo scopo esclusivo di descrivere la configurazione degli oggetti campo del widget della dashboard per il widget *Host card*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Scheda host*

Configurare un widget *Scheda host* che visualizzi queste sezioni: ”Monitoraggio”, ”Disponibilità”, ”Monitorato da”, ”Inventario” e ”Tag”.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

14 Navigatore degli host

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di

configurare il widget *Host navigator* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Host navigator*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Host navigator*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: per configurare più gruppi di host, creare un oggetto campo del widget dashboard per ogni gruppo di host, incrementando il numero nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Gruppi di host (widget)	1	groupids._reference	Invece dell'ID del gruppo di host: `ABCDE._hostgroupids` - impostare un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per i gruppi di host. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Pattern degli host	1	hosts.0	Nome o pattern dell'host. Nota: per configurare più pattern di host, creare un oggetto campo del widget dashboard per ogni pattern di host, incrementando il numero nel nome della proprietà. Per più pattern di host, il parametro *Gruppi di host* deve essere non configurato oppure configurato con almeno un gruppo di host a cui appartengono gli host corrispondenti ai pattern di host configurati. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Stato dell'host	0	status	-1 - *(predefinito)* Qualsiasi; 0 - Abilitato; 1 - Disabilitato. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Tag degli host			
Tipo di valutazione	0	host_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget in una dashboard di template.

Parametro	type	name	value
Nome del tag	1	host_tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato se si configura il widget in una dashboard di template.
Operatore	0	host_tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato se si configura il widget in una dashboard di template.
Valore del tag	1	host_tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato se si configura il widget in una dashboard di template.
Gravità	0	severities.0	0 - Non classificato; 1 - Informazioni; 2 - Avviso; 3 - Media; 4 - Alta; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: per configurare più valori, creare un oggetto campo del widget dashboard per ogni valore, incrementando il numero nel nome della proprietà.
Mostra gli host in manuten-zione	0	maintenance	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Mostra i prob-lemi Raggruppa per	0	show_problems	0 - Tutti; 1 - *(predefinito)* Non soppressi; 2 - Nessuno.

Parametro	type	name	value
Attributo	0	group_by.0.attribute	0 - Gruppo di host; 1 - Valore del tag; 2 - Gravità. Nota: il numero nel nome della proprietà fa riferimento all'ordine dell'attributo nell'elenco degli attributi di raggruppamento. Comportamento del parametro: - *obbligatorio* se si configura *Raggruppa per*
Valore	1	group_by.0.tag_name	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'attributo di raggruppamento impostato nel parametro *Attributo*. Comportamento del parametro: - *obbligatorio* se si configura *Raggruppa per* e *Attributo* è impostato su "Valore del tag"
Limite di host	0	show_lines	I valori possibili vanno da 1 a 9999. Predefinito: 100. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Host navigator*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Navigatore host*

Configurare un widget *Navigatore host* che visualizza gli host raggruppati per gruppo host e, successivamente, per il valore del tag "city".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard

- `dashboard.create`
- `dashboard.update`

15 Scheda item

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Item card* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Item card*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Item card*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Item	4	itemid.0	ID Item. Comportamento del parametro: - *obbligatorio* se *Item (Widget)* non è impostato
Item (Widget)	1	itemid._reference	Invece dell'ID Item: `ABCDE._itemid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per l'item. Comportamento del parametro: - *obbligatorio* se *Item* non è impostato
Mostra	0	sections.0	0 - Descrizione; 1 - Testo di errore; 2 - Metriche; 3 - Dati più recenti; 4 - Tipo di informazione; 5 - Trigger; 6 - Interfaccia host; 7 - Tipo; 8 - Inventario host; 9 - Tag. Nota: Il numero nel nome della proprietà fa riferimento all'ordine della sezione nell'elenco delle sezioni. Per configurare più sezioni, creare un oggetto campo widget dashboard per ciascuna sezione con un numero incrementale nel nome della proprietà.

Parametro	type	name	value
Sovrascrivi host	1	override_hostid._reference	`ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per gli host; `DASHBOARD._hostid` - imposta il selettore host della dashboard come origine dati per gli host. Predefinito: "" (vuoto) Questo parametro non è supportato se si configura il widget in una dashboard template.

Sparkline

I seguenti parametri sono supportati se *Mostra* è impostato su "Dati più recenti".

Parametro	tipo	nome	valore
Larghezza	0	sparkline.width	I valori possibili vanno da 0 a 10. Predefinito: 1.
Riempimento	0	sparkline.fill	I valori possibili vanno da 0 a 10. Predefinito: 3.
Colore	1	sparkline.color	Codice colore esadecimale (ad es., `FF0000`). Predefinito: `42A5F5`.
Intervallo di tempo	1	sparkline.time_period.reference	`DASHBOARD._timeperiod` - imposta il selettore dell'intervallo di tempo del dashboard come origine dei dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro `reference` uguale a `ABCDE`) come origine dei dati. Predefinito: "" (vuoto) In alternativa, puoi impostare l'intervallo di tempo solo nei parametri *Da* e *A*.
Da	1	sparkline.time_period.from	Stringa temporale valida in formato assoluto (`YYYY-MM-DD hh:mm:ss`) o nella sintassi temporale relativa (`now`, `now/d`, `now/w-1w`, ecc.). Predefinito: now-1h. Comportamento del parametro: - *supportato* se *Intervallo di tempo* non è impostato - *obbligatorio* se `sparkline.time_period.to` è impostato
A	1	sparkline.time_period.to	Stringa temporale valida in formato assoluto (`YYYY-MM-DD hh:mm:ss`) o nella sintassi temporale relativa (`now`, `now/d`, `now/w-1w`, ecc.). Predefinito: now. Comportamento del parametro: - *supportato* se *Intervallo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
Dati storici	0	sparkline.history	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Trend.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Item card*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Item card*

Configurare un widget *Item card* che visualizza queste sezioni: ”Description”, ”Latest data”, ”Triggers” e ”Tags”.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

16 Cronologia degli item

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Cronologia item* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Cronologia item*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati dal widget *Cronologia degli item*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale.
Colonne (vedere di seguito)			
Mostra righe	0	show_lines	I valori possibili sono compresi tra 1 e 100. Predefinito: 25.
Sovrascrivi host	1	override_hostid._reference	`ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host; `DASHBOARD._hostid` - imposta il selettore di host del dashboard come origine dati per gli host. Questo parametro non è supportato quando si configura il widget in un dashboard di template.

Parametro	tipo	nome	valore
Configurazione avanzata (vedere di seguito)			
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno del dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Colonne

Le colonne hanno parametri comuni e parametri aggiuntivi a seconda della configurazione del parametro *Item*.

Note:
Per tutti i parametri relativi alle colonne, il numero nel nome della proprietà (ad esempio, columns.0.name) fa riferimento a una colonna per la quale il parametro è configurato.

I seguenti parametri sono supportati per tutte le colonne.

Parametro	type	name	value
Nome	1	columns.0.name	Qualsiasi valore stringa. Comportamento del parametro: - *obbligatorio*
Item	4	columns.0.itemid	ID dell'item. Quando si configura il widget in una dashboard di template, devono essere impostati solo gli item configurati nel template. Comportamento del parametro: - *obbligatorio*
Colore di base	1	columns.0.base_color	Codice colore esadecimale (ad esempio `FF0000`). Predefinito: `""` (vuoto).

I seguenti parametri di colonna sono supportati se l'*Item* configurato è un item di tipo numerico.

Parametro	type	name	value
Visualizzazione	0	columns.0.display	1 - *(predefinito)* Così com'è; 2 - Barra; 3 - Indicatori.
Min	1	columns.0.min	Qualsiasi valore numerico. Comportamento del parametro: - *supportato* se *Display* è impostato su "Bar" o "Indicators"
Max	1	columns.0.max	Qualsiasi valore numerico. Comportamento del parametro: - *supportato* se *Display* è impostato su "Bar" o "Indicators"
Soglie			
Colore	1	columns.0.thresholds.color	Codice colore esadecimale (ad esempio `FF0000`).
Soglia	1	columns.0.thresholds.threshold	Qualsiasi valore numerico. I suffissi (ad esempio "1d", "2w", "4K", "8G") sono supportati.
Dati storici	0	columns.0.history	0 - *(predefinito)* Auto; 1 - Storico; 2 - Trend.

I seguenti parametri delle colonne sono supportati se l'*Item* configurato è di tipo carattere, testo o log.

Parametro	type	name	value
Evidenziazioni			
Evidenziazione	1	columns.0.highlights.0.color	Qualsiasi colore esadecimale (ad esempio, FF0000).
Soglia	1	columns.0.highlights.0.pattern	Qualsiasi espressione regolare.
Visualizzazione	0	columns.0.display	1 - *(predefinito)* Così com'è; 4 - HTML; 5 - Riga singola.
Riga singola	0	columns.0.max_length	I valori possibili vanno da 1 a 500. Predefinito: 100. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostato su "Riga singola"
Usa carattere a spaziatura fissa	0	columns.0.monospace_font	0 - *(predefinito)* Usa il carattere predefinito; 1 - Usa un carattere a spaziatura fissa.
Visualizza l'ora locale	0	columns.0.local_time	0 - *(predefinito)* Visualizza il timestamp; 1 - Visualizza l'ora locale. Comportamento del parametro: - *supportato* se *Item* è impostato su un item di tipo log e *Mostra timestamp* è impostato su "Abilitato"

I seguenti parametri di colonna sono supportati se l'*Item* configurato è un item di tipo binario.

Parametro	type	name	value
Mostra miniatura	1	columns.0.show_thumbnail	0 - *(predefinito)* Disabilitato; 1 - Abilitato.

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati per il widget *Cronologia item*.

Parametro	type	name	value
Nuovi valori	0	sortorder	0 - *(predefinito)* In alto; 1 - In basso.
Mostra timestamp	0	show_timestamp	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Mostra intestazione colonna	0	show_column_header	0 - Disattivata; 1 - Orizzontale; 2 - *(predefinito)* Verticale.
Periodo di tempo	1	time_period._reference	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.

Parametro	type	name	value
Da	1	time_period.from	Stringa temporale valida in sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se è impostato `time_period.to`
A	1	time_period.to	Stringa temporale valida in sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se è impostato `time_period.from`

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Item history*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Cronologia item*

Configurare un widget *Cronologia item* che visualizzi i dati più recenti per due item numerici "42269" e "42270". Inoltre, configurare le colonne degli item in modo che vengano visualizzate verticalmente, con i nomi delle colonne visualizzati orizzontalmente; limitare la visualizzazione a 15 righe di dati e includere una colonna timestamp separata.

Richiesta:

```
{
    "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
}
```


Risposta:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


Vedi anche

- Campo del widget della dashboard
- `dashboard.create`

- `dashboard.update`

17 Navigatore item

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Navigatore item* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Navigatore item*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Item navigator*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: per configurare più gruppi di host, creare un oggetto campo del widget dashboard per ogni gruppo di host, incrementando il numero nel nome della proprietà. Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Gruppi di host (widget)	1	groupids._reference	Invece dell'ID del gruppo di host: `ABCDE._hostgroupids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi di host. Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Host	3	hostids.0	ID dell'host. Nota: per configurare più host, creare un oggetto campo del widget dashboard per ogni host, incrementando il numero nel nome della proprietà. Per più host, il parametro *Gruppi di host* deve essere non configurato oppure configurato con almeno un gruppo di host a cui appartengono gli host configurati. Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Host (widget/dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostid` - impostare il selettore di host della dashboard come origine dati per gli host; `ABCDE._hostid` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato quando si configura il widget su una dashboard di template.

Parametro	type	name	value
Tag degli host			
Tipo di valutazione	0	host_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Nome del tag	1	host_tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Operatore	0	host_tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Valore del tag	1	host_tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Pattern degli item	1	items.0	Nome o pattern dell'item. Nota: per configurare più pattern di item, creare un oggetto campo del widget dashboard per ogni pattern di item, incrementando il numero nel nome della proprietà.
Tag degli item			
Tipo di valutazione	0	item_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome del tag	1	item_tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli item*

Parametro	type	name	value
Operatore	0	item_tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli item*
Valore del tag	1	item_tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli item*
Stato	0	state	-1 - *(predefinito)* Tutti; 0 - Normale; 1 - Non supportato.
Mostra problemi	0	show_problems	0 - Tutti; 1 - *(predefinito)* Non soppressi; 2 - Nessuno.
Raggruppa per			
Attributo	0	group_by.0.attribute	0 - Gruppo di host; 1 - Nome host; 2 - Valore del tag dell'host; 3 - Valore del tag dell'item. Nota: il numero nel nome della proprietà fa riferimento all'ordine dell'attributo nell'elenco degli attributi di raggruppamento. Comportamento del parametro: - *obbligatorio* se si configura *Raggruppa per*
Valore	1	group_by.0.tag_name	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'attributo di raggruppamento impostato nel parametro *Attributo*. Comportamento del parametro: - *obbligatorio* se si configura *Raggruppa per* e *Attributo* è impostato su "Valore del tag dell'host" o "Valore del tag dell'item"
Limite degli item	0	show_lines	I valori possibili sono compresi tra 1 e 9999. Predefinito: 100.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Item navigator*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Navigatore item*

Configurare un widget *Navigatore item* che visualizzi fino a 1000 item raggruppati per host e poi per il valore del tag item "component".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

18 Valore dell'item

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Valore dell'item* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Valore dell'item*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Valore item*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Item	4	itemid.0	ID Item. Comportamento del parametro: - *obbligatorio* se *Item (Widget)* non è impostato
Item (Widget)	1	itemid._reference	Invece dell'ID Item: `ABCDE._itemid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per gli item. Comportamento del parametro: - *obbligatorio* se *Item* non è impostato

Parametro	type	name	value
Mostra	0	show.0	1 - Descrizione; 2 - Valore; 3 - Ora; 4 - Indicatore di variazione; 5 - Sparkline. Predefinito: 1, 2, 3, 4. Nota: per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementato nel nome della proprietà.
Sostituisci host	1	override_hostid._reference	`ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per gli host; `DASHBOARD._hostid` - imposta il selettore host della dashboard come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati per il widget *Valore item*.

Note:
Il numero nel nome della proprietà *Thresholds* (ad esempio thresholds.0.color) fa riferimento alla posizione della soglia in un elenco, ordinato in ordine crescente. Tuttavia, se le soglie sono configurate in un ordine diverso, i valori verranno ordinati in ordine crescente dopo l'aggiornamento della configurazione del widget nel frontend di Zabbix (ad esempio `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametro	type	name	value
Colore di sfondo	1	bg_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).
Thresholds			
Colore	1	thresholds.0.color	Codice colore esadecimale (ad esempio FF0000).
Soglia	1	thresholds.0.threshold	Qualsiasi valore stringa.
Funzione di aggregazione	0	aggregate_function	0 - *(predefinito)* non utilizzata; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Periodo di tempo	1	time_period._reference	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro `reference` uguale a ABCDE) come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*. Comportamento del parametro: - *supportato* se *Funzione di aggregazione* è impostata su "min", "max", "avg", "count", "sum", "first", "last"

Parametro	type	name	value
Da	1	time_period.from	Stringa temporale valida nella sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato e *Funzione di aggregazione* è impostata su ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obbligatorio* se `time_period.to` è impostato
A	1	time_period.to	Stringa temporale valida nella sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato e *Funzione di aggregazione* è impostata su ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obbligatorio* se `time_period.from` è impostato
Dati storici	0	history	0 - *(predefinito)* Auto; 1 - History; 2 - Trends.

Descrizione

I seguenti parametri di configurazione avanzati sono supportati se *Mostra* è impostato su ”Descrizione”.

Parametro	tipo	nome	valore
Descrizione	1	description	Qualsiasi valore stringa, incluse le macro. Macro supportate: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, macro utente. Predefinito: `{ITEM.NAME}`.
Posizione orizzontale	0	desc_h_pos	0 - Sinistra; 1 - *(predefinito)* Centro; 2 - Destra. Due o più elementi (Descrizione, Valore, Ora) non possono condividere la stessa *Posizione orizzontale* e *Posizione verticale*.
Posizione verticale	0	desc_v_pos	0 - In alto; 1 - Al centro; 2 - *(predefinito)* In basso. Due o più elementi (Descrizione, Valore, Ora) non possono condividere la stessa *Posizione orizzontale* e *Posizione verticale*.
Dimensione	0	desc_size	I valori possibili vanno da 1 a 100. Predefinito: 15.
Grassetto	0	desc_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	desc_color	Codice colore esadecimale (ad esempio, `FF0000`). Predefinito: `""` (vuoto).

Valore

I seguenti parametri di configurazione avanzati sono supportati se *Mostra* è impostato su ”Valore”.

Parametro	type	name	value
Posizioni decimali			
Posizioni decimali	0	decimal_places	I valori possibili vanno da 1 a 10. Predefinito: 2.
Dimensione	0	decimal_size	I valori possibili vanno da 1 a 100. Predefinito: 35.
Posizione			
Posizione orizzontale	0	value_h_pos	0 - Sinistra; 1 - *(predefinito)* Centro; 2 - Destra. Due o più elementi (Descrizione, Valore, Ora) non possono condividere la stessa *Posizione orizzontale* e *Posizione verticale*.
Posizione verticale	0	value_v_pos	0 - In alto; 1 - *(predefinito)* Al centro; 2 - In basso. Due o più elementi (Descrizione, Valore, Ora) non possono condividere la stessa *Posizione orizzontale* e *Posizione verticale*.
Dimensione	0	value_size	I valori possibili vanno da 1 a 100. Predefinito: 45.
Grassetto	0	value_bold	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Colore	1	value_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto).
Unità			
Unità (casella di controllo)	0	units_show	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Unità (valore)	1	units	Qualsiasi valore stringa.
Posizione	0	units_pos	0 - Prima del valore; 1 - Sopra il valore; 2 - *(predefinito)* Dopo il valore; 3 - Sotto il valore.
Dimensione	0	units_size	I valori possibili vanno da 1 a 100. Predefinito: 35.
Grassetto	0	units_bold	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Colore	1	units_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto).

Ora

I seguenti parametri di configurazione avanzati sono supportati se *Mostra* è impostato su ”Ora”.

Parametro	type	name	value
Posizione orizzontale	0	time_h_pos	0 - Sinistra; 1 - *(predefinito)* Centro; 2 - Destra. Due o più elementi (Descrizione, Valore, Ora) non possono condividere la stessa *Posizione orizzontale* e *Posizione verticale*.

Parametro	type	name	value
Posizione verticale	0	time_v_pos	0 - *(predefinito)* In alto; 1 - Al centro; 2 - In basso. Due o più elementi (Descrizione, Valore, Ora) non possono condividere la stessa *Posizione orizzontale* e *Posizione verticale*.
Dimensione	0	time_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 15.
Grassetto	0	time_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	time_color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: "" (vuoto).

Indicatore di variazione

I seguenti parametri di configurazione avanzata sono supportati se *Mostra* è impostato su ”Indicatore di variazione”.

Parametro	type	name	value
Colore indicatore di variazione ↑	1	up_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).
Colore indicatore di variazione ↓	1	down_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).
Colore indicatore di variazione ↕	1	updown_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).

Sparkline

I seguenti parametri di configurazione avanzata sono supportati se *Show* è impostato su ”Sparkline”.

Parameter	type	name	value
Width	0	sparkline.width	I valori possibili vanno da 0 a 10. Predefinito: 1.
Fill	0	sparkline.fill	I valori possibili vanno da 0 a 10. Predefinito: 3.
Color	1	sparkline.color	Codice colore esadecimale (ad esempio, FF0000). Predefinito: 42A5F5.
Time period	1	sparkline.time_period.reference	DASHBOARD._timeperiod - imposta il selettore dell’intervallo di tempo della dashboard come origine dei dati; ABCDE._timeperiod - imposta un widget compatibile (con il parametro reference impostato su ABCDE) come origine dei dati. Predefinito: DASHBOARD._timeperiod
From	1	sparkline.time_period.from	In alternativa, puoi impostare l’intervallo di tempo solo nei parametri *From* e *To*. Stringa temporale valida in formato assoluto (YYYY-MM-DD hh:mm:ss) o nella sintassi temporale relativa (now, now/d, now/w-1w, ecc.). Comportamento del parametro: - *obbligatorio* se sparkline.time_period.to è impostato

Parameter	type	name	value
To	1	sparkline.time_period.to	Stringa temporale valida in formato assoluto (`YYYY-MM-DD hh:mm:ss`) o nella sintassi temporale relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *obbligatorio* se `sparkline.time_period.from` è impostato
History data	0	sparkline.history	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Trend.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Item value*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Valore item*

Configurare un widget *Valore item* che visualizzi il valore dell'item per l'item "42266" (disponibilità di Zabbix agent). Inoltre, personalizzare visivamente il widget con più opzioni avanzate, incluso un colore di sfondo dinamico che cambia in base allo stato di disponibilità di Zabbix agent.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`

- `dashboard.update`

19 Mappa

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Mappa* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Mappa*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Map*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.
Map	8	sysmapid.0	ID della Map. Comportamento del parametro: - *obbligatorio* se *Map (Widget)* non è impostato
Map (Widget)	1	sysmapid._reference	`ABCDE._mapid` - imposta un widget *Map navigation tree* (con il relativo parametro *Reference* impostato su "`ABCDE`") come origine dati per le mappe. Comportamento del parametro: - *obbligatorio* se *Map* non è impostato
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

I seguenti esempi mirano a descrivere solo la configurazione degli oggetti campo del widget dashboard per il widget *Mappa*. Per ulteriori informazioni sulla configurazione di un dashboard, vedere `dashboard.create`.

Configurazione di un widget *Mappa*

Configurare un widget *Mappa* che visualizza la mappa "1".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Configurazione di un widget *Mappa* collegato

Configurare un widget *Mappa* collegato a un widget *Albero di navigazione mappa*.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`
- Albero di navigazione della mappa

20 Albero di navigazione della mappa

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Albero di navigazione delle mappe* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Albero di navigazione delle mappe*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Albero di navigazione mappe*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.
Mostra mappe non disponibili	1	show_unavailable	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Riferimento	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

I seguenti parametri sono supportati per la configurazione degli elementi dell'albero di navigazione mappe.

Parametro	type	name	value
Nome	1	navtree.1.name	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà imposta il numero dell'elemento.
Mappa collegata	8	navtree.1.sysmapid	ID della mappa. Nota: Il numero nel nome della proprietà fa riferimento all'elemento a cui è collegata la mappa.
Parametri per la creazione della gerarchia degli elementi	0	navtree.1.parent	Numero dell'elemento padre. Nota: Il numero nel nome della proprietà fa riferimento all'elemento figlio. Il valore della proprietà fa riferimento all'elemento padre.
	0	navtree.1.order	Posizione dell'elemento nell'albero di navigazione mappe. Nota: Il numero nel nome della proprietà fa riferimento al numero dell'elemento. Il valore della proprietà fa riferimento alla posizione dell'elemento nell'albero di navigazione mappe. La posizione dell'elemento padre è determinata all'interno dell'intero albero di navigazione mappe. La posizione dell'elemento figlio è determinata all'interno dell'elemento padre.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Map navigation tree*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Albero di navigazione mappa*

Configurare un widget *Albero di navigazione mappa* che visualizzi il seguente albero di navigazione della mappa:

- Elemento A
  - Elemento A1
  - Elemento A2
- Elemento B
  - Elemento B1 (contiene la mappa collegata "1" che può essere visualizzata in un *widget Mappa* collegato)
  - Elemento B2
- Elemento C

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`
- Mappa

21 Grafico a torta

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Grafico a torta* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Grafico a torta*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Grafico a torta*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.

Set di dati

I seguenti parametri sono supportati per la configurazione di un *Set di dati*.

Note:
Il primo numero nel nome della proprietà (ad esempio, ds.0.hosts.0, ds.0.items.0) rappresenta il set di dati specifico, mentre il secondo numero, se presente, rappresenta l'host o l'item configurato.

Parametro	type	name	value
Tipo di set di dati	0	ds.0.dataset_type	0 - Elenco di item; 1 - * (predefinito)* Pattern di item.
Item	4	ds.0.itemids.0	ID dell'item. Quando si configura il widget su una dashboard di template, devono essere impostati solo gli item configurati sul template. Nota: per configurare più item, creare un oggetto campo widget della dashboard per ogni item, incrementando il numero nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco di item" e *Item (widget)* non è impostato
Item (widget)	1	ds.0.itemids.0._reference	Riferimento dell'ID dell'item: `ABCDE._itemid` - imposta un widget compatibile (con il parametro *Riferimento* impostato su "`ABCDE`") come origine dati per gli item. Nota: per configurare più widget, creare un oggetto campo widget della dashboard per ogni widget, incrementando il numero nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco di item" e *Item* non è impostato
Colore	1	ds.0.color.0	Codice colore esadecimale (ad esempio, `FF0000`). Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco di item"
Tipo di item	0	ds.0.type.0	0 - * (predefinito)* Normale; 1 - Totale. Il valore "Totale" può essere impostato per un solo item nell'intero grafico. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su "Elenco di item"
Pattern di host	1	ds.0.hosts.0	Nome o pattern dell'host (ad esempio, "Zabbix*"). Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Pattern di item" Questo parametro non è supportato quando si configura il widget su una dashboard di template.
Pattern di item	1	ds.0.items.0	Nome o pattern dell'item (ad esempio, "*: Numero di *valori elaborati al secondo"). Quando si configura il widget su una dashboard di template, devono essere impostati solo i pattern degli item configurati sul template. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Pattern di item"

Parametro	type	name	value
Colore	1	ds.0.color	Codice colore esadecimale (ad esempio, FF0000). Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su ”Pattern di item” e *Tavolozza dei colori* non è impostata
Tavolozza dei colori	0	ds.0.color_palette	Indice della tavolozza dei colori. I valori possibili vanno da 0 a 11. Predefinito: 0. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su ”Pattern di item” e *Colore* non è impostato
Funzione di aggregazione	0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - * (predefinito)* last.
Aggregazione del set di dati	0	ds.0.dataset_aggregation	0 - * (predefinito)* nessuna; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum. Comportamento del parametro: - *supportato* se *Tipo di item* è impostato su ”Totale”
Etichetta del set di dati	1	ds.0.data_set_label	Qualsiasi valore stringa. Predefinito: "" (vuoto).
Tag degli item			
Tipo di valutazione	0	ds.0.item_tags_evaltype	0 - * (predefinito)* And/Or; 2 - Or.
Nome del tag	1	ds.0.item_tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all’ordine del tag nell’elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag degli item*
Operatore	0	ds.0.item_tags.0.operator	0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all’ordine del tag nell’elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag degli item*

Parametro	type	name	value
Valore del tag	1	ds.0.item_tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag degli item*

Visualizzazione delle opzioni

I seguenti parametri sono supportati per la configurazione delle *opzioni di visualizzazione*.

Parametro	tipo	nome	valore
Selezione dei dati crono-logici	0	source	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Tendenze.
Disegna	0	draw_type	0 - *(predefinito)* Grafico a torta; 1 - Grafico ad anello.
Larghezza	0	width	20 - 20% del raggio; 30 - 30% del raggio; 40 - 40% del raggio; 50 - *(predefinito)* 50% del raggio. Comportamento del parametro: - *supportato* se *Disegna* è impostato su "Grafico ad anello"
Larghezza del bordo	0	stroke	I valori possibili sono compresi tra 0 e 10. Predefinito: 0. Comportamento del parametro: - *supportato* se *Disegna* è impostato su "Grafico ad anello"
Mostra valore totale	0	total_show	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Disegna* è impostato su "Grafico ad anello"
Dimensione	0	value_size_type	0 - *(predefinito)* Automatica; 1 - Personalizzata. Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su "Abilitato"
Dimensione (valore per la dimensione personalizzata)	0	value_size	I valori possibili sono compresi tra 1 e 100. Predefinito: 20. Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su "Abilitato"
Posizioni decimali	0	decimal_places	I valori possibili sono compresi tra 0 e 6. Predefinito: 2. Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su "Abilitato"
Unità (casella di controllo)	0	units_show	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su "Abilitato"

Parametro	tipo	nome	valore
Unità (valore)	1	units	Qualsiasi valore stringa.
Grassetto	0	value_bold	Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su ”Abilitato” 0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	value_color	Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su ”Abilitato” Codice colore esadecimale (ad esempio, FF0000).
Spazio tra i settori	0	space	Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su ”Abilitato” I valori possibili sono compresi tra 0 e 10. Predefinito: 1.
Unisci i settori più piccoli di N% (casella di controllo)	0	merge	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Unisci i settori più piccoli di N% (valore)	0	merge_percent	I valori possibili sono compresi tra 1 e 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Unisci i settori più piccoli di N%* (casella di controllo) è impostato su ”Abilitato”
Unisci i settori più piccoli di N% (colore)	1	merge_color	Codice colore esadecimale (ad esempio, FF0000). Comportamento del parametro: - *supportato* se *Unisci i settori più piccoli di N%* (casella di controllo) è impostato su ”Abilitato”

Periodo di tempo

I seguenti parametri sono supportati per la configurazione di *Periodo di tempo*.

Parametro	type	name	value
Periodo di tempo	1	time_period._reference	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.
Da	1	time_period.from	Stringa temporale valida in sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
A	1	time_period.to	Stringa temporale valida in sintassi temporale assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.from` è impostato

Legenda

I seguenti parametri sono supportati per la configurazione della *Legenda*.

Parametro	tipo	nome	valore
Mostra legenda	0	legend	0 - Disabilitata; 1 - *(predefinita)* Abilitata.
Mostra valore	0	legend_value	0 - *(predefinita)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata"
Mostra funzione di aggregazione	0	legend_aggregation	0 - *(predefinita)* Disabilitata; 1 - Abilitata. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata"
Righe	0	legend_lines_mode	0 - *(predefinita)* Fisse; 1 - Variabili. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata"
Numero di righe/ Numero massimo di righe	0	legend_lines	I valori possibili sono compresi tra 1 e 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata"
Numero di colonne	0	legend_columns	I valori possibili sono compresi tra 1 e 4. Predefinito: 4. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitata" e *Mostra valore* è impostato su "Disabilitato"

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget del widget *Grafico a torta*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Grafico a torta*

Configurare un widget *Grafico a torta* nel modo seguente:

- 2 set di dati per un totale di 9 item su 1 host.
- Il primo set di dati è di tipo ”Elenco item” ed è composto da 3 item, tutti di tipo ”Normale” e rappresentati da un colore diverso.
- Il secondo set di dati è di tipo ”Pattern di item”, è composto da 6 item, ha un’aggregazione configurata per ciascun item ed è rappresentato da un colore personalizzato.
- Il secondo set di dati ha anche un’etichetta del set di dati personalizzata.
- I dati nel grafico a torta sono visualizzati come grafico ad anello con una larghezza personalizzata e il valore totale con unità al centro.
- I dati nel grafico a torta sono visualizzati e aggregati per un periodo di tempo personalizzato delle ultime 3 ore.
- La legenda del grafico a torta visualizza gli item configurati in 4 righe.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`

- `dashboard.update`

22 Host con problemi

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Host con problemi* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Host con problemi*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Host con problemi*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi host	2	groupids.0	ID del gruppo host. Nota: Per configurare più gruppi host, creare un oggetto campo widget della dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID del gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Riferimento* impostato su "ABCDE") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Escludi gruppi host	2	exclude_groupids.0	ID del gruppo host. Nota: Per escludere più gruppi host, creare un oggetto campo widget della dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host	3	hostids.0	ID dell'host. Nota: Per configurare più host, creare un oggetto campo widget della dashboard per ciascun host con un numero incrementale nel nome della proprietà. Per più host, il parametro *Gruppi host* non deve essere configurato oppure deve essere configurato con almeno un gruppo host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Host (Widget/Dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostids` - imposta un widget compatibile (con il parametro *Riferimento* impostato su "ABCDE") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Problema	1	problem	Nome evento del problema (senza distinzione tra maiuscole e minuscole, nome completo o parte di esso).
Gravità	0	severities.0	0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: Per configurare più valori, creare un oggetto campo widget della dashboard per ciascun valore con un numero incrementale nel nome della proprietà.
Tag problema			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag problema*
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag problema*
Valore tag	1	tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag problema*
Mostra problemi soppressi	0	show_suppressed	0 - *(predefinito)* Disabilitato; 1 - Abilitato.

Parametro	type	name	value
Nascondi gruppi senza problemi	0	hide_empty_groups	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Questo parametro non è supportato se si configura il widget in una dashboard template.
Visualizzazione problemi	0	ext_ack	0 - *(predefinito)* Tutti; 1 - Solo non confermati; 2 - Separati.
Riferimento	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Problem hosts*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Host con problemi*

Configurare un widget *Host con problemi* che visualizzi gli host dei gruppi host "2" e "4" che hanno problemi con un nome che include la stringa "CPU" e che hanno le seguenti severità: "Warning", "Average", "High", "Disaster".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

23 Problemi

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Problems* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Problems*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Problemi*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Mostra	0	show	1 - *(predefinito)* Problemi recenti; 2 - Cronologia; 3 - Problemi.
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: per configurare più gruppi di host, creare un oggetto campo del widget dashboard per ogni gruppo di host, incrementando il numero nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Gruppi di host (widget)	1	groupids._reference	Invece dell'ID del gruppo di host: `ABCDE._hostgroupids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi di host. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Escludi gruppi di host	2	exclude_groupids.0	ID del gruppo di host. Nota: per escludere più gruppi di host, creare un oggetto campo del widget dashboard per ogni gruppo di host, incrementando il numero nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Host	3	hostids.0	ID dell'host. Nota: per configurare più host, creare un oggetto campo del widget dashboard per ogni host, incrementando il numero nel nome della proprietà. Per più host, il parametro *Gruppi di host* deve essere completamente non configurato oppure configurato con almeno un gruppo di host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard di template.

Parametro	tipo	nome	valore
Host (widget/dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - impostare il selettore di host della dashboard come origine dati per gli host; `ABCDE._hostids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard di template.
Problema	1	problem	Nome dell'evento Problema event name (senza distinzione tra maiuscole e minuscole, nome completo o parte del nome).
Gravità	0	severities.0	0 - Non classificato; 1 - Informazioni; 2 - Avviso; 3 - Media; 4 - Alta; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: per configurare più valori, creare un oggetto campo del widget dashboard per ogni valore, incrementando il numero nel nome della proprietà.
Tag dei problemi			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome del tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dei problemi*
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dei problemi*
Valore del tag	1	tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dei problemi*
Mostra tag	0	show_tags	0 - *(predefinito)* Nessuno; 1 - 1; 2 - 2; 3 - 3.

Parametro	tipo	nome	valore
Nome del tag (formato)	0	tag_name_format	0 - *(predefinito)* Completo; 1 - Abbreviato; 2 - Nessuno. Comportamento del parametro: - *supportato* se *Mostra tag* è impostato su ”1”, ”2” o ”3”
Priorità di visualizzazione dei tag	1	tag_priority	Elenco di tag separati da virgole. Comportamento del parametro: - *supportato* se *Mostra tag* è impostato su ”1”, ”2” o ”3”
Mostra dati operativi	0	show_opdata	0 - *(predefinito)* Nessuno; 1 - Separatamente; 2 - Con il nome del problema.
Mostra problemi soppressi	0	show_suppressed	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Stato di riconoscimento	0	acknowledgement_status	0 - *(predefinito)* tutti; 1 - Non riconosciuti; 2 - Riconosciuti.
Da me	0	acknowledged_by_me	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Ordina voci per	0	sort_triggers	1 - Gravità (decrescente); 2 - Host (crescente); 3 - Ora (crescente); 4 - *(predefinito)* Ora (decrescente); 13 - Gravità (crescente); 14 - Host (decrescente); 15 - Problema (crescente); 16 - Problema (decrescente). Per tutti i valori, ad eccezione di ”Ora (decrescente)” e ”Ora (crescente)”, il parametro *Mostra sequenza temporale* deve essere impostato su ”Disabilitato”. I valori ”Host (crescente)” e ”Host (decrescente)” non sono supportati se si configura il widget in una dashboard di template.
Mostra sequenza temporale	0	show_timeline	0 - Disabilitato; 1 - *(predefinito)* Abilitato. Comportamento del parametro: - *supportato* se *Ordina voci per* è impostato su ”Ora (decrescente)” o ”Ora (crescente)”
Evidenzia intera riga	0	highlight_row	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Questo parametro non è supportato nei temi ad alto contrasto.
Mostra righe	0	show_lines	I valori possibili sono compresi tra 1 e 100. Predefinito: 25.

Parametro	tipo	nome	valore
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, ABCDE o JBPNL). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Problems*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Problems*

Configurare un widget *Problems* che visualizzi i problemi per il gruppo host "4" che soddisfano le seguenti condizioni:

- Problemi che hanno un tag con nome "scope" che contiene i valori "performance" o "availability", oppure "capacity".
- Problemi che hanno le seguenti severità: "Warning", "Average", "High", "Disaster".

Inoltre, configurare il widget in modo che mostri i tag e i dati operativi.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

24 Problemi per gravità

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Problemi per gravità* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Problemi per gravità*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Problems by severity*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.

Parametro	type	name	value
Gruppi host	2	groupids.0	ID del gruppo host. Nota: Per configurare più gruppi host, creare un oggetto campo widget della dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID del gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Escludi gruppi host	2	exclude_groupids.0	ID del gruppo host. Nota: Per escludere più gruppi host, creare un oggetto campo widget della dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host	3	hostids.0	ID dell'host. Nota: Per configurare più host, creare un oggetto campo widget della dashboard per ciascun host con un numero incrementale nel nome della proprietà. Per più host, il parametro *Gruppi host* non deve essere configurato oppure deve essere configurato con almeno un gruppo host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host (Widget/Dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Problema	1	problem	Nome evento del problema (senza distinzione tra maiuscole e minuscole, nome completo o parte di esso).
Gravità	0	severities.0	0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: Per configurare più valori, creare un oggetto campo widget della dashboard per ciascun valore con un numero incrementale nel nome della proprietà.
Tag problema			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.

Parametro	type	name	value
Nome tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag.
Operatore	0	tags.0.operator	Comportamento del parametro: - *obbligatorio* se si configurano i *Tag problema* 0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag.
Valore tag	1	tags.0.value	Comportamento del parametro: - *obbligatorio* se si configurano i *Tag problema* Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag.
Mostra	0	show_type	Comportamento del parametro: - *obbligatorio* se si configurano i *Tag problema* 0 - *(predefinito)* Gruppi host; 1 - Totali. Questo parametro non è supportato se si configura il widget in una dashboard template e per impostazione predefinita è impostato su "Totali".
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale. Comportamento del parametro: - *supportato* se *Mostra* è impostato su "Totali"
Mostra dati operativi	0	show_opdata	0 - *(predefinito)* Nessuno; 1 - Separatamente; 2 - Con il nome del problema.
Mostra problemi soppressi	0	show_suppressed	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Nascondi gruppi senza problemi	0	hide_empty_groups	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra* è impostato su "Gruppi host" Questo parametro non è supportato se si configura il widget in una dashboard template.
Visualizzazione problemi	0	ext_ack	0 - *(predefinito)* Tutti; 1 - Solo non riconosciuti; 2 - Separati.

Parametro	type	name	value
Mostra cronologia temporale	0	show_timeline	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere esclusivamente la configurazione degli oggetti campo del widget dashboard per il widget *Problemi per gravità*. Per ulteriori informazioni sulla configurazione di una dashboard, consultare `dashboard.create`.

Configurazione di un widget *Problemi per gravità*

Configurare un widget *Problemi per gravità* che visualizza i totali dei problemi per tutti i gruppi di host.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

25 Grafico a dispersione

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Grafico a dispersione* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Grafico a dispersione*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Scatter plot*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Riferimento	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Set di dati

I seguenti parametri sono supportati per configurare un *Set di dati*.

Note:
Il primo numero nel nome della proprietà (ad esempio, ds.0.hosts.0, ds.0.items.0) rappresenta lo specifico set di dati, mentre il secondo numero, se presente, rappresenta l'host o l'item configurato.

Parametro	type	name	value
Tipo di set di dati ***Set di dati: Elenco item***	0	ds.0.dataset_type	0 - Elenco item; 1 - *(predefinito)* Modelli di item.
Colore	1	ds.0.color.0	Codice colore esadecimale (ad esempio, `FF0000`). Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco item"
Asse X	4	ds.0.x_axis_itemids.0	ID item. Quando si configura il widget in una dashboard template, devono essere impostati solo gli item configurati nel template. Nota: Per configurare più item, creare un oggetto campo widget dashboard per ciascun item con un numero incrementato nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco item" e *Asse X (Widget)* non è impostato
Asse X (Widget)	1	ds.0.x_axis_itemids.0	ID item: `ABCDE._itemid` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli item. Nota: Per configurare più widget, creare un oggetto campo widget dashboard per ciascun widget con un numero incrementato nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco item" e *Asse X* non è impostato
Asse Y	4	ds.0.y_axis_itemids.0	ID item. Quando si configura il widget in una dashboard template, devono essere impostati solo gli item configurati nel template. Nota: Per configurare più item, creare un oggetto campo widget dashboard per ciascun item con un numero incrementato nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco item" e *Asse Y (Widget)* non è impostato

Parametro	type	name	value
Asse Y (Widget)	1	ds.0.y_axis_itemids.0	Inserisci l'ID item: `ABCDE._itemid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per gli item. Nota: Per configurare più widget, creare un oggetto campo widget dashboard per ciascun widget con un numero incrementato nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Elenco item" e *Asse Y* non è impostato
Sostituisci host	1	ds.0.override_hostid.0	`ABCDE._hostid` - imposta un widget compatibile (con il parametro "Reference" impostato su "ABCDE") come origine dati per gli host; `DASHBOARD._hostid` - imposta il selettore host della dashboard come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Set di dati: Modelli di item			
Colore	1	ds.0.color	Codice colore esadecimale (ad esempio, `FF0000`). Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su "Modelli di item" e *Palette colori* non è impostata
Palette colori	0	ds.0.color_palette	Indice della palette colori. Intervallo di valori possibili: 0-11. Predefinito: 0. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su "Modelli di item" e *Colore* non è impostato
Modelli host	1	ds.0.hosts.0	Nome o modello di host (ad esempio, "Zabbix*"). Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Modelli di item" Questo parametro non è supportato se si configura il widget in una dashboard template.
Modelli item asse X	1	ds.0.x_axis_items.0	Nome o modello di item (ad esempio, "*: Number of processed *values per second"). Quando si configura il widget in una dashboard template, devono essere impostati solo i modelli per gli item configurati nel template. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Modelli di item"
Modelli item asse Y	1	ds.0.y_axis_items.0	Nome o modello di item (ad esempio, "*: Number of processed *values per second"). Quando si configura il widget in una dashboard template, devono essere impostati solo i modelli per gli item configurati nel template. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su "Modelli di item"

Parametro	type	name	value
Gruppi host	2	groupids.0	ID gruppo host. Nota: Per configurare più gruppi host, creare un oggetto campo widget dashboard per ciascun gruppo host con un numero incrementato nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Sostituisci host	1	ds.0.override_hostid._reference	`ABCDE._hostid` - imposta un widget compatibile (con il parametro "Reference" impostato su "`ABCDE`") come origine dati per gli host; `DASHBOARD._hostid` - imposta il selettore host della dashboard come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Tag host			
Tipo di valutazione	0	evaltype_host	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget in una dashboard template.
Nome tag	1	host_tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Operatore	0	host_tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Valore tag	1	host_tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Marcatore	0	ds.0.marker	0 - *(predefinito)* Ellisse; 1 - Quadrato; 2 - Triangolo; 3 - Rombo; 4 - Stella; 5 - Croce.
Dimensione marcatore	0	ds.0.marker_size	0 - *(predefinito)* Piccolo (6px); 1 - Medio (9px); 2 - Grande (15px).
Spostamento temporale	1	ds.0.timeshift	Stringa temporale valida (ad esempio, 3600, 1h, ecc.). È possibile usare i suffissi temporali. Sono consentiti anche valori negativi. Predefinito: "" (vuoto).
Intervallo di aggregazione	1	ds.0.aggregate_interval	Stringa temporale valida (ad esempio, 3600, 1h, ecc.). È possibile usare i suffissi temporali. Predefinito: 15m.
Funzione di aggregazione	0	ds.0.aggregate_function	0 - non utilizzata; 1 - min; 2 - max; 3 - *(predefinito)* avg; 4 - count; 5 - sum; 6 - first; 7 - last.

Opzioni di visualizzazione

I seguenti parametri sono supportati per la configurazione di *Opzioni di visualizzazione*.

Parametro	type	name	value
Selezione dei dati storici	0	source	0 - *(predefinito)* Auto; 1 - Storico; 2 - Trend.
Nomi host nelle etichette	0	show_hostnames	0 - *(predefinito)* Auto; 1 - Mostra; 2 - Nascondi. Questo parametro non è supportato se si configura il widget in una dashboard di template.

Periodo di tempo

I seguenti parametri sono supportati per la configurazione di *Periodo di tempo*.

Parametro	type	name	value
Periodo di tempo	1	time_period._referenceDASHBOARD._timeperiod	DASHBOARD._timeperiod - imposta il selettore del periodo di tempo della dashboard come origine dati; ABCDE._timeperiod - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati. Predefinito: DASHBOARD._timeperiod In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.
Da	1	time_period.from	Stringa temporale valida nella sintassi temporale assoluta (YYYY-MM-DD hh:mm:ss) o relativa (now, now/d, now/w-1w, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se time_period.to è impostato
A	1	time_period.to	Stringa temporale valida nella sintassi temporale assoluta (YYYY-MM-DD hh:mm:ss) o relativa (now, now/d, now/w-1w, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se time_period.from è impostato

Assi

I seguenti parametri sono supportati per la configurazione di *Assi*.

Parametro	type	name	value
Asse X	0	x_axis	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Asse Y	0	y_axis	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Min	1	x_axis_min y_axis_min	Qualsiasi valore numerico. Predefinito: "" (vuoto).
Max	1	x_axis_max y_axis_max	Qualsiasi valore numerico. Predefinito: "" (vuoto).
Unità (tipo)	0	x_axis_units y_axis_units	0 - *(predefinito)* Automatico; 1 - Statico.
Unità (valore)	1	x_axis_static_units y_axis_static_units	Qualsiasi valore stringa. Predefinito: "" (vuoto).

Legenda

I seguenti parametri sono supportati per la configurazione della *Legenda*.

Parametro	type	name	value
Mostra legenda	0	legend	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Mostra funzione di aggregazione	0	legend_aggregation	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitato"

Parametro	type	name	value
Righe	0	legend_lines_mode	0 - *(predefinito)* Fisso; 1 - Variabile. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitato”
Numero di righe/ Numero massimo di righe	0	legend_lines	Intervallo di valori possibili: 1-10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitato”
Numero di colonne	0	legend_columns	Intervallo di valori possibili: 1-4. Predefinito: 4. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitato”

Soglie

I seguenti parametri sono supportati per la configurazione delle *Soglie*.

Note:
Il numero nel nome della proprietà *Thresholds* (ad esempio, thresholds.0.color) fa riferimento alla posizione della soglia in un elenco, ordinato in ordine crescente. Tuttavia, se le soglie sono configurate in un ordine diverso, i valori verranno ordinati in ordine crescente dopo l'aggiornamento della configurazione del widget nel frontend di Zabbix (ad esempio, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametro	type	name	value
Interpolazione del colore	0	interpolation	0 - *(predefinito)* Disabilitata; 1 - Abilitata.
Colore	1	thresholds.0.color	Codice colore esadecimale (ad esempio, FF0000).
Asse X	1	thresholds.0.x_axis_threshold	Qualsiasi valore numerico. Sono supportati i suffissi (ad esempio, ”1d”, ”2w”, ”4K”, ”8G”).
Asse Y	1	thresholds.0.y_axis_threshold	Qualsiasi valore numerico. Sono supportati i suffissi (ad esempio, ”1d”, ”2w”, ”4K”, ”8G”).

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Scatter plot*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Grafico a dispersione*

Configurare un widget *Grafico a dispersione* nel modo seguente:

- 5 set di dati di tipo ”Pattern di item”, ciascuno con aggregazione predefinita (avg, 15m) e marcatori diversi.
- La legenda visualizza gli item configurati in una singola riga e due colonne.
- Le soglie sono configurate per colorare di rosso un marcatore quando il suo valore raggiunge 80 su qualsiasi asse.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

26 Report SLA

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di

configurare il widget *Report SLA* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Report SLA*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati dal widget *Report SLA*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - *(predefinito)* Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
SLA	10	slaid.0	ID SLA. Comportamento del parametro: - *obbligatorio*
Servizio	9	serviceid.0	ID del servizio.
Mostra periodi	0	show_periods	I valori possibili vanno da 1 a 100. Predefinito: 20.
Da	1	date_from	Stringa di data valida nel formato `YYYY-MM-DD`. Sono supportate le date relative con i modificatori `d`, `w`, `M`, `y` (ad esempio `now`, `now/d`, `now/w-1w`, ecc.).
A	1	date_to	Stringa di data valida nel formato `YYYY-MM-DD`. Sono supportate le date relative con i modificatori `d`, `w`, `M`, `y` (ad esempio `now`, `now/d`, `now/w-1w`, ecc.).

Esempi

Gli esempi seguenti hanno lo scopo esclusivo di descrivere la configurazione degli oggetti campo del widget della dashboard per il widget *Report SLA*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Report SLA*

Configurare un widget *Report SLA* che visualizza il report SLA per il servizio "2" dell'SLA "4" per il periodo degli ultimi 30 giorni.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

27 Informazioni di sistema

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Informazioni di sistema* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Informazioni di sistema*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *System Information*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - *(predefinito)* 15 minuti.
Show	0	info_type	0 - *(predefinito)* Statistiche di sistema; 1 - Nodi ad alta disponibilità.
Show software update check details	0	show_software_update_check_details	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Parameter behavior: - *supported* se `AllowSoftwareUpdateCheck` nella configurazione del server Zabbix è abilitato e *Show* è impostato su "System stats"

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *System information*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Informazioni di sistema*

Configurare un widget *Informazioni di sistema* che visualizzi le statistiche di sistema con un intervallo di aggiornamento di 10 minuti e il controllo degli aggiornamenti software abilitato.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

28 Host principali

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *Top hosts* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Top hosts*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Top hosts*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: per configurare più gruppi di host, creare un oggetto campo del widget della dashboard per ogni gruppo di host, incrementando il numero nel nome della proprietà. Questo parametro non è supportato se si configura il widget su una dashboard di template.
Gruppi di host (widget)	1	groupids._reference	Invece dell'ID del gruppo di host: `ABCDE._hostgroupids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi di host. Questo parametro non è supportato se si configura il widget su una dashboard di template.
Host	3	hostids.0	ID dell'host. Nota: per configurare più host, creare un oggetto campo del widget della dashboard per ogni host, incrementando il numero nel nome della proprietà. Per più host, il parametro *Gruppi di host* deve essere completamente non configurato oppure configurato con almeno un gruppo di host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget su una dashboard di template.
Host (widget/dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - impostare il selettore di host della dashboard come origine dati per gli host; `ABCDE._hostids` - impostare un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget su una dashboard di template.
Tag degli host			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget su una dashboard di template.
Nome del tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato se si configura il widget su una dashboard di template.

Parametro	tipo	nome	valore
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato se si configura il widget su una dashboard di template.
Valore del tag	1	tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag degli host* Questo parametro non è supportato se si configura il widget su una dashboard di template.
Mostra gli host in manuten-zione	0	maintenance	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colonne (vedere sotto)			
Ordina per	0	column	Valore numerico della colonna tra quelle configurate.
Ordinamento	0	order	2 - *(predefinito)* Primi N; 3 - Ultimi N.
Limite host	0	show_lines	I valori possibili sono compresi tra 1 e 1000. Predefinito: 10. Questo parametro non è supportato se si configura il widget su una dashboard di template.

Colonne

Le colonne hanno parametri comuni e parametri aggiuntivi a seconda della configurazione del parametro *Data*.

Note:
Per tutti i parametri relativi alle colonne, il numero nel nome della proprietà (ad esempio columns.0.name) fa riferimento a una colonna per la quale il parametro è configurato.

I seguenti parametri sono supportati per tutte le colonne.

Parametro	type	name	value
Nome	1	columns.0.name	Qualsiasi valore stringa. Comportamento del parametro: - *obbligatorio*

Parametro	type	name	value
Dati	0	columns.0.data	1 - Valore dell'item; 2 - Nome dell'host; 3 - Testo. Comportamento del parametro: - *obbligatorio*
Colore di base	1	columns.0.base_color	Codice colore esadecimale (ad esempio FF0000). Comportamento del parametro: - *obbligatorio*

Valore dell'item

I seguenti parametri sono supportati se *Dati* è impostato su "Valore dell'item".

Note:
Il primo numero nel nome della proprietà *Soglie* (ad esempio, columnsthresholds.0.color.0) fa riferimento alla colonna per la quale sono configurate le soglie, mentre il secondo numero fa riferimento alla posizione della soglia in un elenco ordinato in modo crescente. Tuttavia, se le soglie sono configurate in un ordine diverso, i valori verranno ordinati in modo crescente dopo l'aggiornamento della configurazione del widget nel frontend di Zabbix (ad esempio, `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Parametro	tipo	nome	valore
Item	1	columns.0.item	Nome dell'item valido. Quando si configura il widget in una dashboard del template, devono essere impostati solo gli item configurati nel template.
Visualizza il valore dell'item come	0	columns.0.display_value_as	0 - *(predefinito)* Numerico; 1 - Testo; 2 - Binario.
Visualizzazione	0	columns.0.display	1 - *(predefinito)* Così com'è; 2 - Barra; 3 - Indicatori; 6 - Sparkline. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico"
Min	1	columns.0.min	Qualsiasi valore numerico. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Barra" o "Indicatori"
Max	1	columns.0.max	Qualsiasi valore numerico. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Barra" o "Indicatori"
Posizioni decimali	0	columns.0.decimal_places	I valori possibili vanno da 0 a 10. Predefinito: 2. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico"
Sparkline			

Parametro	tipo	nome	valore
Larghezza	0	columns.0.sparkline.width	I valori possibili vanno da 0 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline"
Riempimento	0	columns.0.sparkline.fill	I valori possibili vanno da 0 a 10. Predefinito: 3. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline"
Colore	1	columns.0.sparkline.color	Codice colore esadecimale (ad esempio, `FF0000`). Predefinito: `42A5F5`. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline"
Periodo di tempo	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro `reference` uguale a `ABCDE`) come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline"
Da	1	columns.0.sparkline.time_period.from	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline" - *obbligatorio* se `columns.0.sparkline.time_period.to` è impostato
A	1	columns.0.sparkline.time_period.to	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline" - *obbligatorio* se `columns.0.sparkline.time_period.from` è impostato
Dati storici	0	columns.0.sparkline.history	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Trend. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico" e *Visualizzazione* è impostato su "Sparkline"

Soglie

Parametro	tipo	nome	valore
Colore	1	columnsthresholds.0.color	Codice colore esadecimale (ad esempio, FF0000). Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico"
Soglia	1	columnsthresholds.0.threshold	Qualsiasi valore stringa. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numerico"
Evidenziazioni			
Colore	1	columns.0.highlights.color	Codice colore esadecimale (ad esempio, FF0000). Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Testo"
Pattern	1	columns.0.highlights.pattern	Qualsiasi valore stringa. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Testo"
Mostra miniatura	0	columns.0.show_thumbnail	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Binario"
Funzione di aggregazione	0	columns.0.aggregate_function	0 - *(predefinito)* non utilizzata; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Periodo di tempo	1	columns.0.time_period.reference	DASHBOARD - `_timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro `reference` uguale a `ABCDE`) come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*. Comportamento del parametro: - *supportato* se *Funzione di aggregazione* è impostato su "min", "max", "avg", "count", "sum", "first", "last"
Da	1	columns.0.time_period.from	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato e *Funzione di aggregazione* è impostato su "min", "max", "avg", "count", "sum", "first", "last" - *obbligatorio* se `columns.0.time_period.to` è impostato

Parametro	tipo	nome	valore
A	1	columns.0.time_period.to	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato e *Funzione di aggregazione* è impostato su ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obbligatorio* se `columns.0.time_period.from` è impostato
Dati storici	0	columns.0.history	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Trend. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su ”Numerico”
Riferimento	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Testo

I seguenti parametri sono supportati se *Data* è impostato su ”Text”.

Parameter	type	name	value
Text	1	columns.0.text	Qualsiasi valore stringa, incluse le macro. Macro supportate: `{HOST.*}`, `{INVENTORY.*}`. Comportamento del parametro: - *obbligatorio* se *Data* è impostato su ”Text”

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Top hosts*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Top hosts*

Configurare un widget *Top hosts* che visualizzi i principali host in base all'utilizzo della CPU nel gruppo host ”4”. Inoltre, configurare le seguenti colonne personalizzate: ”Nome host”, ”Utilizzo CPU in %”, ”Media 1m”, ”Media 5m”, ”Media 15m”, ”Processi”.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

29 Top item

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Top items* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Top items*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Top items*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi host	2	groupids.0	ID del gruppo host. Nota: Per configurare più gruppi host, creare un oggetto campo widget dashboard per ciascun gruppo host con un numero incrementale nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID del gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host	3	hostids.0	ID dell'host. Nota: Per configurare più host, creare un oggetto campo widget dashboard per ciascun host con un numero incrementale nel nome della proprietà. Per più host, il parametro *Gruppi host* non deve essere configurato oppure deve essere configurato con almeno un gruppo host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host (Widget/Dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Tag host			
Tipo di valutazione	0	host_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget in una dashboard template.
Nome tag	1	host_tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Operatore	0	host_tags.0.operator	0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.

Parametro	type	name	value
Valore tag	1	host_tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale.
Mostra problemi Item (vedere sotto)	0	show_problems	0 - Tutti; 1 - *(predefinito)* Non soppressi; 2 - Nessuno.

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati per il widget *Top items*.

Parametro	type	name	value
Ordinamento degli host			
Ordina per	0	host_ordering_order_by	0 - *(predefinito)* Nome dell'host; 3 - Valore dell'item.
Pattern degli item	1	host_ordering_item.0	Nome o pattern dell'item (ad esempio, "*: Numero di *valori elaborati al secondo"). Nota: per configurare più pattern di item, creare un oggetto campo del widget dashboard per ogni pattern di item, utilizzando un numero incrementato nel nome della proprietà. Quando si configura il widget in una dashboard del template, impostare solo i pattern degli item configurati nel template. Comportamento del parametro: - *obbligatorio* se *Ordina per* è impostato su "Valore dell'item"
Ordine	0	host_ordering_order	2 - *(predefinito)* Primi N; 3 - Ultimi N.
Limite	0	host_ordering_limit	I valori possibili sono compresi tra 1 e 100. Predefinito: 10.
Ordinamento degli item			
Ordina per	0	item_ordering_order_by	1 - Host; 2 - Nome dell'item; 3 - *(predefinito)* Valore dell'item.
Pattern degli host	1	item_ordering_host.0	Nome o pattern dell'host. Nota: per configurare più pattern di host, creare un oggetto campo del widget dashboard per ogni pattern di host, utilizzando un numero incrementato nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Ordina per* è impostato su "Host"

Parametro	type	name	value
Ordine	0	item_ordering_order	2 - *(predefinito)* Primi N; 3 - Ultimi N.
Limite	0	item_ordering_limit	I valori possibili sono compresi tra 1 e 100. Predefinito: 10.

Colonne

Note:
Il primo numero nel nome della proprietà (ad es., columns.0.items.0, columns.0.item_tags_evaltype) rappresenta la colonna specifica, mentre il secondo numero, se presente, rappresenta l'entità configurata (ad es., pattern di item, tag).

Parametro	tipo	nome	valore
Pattern di item	1	columns.0.items.0	Nome o pattern dell'item (ad es., "*: Numero di *valori elaborati al secondo"). Nota: per configurare più pattern di item, creare un oggetto campo del widget dashboard per ogni pattern di item, incrementando il secondo numero nel nome della proprietà. Quando si configura il widget in una dashboard di template, impostare solo i pattern degli item configurati nel template. Comportamento del parametro: - *obbligatorio*
Tag degli item			
Tipo di valutazione	0	columns.0.item_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome del tag	1	columns.0.item_tags.0.tag	Qualsiasi valore stringa. Nota: il secondo numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag degli item*
Operatore	0	columns.0.item_tags.0.operator	**0** - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il secondo numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag degli item*
Valore del tag	1	columns.0.item_tags.0.value	Qualsiasi valore stringa. Nota: il secondo numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag degli item*
Colore di base	1	columns.0.base_color	Codice colore esadecimale (ad es., FF0000).

Parametro	tipo	nome	valore
Visualizza valore come	0	columns.0.display_value_as	1 - *(predefinito)* Numerico; 2 - Testo.
Visualizzazione	0	columns.0.display	1 - *(predefinito)* Così com'è; 2 - Barra; 3 - Indicatori; 6 - Sparkline.
Min	1	columns.0.min	Qualsiasi valore numerico. Sono supportati i suffissi (ad es., "1d", "2w", "4K", "8G"). Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico" e *Visualizzazione* è impostata su "Barra" o "Indicatori"
Max	1	columns.0.max	Qualsiasi valore numerico. Sono supportati i suffissi (ad es., "1d", "2w", "4K", "8G"). Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico" e *Visualizzazione* è impostata su "Barra" o "Indicatori"
Sparkline			
Larghezza	0	columns.0.sparkline.width	I valori possibili sono compresi tra 0 e 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Riempimento	0	columns.0.sparkline.fill	I valori possibili sono compresi tra 0 e 10. Predefinito: 3. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Colore	1	columns.0.sparkline.color	Codice colore esadecimale (ad es., `FF0000`). Predefinito: `42A5F5`. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Intervallo di tempo	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - imposta il selettore dell'intervallo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro `reference` uguale a `ABCDE`) come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare l'intervallo di tempo solo nei parametri *Da* e *A*. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Da	1	columns.0.sparkline.time_period.from	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline" - *obbligatorio* se `columns.0.sparkline.time_period.to` è impostato

Parametro	tipo	nome	valore
A	1	columns.0.sparkline.time_period.to	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su ”Sparkline” - *obbligatorio* se `columns.0.sparkline.time_period.from` è impostato
Dati storici	0	columns.0.sparkline.history	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Trend. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su ”Sparkline”
Soglie			
Colore	1	columns.0.thresholds.color	Codice colore esadecimale (ad es., `FF0000`). Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su ”Numerico”
Soglia	1	columns.0.thresholds.value	Un valore numerico. Sono supportati i suffissi (ad es., ”1d”, ”2w”, ”4K”, ”8G”). Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su ”Numerico”
Evidenziazioni			
Evidenziazione	1	columns.0.highlights.color	Codice colore esadecimale (ad es., `FF0000`). Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su ”Testo”
Soglia	1	columns.0.highlights.pattern	Qualsiasi espressione regolare. Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su ”Testo”
Posizioni decimali	0	columns.0.decimal_places	I valori possibili sono compresi tra 0 e 10. Predefinito: 2.
Configurazione avanzata			
Funzione di aggregazione	0	columns.0.aggregate_function	Se *Visualizza valore come* è impostato su ”Numerico”: 0 - *(predefinito)* non utilizzata; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last. Se *Visualizza valore come* è impostato su ”Testo”: 0 - *(predefinito)* non utilizzata; 4 - count; 6 - first; 7 - last.

Parametro	tipo	nome	valore
Intervallo di tempo	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - imposta il selettore dell'intervallo di tempo della dashboard come origine dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro `reference` uguale a `ABCDE`) come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare l'intervallo di tempo solo nei parametri *Da* e *A*. Comportamento del parametro: - *supportato* se *Funzione di aggregazione* è impostata su "min", "max", "avg", "count", "sum", "first", "last"
Da	1	columns.0.time_period.from	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Intervallo di tempo* non è impostato e *Funzione di aggregazione* è impostata su "min", "max", "avg", "count", "sum", "first", "last" - *obbligatorio* se `columns.0.time_period.to` è impostato
A	1	columns.0.time_period.to	Stringa temporale valida nella sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Intervallo di tempo* non è impostato e *Funzione di aggregazione* è impostata su "min", "max", "avg", "count", "sum", "first", "last" - *obbligatorio* se `columns.0.time_period.from` è impostato
Dati storici	0	columns.0.history	0 - *(predefinito)* Automatico; 1 - Cronologia; 2 - Trend. Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico"
Aggrega	0	columns.0.aggregate_column_name	0 - *(predefinito)* Ogni item 1 - Combinati. Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico" e *Visualizzazione* è impostata su "Così com'è", "Barra" o "Indicatori".
Funzione di aggregazione combinata	0	columns.0.column_aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum *(predefinito)*. Comportamento del parametro: - *obbligatorio* se si configura *Aggrega*.
Nome della colonna combinata	1	columns.0.combined_column_name	Qualsiasi valore stringa. Comportamento del parametro: - *obbligatorio* se si configura *Aggrega*.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Top items*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Top items*

Configurare un widget *Top items* che visualizzi i dati per l'host "10084" e solo per gli item per i quali il tag con nome "component" contiene il valore "cpu". Inoltre, visualizzare i dati con gli host posizionati in alto e utilizzare una barra indicatrice colorata per la rappresentazione delle celle.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

30 trigger principali

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Top triggers* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Top triggers*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati dal widget *Top triggers*.

Parametro	tipo	nome	valore
Intervallo di aggiornamento	0	rf_rate	0 - *(predefinito)* Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.

Parametro	tipo	nome	valore
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: per configurare più gruppi di host, creare un oggetto campo del widget dashboard per ogni gruppo di host, incrementando il numero nel nome della proprietà. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Host	3	hostids.0	ID dell'host. Nota: per configurare più host, creare un oggetto campo del widget dashboard per ogni host, incrementando il numero nel nome della proprietà. Per più host, il parametro *Gruppi di host* deve essere non configurato oppure configurato con almeno un gruppo di host a cui appartengono gli host configurati. Questo parametro non è supportato quando si configura il widget in una dashboard di template.
Problema	1	problem	Nome dell'evento del problema (senza distinzione tra maiuscole e minuscole, nome completo o parte di esso).
Gravità	0	severities.0	Gravità dei trigger. 0 - Non classificato; 1 - Informazioni; 2 - Avviso; 3 - Media; 4 - Alta; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: per configurare più valori, creare un oggetto campo del widget dashboard per ogni valore, incrementando il numero nel nome della proprietà.
Tag dei problemi			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome del tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dei problemi*
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dei problemi*

Parametro	tipo	nome	valore
Valore del tag	1	tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine del tag nell'elenco di valutazione dei tag.
Periodo di tempo	1	time_period._referenceDASHBOARD	Comportamento del parametro: - *obbligatorio* se si configura *Tag dei problemi* `DASHBOARD._timeperiod` - imposta il selettore del periodo di tempo della dashboard come origine dei dati; `ABCDE._timeperiod` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dei dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.
Da	1	time_period.from	Stringa temporale valida in sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
A	1	time_period.to	Stringa temporale valida in sintassi assoluta (`YYYY-MM-DD hh:mm:ss`) o relativa (`now`, `now/d`, `now/w-1w`, ecc.). Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.from` è impostato
Limite dei trig-ger	0	show_lines	I valori possibili sono compresi tra 1 e 100. Predefinito: 10.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Top triggers*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Top triggers*

Configurare un widget *Top triggers* che visualizza i primi 5 trigger per il gruppo host "4" con il conteggio di tutti i problemi per ciascun trigger. Il widget visualizza solo i trigger con gravità "Warning", "Average", "High" o "Disaster", e i problemi che hanno un tag con nome "scope" che contiene i valori "performance" o "availability", oppure "capacity".

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

31 Panoramica dei trigger

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Panoramica dei trigger* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Panoramica dei trigger*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Panoramica trigger*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Mostra	0	show	1 - *(predefinito)* Problemi recenti; 2 - Qualsiasi; 3 - Problemi.
Gruppi host	2	groupids.0	ID del gruppo host. Nota: per configurare più gruppi host, creare un oggetto campo widget dashboard per ciascun gruppo host con un numero incrementato nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi host (Widget)	1	groupids._reference	Invece dell'ID del gruppo host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Riferimento* impostato su "`ABCDE`") come origine dati per i gruppi host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host	3	hostids.0	ID dell'host. Nota: per configurare più host, creare un oggetto campo widget dashboard per ciascun host con un numero incrementato nel nome della proprietà. Per più host, il parametro *Gruppi host* non deve essere configurato affatto oppure deve essere configurato con almeno un gruppo host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host (Widget/Dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostids` - imposta un widget compatibile (con il parametro *Riferimento* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Tag problema			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.

Parametro	type	name	value
Nome tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine dei tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag problema*
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: il numero nel nome della proprietà fa riferimento all'ordine dei tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag problema*
Valore tag	1	tags.0.value	Qualsiasi valore stringa. Nota: il numero nel nome della proprietà fa riferimento all'ordine dei tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag problema*
Mostra problemi soppressi	0	show_suppressed	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Trigger overview*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Panoramica trigger*

Configurare un widget *Panoramica trigger* che visualizzi gli stati dei trigger per tutti i gruppi host che hanno trigger con un tag con nome "scope" e contenente il valore "availability".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Trigger overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
```

```
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "availability"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

32 URL

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget della dashboard consentono di configurare il widget *URL* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *URL*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *URL*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - *(predefinito)* Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
URL	1	url	Stringa URL valida. Comportamento del parametro: - *obbligatorio*
Sostituisci host	1	override_hostid._reference	`ABCDE._hostid` - imposta un widget compatibile (con il parametro *Reference* impostato su "ABCDE") come origine dati per gli host; `DASHBOARD._hostid` - imposta il selettore host della dashboard come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *URL*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *URL*

Configurare un widget *URL* che visualizza la home page del manuale di Zabbix.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

33 Monitoraggio web

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti dei campi del widget dashboard consentono di configurare il widget *Monitoraggio web* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono convalidate durante la creazione o l'aggiornamento di una dashboard. Ciò consente agli utenti di modificare i widget integrati e creare widget personalizzati, ma introduce anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o il corretto aggiornamento del widget *Monitoraggio web*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Monitoraggio web*.

Parametro	type	name	value
Intervallo di aggiornamento	0	rf_rate	0 - Nessun aggiornamento; 10 - 10 secondi; 30 - 30 secondi; 60 - *(predefinito)* 1 minuto; 120 - 2 minuti; 600 - 10 minuti; 900 - 15 minuti.
Gruppi di host	2	groupids.0	ID del gruppo di host. Nota: Per configurare più gruppi di host, creare un oggetto campo widget dashboard per ciascun gruppo di host con un numero incrementato nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Gruppi di host (Widget)	1	groupids._reference	Invece dell'ID del gruppo di host: `ABCDE._hostgroupids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per i gruppi di host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Escludi gruppi di host	2	exclude_groupids.0	ID del gruppo di host. Nota: Per escludere più gruppi di host, creare un oggetto campo widget dashboard per ciascun gruppo di host con un numero incrementato nel nome della proprietà. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host	3	hostids.0	ID dell'host. Nota: Per configurare più host, creare un oggetto campo widget dashboard per ciascun host con un numero incrementato nel nome della proprietà. Per più host, il parametro *Gruppi di host* non deve essere configurato affatto oppure deve essere configurato con almeno un gruppo di host a cui appartengono gli host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.
Host (Widget/Dashboard)	1	hostids._reference	Invece dell'ID dell'host: `DASHBOARD.hostids` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostids` - imposta un widget compatibile (con il parametro *Reference* impostato su "`ABCDE`") come origine dati per gli host. Questo parametro non è supportato se si configura il widget in una dashboard template.
Tag dello scenario			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome tag	1	tags.0.tag	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine dei tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dello scenario*

Parametro	type	name	value
Operatore	0	tags.0.operator	0 - Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste. Nota: Il numero nel nome della proprietà fa riferimento all'ordine dei tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dello scenario*
Valore tag	1	tags.0.value	Qualsiasi valore stringa. Nota: Il numero nel nome della proprietà fa riferimento all'ordine dei tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag dello scenario*
Mostra host in manuten-zione	0	maintenance	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Reference	1	reference	Qualsiasi valore stringa composto da 5 caratteri (ad esempio, `ABCDE` o `JBPNL`). Questo valore deve essere univoco all'interno della dashboard a cui appartiene il widget. Comportamento del parametro: - *obbligatorio*

Esempi

Gli esempi seguenti hanno lo scopo di descrivere solo la configurazione degli oggetti campo del widget della dashboard per il widget *Web monitoring*. Per ulteriori informazioni sulla configurazione di una dashboard, vedere `dashboard.create`.

Configurazione di un widget *Monitoraggio web*

Configurare un widget *Monitoraggio web* che visualizza un riepilogo dello stato degli scenari di monitoraggio web attivi per il gruppo host "4".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Campo del widget della dashboard
- `dashboard.create`
- `dashboard.update`

Dashboard del template

Questa classe è progettata per funzionare con i dashboard del template.

Riferimenti agli oggetti:

- Dashboard del template
- Pagina del dashboard del template
  - Widget del dashboard del template
    - Campo del widget del dashboard del template

Metodi disponibili:

- templatedashboard.create - crea nuovi dashboard del template
- templatedashboard.delete - elimina dashboard del template
- templatedashboard.get - recupera dashboard del template
- templatedashboard.update - aggiorna dashboard del template

Oggetto template della dashboard

I seguenti oggetti sono direttamente correlati all'API `templateashboard`.

Dashboard del template

L'oggetto dashboard del template ha le seguenti proprietà.

Property	Type	Description
dashboardid	ID	ID del dashboard del template. Property behavior: - *read-only* - *required* for update operations
name	string	Nome del dashboard del template. Property behavior: - *required* for create operations
templateid	ID	ID del template a cui appartiene il dashboard. Property behavior: - *constant* - *required* for create operations
display_period	integer	Intervallo predefinito della presentazione automatica (in secondi). Possible values: 10, 30, 60, 120, 600, 1800, 3600. Default: 30.
auto_start	integer	Avvio automatico della presentazione automatica. Possible values: 0 - non avviare automaticamente la presentazione automatica; 1 - *(default)* avviare automaticamente la presentazione automatica.
uuid	string	Identificatore univoco universale, usato per collegare i dashboard del template importati a quelli già esistenti. Generato automaticamente, se non specificato.

Pagina dashboard del template

L'oggetto pagina dashboard del template ha le seguenti proprietà.

Property	Type	Description
dashboard_pageid	ID	ID della pagina dashboard. Property behavior: - *read-only*
name	string	Nome della pagina dashboard. Default: stringa vuota.
display_period	integer	Intervallo della presentazione della dashboard (in secondi). Valori possibili: 0, 10, 30, 60, 120, 600, 1800, 3600. Default: 0 (verrà usato l'intervallo predefinito della presentazione).
widgets	array	Array degli oggetti template dashboard widget.

Widget dashboard del template

L'oggetto widget dashboard del template ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
widgetid	ID	ID del widget dashboard. Comportamento della proprietà: - *sola lettura*

Proprietà	Tipo	Descrizione
type	string	Tipo del widget dashboard. Valori possibili: actionlog - Log delle azioni; clock - Orologio; discovery - Stato del discovery; favgraphs - Grafici preferiti; favmaps - Mappe preferite; gauge - Indicatore; graph - Grafico (classico); graphprototype - Prototipo di grafico; honeycomb - Nido d'ape; hostavail - Disponibilità host; hostcard - Scheda host; hostnavigator - Navigatore host; itemcard - Scheda item; itemnavigator - Navigatore item; item - Valore item; map - Mappa; navtree - Albero di navigazione delle mappe; piechart - Grafico a torta; plaintext - Testo semplice; problemhosts - Host con problemi; problems - Problemi; problemsbysv - Problemi per gravità; scatterplot - Grafico a dispersione; slareport - Report SLA; svggraph - Grafico; systeminfo - Informazioni di sistema; tophosts - Top host; topitems - Top item; toptriggers - Top trigger; trigover - Panoramica trigger; url - URL; web - Monitoraggio web. Comportamento della proprietà: - *obbligatoria*
name	string	Nome personalizzato del widget.
x	integer	Posizione orizzontale dal lato sinistro della dashboard. I valori possibili vanno da 0 a 71.
y	integer	Posizione verticale dalla parte superiore della dashboard. I valori possibili vanno da 0 a 63.
width	integer	Larghezza del widget. I valori possibili vanno da 1 a 72.
height	integer	Altezza del widget. I valori possibili vanno da 1 a 64.
view_mode	integer	Modalità di visualizzazione del widget. Valori possibili: 0 - *(predefinita)* visualizzazione predefinita del widget; 1 - con intestazione nascosta;
fields	array	Array degli oggetti campo del widget dashboard del template.

Campo widget della dashboard del template

L'oggetto campo widget della dashboard del template ha le seguenti proprietà.

Proprietà	Type	Descrizione
type	integer	Tipo del campo del widget. Valori possibili: 0 - Intero; 1 - Stringa; 4 - Item; 5 - Prototipo di item; 6 - Grafico; 7 - Prototipo di grafico; 8 - Mappa; 9 - Servizio; 10 - SLA; 11 - Utente; 12 - Azione; 13 - Tipo di supporto. Comportamento della proprietà: - *required*
name	string	Nome del campo del widget. Valori possibili: vedere Campi del widget della dashboard. Si noti che alcuni parametri relativi all'host (ad esempio, *Gruppi host*, *Escludi gruppi host* e *Host* nel widget *Problems*, *Gruppi host* nel widget *Host availability*, ecc.) non sono disponibili quando si configura il widget in una dashboard del template. Questo perché le dashboard del template visualizzano dati solo dall'host a cui il template è collegato. Comportamento della proprietà: - *required*
value	mixed	Valore del campo del widget in base al tipo. Valori possibili: vedere Campi del widget della dashboard. Si noti che alcuni parametri relativi all'host (ad esempio, *Gruppi host*, *Escludi gruppi host* e *Host* nel widget *Problems*, *Gruppi host* nel widget *Host availability*, ecc.) non sono disponibili quando si configura il widget in una dashboard del template. Questo perché le dashboard del template visualizzano dati solo dall'host a cui il template è collegato. Comportamento della proprietà: - *required*

templatedashboard.create

Descrizione

`object templatedashboard.create(object/array templateDashboards)`

Questo metodo consente di creare nuovi dashboard del template.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Dashboard del template da creare.

Oltre alle proprietà standard del dashboard del template, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
pages	array	Pagine del dashboard del template da creare per il dashboard. Le pagine del dashboard saranno ordinate nello stesso ordine in cui sono specificate. Comportamento del parametro: - *obbligatorio*

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID del modello dashboard creato sotto la proprietà `dashboardids`. L'ordine degli ID restituiti corrisponde all'ordine dei modelli dashboard passati.

Esempi

Creazione di una dashboard del template

Creare una dashboard del template denominata "Graphs" con un widget Graph in una singola pagina della dashboard.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

Vedi anche

- Pagina dashboard del template
- Widget della dashboard del template
- Campo del widget della dashboard del template

Fonte

CTemplateDashboard::create() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Descrizione

`object templatedashboard.delete(array templateDashboardIds)`

Questo metodo consente di eliminare i dashboard del template.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

ID (`array`) dei modelli dashboard da eliminare.

Valori restituiti

(`object`) Restituisce un oggetto contenente gli ID dei modelli dashboard eliminati sotto la proprietà `dashboardids`.

Esempi

Eliminazione di più dashboard di template

Eliminare due dashboard di template.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateDashboard::delete() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

Descrizione

`integer/array templatedashboard.get(object parameters)`

Il metodo consente di recuperare i dashboard del template in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
dashboardids	ID/array	Restituisce solo i dashboard di template con gli ID specificati.
templateids	ID/array	Restituisce solo i dashboard di template che appartengono ai template specificati.
selectPages	query	Restituisce una proprietà `pages` con le pagine del dashboard di template, ordinate correttamente.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `dashboardid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero delle dashboard del template

Recupera tutte le dashboard del template con i widget per un template specificato.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Pagina della dashboard del template
- Widget della dashboard del template
- Campo del widget della dashboard del template

Fonte

CTemplateDashboard::get() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

Descrizione

`object templatedashboard.update(object/array templateDashboards)`

Questo metodo consente di aggiornare i dashboard del template esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

(object/array) Proprietà del dashboard del template da aggiornare.

La proprietà dashboardid deve essere specificata per ciascun dashboard, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate.

Oltre alle proprietà standard del dashboard del template, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
pages	array	Pagine del dashboard del template per sostituire le pagine del dashboard esistenti. Le pagine del dashboard vengono aggiornate tramite la proprietà dashboard_pageid. Le nuove pagine del dashboard verranno create per gli oggetti senza la proprietà dashboard_pageid e le pagine del dashboard esistenti verranno eliminate se non riutilizzate. Le pagine del dashboard saranno ordinate nello stesso ordine in cui sono specificate. Verranno aggiornate solo le proprietà specificate delle pagine del dashboard. Per la proprietà pages è richiesto almeno un oggetto pagina del dashboard.

Valori restituiti

(object) Restituisce un oggetto contenente gli ID dei modelli dashboard aggiornati sotto la proprietà dashboardids.

Esempi

Rinominare una dashboard di template

Rinominare una dashboard di template in "Performance graphs".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

Aggiornamento delle pagine della dashboard del template

Rinominare la prima pagina della dashboard, sostituire i widget nella seconda pagina della dashboard e aggiungere una nuova pagina come terza. Eliminare tutte le altre pagine della dashboard.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Vedi anche

- Widget dashboard del template
- Campo del widget dashboard del template

Fonte

CTemplateDashboard::update() in *ui/include/classes/api/services/CTemplateDashboard.php*.

Directory utenti

Questa classe è progettata per funzionare con le directory utenti.

Riferimenti agli oggetti:

- Directory utenti
  - Mappature dei tipi di supporto
  - Mappature dei gruppi di provisioning

Metodi disponibili:

- userdirectory.create - crea una nuova directory utenti
- userdirectory.delete - elimina la directory utenti
- userdirectory.get - recupera la directory utenti
- userdirectory.update - aggiorna la directory utenti
- userdirectory.test - verifica la connessione della directory utenti

Oggetto directory utente

I seguenti oggetti sono direttamente correlati all'API `userdirectory`.

Directory utente

L'oggetto directory utente dispone delle proprietà seguenti.

Proprietà	Tipo	Descrizione
userdirectoryid	ID	ID della directory utente. Se una directory utente viene eliminata, il valore della proprietà `userdirectoryid` dell'oggetto User viene impostato su "0" per tutti gli utenti collegati alla directory utente eliminata. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
idp_type	integer	Tipo di protocollo di autenticazione utilizzato dall'identity provider per la directory utente. È possibile avere una sola directory utente di tipo SAML. Valori possibili: 1 - Directory utente di tipo LDAP; 2 - Directory utente di tipo SAML. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
group_name	string	Attributo della directory utente LDAP/SAML che contiene il nome del gruppo utilizzato per mappare i gruppi tra la directory utente LDAP/SAML e Zabbix. Esempio: *cn* Comportamento della proprietà: - *obbligatoria* se `provision_status` è impostato su "Enabled" e `saml_jit_status` dell'oggetto Authentication è impostato su "Enabled for configured SAML IdPs"
user_username	string	Attributo della directory utente LDAP/SAML (anche attributo SCIM se `scim_status` è impostato su "SCIM provisioning is enabled") che contiene il nome dell'utente, utilizzato come valore della proprietà `name` dell'oggetto User quando l'utente viene sottoposto a provisioning. Esempi: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Attributo della directory utente LDAP/SAML (anche attributo SCIM se `scim_status` è impostato su "SCIM provisioning is enabled") che contiene il cognome dell'utente, utilizzato come valore della proprietà `surname` dell'oggetto User quando l'utente viene sottoposto a provisioning. Esempi: *sn*, *surname*, *lastName*
provision_status	integer	Stato del provisioning della directory utente. Valori possibili: 0 - *(predefinito)* Disabilitato (il provisioning degli utenti creati da questa directory utente è disabilitato); 1 - Abilitato (il provisioning degli utenti creati da questa directory utente è abilitato; inoltre, lo stato del provisioning LDAP o SAML (`ldap_jit_status` o `saml_jit_status` dell'oggetto Authentication) deve essere abilitato).
provision_groups	array	Array di oggetti mappatura dei gruppi di provisioning per mappare il modello del gruppo utente LDAP/SAML al gruppo utenti e al ruolo utente Zabbix. Comportamento della proprietà: - *obbligatoria* se `provision_status` è impostato su "Enabled"

Proprietà	Tipo	Descrizione
provision_media	array	Array di oggetti mappatura dei tipi di supporto per mappare gli attributi dei supporti LDAP/SAML dell'utente (ad esempio, l'e-mail) ai supporti utente Zabbix per l'invio delle notifiche.
Proprietà specifiche di LDAP:		
name	string	Nome univoco della directory utente. Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su "User directory of type LDAP"
host	string	Nome host, IP o URI del server LDAP. L'URI deve contenere lo schema (`ldap://` o `ldaps://`), l'host e, facoltativamente, la porta. Esempi: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su "User directory of type LDAP"
port	integer	Porta del server LDAP. Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su "User directory of type LDAP"
base_dn	string	Percorso di base della directory utente LDAP agli account utente. Esempi: *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (per OpenLDAP) *DC=company,DC=com* (per Microsoft Active Directory) *uid=%{user},dc=example,dc=com* (per il binding diretto dell'utente; il segnaposto "*%{user}*" è obbligatorio) Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su "User directory of type LDAP"
search_attribute	string	Attributo della directory utente LDAP utilizzato per identificare l'account utente in base alle informazioni fornite nella richiesta di accesso. Esempi: *uid* (per OpenLDAP) *sAMAccountName* (per Microsoft Active Directory) Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su "User directory of type LDAP"

Proprietà	Tipo	Descrizione
bind_dn	string	Account del server LDAP utilizzato per il binding e la ricerca nel server LDAP. Per il binding diretto dell'utente e il binding anonimo, `bind_dn` deve essere vuoto. Esempi: *uid=ldap_search,ou=system* (per OpenLDAP) *CN=ldap_search,OU=user_group,DC=company,DC=com* (per Microsoft Active Directory) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
bind_password	string	Password LDAP dell'account utilizzato per il binding e la ricerca nel server LDAP. Per il binding diretto dell'utente e il binding anonimo, `bind_password` deve essere vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
description	string	Descrizione della directory utente. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
group_basedn	string	Percorso di base della directory utente LDAP ai gruppi; utilizzato per configurare un controllo dell'appartenenza dell'utente nella directory utente LDAP. Ignorato durante il provisioning di un utente se `group_membership` è impostato. Esempio: *ou=Groups,dc=example,dc=com* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"

Proprietà	Tipo	Descrizione
group_filter	string	Stringa di filtro per recuperare i gruppi della directory utente LDAP di cui l'utente è membro; utilizzata per configurare un controllo dell'appartenenza dell'utente nella directory utente LDAP. Ignorata durante il provisioning di un utente se `group_membership` è impostato. Segnaposto `group_filter` supportati: *%{attr}* - attributo di ricerca (sostituito dal valore della proprietà `search_attribute`); *%{groupattr}* - attributo del gruppo (sostituito dal valore della proprietà `group_member`); *%{host}* - nome host, IP o URI del server LDAP (sostituito dal valore della proprietà `host`); *%{user}* - nome utente Zabbix. Predefinito: *(%{groupattr}=%{user})* Esempi: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* corrisponderà a "User1" se un oggetto gruppo LDAP contiene l'attributo "*member*" con il valore "*uid=User1,ou=Users,dc=example,dc=com*" e restituirà il gruppo di cui "User1" è membro; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* corrisponderà a "User1" se un oggetto gruppo LDAP contiene l'attributo specificato nella proprietà `group_member` con il valore "*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*" e restituirà il gruppo di cui "User1" è membro. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
group_member	string	Attributo della directory utente LDAP che contiene informazioni sui membri del gruppo; utilizzato per configurare un controllo dell'appartenenza dell'utente nella directory utente LDAP. Ignorato durante il provisioning di un utente se `group_membership` è impostato. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
group_membership	string	Attributo della directory utente LDAP che contiene informazioni sui gruppi a cui appartiene un utente. Esempio: *memberOf* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
search_filter	string	Stringa di filtro personalizzata utilizzata per individuare e autenticare un utente in una directory utente LDAP in base alle informazioni fornite nella richiesta di accesso. Segnaposto `search_filter` supportati: *%{attr}* - nome dell'attributo di ricerca (ad esempio, *uid*, *sAMAccountName*); *%{user}* - nome utente Zabbix. Predefinito: *(%{attr}=%{user})* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"

Proprietà	Tipo	Descrizione
start_tls	integer	Opzione di configurazione del server LDAP che consente di proteggere la comunicazione con il server LDAP utilizzando Transport Layer Security (TLS). Si noti che `start_tls` deve essere impostato su ”Disabled” per gli host che utilizzano il protocollo `ldaps://`. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su ”User directory of type LDAP”
user_ref_attr	string	Attributo della directory utente LDAP utilizzato per fare riferimento a un oggetto utente. Il valore di `user_ref_attr` viene utilizzato per ottenere i valori dall’attributo specificato nella directory utente e inserirli al posto del segnaposto *%{ref}* nella stringa `group_filter`. Esempi: *cn*, *uid*, *member*, *uniqueMember* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su ”User directory of type LDAP”
Proprietà specifiche di SAML:		
idp_entityid	string	URI che identifica l’identity provider e viene utilizzato per comunicare con l’identity provider nei messaggi SAML. Esempio: *https://idp.example.com/idp* Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su ”User directory of type SAML”
sp_entityid	string	URL o qualsiasi stringa che identifica il service provider dell’identity provider. Esempi: *https://idp.example.com/sp* *zabbix* Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su ”User directory of type SAML”
username_attribute	string	Attributo della directory utente SAML (anche attributo SCIM se `scim_status` è impostato su ”SCIM provisioning is enabled”) che contiene il nome utente dell’utente, confrontato con il valore della proprietà `username` dell’oggetto User durante l’autenticazione. Esempi: *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* Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su ”User directory of type SAML”
sso_url	string	URL del servizio SAML single sign-on dell’identity provider, a cui Zabbix invierà le richieste di autenticazione SAML. Esempio: *http://idp.example.com/idp/sso/saml* Comportamento della proprietà: - *obbligatoria* se `idp_type` è impostato su ”User directory of type SAML”

Proprietà	Tipo	Descrizione
slo_url	string	URL del servizio SAML single log-out dell'identity provider, a cui Zabbix invierà le richieste di disconnessione SAML. Esempio: *https://idp.example.com/idp/slo/saml* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
encrypt_nameid	integer	Indica se il name ID SAML deve essere crittografato. Valori possibili: 0 - *(predefinito)* Non crittografare il name ID; 1 - Crittografare il name ID. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
encrypt_assertions	integer	Indica se le asserzioni SAML devono essere crittografate. Valori possibili: 0 - *(predefinito)* Non crittografare le asserzioni; 1 - Crittografare le asserzioni. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
nameid_format	string	Formato del name ID del service provider dell'identity provider SAML. Esempi: *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* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
scim_status	integer	Indica se il provisioning SCIM per SAML è abilitato o disabilitato. Valori possibili: 0 - *(predefinito)* Il provisioning SCIM è disabilitato; 1 - Il provisioning SCIM è abilitato. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
sign_assertions	integer	Indica se le asserzioni SAML devono essere firmate con una firma SAML. Valori possibili: 0 - Non firmare le asserzioni; 1 - *(predefinito)* Firmare le asserzioni. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
sign_authn_requests	integer	Indica se le richieste AuthN SAML devono essere firmate con una firma SAML. Valori possibili: 0 - Non firmare le richieste AuthN; 1 - *(predefinito)* Firmare le richieste AuthN. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"

Proprietà	Tipo	Descrizione
sign_messages	integer	Indica se i messaggi SAML devono essere firmati con una firma SAML. Valori possibili: 0 - Non firmare i messaggi; 1 - *(predefinito)* Firmare i messaggi. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
sign_logout_requests	integer	Indica se le richieste di disconnessione SAML devono essere firmate con una firma SAML. Valori possibili: 0 - Non firmare le richieste di disconnessione; 1 - *(predefinito)* Firmare le richieste di disconnessione. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
sign_logout_responses	integer	Indica se le risposte di disconnessione SAML devono essere firmate con una firma SAML. Valori possibili: 0 - Non firmare le risposte di disconnessione; 1 - *(predefinito)* Firmare le risposte di disconnessione. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
idp_certificate	string	Contenuto del certificato del Service Provider (SP) per la configurazione del servizio SAML single sign-on (SSO). Esempi: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` nel file di configurazione del frontend - *sola scrittura*
sp_private_key	string	Contenuto della chiave privata del Service Provider (SP) per la configurazione del servizio SAML single sign-on (SSO). Fornisce autenticazione sicura e scambio di dati con l'Identity Provider (IdP). Esempi: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` nel file di configurazione del frontend - *sola scrittura*

Proprietà	Tipo	Descrizione
sp_certificate	string	Contenuto del certificato del Service Provider (SP) per la configurazione del servizio SAML single sign-on (SSO). Esempi: *-----BEGIN CERTIFICATE-----* * MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` nel file di configurazione del frontend - *sola scrittura*
idp_certificate_hash	string	Hash md5 del valore di idp_certificate. Restituisce una stringa vuota per un idp_certificate vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` nel file di configurazione del frontend - *sola lettura*
sp_private_key_hash	string	Hash md5 del valore di sp_private_key. Restituisce una stringa vuota per uno sp_private_key vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` nel file di configurazione del frontend - *sola lettura*
sp_certificate_hash	string	Hash md5 del valore di sp_certificate. Restituisce una stringa vuota per uno sp_certificate vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` nel file di configurazione del frontend - *sola lettura*

Mappature dei tipi di supporto

L'oggetto delle mappature dei tipi di supporto ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
userdirectory_mediaid	ID	ID della mappatura del tipo di supporto. Comportamento della proprietà: - *sola lettura*
name	string	Nome visibile nell'elenco delle mappature dei tipi di supporto. Comportamento della proprietà: - *obbligatoria*
mediatypeid	ID	ID del tipo di supporto da creare; utilizzato come valore per la proprietà `mediatypeid` dell'oggetto Media. Comportamento della proprietà: - *obbligatoria*

Proprietà	Tipo	Descrizione
attribute	string	Attributo della directory utenti LDAP/SAML (anche attributo SCIM se `scim_status` è impostato su "SCIM provisioning is enabled") che contiene il supporto dell'utente (ad esempio, *user@example.com*), utilizzato come valore per la proprietà `sendto` dell'oggetto Media. Se presente nei dati ricevuti dal provider di identità LDAP/SAML e il valore non è vuoto, questo attiverà la creazione del supporto per l'utente sottoposto a provisioning. Comportamento della proprietà: - *obbligatoria*
active	integer	Valore della proprietà `active` del supporto utente quando il supporto viene creato per l'utente sottoposto a provisioning. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.
severity	integer	Valore della proprietà `severity` del supporto utente quando il supporto viene creato per l'utente sottoposto a provisioning. Predefinito: 63.
period	string	Valore della proprietà `period` del supporto utente quando il supporto viene creato per l'utente sottoposto a provisioning. Predefinito: 1-7,00:00-24:00.

Mappature dei gruppi di provisioning

Le mappature dei gruppi di provisioning hanno le seguenti proprietà.

Proprietà	Tipo	Descrizione
name	string	Nome completo di un gruppo (ad esempio, *Zabbix administrators*) nella directory utenti LDAP/SAML (anche SCIM se `scim_status` è impostato su "SCIM provisioning is enabled"). Supporta il carattere jolly "*". Univoco tra tutte le mappature dei gruppi di provisioning. Comportamento della proprietà: - *required*
roleid	ID	ID del ruolo utente da assegnare all'utente. Se corrispondono più mappature dei gruppi di provisioning, all'utente viene assegnato il ruolo con il tipo di utente più alto (*User*, *Admin* o *Super admin*). Se sono presenti più ruoli con lo stesso tipo di utente, all'utente viene assegnato il primo ruolo (ordinato in ordine alfabetico). Comportamento della proprietà: - *required*
user_groups	array	Array di oggetti ID dei gruppi utenti di Zabbix. Ogni oggetto ha le seguenti proprietà: `usrgrpid` - `(ID)` ID del gruppo utenti di Zabbix da assegnare all'utente. Se corrispondono più mappature dei gruppi di provisioning, all'utente vengono assegnati i gruppi utenti di Zabbix di tutte le mappature corrispondenti. Comportamento della proprietà: - *required*

userdirectory.create

Descrizione

`object userdirectory.create(object/array userDirectory)`

Questo metodo consente di creare nuovi directory utente.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*.

Parametri

`(object/array)` Directory utente da creare.

Il metodo accetta directory utente con le proprietà standard della directory utente.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle directory utente create nella proprietà `userdirectoryids`. L'ordine degli ID restituiti corrisponde all'ordine delle directory utente passate.

Esempi

Creazione di una directory utenti

Creare una directory utenti per autenticare gli utenti con StartTLS tramite LDAP. Si noti che, per autenticare gli utenti tramite LDAP, l'autenticazione LDAP deve essere abilitata.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Creazione di una directory utenti (provisioning JIT abilitato)

Creare una directory utenti per autenticare gli utenti tramite LDAP (con provisioning JIT abilitato). Si noti che, per autenticare gli utenti tramite LDAP, l'autenticazione LDAP deve essere abilitata.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CUserDirectory::create() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

Descrizione

`object userdirectory.delete(array userDirectoryIds)`

Questo metodo consente di eliminare le directory utente. Una directory utente non può essere eliminata quando è utilizzata direttamente da almeno un gruppo di utenti.
 La directory utente LDAP predefinita non può essere eliminata quando `authentication.ldap_configured` è impostato su 1 o quando rimangono altre directory utente.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*.

Parametri

"(array)" ID delle directory utente da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei proxy eliminati sotto la proprietà `proxyids`.

Esempi

Eliminazione di più directory utente

Elimina due directory utente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

Fonte

CUserDirectory::delete() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

Descrizione

`integer/array userdirectory.get(object parameters)`

Il metodo consente di recuperare le directory utente in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Super admin*.

Parametri

(object) Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
userdirectoryids	ID/array	Restituisce solo le directory utente con gli ID specificati.
selectUsrgrps	query	Restituisce una proprietà `usrgrps` con i gruppi utenti associati a una directory utente. Supporta `count`.
selectProvisionMedia	query	Restituisce una proprietà `provision_media` con le mappature dei tipi di supporto associate a una directory utente.
selectProvisionGroups	query	Restituisce una proprietà `provision_groups` con le mappature dei gruppi di provisioning associate a una directory utente.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `name`.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori. Supporta le proprietà: `userdirectoryid`, `idp_type`, `provision_status`.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Proprietà supportate: `name`, `description`. La directory utente di tipo SAML avrà un valore vuoto sia per la proprietà `name` sia per la proprietà `description`. Entrambe le proprietà possono essere modificate con l'operazione `userdirectory.update`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

(integer/array) Restituisce:

- un array di oggetti
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`

Esempi

Recupero delle directory utente

Recupera tutte le directory utente con proprietà aggiuntive che mostrano le mappature dei tipi di supporto e le mappature dei gruppi di provisioning associate a ciascuna directory utente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Guarda anche

- Gruppo utenti

Fonte

CUserDirectory::get() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

Descrizione

`object userdirectory.test(array userDirectory)`

Questo metodo consente di testare le impostazioni di connessione della directory utente.

Note:
Questo metodo consente anche di verificare quali dati configurati corrispondono alle impostazioni della directory utente per il provisioning degli utenti (ad esempio, quale ruolo utente, quali gruppi utente e quali media utente verranno assegnati all'utente). Per questo tipo di test, la richiesta API deve essere effettuata per una user directory che abbia `provision_status` impostato su enabled.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*.

Parametri

`(object)` Proprietà della directory utenti.

Poiché l'API `userdirectory.get` non restituisce il campo `bind_password`, devono essere forniti `userdirectoryid` e/o `bind_password`.
 Oltre alle proprietà standard della directory utenti, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
test_username	string	Nome utente da testare nella directory utenti.
test_password	string	Password associata al nome utente da testare nella directory utenti.

Valori restituiti

"(bool)" Restituisce vero in caso di successo.

Esempi

Test della directory utenti per un utente esistente

Testa la directory utenti ”3” per ”user1”.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Test della directory utenti per un utente inesistente

Test della directory utenti ”3” per l’utente inesistente ”user2”.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Errore dell'applicazione.",
        "data": "Nome utente o password non corretti oppure account temporaneamente bloccato."
    },
    "id": 1
}
```

Test della directory utenti per il provisioning degli utenti

Testa la user directory ”3” per verificare quali dati configurati corrispondono alle impostazioni della directory utenti per il provisioning di ”user3” (ad esempio, quale ruolo utente, gruppi utenti e supporti utente verranno assegnati all’utente).

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

Fonte

CUserDirectory::test() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

Descrizione

`object userdirectory.update(object/array userDirectory)`

Questo metodo consente di aggiornare le directory utente esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*.

Parametri

`(object/array)` Proprietà della directory utenti da aggiornare.

La proprietà `userdirectoryid` deve essere definita per ogni directory utenti, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Valori restituiti

”(object)” Restituisce un oggetto contenente gli ID delle directory utente aggiornate nella proprietà ”userdirectoryids”.

Esempi

Aggiornare la password di bind per una directory utenti

Impostare una nuova password di bind per una directory utenti.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Aggiornare le mappature per la directory utenti

Aggiorna le mappature dei gruppi di provisioning e le mappature dei tipi di supporto per la directory utenti ”2”.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CUserDirectory::update() in *ui/include/classes/api/services/CUserDirectory.php*.

Dispositivo

Questa classe è progettata per funzionare con dispositivi mobili da utilizzare con l'app mobile Zabbix.

Riferimenti agli oggetti:

- Device

Metodi disponibili:

- device.get - recupera i dispositivi
- device.offboard - rimuove i dispositivi (offboarding)

Oggetto device

L'oggetto device rappresenta un dispositivo mobile da utilizzare con l'app mobile Zabbix.

Property	Type	Description
deviceid	ID	ID del device (interno, ad esempio 1, 2). Property behavior: - Sola lettura
userid	ID	ID dell'utente del device. Property behavior: - Costante
uuid	string	Identificatore univoco universale del device (UUIDv7) (esterno, ad esempio `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`). Property behavior: - Sola lettura
name	string	Nome del device. Property behavior: - Sola lettura
status	integer	Stato del device. Valori possibili: 0 - Nuovo 1 - Attivo 2 - Non assegnato Property behavior: - Sola lettura
activated_at	timestamp	Data e ora in cui il device è stato aggiunto nell'app mobile, oppure 0 se non è ancora stato aggiunto. Property behavior: - Sola lettura

Property	Type	Description
lastaccess	timestamp	Data e ora dell'ultima autenticazione del device nell'app mobile, oppure 0 se non è ancora stato autenticato. Property behavior: - Sola lettura

device.get

Questo metodo consente di recuperare i dispositivi mobili da utilizzare con la Zabbix mobile app.

Il metodo è disponibile per gli utenti *User*, *Admin* e *Super admin* se `EnableMobileDevices` è abilitato e l'utente dispone dell'accesso ai dispositivi.
L'accesso ai metodi API può essere revocato nella configurazione del ruolo utente.

Gli utenti *User* e *Admin* possono recuperare solo i propri dispositivi.

Firma del metodo: `integer/array device.get(object parameters)`

Origine del metodo: `CDevice::get()` in `ui/include/classes/api/services/CDevice.php`

Parametri

Parametro	Tipo	Descrizione
deviceids	ID/array	Restituisce solo i dispositivi con gli ID dispositivo specificati.
userids	ID/array	Restituisce solo i dispositivi appartenenti agli utenti con gli ID utente specificati.
roleids	ID/array	Restituisce solo i dispositivi appartenenti agli utenti con gli ID dei ruoli utente specificati.
usrgrpids	ID/array	Restituisce solo i dispositivi appartenenti agli utenti con gli ID dei gruppi utente specificati.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `deviceid`, `name`.
countOutput	boolean	Questi parametri sono descritti nella documentazione di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

Restituisce un array di oggetti oppure il numero di oggetti recuperati (se viene utilizzato `countOutput`).

Esempi

Recuperare i dispositivi

Recupera i dispositivi appartenenti agli utenti con questi ID utente: 1, 2, 3.
 Nota che l'utente 2 (guest) non dispone di dispositivi e che il dispositivo 3 non è ancora stato aggiunto o autenticato nell'app mobile Zabbix.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
```

```
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
        },
        {
            "deviceid": "2",
            "userid": "1",
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```

device.offboard

Questo metodo consente di rimuovere (disassociare) i dispositivi mobili da utilizzare con l'app mobile Zabbix.

Il metodo è disponibile per gli utenti *User*, *Admin* e *Super admin* se `EnableMobileDevices` è abilitato e l'utente dispone dell'accesso ai dispositivi.
L'accesso ai metodi API può essere revocato nella configurazione del ruolo utente.

Gli utenti *User* e *Admin* possono rimuovere solo i propri dispositivi.

Firma del metodo: `object device.offboard(object parameters)`

Origine del metodo: `CDevice::offboard()` in `ui/include/classes/api/services/CDevice.php`

Parametri

Parametro	Tipo	Descrizione
uuid	string	Identificatore univoco universale del dispositivo (UUIDv7). Corrisponde a `deviceid`.
force	boolean	Impostare su `true` per ignorare l'errore "Device not found" e rimuovere forzatamente il dispositivo. Predefinito: `false`

Valori restituiti

Restituisce un oggetto contenente l'UUID del dispositivo rimosso.

Esempi

Rimuovi dispositivo

Rimuove un dispositivo con l'UUID specificato, ignorando l'errore "Dispositivo non trovato".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

Elemento del grafico

Questa classe è progettata per lavorare con gli elementi del grafico.

Riferimenti agli oggetti:

- Elemento del grafico

Metodi disponibili:

- graphitem.get - recupera gli elementi del grafico

Oggetto elemento del grafico

I seguenti oggetti sono direttamente correlati all'API `graphitem`.

Elemento del grafico

> **Note:**
> Gli elementi del grafico possono essere modificati solo tramite l'API `graph`.

L'oggetto elemento del grafico ha le seguenti proprietà.

Property	Type	Description
gitemid	ID	ID dell'elemento del grafico. Property behavior: - *sola lettura*
color	string	Colore di disegno dell'elemento del grafico come codice colore esadecimale. Property behavior: - *obbligatorio* per le operazioni di creazione

Property	Type	Description
itemid	ID	ID dell'item. Property behavior: - *obbligatorio* per le operazioni di creazione
calc_fnc	integer	Valore dell'item che verrà visualizzato. Valori possibili: 1 - valore minimo; 2 - *(predefinito)* valore medio; 4 - valore massimo; 7 - tutti i valori; 9 - ultimo valore, usato solo dai grafici a torta ed esplosi.
drawtype	integer	Stile di disegno dell'elemento del grafico. Valori possibili: 0 - *(predefinito)* linea; 1 - area riempita; 2 - linea spessa; 3 - punto; 4 - linea tratteggiata; 5 - linea sfumata.
graphid	ID	ID del grafico a cui appartiene l'elemento del grafico.
sortorder	integer	Posizione dell'item nel grafico. Predefinito: inizia da "0" e aumenta di uno per ogni voce.
type	integer	Tipo di elemento del grafico. Valori possibili: 0 - *(predefinito)* semplice; 2 - somma del grafico, usato solo dai grafici a torta ed esplosi.
yaxisside	integer	Lato del grafico in cui verrà disegnata la scala Y dell'elemento del grafico. Valori possibili: 0 - *(predefinito)* lato sinistro; 1 - lato destro.

graphitem.get

Descrizione

`integer/array graphitem.get(object parameters)`

Il metodo consente di recuperare gli elementi del grafico in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
graphids	ID/array	Restituisce solo i graph item che appartengono ai grafici specificati.
itemids	ID/array	Restituisce solo i graph item con gli ID item specificati.

Parametro	Tipo	Descrizione
type	integer	Restituisce solo i graph item del tipo specificato. Per un elenco dei tipi di graph item supportati, fare riferimento alla pagina dell'oggetto graph item.
selectGraphs	query	Restituisce una proprietà `graphs` con un array di grafici a cui l'item appartiene.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `gitemid`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero degli item del grafico da un grafico

Recupera tutti gli item del grafico utilizzati in un grafico con informazioni aggiuntive sull'item e sull'host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Guarda anche

- Grafico

Fonte

CGraphItem::get() in *ui/include/classes/api/services/CGraphItem.php*.

Espressione regolare

Questa classe è progettata per funzionare con le espressioni regolari globali.

Riferimenti agli oggetti:

- Espressione regolare
- Espressioni

Metodi disponibili:

- regexp.create - crea nuove espressioni regolari
- regexp.delete - elimina espressioni regolari
- regexp.get - recupera espressioni regolari
- regexp.update - aggiorna espressioni regolari

Oggetto espressione regolare

I seguenti oggetti sono direttamente correlati all'API `regexp`.

Espressione regolare

L'oggetto espressione regolare globale ha le seguenti proprietà.

Property	Type	Description
regexpid	ID	ID dell'espressione regolare. Property behavior: - *read-only* - *required* for update operations.
name	string	Nome dell'espressione regolare. Property behavior: - *required* for create operations.
description	text	Descrizione dell'espressione regolare.
test_string	string	Stringa di test.

Espressioni

L’oggetto expressions ha le seguenti proprietà.

Property	Type	Description
expression	string	Espressione regolare. Property behavior: - *required*
expression_type	integer	Tipo di espressione regolare. Valori possibili: 0 - Contiene la stringa; 1 - Contiene una qualsiasi sottostringa dall’elenco; 2 - Non contiene la stringa; 3 - Corrisponde all’espressione regolare; 4 - Non corrisponde all’espressione regolare. Property behavior: - *required*
exp_delimiter	string	Delimitatore dell’espressione. Valore predefinito: `","`. Valori possibili: `","` o `"."`, oppure `"/"`. Property behavior: - *supported* if `expression_type` is set to *Contains any substring from list*.
case_sensitive	integer	Distinzione tra maiuscole e minuscole. Valore predefinito: 0. Valori possibili: 0 - Non fa distinzione tra maiuscole e minuscole; 1 - Fa distinzione tra maiuscole e minuscole.

regexp.create

Descrizione

`object regexp.create(object/array regularExpressions)`

Questo metodo consente di creare nuove espressioni regolari globali.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente

Parametri

`(object/array)` Espressioni regolari da creare.

Oltre alle proprietà standard, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
expressions	array	Opzioni delle espressioni. Comportamento del parametro: - *obbligatorio*

Valori di ritorno

`(oggetto)` Restituisce un oggetto contenente gli ID delle espressioni regolari create sotto la proprietà `regexpids`.

Esempi

Creazione di una nuova espressione regolare globale.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "Espressione regolare globale usata per individuare e identificare le risorse di me
        "test_string": "/boot",
        "expressions": [
            {
                "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$"
                "expression_type": "4",
                "case_sensitive": "1"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "9"
        ]
    },
    "id": 1
}
```

Fonte

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Descrizione

`object regexp.delete(array regexpids)`

Questo metodo consente di eliminare le espressioni regolari globali.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle espressioni regolari da eliminare.

Valori di ritorno

(oggetto) Restituisce un oggetto contenente gli ID delle espressioni regolari cancellate sotto la proprietà regexpids.

Esempi

Eliminazione di più espressioni regolari globali.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

Fonte

CRegexp::delete() in *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Descrizione

`integer/array regexp.get(object parameters)`

Il metodo consente di recuperare le espressioni regolari globali in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per *Super Admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

(object) Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
regexpids	ID/array	Restituisce solo le espressioni regolari con gli ID specificati.
selectExpressions	query	Restituisce una proprietà `expressions`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `regexpid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

Parametro	Tipo	Descrizione
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero delle espressioni regolari globali.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid","name","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "regexpid": "1",
            "name": "File systems for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Network interfaces for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
                    "expression_type": "4"
                },
                {
```

```
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
    },
    "id": 1
}
```

Fonte

CRegexp::get() in *ui/include/classes/api/services/CRegexp.php*.

regexp.update

Descrizione

`object regexp.update(object/array regularExpressions)`

Questo metodo consente di aggiornare le espressioni regolari globali esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà dell'espressione regolare da aggiornare.

La proprietà `regexpid` deve essere definita per ogni oggetto, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
expressions	array	Opzioni delle espressioni.

Valori di ritorno

`(oggetto)` Restituisce un oggetto contenente gli ID delle espressioni regolari aggiornate sotto la proprietà `regexpids`.

Esempi

Aggiornamento dell'espressione regolare globale per la scoperta dei file system.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
        "regexpid": "1",
        "name": "File systems for discovery",
        "description": "Espressione regolare globale usata per individuare e identificare i file system su
        "test_string": "",
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
```

```
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CRegexp::update() in *ui/include/classes/api/services/CRegexp.php*.

Evento

Questa classe è progettata per lavorare con gli eventi.

Riferimenti agli oggetti:

- Evento
- Tag evento
- URL del tipo di supporto

Metodi disponibili:

- event.get - recupera gli eventi
- event.acknowledge - conferma gli eventi

Oggetto evento

I seguenti oggetti sono direttamente correlati all'API `event`.

Evento

> **Note:**
> Gli eventi vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

L'oggetto evento ha le seguenti proprietà.

Property	Type	Description
eventid	ID	ID dell'evento.
source	integer	Tipo di evento. Valori possibili: 0 - evento creato da un trigger; 1 - evento creato da una regola di discovery; 2 - evento creato da autoregistrazione attiva di agent; 3 - evento interno; 4 - evento creato durante l'aggiornamento dello stato del servizio.

Property	Type	Description
object	integer	Tipo di oggetto correlato all'evento. Valori possibili se `source` è impostato su "evento creato da un trigger": 0 - trigger. Valori possibili se `source` è impostato su "evento creato da una regola di discovery": 1 - host scoperto; 2 - servizio scoperto. Valori possibili se `source` è impostato su "evento creato da autoregistrazione attiva di agent": 3 - host autoregistrato. Valori possibili se `source` è impostato su "evento interno": 0 - trigger; 4 - item; 5 - regola LLD. Valori possibili se `source` è impostato su "evento creato durante l'aggiornamento dello stato del servizio": 6 - servizio.
objectid	ID	ID dell'oggetto correlato.
acknowledged	integer	Indica se l'evento è stato riconosciuto.
clock	timestamp	Ora in cui l'evento è stato creato.
ns	integer	Nanosecondi al momento della creazione dell'evento.
name	string	Nome dell'evento risolto.
value	integer	Stato dell'oggetto correlato. Valori possibili se `source` è impostato su "evento creato da un trigger" o "evento creato durante l'aggiornamento dello stato del servizio": 0 - OK; 1 - problema. Valori possibili se `source` è impostato su "evento creato da una regola di discovery": 0 - host o servizio attivo; 1 - host o servizio non attivo; 2 - host o servizio scoperto; 3 - host o servizio perso. Valori possibili se `source` è impostato su "evento interno": 0 - stato "normale"; 1 - stato "sconosciuto" o "non supportato". Comportamento della proprietà: - *supportato* se `source` è impostato su "evento creato da un trigger", "evento creato da una regola di discovery", "evento interno" o "evento creato durante l'aggiornamento dello stato del servizio"
severity	integer	Gravità corrente dell'evento. Valori possibili: 0 - non classificato; 1 - informazione; 2 - avviso; 3 - media; 4 - alta; 5 - disastro.
r_eventid	ID	ID dell'evento di ripristino.

Property	Type	Description
c_eventid	ID	ID dell'evento utilizzato per sovrascrivere (chiudere) l'evento corrente in base alla regola di correlazione globale. Vedere `correlationid` per identificare la regola di correlazione esatta. Questo parametro è definito solo quando l'evento viene chiuso da una regola di correlazione globale.
cause_eventid	ID	ID dell'evento causa.
correlationid	ID	ID della regola di correlazione che ha generato la chiusura del problema. Questo parametro è definito solo quando l'evento viene chiuso da una regola di correlazione globale.
userid	ID	ID dell'utente che ha chiuso l'evento (se l'evento è stato chiuso manualmente).
suppressed	integer	Indica se l'evento è soppresso. Valori possibili: 0 - l'evento è nello stato normale; 1 - l'evento è soppresso.
opdata	string	Dati operativi con macro espanse.
urls	array	URL del tipo di supporto attivi.

Tag evento

L'oggetto tag evento ha le seguenti proprietà.

Property	Type	Description
tag	string	Nome del tag evento.
value	string	Valore del tag evento.

URL del tipo di supporto

L'oggetto URL del tipo di supporto ha le seguenti proprietà.

Proprietà	Type	Descrizione
name	string	Nome dell'URL definito per il tipo di supporto.
url	string	Valore dell'URL definito per il tipo di supporto.

I risultati conterranno voci solo per i tipi di supporto attivi con voce del menu eventi abilitata. Le macro utilizzate nelle proprietà verranno espanse, ma se una delle proprietà contiene una macro non espansa, entrambe le proprietà verranno escluse dai risultati. Per le macro supportate, vedere *Macro supportate*.

event.acknowledge

Descrizione

`object event.acknowledge(object/array parameters)`

Questo metodo consente di aggiornare gli eventi. È possibile eseguire le seguenti azioni di aggiornamento:

- Chiudere l'evento. Se l'evento è già risolto, questa azione verrà ignorata.
- Acknowledge dell'evento. Se l'evento è già acknowledged, questa azione verrà ignorata.
- Rimuovere l'acknowledge dell'evento. Se l'evento non è acknowledged, questa azione verrà ignorata.
- Aggiungere un messaggio.
- Modificare la severità dell'evento. Se l'evento ha già la stessa severità, questa azione verrà ignorata.
- Sopprimere l'evento. Se l'evento è già soppresso, questa azione verrà ignorata.
- Rimuovere la soppressione dell'evento. Se l'evento non è soppresso, questa azione verrà ignorata.
- Modificare il rank dell'evento in causa. Se l'evento è già classificato come causa, questa azione verrà ignorata.
- Modificare il rank dell'evento in sintomo. Se l'evento è già classificato come sintomo, questa azione verrà ignorata.

Attention:
Possono essere aggiornati solo gli eventi di trigger.
 Possono essere aggiornati solo gli eventi di problema.
 Per chiudere l'evento o modificare la severità dell'evento sono necessari i diritti di lettura/scrittura per il trigger.
 Per chiudere un evento, la chiusura manuale deve essere consentita nel trigger.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Parametri contenenti gli ID degli eventi e le operazioni di aggiornamento da eseguire.

Parameter	Type	Description
eventids	ID/array	ID degli eventi da riconoscere. Comportamento del parametro: - *obbligatorio*
action	integer	Azione o azioni di aggiornamento dell'evento. Valori bitmap possibili: 1 - chiudere il problema; 2 - riconoscere l'evento; 4 - aggiungere un messaggio; 8 - modificare la gravità; 16 - annullare il riconoscimento dell'evento; 32 - sopprimere l'evento; 64 - annullare la soppressione dell'evento; 128 - modificare la classificazione dell'evento in causa; 256 - modificare la classificazione dell'evento in sintomo. Questo è un campo con maschera di bit; è accettabile qualsiasi somma dei valori bitmap possibili (ad esempio, 6 per riconoscere un evento e lasciare un messaggio, oppure 34 per riconoscere e sopprimere un evento). Comportamento del parametro: - *obbligatorio*
cause_eventid	ID	ID dell'evento causa. Comportamento del parametro: - *obbligatorio* se `action` contiene il bit "modificare la classificazione dell'evento in sintomo"
message	string	Testo del messaggio. Comportamento del parametro: - *obbligatorio* se `action` contiene il bit "aggiungere un messaggio"
severity	integer	Nuova gravità per gli eventi. Valori possibili: 0 - non classificato; 1 - informazioni; 2 - avviso; 3 - media; 4 - alta; 5 - disastro. Comportamento del parametro: - *obbligatorio* se `action` contiene il bit "modificare la gravità"

Parameter	Type	Description
suppress_until	integer	Timestamp Unix fino al quale l'evento deve essere soppresso. Se impostato su "0", la soppressione sarà indefinita. Comportamento del parametro: - *obbligatorio* se `action` contiene il bit "sopprimere l'evento"

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli eventi aggiornati nella proprietà `eventids`.

Esempi

Riconoscimento di un evento

Riconosci un singolo evento e lascia un messaggio.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
        "message": "Problema risolto."
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

Modifica della gravità dell'evento

Modificare la gravità per più eventi e lasciare un messaggio.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
```

```
            "20428"
        ]
    },
    "id": 1
}
```

Sopprimere un evento indefinitamente

Sopprimere un singolo evento senza una data di scadenza.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Riattivazione della soppressione di un evento

Rimuove la soppressione di un singolo evento precedentemente soppresso.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Confermare e sopprimere un evento

Conferma uno o più eventi, aggiungi un messaggio e sopprimili nella stessa richiesta.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Suppressing during planned maintenance.",
        "suppress_until": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427,
            20428
        ]
    },
    "id": 1
}
```

Fonte

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

Descrizione

`integer/array event.get(object parameters)`

Il metodo consente di recuperare gli eventi in base ai parametri specificati.

Attention:
Questo metodo può restituire eventi di un'entità eliminata se tali eventi non sono ancora stati rimossi dall'housekeeper.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parameter	Type	Description
eventids	ID/array	Restituisce solo gli eventi con gli ID specificati.
groupids	ID/array	Restituisce solo gli eventi creati da oggetti che appartengono ai gruppi di host specificati.
hostids	ID/array	Restituisce solo gli eventi creati da oggetti che appartengono agli host specificati.
objectids	ID/array	Restituisce solo gli eventi creati dagli oggetti specificati.

Parameter	Type	Description
source	integer	Restituisce solo gli eventi con il tipo specificato. Per l'elenco dei tipi di evento supportati, fare riferimento alla pagina dell'oggetto event. Predefinito: 0 - eventi trigger.
object	integer	Restituisce solo gli eventi creati da oggetti del tipo specificato. Per l'elenco dei tipi di oggetto supportati, fare riferimento alla pagina dell'oggetto event. Predefinito: 0 - trigger.
acknowledged	boolean	Se impostato su `true`, restituisce solo gli eventi riconosciuti.
action	integer	Restituisce solo gli eventi per i quali sono state eseguite le azioni di aggiornamento dell'evento specificate. Per più azioni, usare la somma di qualsiasi valore bitmap accettabile come bitmask (ad esempio, 34 per riconoscere e sopprimere l'evento).
action_userids	ID/array	Restituisce solo gli eventi con gli ID specificati degli utenti che hanno eseguito le azioni di aggiornamento dell'evento.
suppressed	boolean	Se impostato su `true`, restituisce solo gli eventi soppressi.
symptom	boolean	Se impostato su `true`, restituisce solo gli eventi sintomo.
severities	integer/array	Restituisce solo gli eventi con le severità specificate. Si applica solo se object è trigger.
trigger_severities	integer/array	Restituisce solo gli eventi con le severità del trigger specificate. Si applica solo se object è trigger.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
tags	array	Restituisce solo gli eventi con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti gli eventi. Valori possibili per operator: 0 - *(predefinito)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
eventid_from	string	Restituisce solo gli eventi con ID maggiori o uguali all'ID specificato.
eventid_till	string	Restituisce solo gli eventi con ID minori o uguali all'ID specificato.
time_from	timestamp	Restituisce solo gli eventi creati a partire dal momento specificato.
time_till	timestamp	Restituisce solo gli eventi creati fino al momento specificato.
problem_time_from	timestamp	Restituisce solo gli eventi che si trovavano nello stato di problema a partire da `problem_time_from`, indipendentemente dal loro stato attuale. Si applica solo quando la source è un evento trigger e object è trigger. Questo parametro è obbligatorio se è specificato `problem_time_till`.
problem_time_till	timestamp	Restituisce solo gli eventi che si trovavano nello stato di problema fino a `problem_time_till`, indipendentemente dal loro stato attuale. Si applica solo quando la source è un evento trigger e object è trigger. Questo parametro è obbligatorio se è specificato `problem_time_from`.
value	integer/array	Restituisce solo gli eventi con i valori specificati.

Parameter	Type	Description
selectAcknowledges	query	Restituisce una proprietà `acknowledges` con gli aggiornamenti dell'evento. Gli aggiornamenti dell'evento sono ordinati in ordine cronologico inverso. L'oggetto di aggiornamento dell'evento ha le seguenti proprietà: `acknowledgeid` - `(ID)` ID del riconoscimento; `userid` - `(ID)` ID dell'utente che ha aggiornato l'evento; `clock` - `(timestamp)` momento in cui l'evento è stato aggiornato; `message` - `(string)` testo del messaggio; `action` - `(integer)` azione di aggiornamento eseguita, vedere `event.acknowledge`; `old_severity` - `(integer)` severità dell'evento prima di questa azione di aggiornamento; `new_severity` - `(integer)` severità dell'evento dopo questa azione di aggiornamento; `suppress_until` - `(timestamp)` momento fino al quale l'evento sarà soppresso; `taskid` - `(ID)` ID del task se l'evento corrente è in fase di cambio di priorità; `maintenanceid` - `(ID)` ID della manutenzione; `username` - `(string)` nome utente dell'utente che ha aggiornato l'evento; `name` - `(string)` nome dell'utente che ha aggiornato l'evento; `surname` - `(string)` cognome dell'utente che ha aggiornato l'evento. Supporta `count`.
selectAlerts	query	Restituisce una proprietà `alerts` con gli alert generati dall'evento. Gli alert sono ordinati in ordine cronologico inverso.
selectHosts	query	Restituisce una proprietà `hosts` con gli host contenenti l'oggetto che ha creato l'evento. Supportato solo per eventi generati da trigger, item o regole LLD.
selectRelatedObject	query	Restituisce una proprietà `relatedObject` con l'oggetto che ha creato l'evento. Il tipo di oggetto restituito dipende dal tipo di evento.
selectSuppressionData	query	Restituisce una proprietà `suppression_data` con l'elenco delle manutenzioni attive e delle soppressioni manuali: `maintenanceid` - `(ID)` ID della manutenzione; `userid` - `(ID)` ID dell'utente che ha soppresso l'evento; `suppress_until` - `(integer)` tempo fino al quale l'evento è soppresso.
selectTags	query	Restituisce una proprietà `tags` con i tag dell'evento.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta le proprietà di tipo `text` data type.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `eventid`, `objectid`, `clock`. Valori possibili se usato insieme a `groupBy`: `objectid`. Valori possibili se usato insieme a `countOutput` e `groupBy`: `objectid`, `rowscount`.
groupBy	string/array	Raggruppa i risultati in base alle proprietà specificate. Le proprietà specificate saranno restituite nei risultati. Valori possibili: `objectid`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	

Parameter	Type	Description
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti
- il numero di oggetti recuperati, se è stato usato il parametro `countOutput`, ma non è stato usato il parametro `groupBy`
- un array di oggetti con i risultati dell'aggregazione, se è stato usato il parametro `groupBy`

Esempi

Recupero degli eventi del trigger

Recupera gli eventi più recenti dal trigger "22395".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
```

```
            "suppression_data": [],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "1",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "maintenanceid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [
                {
```

```
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


Recupero degli eventi per intervallo di tempo

Recupera tutti gli eventi creati tra il 17 e il 18 ottobre 2012, in ordine cronologico inverso.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Risposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
```

```
            "ns": "0",
            "name": "Less than 25% free in the history 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 processes more than 75% busy",
            "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 processes more than 75% busy",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


Recupero degli eventi riconosciuti da un utente specificato

Recupero degli eventi riconosciuti dall'utente con ID=10

Richiesta:


```
{
    "jsonrpc": "2.0",
```

```
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

Recupero dei principali trigger in base al numero di eventi di problema

Recupera i 5 trigger principali con gravità ”Warning”, ”Average”, ”High” o ”Disaster”, insieme al numero di eventi di problema entro un intervallo di tempo specificato.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Alert
- Item
- Host
- Regola LLD
- Servizio
- Trigger

Fonte

CEvent::get() in *ui/include/classes/api/services/CEvent.php*.

Grafico

Questa classe è progettata per lavorare con i grafici.

Riferimenti degli oggetti:

- Grafico

Metodi disponibili:

- graph.create - crea nuovi grafici
- graph.delete - elimina i grafici
- graph.get - recupera i grafici
- graph.update - aggiorna i grafici

Oggetto grafico

I seguenti oggetti sono direttamente correlati all'API `graph`.

Grafico

L'oggetto grafico ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
graphid	ID	ID del grafico. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
height	integer	Altezza del grafico in pixel. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
name	string	Nome del grafico. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
width	integer	Larghezza del grafico in pixel. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
flags	integer	Origine del grafico. Valori possibili: 0 - *(predefinito)* un grafico semplice; 4 - un grafico convertito da prototipo. Comportamento della proprietà: - *sola lettura*
graphtype	integer	Tipo di layout del grafico. Valori possibili: 0 - *(predefinito)* normale; 1 - in pila; 2 - a torta; 3 - esploso.
percent_left	float	Percentile sinistro. Predefinito: 0.
percent_right	float	Percentile destro. Predefinito: 0.
show_3d	integer	Indica se mostrare i grafici a torta ed esplosi in 3D. Valori possibili: 0 - *(predefinito)* mostra in 2D; 1 - mostra in 3D.
show_legend	integer	Indica se mostrare la legenda nel grafico. Valori possibili: 0 - nascondi; 1 - *(predefinito)* mostra.
show_work_period	integer	Indica se mostrare l'orario di lavoro nel grafico. Valori possibili: 0 - nascondi; 1 - *(predefinito)* mostra.

Proprietà	Tipo	Descrizione
show_triggers	integer	Indica se mostrare la linea del trigger nel grafico. Valori possibili: 0 - nascondi; 1 - *(predefinito)* mostra.
templateid	ID	ID del grafico del template padre. Comportamento della proprietà: - *sola lettura*
yaxismax	float	Il valore massimo fisso per l'asse Y. Predefinito: 100.
yaxismin	float	Il valore minimo fisso per l'asse Y. Predefinito: 0.
ymax_itemid	ID	ID dell'item utilizzato come valore massimo per l'asse Y. Se un utente non ha accesso all'item specificato, il grafico viene visualizzato come se `ymax_type` fosse impostato su "calculated".
ymax_type	integer	Metodo di calcolo del valore massimo per l'asse Y. Valori possibili: 0 - *(predefinito)* calcolato; 1 - fisso; 2 - item.
ymin_itemid	ID	ID dell'item utilizzato come valore minimo per l'asse Y. Se un utente non ha accesso all'item specificato, il grafico viene visualizzato come se `ymin_type` fosse impostato su "calculated".
ymin_type	integer	Metodo di calcolo del valore minimo per l'asse Y. Valori possibili: 0 - *(predefinito)* calcolato; 1 - fisso; 2 - item.
uuid	string	Identificatore univoco universale, utilizzato per collegare i grafici importati a quelli già esistenti. Generato automaticamente, se non specificato. Comportamento della proprietà: - *supportato* se il grafico appartiene a un template

graph.create

Descrizione

`object graph.create(object/array graphs)`

Questo metodo consente di creare nuovi graph.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Grafici da creare.

Oltre alle proprietà standard del grafico, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
gitems	array	Item del grafico da creare per il grafico. Comportamento del parametro: *- obbligatorio*

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID dei grafici creati sotto la proprietà `graphids`. L'ordine degli ID restituiti corrisponde a ordine dei grafici passati.

Esempi

Creazione di un grafico

Creare un grafico con due item.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Vedi anche

- Item del grafico

Fonte

CGraph::create() in *ui/include/classes/api/services/CGraph.php*.

graph.delete

Descrizione

`object graph.delete(array graphIds)`

Questo metodo consente di eliminare i graph.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei grafici da eliminare.

Restituisce i valori

`(oggetto)` Restituisce un oggetto contenente gli ID dei grafici eliminati sotto la proprietà `graphids`.

Esempi

Eliminazione di più grafici

Eliminare due grafici.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Fonte

CGraph::delete() in *ui/include/classes/api/services/CGraph.php*.

graph.get

Descrizione

`integer/array graph.get(object parameters)`

Il metodo consente di recuperare i graph in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
graphids	ID/array	Restituisce solo i grafici con gli ID specificati.
groupids	ID/array	Restituisce solo i grafici che appartengono a host o template nei gruppi di host o gruppi di template specificati.

Parametro	Type	Descrizione
templateids	ID/array	Restituisce solo i grafici che appartengono ai template specificati.
hostids	ID/array	Restituisce solo i grafici che appartengono agli host specificati.
itemids	ID/array	Restituisce solo i grafici che contengono gli item specificati.
templated	boolean	Se impostato su `true`, restituisce solo i grafici che appartengono ai template.
inherited	boolean	Se impostato su `true`, restituisce solo i grafici ereditati da un template.
expandName	flag	Espande le macro nel nome del grafico.
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery del grafico. L'oggetto di discovery del grafico collega un grafico rilevato a un prototipo di grafico da cui è stato rilevato. Ha le seguenti proprietà: `parent_graphid` - `(ID)` ID del prototipo di grafico da cui è stato creato il grafico; `status` - `(int)` stato della discovery del grafico: 0 - *(predefinito)* il grafico è rilevato, 1 - il grafico non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un grafico non più rilevato verrà eliminato.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola di low-level discovery che ha creato il grafico.
selectHostGroups	query	Restituisce una proprietà `hostgroups` con i gruppi di host a cui appartiene il grafico.
selectTemplateGroups	query	Restituisce una proprietà `templategroups` con i gruppi di template a cui appartiene il grafico.
selectTemplates	query	Restituisce una proprietà `templates` con i template a cui appartiene il grafico.
selectHosts	query	Restituisce una proprietà `hosts` con gli host a cui appartiene il grafico.
selectItems	query	Restituisce una proprietà `items` con gli item utilizzati nel grafico.
selectGraphItems	query	Restituisce una proprietà `gitems` con gli item utilizzati nel grafico.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text` data type. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene il grafico; `hostid` - ID dell'host a cui appartiene il grafico.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `graphid`, `name`, `graphtype`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGraphDiscovery	query	Restituisce una proprietà `graphDiscovery` con l'oggetto di discovery del grafico. Gli oggetti di discovery del grafico collegano il grafico a un prototipo di grafico da cui è stato creato. Questa query è **deprecata**, usare invece `selectDiscoveryData`.

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei grafici dagli host

Recupera tutti i grafici dall'host "10107" e ordinali per nome.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Regola di individuazione
- Item del grafico
- Item
- Host
- Gruppo di host
- Template
- Gruppo di template

Fonte

CGraph::get() in *ui/include/classes/api/services/CGraph.php*.

graph.update

Descrizione

`object graph.update(object/array graphs)`

Questo metodo consente di aggiornare i grafici esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del grafico da aggiornare.

La proprietà `graphid` deve essere definita per ciascun grafico, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del grafico, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
gitems	array	Item del grafico per sostituire gli item del grafico esistenti. Se un item del grafico ha la proprietà `gitemid` definita, verrà aggiornato; in caso contrario verrà creato un nuovo item del grafico.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID dei grafici aggiornati sotto la proprietà `graphids`.

Esempi

Impostazione del massimo per la scala Y

Impostare il massimo della scala Y su un valore fisso di 100.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Fonte

CGraph::update() in *ui/include/classes/api/services/CGraph.php*.

Gruppo di proxy

Questa classe è progettata per funzionare con i gruppi di proxy.

Riferimenti agli oggetti:

- Gruppo di proxy

Metodi disponibili:

- proxygroup.create - crea nuovi gruppi di proxy
- proxygroup.delete - elimina gruppi di proxy
- proxygroup.get - recupera gruppi di proxy
- proxygroup.update - aggiorna gruppi di proxy

Oggetto gruppo di proxy

I seguenti oggetti sono direttamente correlati all'API `proxygroup`.

Gruppo di proxy

L'oggetto del gruppo di proxy ha le seguenti proprietà.

Property	Type	Description
proxy_groupid	ID	ID del gruppo di proxy. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del gruppo di proxy. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
description	text	Descrizione del gruppo di proxy.
failover_delay	string	Periodo durante il quale un proxy nel gruppo di proxy deve comunicare con Zabbix server per essere considerato online. Sono supportati i suffissi temporali (ad esempio, `30s`, `1m`). Sono supportate le macro utente. Valori possibili: 10s-15m. Predefinito: 1m.
min_online	string	Numero minimo di proxy online richiesti per mantenere online il gruppo di proxy. Sono supportate le macro utente. Intervallo di valori possibili: 1-1000. Predefinito: 1.
state	integer	Stato del gruppo di proxy. Valori possibili: 0 - Sconosciuto; 1 - Offline; 2 - In ripristino; 3 - Online; 4 - In peggioramento. Comportamento della proprietà: - *sola lettura*

proxygroup.create

Descrizione

`object proxygroup.create(object/array proxyGroups)`

Questo metodo consente di creare nuovi gruppi di proxy.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Gruppi proxy da creare.

Il metodo accetta gruppi proxy con le proprietà standard dei gruppi proxy.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di proxy creati nella proprietà `proxy_groupids`. L'ordine degli ID restituiti corrisponde all'ordine dei gruppi di proxy passati.

Esempi

Creare un gruppo di proxy

Crea un gruppo di proxy con impostazioni personalizzate.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "proxygroup.create",
"params": {
"name": "Proxy group",
"failover_delay": "5m",
"min_online": "10"
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"proxy_groupids": [
"5"
]
},
"id": 1
}
```

Sorgente

CProxyGroup::create() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Descrizione

`object proxygroup.delete(array proxyGroupIds)`

Questo metodo consente di eliminare i gruppi di proxy.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei gruppi di proxy da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di proxy eliminati nella proprietà `proxy_groupids`.

Esempi

Eliminare molteplici gruppi di proxy

Elimina due gruppi di proxy.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "proxygroup.delete",
"params": [
"5",
"10"
],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"proxy_groupids": [
"5",
"10"
]
},
"id": 1
}
```

Sorgente

CProxyGroup::delete() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Descrizione

`integer/array proxygroup.get(object parameters)`

Il metodo consente di recuperare i gruppi di proxy in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente. Vengono restituiti solo i gruppi di proxy consentiti dalla *Proxy access list* dei gruppi di utenti dell'utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
proxy_groupids	ID/array	Restituisce solo i gruppi di proxy con gli ID specificati.
proxyids	ID/array	Restituisce solo i gruppi di proxy che contengono i proxy specificati.
selectProxies	query	Restituisce una proprietà `proxies` con i proxy che appartengono al gruppo di proxy. Supporta `count`.

Parametro	Type	Descrizione
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `proxy_groupid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(intero/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recuperare tutti i gruppi di proxy

Recupera tutti i gruppi di proxy configurati con i relativi proxy.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "proxygroup.get",
"params": {
"output": "extend",
"selectProxies": ["proxyid", "name"]
},
"id": 1
}
```

Risposta:

```
{
"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
}
```


Vedi anche

- Proxy

Sorgente

CProxyGroup::get() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Descrizione

`object proxygroup.update(object/array proxyGroups)`

Questo metodo consente di aggiornare i gruppi proxy esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. I permessi per richiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del gruppo proxy da aggiornare.

La proprietà `proxy_groupid` deve essere definita per ciascun gruppo proxy; tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Il metodo accetta gruppi proxy con le proprietà standard del gruppo proxy.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi proxy aggiornati nella proprietà `proxy_groupids`.

Esempi

Modificare il numero minimo di proxy online

Modifica il numero minimo di proxy online necessari affinché il gruppo sia online.

Richiesta:


```
{
"jsonrpc": "2.0",
"method": "proxygroup.update",
```

```
"params": {
"proxy_groupid": "5",
"min_online": "3"
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"proxy_groupids": [
"5"
]
},
"id": 1
}
```

Sorgente

CProxyGroup::update() in *ui/include/classes/api/services/CProxyGroup.php*.

Gruppo di template

Questa classe è progettata per funzionare con i gruppi di template.

Riferimenti agli oggetti:

- Gruppo di template

Metodi disponibili:

- templategroup.create - crea nuovi gruppi di template
- templategroup.delete - elimina gruppi di template
- templategroup.get - recupera gruppi di template
- templategroup.massadd - aggiunge oggetti correlati ai gruppi di template
- templategroup.massremove - rimuove oggetti correlati dai gruppi di template
- templategroup.propagate - propaga i permessi ai sottogruppi dei gruppi di template
- templategroup.update - aggiorna gruppi di template

Oggetto gruppo di template

I seguenti oggetti sono direttamente correlati all'API `templategroup`.

Gruppo di template

L'oggetto gruppo di template ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
groupid	ID	ID del gruppo di template. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del gruppo di template. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
uuid	string	Identificatore univoco universale, utilizzato per collegare i gruppi di template importati a quelli già esistenti. Generato automaticamente, se non specificato.

templategroup.create

Descrizione

`object templategroup.create(object/array templateGroups)`

Questo metodo consente di creare nuovi gruppi di template.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object/array)` Gruppi di template da creare. Il metodo accetta gruppi di template con le proprietà standard dei gruppi di template.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID del modello creato gruppi nella proprietà `groupids`. L'ordine degli ID restituiti corrisponde all'ordine dei gruppi di modelli passati.

Esempi

Creazione di un gruppo di template

Creare un gruppo di template chiamato "Templates/Databases".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::create() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Descrizione

`object templategroup.delete(array templateGroupIds)`

Questo metodo consente di eliminare i gruppi di template.

Un gruppo di template non può essere eliminato se contiene template che appartengono solo a questo gruppo.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

"(array)" ID dei gruppi di modelli da eliminare.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID del modello eliminato gruppi nella proprietà `groupids`.

Esempi

Eliminazione di più gruppi di template

Elimina due gruppi di template.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::delete() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

Descrizione

`integer/array templategroup.get(object parameters)`

Il metodo consente di recuperare i gruppi di template in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
graphids	ID/array	Restituisce solo i gruppi di template che contengono template con i grafici specificati.
groupids	ID/array	Restituisce solo i gruppi di template con gli ID dei gruppi di template specificati.
templateids	ID/array	Restituisce solo i gruppi di template che contengono i template specificati.

Parametro	Type	Descrizione
triggerids	ID/array	Restituisce solo i gruppi di template che contengono template con i trigger specificati.
with_graphs	boolean	Restituisce solo i gruppi di template che contengono template con grafici.
with_graph_prototypes	boolean	Restituisce solo i gruppi di template che contengono template con prototipi di grafico.
with_httptests	boolean	Restituisce solo i gruppi di template che contengono template con controlli web.
with_items	boolean	Restituisce solo i gruppi di template che contengono template con item. Sovrascrive i parametri `with_simple_graph_items`.
with_item_prototypes	boolean	Restituisce solo i gruppi di template che contengono template con prototipi di item. Sovrascrive il parametro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Restituisce solo i gruppi di template che contengono template con prototipi di item, abilitati per la creazione e con tipo di informazione numerico.
with_simple_graph_items	boolean	Restituisce solo i gruppi di template che contengono template con item numerici.
with_templates	boolean	Restituisce solo i gruppi di template che contengono template.
with_triggers	boolean	Restituisce solo i gruppi di template che contengono template con trigger.
selectTemplates	query	Restituisce una proprietà `templates` con i template che appartengono al gruppo di template. Supporta `count`.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectTemplates` - i risultati saranno ordinati per `template`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `groupid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei dati per nome

Recupera tutti i dati relativi a due gruppi di template denominati ”Templates/Databases” e ”Templates/Modules”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

Guarda anche

- Modello

Fonte

CTemplateGroup::get() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Descrizione

`object templategroup.massadd(object parameters)`

Questo metodo consente di aggiungere simultaneamente più oggetti correlati a tutti i gruppi di template specificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID dei gruppi di template da aggiornare e gli oggetti da aggiungere a tutti i gruppi di template.

Il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
groups	object/array	Gruppi di template da aggiornare. I gruppi di template devono avere definita solo la proprietà `groupid`. Comportamento dei parametri: - *obbligatorio*
templates	object/array	Template da aggiungere a tutti i gruppi di template. I template devono avere definita solo la proprietà `templateid`. Comportamento dei parametri: - *obbligatorio*

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di template aggiornati nella proprietà `groupids`.

Esempi

Aggiunta di template ai gruppi di template

Aggiungere due template ai gruppi di template con ID 12 e 13.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

Guarda anche

- Modello

Fonte

CTemplateGroup::massAdd() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Descrizione

`object templategroup.massremove(object parameters)`

Questo metodo consente di rimuovere gli oggetti correlati da più gruppi di template.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID dei gruppi di template da aggiornare e gli oggetti che devono essere rimossi.

Parametro	Type	Descrizione
groupids	ID/array	ID dei gruppi di template da aggiornare. Comportamento del parametro: - *obbligatorio*
templateids	ID/array	ID dei template da rimuovere da tutti i gruppi di template. Comportamento del parametro: - *obbligatorio*

Restituisce i valori

"(object)" Restituisce un oggetto contenente gli ID del modello aggiornato gruppi nella proprietà `groupids`.

Esempi

Rimozione di template dai gruppi di template

Rimuovere due template dai gruppi di template specificati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
```

```
        ]
    },
    "id": 1
}
```

Sorgente

CTemplateGroup::massRemove() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Descrizione

`object templategroup.propagate(object parameters)`

Questo metodo consente di applicare i permessi a tutti i sottogruppi dei gruppi di template.

Note:
Questo metodo è disponibile solo per i tipi di utente *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
groups	object/array	Gruppi di template da propagare. I gruppi di template devono avere definita solo la proprietà `groupid`. Comportamento del parametro: - *obbligatorio*
permissions	boolean	Impostare `true` se è necessario propagare i permessi. Comportamento del parametro: - *obbligatorio*

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di template propagati

nella proprietà `groupids`.

Esempi

Propagazione dei permessi del gruppo template ai relativi sottogruppi.

Propaga i permessi del gruppo template ai relativi sottogruppi.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

Vedi anche

- templategroup.update
- templategroup.massadd
- Template

Fonte

CTemplateGroup::propagate() in

ui/include/classes/api/services/CTemplateGroup.php.

templategroup.update

Descrizione

`object templategroup.update(object/array templateGroups)`

Questo metodo consente di aggiornare i template group esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del gruppo di template da aggiornare.

La proprietà `groupid` deve essere definita per ciascun gruppo di template, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate, tutte le altre rimarranno invariate.

Restituisce i valori

"(object)" Restituisce un oggetto contenente gli ID del modello aggiornato gruppi nella proprietà `groupids`.

Esempi

Rinominare un gruppo di template

Rinominare un gruppo di template in "Templates/Databases"

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::update() in *ui/include/classes/api/services/CTemplateGroup.php*.

Gruppo host

Questa classe è progettata per funzionare con i gruppi host.

Riferimenti agli oggetti:

- Gruppo host

Metodi disponibili:

- hostgroup.create - crea nuovi gruppi host
- hostgroup.delete - elimina gruppi host
- hostgroup.get - recupera gruppi host
- hostgroup.massadd - aggiunge oggetti correlati ai gruppi host
- hostgroup.massremove - rimuove oggetti correlati dai gruppi host
- hostgroup.propagate - propaga autorizzazioni e filtri tag ai sottogruppi dei gruppi host
- hostgroup.update - aggiorna gruppi host

Oggetto gruppo di host

I seguenti oggetti sono direttamente correlati all'API `hostgroup`.

Gruppo host

L'oggetto gruppo host ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
groupid	ID	ID del gruppo host. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del gruppo host. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
flags	integer	Origine del gruppo host. Valori possibili: 0 - un gruppo host normale; 4 - un gruppo host individuato automaticamente. Comportamento della proprietà: - *sola lettura*
uuid	string	Identificatore univoco universale, utilizzato per collegare i gruppi host importati a quelli già esistenti. Generato automaticamente, se non specificato.

hostgroup.create

Descrizione

`object hostgroup.create(object/array hostGroups)`

Questo metodo consente di creare nuovi gruppi di host.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Gruppi host da creare.

Il metodo accetta gruppi host con le proprietà standard dei gruppi host.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di host creati nella proprietà `groupids`. L'ordine degli ID restituiti corrisponde all'ordine dei gruppi di host passati.

Esempi

Creare un gruppo di host

Crea un gruppo di host chiamato "Linux servers".

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "hostgroup.create",
"params": {
"name": "Linux servers"
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"groupids": [
"107819"
]
},
"id": 1
}
```

Sorgente

CHostGroup::create() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

Descrizione

`object hostgroup.delete(array hostGroupIds)`

Questo metodo consente di eliminare i gruppi host.

Un gruppo host non può essere eliminato se:

- contiene host che appartengono solo a questo gruppo;
- è contrassegnato come interno;
- è utilizzato da un host prototype;
- è utilizzato in uno script globale;
- è utilizzato in una condizione di correlazione.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei gruppi di host da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di host eliminati nella proprietà `groupids`.

Esempi

Eliminare molteplici gruppi di host

Elimina due gruppi di host.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "hostgroup.delete",
"params": [
"107824",
"107825"
],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"groupids": [
"107824",
"107825"
]
},
"id": 1
}
```

Sorgente

CHostGroup::delete() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

Descrizione

`integer/array hostgroup.get(object parameters)`

Il metodo consente di recuperare i gruppi di host in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
graphids	ID/array	Restituisce solo i gruppi host che contengono host con i grafici specificati.
groupids	ID/array	Restituisce solo i gruppi host con gli ID dei gruppi host specificati.
hostids	ID/array	Restituisce solo i gruppi host che contengono gli host specificati.
maintenanceids	ID/array	Restituisce solo i gruppi host interessati dalle maintenance specificate.
triggerids	ID/array	Restituisce solo i gruppi host che contengono host con i trigger specificati.

Parametro	Tipo	Descrizione
with_graphs	boolean	Restituisce solo i gruppi host che contengono host con grafici.
with_graph_prototypes	boolean	Restituisce solo i gruppi host che contengono host con prototipi di grafico.
with_hosts	boolean	Restituisce solo i gruppi host che contengono host.
with_httptests	boolean	Restituisce solo i gruppi host che contengono host con controlli web. Sovrascrive il parametro `with_monitored_httptests`.
with_items	boolean	Restituisce solo i gruppi host che contengono host con item. Sovrascrive i parametri `with_monitored_items` e `with_simple_graph_items`.
with_item_prototypes	boolean	Restituisce solo i gruppi host che contengono host con prototipi di item. Sovrascrive il parametro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Restituisce solo i gruppi host che contengono host con prototipi di item abilitati per la creazione e con tipo di informazione numerico.
with_monitored_httptests	boolean	Restituisce solo i gruppi host che contengono host con controlli web abilitati.
with_monitored_hosts	boolean	Restituisce solo i gruppi host che contengono host monitorati.
with_monitored_items	boolean	Restituisce solo i gruppi host che contengono host con item abilitati. Sovrascrive il parametro `with_simple_graph_items`.
with_monitored_triggers	boolean	Restituisce solo i gruppi host che contengono host con trigger abilitati. Anche tutti gli item utilizzati nel trigger devono essere abilitati.
with_simple_graph_items	boolean	Restituisce solo i gruppi host che contengono host con item numerici.
with_triggers	boolean	Restituisce solo i gruppi host che contengono host con trigger. Sovrascrive il parametro `with_monitored_triggers`.
selectDiscoveryRules	query	Restituisce una proprietà `discoveryRules` con le regole LLD che hanno rilevato il gruppo host.
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con gli oggetti di discovery del gruppo host. Ogni oggetto di discovery del gruppo host è un prototipo di gruppo host collegato al gruppo host rilevato e ha le seguenti proprietà: `parent_group_prototypeid` - `(ID)` ID del prototipo di gruppo host da cui è stato rilevato il gruppo host; `name` - `(string)` nome del prototipo di gruppo host; `ts_delete` - `(timestamp)` momento in cui il gruppo host non più rilevato verrà eliminato; `status` - `(int)` stato della discovery del gruppo host: 0 - *(predefinito)* il gruppo host è rilevato, 1 - il gruppo host non è più rilevato.
selectHostPrototypes	query	Restituisce una proprietà `hostPrototypes` con i prototipi host che hanno rilevato questo gruppo host.
selectHosts	query	Restituisce una proprietà `hosts` con gli host che appartengono al gruppo host. Supporta `count`.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectHosts` - i risultati saranno ordinati per `host`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `groupid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

Parametro	Tipo	Descrizione
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroupDiscoveries	query	Restituisce una proprietà `groupDiscoveries` con gli oggetti di discovery del gruppo host. Ogni oggetto di discovery del gruppo host è un prototipo di gruppo host collegato al gruppo host rilevato. Questa query è **deprecata**, usare invece `selectDiscoveryData`.

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`

Esempi

Recupero dei dati per nome

Recupera tutti i dati relativi a due gruppi di host denominati ”Zabbix servers” e ”Linux servers”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

Vedi anche

- Host

Sorgente

CHostGroup::get() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

Descrizione

`object hostgroup.massadd(object parameters)`

Questo metodo consente di aggiungere simultaneamente più oggetti correlati a tutti i gruppi host specificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID dei gruppi host da aggiornare e gli oggetti da aggiungere a tutti i gruppi host.

Il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groups	object/array	Gruppi host da aggiornare. I gruppi host devono avere definita solo la proprietà `groupid`. Comportamento dei parametri: - *obbligatorio*
hosts	object/array	Host da aggiungere a tutti i gruppi host. Gli host devono avere definita solo la proprietà `hostid`.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di host aggiornati nella proprietà `groupids`.

Esempi

Aggiungere host ai gruppi di host

Aggiunge due host ai gruppi host con ID 5 e 6.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "hostgroup.massadd",
"params": {
"groups": [
{
"groupid": "5"
},
{
"groupid": "6"
}
],
"hosts": [
{
"hostid": "30050"
},
{
"hostid": "30001"
```

```
}
]
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"groupids": [
"5",
"6"
]
},
"id": 1
}
```

Vedi anche

- Host

Sorgente

CHostGroup::massAdd() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

Descrizione

`object hostgroup.massremove(object parameters)`

Questo metodo consente di rimuovere gli oggetti correlati da più gruppi host.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID dei gruppi di host da aggiornare e gli oggetti che devono essere rimossi.

Parametro	Tipo	Descrizione
groupids	ID/array	ID dei gruppi di host da aggiornare. Comportamento del parametro: - *obbligatorio*
hostids	ID/array	ID degli host da rimuovere da tutti i gruppi di host.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di host aggiornati nella proprietà `groupids`.

Esempi

Rimuovere host dai gruppi di host

Rimuove due host dai gruppi di host specificati.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "hostgroup.massremove",
"params": {
"groupids": [
```

```
"5",
"6"
],
"hostids": [
"30050",
"30001"
]
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"groupids": [
"5",
"6"
]
},
"id": 1
}
```

Sorgente

CHostGroup::massRemove() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

Descrizione

`object hostgroup.propagate(object parameters)`

Questo metodo consente di applicare permessi e filtri dei tag a tutti i sottogruppi di un host group.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
groups	object/array	Gruppi di host da propagare. I gruppi di host devono avere la proprietà `groupid` definita. Comportamento dei parametri: - *obbligatorio*
permissions	boolean	Impostare su "true" per propagare i permessi. Comportamento dei parametri: - *obbligatorio* se `tag_filters` non è impostato
tag_filters	boolean	Impostare su "true" per propagare i filtri tag. Comportamento dei parametri: - *obbligatorio* se `permissions` non è impostato

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di host propagati sotto la proprietà `groupids`.

Esempi

Propagazione dei permessi del gruppo host e dei filtri tag ai suoi sottogruppi.

Propaga i permessi del gruppo host e i filtri tag ai suoi sottogruppi.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

Vedi anche

- hostgroup.update
- hostgroup.massadd
- Host

Sorgente

CHostGroup::propagate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

Descrizione

`object hostgroup.update(object/array hostGroups)`

Questo metodo consente di aggiornare i gruppi di host esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del gruppo host da aggiornare.

La proprietà `groupid` deve essere definita per ciascun gruppo host, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate, tutte le altre rimarranno invariate.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi di host aggiornati nella proprietà `groupids`.

Esempi

Rinominare un gruppo di host

Rinomina un gruppo di host in ”Linux hosts”.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "hostgroup.update",
"params": {
"groupid": "7",
"name": "Linux hosts"
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"groupids": [
"7"
]
},
"id": 1
}
```

Rinomina di più gruppi di host

Rinominare due gruppi di host per includere le informazioni sulla regione nei loro nomi.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

Sorgente

CHostGroup::update() in *ui/include/classes/api/services/CHostGroup.php*.

Gruppo utenti

Questa classe è progettata per funzionare con i gruppi utenti.

Riferimenti agli oggetti:

- Gruppo utenti
- Permesso
- Permesso basato su tag

Metodi disponibili:

- usergroup.create - crea nuovi gruppi utenti
- usergroup.delete - elimina gruppi utenti
- usergroup.get - recupera gruppi utenti
- usergroup.update - aggiorna gruppi utenti

Oggetto gruppo utenti

I seguenti oggetti sono direttamente correlati all'API `usergroup`.

Gruppo di utenti

L'oggetto gruppo di utenti dispone delle seguenti proprietà.

Property	Type	Description
usrgrpid	ID	ID del gruppo di utenti. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del gruppo di utenti. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
debug_mode	integer	Indica se la modalità di debug è abilitata o disabilitata. Valori possibili: 0 - disabilitata *(predefinito)*; 1 - abilitata.
gui_access	integer	Metodo di autenticazione del frontend per gli utenti del gruppo. Valori possibili: 0 - usa il metodo di autenticazione predefinito del sistema *(predefinito)*; 1 - usa l'autenticazione interna; 2 - usa l'autenticazione LDAP; 3 - disabilita l'accesso al frontend.
mfa_status	integer	Indica se MFA è abilitata o disabilitata per gli utenti del gruppo. Valori possibili: 0 - disabilitata (per tutti i metodi MFA configurati); 1 - abilitata (per tutti i metodi MFA configurati).
mfaid	ID	Metodo MFA utilizzato per gli utenti del gruppo. Comportamento della proprietà: - *supportata* per gli utenti di tipo *Super admin* - *supportata* se `mfa_status` dell'oggetto Authentication è impostato su "Enabled"
proxy_mode	integer	Modalità per gestire l'accesso ai proxy non inclusi nei gruppi di proxy. Valori possibili: 0 - elenco di negazione; 1 - elenco di autorizzazione *(predefinito)*. Comportamento della proprietà: - *supportata* per gli utenti di tipo *Super admin*.

Property	Type	Description
proxy_group_mode	integer	Modalità per gestire l'accesso ai gruppi di proxy e ai relativi proxy. Valori possibili: 0 - elenco di negazione; 1 - elenco di autorizzazione *(predefinito)*. Comportamento della proprietà: - *supportata* per gli utenti di tipo *Super admin*.
users_status	integer	Indica se il gruppo di utenti è abilitato o disabilitato. Per gli utenti deprovisioned, il gruppo di utenti non può essere abilitato. Valori possibili: 0 - abilitato *(predefinito)*; 1 - disabilitato.
userdirectoryid	ID	ID della directory utenti utilizzata per l'autenticazione. Comportamento della proprietà: - *supportata* per gli utenti di tipo *Super admin* - *supportata* se `gui_access` è impostato su "usa il metodo di autenticazione predefinito del sistema" o "usa l'autenticazione LDAP"

Permesso

L'oggetto permission ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
id	ID	ID del gruppo host o del gruppo template a cui aggiungere il permesso. Comportamento della proprietà: - *obbligatorio*
permission	integer	Livello di accesso al gruppo host o al gruppo template. Valori possibili: 0 - accesso negato; 2 - accesso in sola lettura; 3 - accesso in lettura-scrittura. Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione

Permesso basato su tag

L'oggetto del permesso basato su tag ha le seguenti proprietà.

Proprietà	Type	Descrizione
groupid	ID	ID del gruppo host a cui aggiungere il permesso. Comportamento della proprietà: - *obbligatorio*
tag	string	Nome del tag.
value	string	Valore del tag.

APM-based permission

The APM-based permission object has the following properties.

Property	Type	Description
key	string	APM resource attribute key. Property behavior: - *required*
value	string	APM resource attribute value.
signal_type	integer	APM resource attribute category. Possible values: -1 - *(default)* All; 0 - Traces; 1 - Metrics; 2 - Logs.

usergroup.create

Descrizione

`object usergroup.create(object/array userGroups)`

Questo metodo consente di creare nuovi gruppi di utenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Gruppi di utenti da creare.

Oltre alle proprietà standard dei gruppi di utenti, il metodo accetta i seguenti parametri.

Parameter	Type	Description
hostgroup_rights	object/array	Autorizzazioni per i gruppi di host da assegnare al gruppo di utenti.
proxies	object/array	proxy da collegare al gruppo di utenti. I proxy devono avere definita solo la proprietà `proxyid`.
proxy_groups	object/array	Gruppi di proxy da collegare al gruppo di utenti. I gruppi di proxy devono avere definita solo la proprietà `proxy_groupid`.
templategroup_rights	object/array	Autorizzazioni per i gruppi di template da assegnare al gruppo di utenti.
tag_filters	array	Autorizzazioni basate sui tag da assegnare al gruppo di utenti.
users	object/array	Utenti da aggiungere al gruppo di utenti. L'utente deve avere definita solo la proprietà `userid`.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei gruppi utenti creati nella proprietà `usrgrpids`. L'ordine degli ID restituiti corrisponde all'ordine dei gruppi utenti passati.

Esempi

Creazione di un gruppo utenti

Creare un gruppo utenti *Operation managers* con accesso negato al gruppo host "2" e aggiungervi un utente.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
```

```
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

Guarda anche

- Permessi

Sorgente

CUserGroup::create() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

Descrizione

`object usergroup.delete(array userGroupIds)`

Questo metodo consente di eliminare i gruppi di utenti.

Attention:
Il gruppo di utenti Deprovisioned (il gruppo di utenti specificato per `disabled_usrgrpid` in Authentication) non può essere eliminato.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Parametri

"(array)" ID dei gruppi utenti da eliminare.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID dell'utente eliminato gruppi sotto la proprietà `usrgrids`.

Esempi

Eliminazione di più gruppi di utenti

Eliminare due gruppi di utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
```

```
        "21"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

Fonte

CUserGroup::delete() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

Descrizione

`integer/array usergroup.get(object parameters)`

Il metodo consente di recuperare i gruppi di utenti in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
mfaids	ID/array	Restituisce solo i gruppi di utenti con i metodi MFA specificati. Comportamento del parametro: - *supportato* per gli utenti di tipo *Super admin*
mfa_status	integer	Restituisce solo i gruppi di utenti con lo stato MFA specificato. Per un elenco degli stati supportati, consultare la pagina dei gruppi di utenti.
proxyids	ID/array	Restituisce solo i gruppi di utenti che contengono i proxy specificati. Comportamento del parametro: - *supportato* per gli utenti di tipo *Super admin*.
proxy_groupids	ID/array	Restituisce solo i gruppi di utenti che contengono i gruppi di proxy specificati. Comportamento del parametro: - *supportato* per gli utenti di tipo *Super admin*.
selectProxies	ID/array	Restituisce una proprietà `proxies` con i proxy del gruppo di utenti. Comportamento del parametro: - *supportato* per gli utenti di tipo *Super admin*.

Parametro	Tipo	Descrizione
selectProxyGroups	ID/array	Restituisce una proprietà `proxy_groups` con i gruppi di proxy del gruppo di utenti. Comportamento del parametro: - *supportato* per gli utenti di tipo *Super admin*.
status	integer	Restituisce solo i gruppi di utenti con lo stato specificato. Per un elenco degli stati supportati, consultare la pagina dei gruppi di utenti.
userids	ID/array	Restituisce solo i gruppi di utenti che contengono gli utenti specificati.
usrgrpids	ID/array	Restituisce solo i gruppi di utenti con gli ID specificati.
selectTagFilters	query	Restituisce le autorizzazioni basate sui tag del gruppo di utenti nella proprietà `tag_filters`.
selectUsers	query	Restituisce gli utenti del gruppo di utenti nella proprietà `users`. Per le limitazioni basate sul tipo di utente, consultare `user.get`.
selectHostGroupRights	query	Restituisce le autorizzazioni del gruppo di utenti sui gruppi di host nella proprietà `hostgroup_rights`. Per un elenco dei livelli di accesso ai gruppi di host, consultare la pagina dei gruppi di utenti.
selectTemplateGroupRights	query	Restituisce le autorizzazioni del gruppo di utenti sui gruppi di template nella proprietà `templategroup_rights`. Per un elenco dei livelli di accesso ai gruppi di template, consultare la pagina dei gruppi di utenti.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni.
output	query	Proprietà dell'oggetto gruppo di utenti da restituire. Gli utenti di tipo *Admin* e *User* possono recuperare solo le seguenti proprietà: `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Predefinito: `extend`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `usrgrpid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei gruppi utenti abilitati

Recupera tutti i gruppi utenti abilitati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix administrators",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Guests",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Enabled debug mode",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "No access to the frontend",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Read only",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
```

```
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Deny",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        }
    ],
    "id": 1
}
```


Guarda anche

- Utente

Fonte

CUserGroup::get() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Descrizione

`object usergroup.update(object/array userGroups)`

Questo metodo consente di aggiornare i gruppi di utenti esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà dei gruppi di utenti da aggiornare.

La proprietà `usrgrpid` deve essere definita per ogni gruppo di utenti; tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, mentre tutte le altre rimarranno invariate.

Oltre alle proprietà standard dei gruppi di utenti, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
hostgroup_rights	object/array	Autorizzazioni per i gruppi di host, che sostituiranno le autorizzazioni attualmente assegnate al gruppo di utenti.
proxies	object/array	proxy da collegare al gruppo di utenti. I proxy devono avere definita solo la proprietà `proxyid`.
proxy_groups	object/array	Gruppi di proxy da collegare al gruppo di utenti. I gruppi di proxy devono avere definita solo la proprietà `proxy_groupid`.
templategroup_rights	object/array	Autorizzazioni per i gruppi di template, che sostituiranno le autorizzazioni attualmente assegnate al gruppo di utenti.
tag_filters	array	Autorizzazioni basate sui tag che sostituiranno le autorizzazioni attualmente assegnate al gruppo di utenti.
users	object/array	Utenti che sostituiranno gli utenti attualmente assegnati al gruppo di utenti. L'utente deve avere definita solo la proprietà `userid`.】【。

Restituisce valori

"(object)" Restituisce un oggetto contenente gli ID dell'utente aggiornato gruppi sotto la proprietà `usrgrids`.

Esempi

Abilitazione di un gruppo utenti e aggiornamento dei permessi

Abilitare un gruppo utenti e fornirgli l'accesso in lettura-scrittura ai gruppi host "2" e "4".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

Guarda anche

- Permessi

Fonte

CUserGroup::update() in *ui/include/classes/api/services/CUserGroup.php*.

Host

Questa classe è progettata per funzionare con gli host.

Riferimenti agli oggetti:

- Host
- Inventario host
- Tag host

Metodi disponibili:

- host.create - crea nuovi host
- host.delete - elimina host
- host.get - recupera host
- host.massadd - aggiunge oggetti correlati agli host
- host.massremove - rimuove oggetti correlati dagli host
- host.update - aggiorna host

Oggetto host

I seguenti oggetti sono direttamente correlati all'API host.

Host

L'oggetto host ha le seguenti proprietà.

Property	Type	Description
hostid	ID	ID dell'host. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
host	string	Nome tecnico dell'host. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
description	text	Descrizione dell'host.
flags	integer	Origine dell'host. Valori possibili: 0 - un host semplice; 4 - un host convertito da un prototipo. Comportamento della proprietà: - *sola lettura*
inventory_mode	integer	Modalità di popolamento dell'inventario dell'host. Valori possibili: -1 - *(predefinita)* disabilitata; 0 - manuale; 1 - automatica.
ipmi_authtype	integer	Algoritmo di autenticazione IPMI. Valori possibili: -1 - *(predefinito)* predefinito; 0 - nessuno; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+.
ipmi_password	string	Password IPMI.
ipmi_privilege	integer	Livello di privilegio IPMI. Valori possibili: 1 - callback; 2 - *(predefinito)* utente; 3 - operatore; 4 - amministratore; 5 - OEM.
ipmi_username	string	Nome utente IPMI.
maintenance_from	timestamp	Ora di inizio della manutenzione effettiva. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
maintenance_status	integer	Stato della manutenzione effettiva. Valori possibili: 0 - *(predefinito)* nessuna manutenzione; 1 - manutenzione attiva. Comportamento della proprietà: - *sola lettura*
maintenance_type	integer	Tipo di manutenzione effettiva. Valori possibili: 0 - *(predefinito)* manutenzione con raccolta dati; 1 - manutenzione senza raccolta dati. Comportamento della proprietà: - *sola lettura*
maintenanceid	ID	ID della manutenzione attualmente attiva sull'host. Comportamento della proprietà: - *sola lettura*
name	string	Nome visibile dell'host. Predefinito: valore della proprietà `host`.
monitored_by	integer	Origine utilizzata per monitorare l'host. Valori possibili: 0 - *(predefinito)* Zabbix server; 1 - proxy; 2 - gruppo di proxy. Comportamento della proprietà: - *obbligatoria* se `select_server_for_monitoring` è impostato su "Disabilitato".
proxyid	ID	ID del proxy utilizzato per monitorare l'host. Comportamento della proprietà: - *obbligatoria* se `monitored_by` è impostato su "Proxy"
proxy_groupid	ID	ID del gruppo di proxy utilizzato per monitorare l'host. Comportamento della proprietà: - *obbligatoria* se `monitored_by` è impostato su "Gruppo di proxy"
status	integer	Stato e funzione dell'host. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.
tls_connect	integer	Connessioni all'host. Valori possibili: 1 - *(predefinito)* nessuna crittografia; 2 - PSK; 4 - certificato.
tls_accept	integer	Connessioni dall'host. Valori bitmap possibili: 1 - *(predefinito)* nessuna crittografia; 2 - PSK; 4 - certificato. Questo è un campo con maschera di bit; è accettabile qualsiasi somma dei valori bitmap possibili (ad esempio, 6 per PSK e certificato).

Property	Type	Description
tls_issuer	string	Emittente del certificato.
tls_subject	string	Oggetto del certificato.
tls_psk_identity	string	Identità PSK; deve essere associata a una sola PSK (tra autoregistrazione, host e proxy). Non includere informazioni sensibili nell'identità PSK, poiché viene inviata non crittografata sulla rete per informare il destinatario quale PSK utilizzare. Comportamento della proprietà: - *sola scrittura* - *obbligatoria* se `tls_connect` è impostato su "PSK" oppure se `tls_accept` contiene il bit "PSK"
tls_psk	string	Chiave precondivisa (PSK); deve contenere almeno 32 cifre esadecimali. Comportamento della proprietà: - *sola scrittura* - *obbligatoria* se `tls_connect` è impostato su "PSK" oppure se `tls_accept` contiene il bit "PSK"
active_available	integer	Stato di disponibilità dell'interfaccia active dell'host. Valori possibili: 0 - lo stato dell'interfaccia è sconosciuto; 1 - l'interfaccia è disponibile; 2 - l'interfaccia non è disponibile. Comportamento della proprietà: - *sola lettura*
assigned_proxyid	ID	ID del proxy assegnato da Zabbix server, se l'host è monitorato da un gruppo di proxy. Comportamento della proprietà: - *sola lettura*

Inventario host

L'oggetto inventario host ha le seguenti proprietà.

Note:
Ogni proprietà ha il proprio numero ID univoco, che viene utilizzato per associare i campi dell'inventario host agli item.

ID	Proprietà	Type	Descrizione	Lunghezza massima
4	alias	string	Alias.	128 caratteri
11	asset_tag	string	Tag asset.	64 caratteri
28	chassis	string	Chassis.	64 caratteri
23	contact	string	Persona di contatto.	65535 caratteri per database SQL
32	contract_number	string	Numero di contratto.	64 caratteri
47	date_hw_decomm	string	Data di dismissione dell'HW.	64 caratteri
46	date_hw_expiry	string	Data di scadenza della manutenzione HW.	64 caratteri
45	date_hw_install	string	Data di installazione dell'HW.	64 caratteri
44	date_hw_purchase	string	Data di acquisto dell'HW.	64 caratteri
34	deployment_status	string	Stato di distribuzione.	64 caratteri
14	hardware	string	Hardware.	255 caratteri
15	hardware_full	string	Dettagli hardware.	65535 caratteri per database SQL
39	host_netmask	string	Maschera di sottorete del host.	39 caratteri
38	host_networks	string	Reti del host.	65535 caratteri per database SQL
40	host_router	string	Router del host.	39 caratteri
30	hw_arch	string	Architettura HW.	32 caratteri

ID	Proprietà	Type	Descrizione	Lunghezza massima
33	installer_name	string	Nome dell'installatore.	64 caratteri
24	location	string	Posizione.	65535 caratteri per database SQL
25	location_lat	string	Latitudine della posizione.	16 caratteri
26	location_lon	string	Longitudine della posizione.	16 caratteri
12	macaddress_a	string	Indirizzo MAC A.	64 caratteri
13	macaddress_b	string	Indirizzo MAC B.	64 caratteri
29	model	string	Modello.	64 caratteri
3	name	string	Nome.	128 caratteri
27	notes	string	Note.	65535 caratteri per database SQL
41	oob_ip	string	Indirizzo IP OOB.	39 caratteri
42	oob_netmask	string	Maschera di sottorete del host OOB.	39 caratteri
43	oob_router	string	Router OOB.	39 caratteri
5	os	string	Nome del SO.	128 caratteri
6	os_full	string	Nome dettagliato del SO.	255 caratteri
7	os_short	string	Nome breve del SO.	128 caratteri
61	poc_1_cell	string	Numero di cellulare del POC primario.	64 caratteri
58	poc_1_email	string	Email primaria.	128 caratteri
57	poc_1_name	string	Nome del POC primario.	128 caratteri
63	poc_1_notes	string	Note del POC primario.	65535 caratteri per database SQL
59	poc_1_phone_a	string	Telefono A del POC primario.	64 caratteri
60	poc_1_phone_b	string	Telefono B del POC primario.	64 caratteri
62	poc_1_screen	string	Nome schermata del POC primario.	64 caratteri
68	poc_2_cell	string	Numero di cellulare del POC secondario.	64 caratteri
65	poc_2_email	string	Email secondaria.	128 caratteri
64	poc_2_name	string	Nome del POC secondario.	128 caratteri
70	poc_2_notes	string	Note del POC secondario.	65535 caratteri per database SQL
66	poc_2_phone_a	string	Telefono A del POC secondario.	64 caratteri
67	poc_2_phone_b	string	Telefono B del POC secondario.	64 caratteri
69	poc_2_screen	string	Nome schermata del POC secondario.	64 caratteri
8	serialno_a	string	Numero di serie A.	64 caratteri
9	serialno_b	string	Numero di serie B.	64 caratteri
48	site_address_a	string	Indirizzo del sito A.	128 caratteri
49	site_address_b	string	Indirizzo del sito B.	128 caratteri
50	site_address_c	string	Indirizzo del sito C.	128 caratteri
51	site_city	string	Città del sito.	128 caratteri
53	site_country	string	Paese del sito.	64 caratteri
56	site_notes	string	Note del sito.	65535 caratteri per database SQL
55	site_rack	string	Posizione rack del sito.	128 caratteri
52	site_state	string	Stato/regione del sito.	64 caratteri
54	site_zip	string	CAP/codice postale del sito.	64 caratteri
16	software	string	Software.	255 caratteri
18	software_app_a	string	Applicazione software A.	64 caratteri
19	software_app_b	string	Applicazione software B.	64 caratteri
20	software_app_c	string	Applicazione software C.	64 caratteri
21	software_app_d	string	Applicazione software D.	64 caratteri
22	software_app_e	string	Applicazione software E.	64 caratteri
17	software_full	string	Dettagli software.	65535 caratteri per database SQL
10	tag	string	Tag.	64 caratteri
1	type	string	Tipo.	64 caratteri
2	type_full	string	Dettagli del tipo.	64 caratteri
35	url_a	string	URL A.	2048 caratteri
36	url_b	string	URL B.	2048 caratteri
37	url_c	string	URL C.	2048 caratteri
31	vendor	string	Fornitore.	64 caratteri

Tag host

L'oggetto tag host ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tag	string	Nome del tag host. Comportamento della proprietà: - *obbligatoria*
value	string	Valore del tag host. Comportamento della proprietà: - *sola lettura*.
automatic	integer	Tipo di tag host. Valori possibili: 0 - *(predefinito)* manuale (tag creato dall'utente); 1 - automatico (tag creato dal low-level discovery) Comportamento della proprietà: - *sola lettura*
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template. Comportamento della proprietà: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.

host.create

Descrizione

`object host.create(object/array hosts)`

Questo metodo consente di creare nuovi host.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Host da creare.

Oltre alle proprietà standard dell'host, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groups	object/array	Gruppi di host a cui aggiungere l'host. I gruppi di host devono avere definita solo la proprietà `groupid`. Comportamento del parametro: - *obbligatorio*
interfaces	object/array	Interfacce da creare per l'host.
tags	object/array	Tag dell'host.
templates	object/array	Template da collegare all'host. I template devono avere definita solo la proprietà `templateid`.
macros	object/array	Macro utente da creare per l'host.
inventory	object	Proprietà dell'inventario dell'host.

Valori restituiti

(object) Restituisce un oggetto contenente gli ID degli host creati nella proprietà hostids. L'ordine degli ID restituiti corrisponde all'ordine degli host passati.

Esempi

Creazione di un host

Creare un host chiamato "Linux server" con un'interfaccia IP e tag, aggiungerlo a un gruppo, collegargli un template e impostare gli indirizzi MAC nell' inventario dell'host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
                "value": "123321"
            },
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

Creazione di un host con interfaccia SNMP

Creare un host denominato "SNMP host" con un'interfaccia SNMPv3 con dettagli.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

Creazione di un host con crittografia PSK

Crea un host chiamato "PSK host" con la crittografia PSK configurata. Nota che l'host deve essere preconfigurato per usare PSK.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Creazione di un host monitorato da un proxy

Creare un host che sarà monitorato dal proxy con ID ”1”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
```

```
            "10591"
        ]
    },
    "id": 1
}
```

Creazione di un host monitorato da un gruppo di proxy

Creare un host che sarà monitorato dal gruppo di proxy con ID "1".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

Vedi anche

- Gruppo host
- Template
- Macro utente
- Interfaccia host
- Inventario host
- Tag host
- Proxy
- Gruppo proxy

Sorgente

CHost::create() in *ui/include/classes/api/services/CHost.php*.

host.delete

Descrizione

`object host.delete(array hosts)`

Questo metodo consente di eliminare gli host.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID degli host da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli host eliminati nella proprietà `hostids`.

Esempi

Eliminare molteplici host

Elimina due host.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "host.delete",
"params": [
"13",
"32"
],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"hostids": [
"13",
"32"
]
},
"id": 1
}
```

Sorgente

CHost::delete() in *ui/include/classes/api/services/CHost.php*.

host.get

Descrizione

`integer/array host.get(object parameters)`

Il metodo consente di recuperare gli host in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
groupids	ID/array	Restituisce solo gli host che appartengono ai gruppi specificati.
dserviceids	ID/array	Restituisce solo gli host correlati ai servizi rilevati specificati.
graphids	ID/array	Restituisce solo gli host che hanno i grafici specificati.
hostids	ID/array	Restituisce solo gli host con gli ID host specificati.
httptestids	ID/array	Restituisce solo gli host che hanno i controlli web specificati.
interfaceids	ID/array	Restituisce solo gli host che utilizzano le interfacce specificate.
itemids	ID/array	Restituisce solo gli host che hanno gli item specificati.

Parametro	Tipo	Descrizione
maintenanceids	ID/array	Restituisce solo gli host interessati dalle manutenzioni specificate.
monitored_hosts	flag	Restituisce solo gli host monitorati.
proxyids	ID/array	Restituisce solo gli host monitorati dai proxy specificati.
proxy_groupids	ID/array	Restituisce solo gli host monitorati dai gruppi di proxy specificati.
templated_hosts	flag	Restituisce sia host che template.
templateids	ID/array	Restituisce solo gli host collegati ai template specificati.
triggerids	ID/array	Restituisce solo gli host che hanno i trigger specificati.
with_items	flag	Restituisce solo gli host che hanno item. Sovrascrive i parametri `with_monitored_items` e `with_simple_graph_items`.
with_item_prototypes	flag	Restituisce solo gli host che hanno prototipi di item. Sovrascrive il parametro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Restituisce solo gli host che hanno prototipi di item abilitati per la creazione e con tipo di informazione numerico.
with_graphs	flag	Restituisce solo gli host che hanno grafici.
with_graph_prototypes	flag	Restituisce solo gli host che hanno prototipi di grafico.
with_httptests	flag	Restituisce solo gli host che hanno controlli web. Sovrascrive il parametro `with_monitored_httptests`.
with_monitored_httptests	flag	Restituisce solo gli host che hanno controlli web abilitati.
with_monitored_items	flag	Restituisce solo gli host che hanno item abilitati. Sovrascrive il parametro `with_simple_graph_items`.
with_monitored_triggers	flag	Restituisce solo gli host che hanno trigger abilitati. Anche tutti gli item utilizzati nel trigger devono essere abilitati.
with_simple_graph_items	flag	Restituisce solo gli host che hanno item con tipo di informazione numerico.
with_triggers	flag	Restituisce solo gli host che hanno trigger. Sovrascrive il parametro `with_monitored_triggers`.
withProblemsSuppressed	boolean	Se impostato su `true`, restituisce solo gli host con problemi soppressi.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
severities	integer/array	Restituisce gli host che hanno solo problemi con le gravità specificate. Si applica solo se l'oggetto problema è un trigger.
tags	object/array	Restituisce solo gli host con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti gli host. Valori possibili di operator: 0 - *(predefinito)* Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
inheritedTags	boolean	Restituisce gli host che hanno i `tags` specificati anche in tutti i template collegati. Valori possibili: `true` - anche i template collegati devono avere i `tags` specificati; `false` - *(predefinito)* i tag dei template collegati vengono ignorati.

Parametro	Tipo	Descrizione
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di individuazione host. L'oggetto di individuazione host collega un host rilevato a un prototipo di host dal quale è stato rilevato. Ha le seguenti proprietà: `host` - `(string)` ID dell'host; `parent_hostid` - `(string)` ID del prototipo di host dal quale l'host è stato creato; `status` - `(int)` stato di individuazione dell'host: 0 - *(predefinito)* l'host è rilevato, 1 - l'host non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un host non più rilevato verrà eliminato; `ts_disable` - `(timestamp)` momento in cui un host non più rilevato verrà disabilitato; `disable_source` - `(int)` indicatore che specifica se l'host è stato disabilitato da una regola LLD o manualmente: 0 - *(predefinito)* disabilitato automaticamente, 1 - disabilitato da una regola LLD.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola di low-level discovery che ha creato l'host (dal prototipo di host nel monitoraggio VMware).
selectDiscoveryRules	query	Restituisce una proprietà `discoveryRules` con le regole LLD dell'host. Supporta `count`.
selectGraphs	query	Restituisce una proprietà `graphs` con i grafici dell'host. Supporta `count`.
selectHostGroups	query	Restituisce una proprietà `hostgroups` con i dati dei gruppi host a cui l'host appartiene.
selectHttpTests	query	Restituisce una proprietà `httpTests` con gli scenari web dell'host. Supporta `count`.
selectInterfaces	query	Restituisce una proprietà `interfaces` con le interfacce dell'host. Supporta `count`.
selectInventory	query	Restituisce una proprietà `inventory` con i dati dell'inventario dell'host.
selectItems	query	Restituisce una proprietà `items` con gli item dell'host. Supporta `count`.
selectMacros	query	Restituisce una proprietà `macros` con le macro dell'host.
selectParentTemplates	query	Restituisce una proprietà `parentTemplates` con i template a cui l'host è collegato. Oltre ai campi dell'oggetto Template, contiene `link_type` - `(integer)` il modo in cui il template è collegato all'host. Valori possibili: 0 - *(predefinito)* collegato manualmente; 1 - collegato automaticamente da LLD. Supporta `count`.
selectDashboards	query	Restituisce una proprietà `dashboards`. Supporta `count`.
selectTags	query	Restituisce una proprietà `tags` con i tag dell'host.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag presenti in tutti i template collegati all'host.
selectTriggers	query	Restituisce una proprietà `triggers` con i trigger dell'host. Supporta `count`.

Parametro	Tipo	Descrizione
selectValueMaps	query	Restituisce una proprietà `valuemaps` con le mappe di valori dell'host.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta le proprietà del tipo di dati `text`. Supporta proprietà aggiuntive: proprietà di Host interface.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectParentTemplates` - i risultati saranno ordinati per `host`; `selectInterfaces`; `selectItems` - ordinati per `name`; `selectDiscoveryRules` - ordinati per `name`; `selectTriggers` - ordinati per `description`; `selectGraphs` - ordinati per `name`; `selectDashboards` - ordinati per `name`.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Supporta solo le proprietà del tipo di dati `string` e `text`. Supporta proprietà aggiuntive: proprietà di Host interface.
searchInventory	object	Restituisce gli host che hanno dati di inventario corrispondenti al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Supporta solo le proprietà del tipo di dati `string` e `text`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Restituisce una proprietà `discoveries` con le regole di low-level discovery dell'host. Supporta `count`. Questa query è **deprecata**; utilizzare invece `selectDiscoveryRules`.

Parametro	Tipo	Descrizione
selectHostDiscovery	query	Restituisce una proprietà `hostDiscovery` con i dati dell'oggetto di individuazione host. Questa query è **deprecata**; utilizzare invece `selectDiscoveryData`.

Valori restituiti

`(intero/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei dati per nome

Recupera tutti i dati relativi a due host denominati "Zabbix server" e "Linux server".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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": "Il server di monitoraggio Zabbix.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
```

```
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```

Recupero dei gruppi di host

Recupera i gruppi di host di cui l'host "Zabbix server" è membro.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Recupero dei template collegati

Recupera gli ID e i nomi dei template collegati all'host "10084".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

Recupero degli host per template

Recupera gli host a cui è collegato il template "10001" (*Linux by Zabbix agent*).

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Ricerca per dati di inventario dell'host

Recupera gli host che contengono "Linux" nel campo "OS" dell'inventario dell'host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
```

```
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

Ricerca per tag host

Recupera gli host che hanno il tag ”host-name” uguale a ”linux-server”.

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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Recupera gli host che hanno questi tag non solo a livello di host ma anche nei loro template padre collegati.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
```

```
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Recupera gli host escludendo quelli con il tag ”component”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Ricerca di un host con tag e tag del template

Recupera un host con i tag e tutti i tag collegati ai template padre.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Ricerca degli host per gravità del problema

Recupera gli host che hanno problemi di livello ”Disaster”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

Recupera gli host che hanno problemi di livello ”Average” e ”High”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
```

```
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

Recupero degli item del host

Recupera gli item per il host "Zabbix server". La richiesta è limitata alle proprietà dell'item `itemid`, `name` e `status`. A causa delle grandi dimensioni della risposta, nell'esempio viene mostrato solo un sottoinsieme degli item.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Host name of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix agent ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Maximum number of open file descriptors",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maximum number of processes",
          "status": "0"
        },
        {
          "itemid": "69869",
          "name": "Interface enp0s3: Inbound packets discarded",
```

```
            "status": "0"
          },
          {
            "itemid": "69870",
            "name": "Interface enp0s3: Inbound packets with errors",
            "status": "0"
          },
          {
            "itemid": "69871",
            "name": "Interface enp0s3: Bits received",
            "status": "0"
          },
          {
            "itemid": "69872",
            "name": "Interface enp0s3: Outbound packets discarded",
            "status": "0"
          },
          {
            "itemid": "69873",
            "name": "Interface enp0s3: Outbound packets with errors",
            "status": "0"
          },
          {
            "itemid": "69874",
            "name": "Interface enp0s3: Bits sent",
            "status": "0"
          },
          {
            "itemid": "42253",
            "name": "Number of processes",
            "status": "0"
          },
          {
            "itemid": "42254",
            "name": "Number of running processes",
            "status": "0"
          },
          {
            "itemid": "42255",
            "name": "System boot time",
            "status": "0"
          }
        ]
      }
    ],
    "id": 1
}
```


Vedi anche

- Gruppo host
- Template
- Macro utente
- Interfaccia host
- Proxy
- Gruppo proxy

Sorgente

CHost::get() in *ui/include/classes/api/services/CHost.php*.

host.massadd

Descrizione

`object host.massadd(object parameters)`

Questo metodo consente di aggiungere simultaneamente più oggetti correlati a tutti gli host specificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID degli host da aggiornare e gli oggetti da aggiungere a tutti gli host.

Il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
hosts	object/array	Host da aggiornare. Gli host devono avere definita solo la proprietà `hostid`. Comportamento dei parametri: - *obbligatorio*
groups	object/array	Gruppi di host da aggiungere agli host specificati. I gruppi di host devono avere definita solo la proprietà `groupid`.
interfaces	object/array	Interfacce host da creare per gli host specificati.
macros	object/array	Macro utente da creare per gli host specificati.
templates	object/array	Template da collegare agli host specificati. I template devono avere definita solo la proprietà `templateid`.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli host aggiornati sotto la proprietà `hostids`.

Esempi

Aggiunta di macro

Aggiungere due nuove macro a due host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Descrizione di test"
            }
        ]
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

Vedi anche

- host.update
- Gruppo host
- Template
- Macro utente
- Interfaccia host

Fonte

CHost::massAdd() in *ui/include/classes/api/services/CHost.php*.

host.massremove

Descrizione

`object host.massremove(object parameters)`

Questo metodo consente di rimuovere gli oggetti correlati da più host.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID degli host da aggiornare e gli oggetti che devono essere rimossi.

Parametro	Type	Descrizione
hostids	ID/array	ID degli host da aggiornare. Comportamento del parametro: - *obbligatorio*
groupids	ID/array	ID dei gruppi di host da cui rimuovere gli host specificati.
interfaces	object/array	Interfacce host da rimuovere dagli host specificati. L'oggetto interfaccia host deve avere definite solo le proprietà `ip`, `dns` e `port`.
macros	string/array	Macro utente da eliminare dagli host specificati.
templateids	ID/array	ID dei template da scollegare dagli host specificati.
templateids_clear	ID/array	ID dei template da scollegare e rimuovere dagli host specificati.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli host aggiornati sotto la proprietà `hostids`.

Esempi

Scollegamento dei template

Scollega un template da due host ed elimina tutte le entità templateizzate.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

Vedi anche

- host.update
- Macro utente
- Interfaccia host

Fonte

CHost::massRemove() in *ui/include/classes/api/services/CHost.php*.

host.update

Descrizione

`object host.update(object/array hosts)`

Questo metodo consente di aggiornare gli host esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del host da aggiornare.

La proprietà `hostid` deve essere definita per ogni host, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate, tutte le altre rimarranno invariate.

Si noti tuttavia che l'aggiornamento del nome tecnico del host aggiornerà anche il nome visibile del host (se non specificato separatamente) con il valore del nome tecnico del host.

Oltre alle proprietà standard del host, il metodo accetta i seguenti parametri.

Parameter	Type	Description
groups	object/array	Gruppi di host per sostituire i gruppi di host correnti a cui il host appartiene. Tutti i gruppi di host non elencati nella richiesta verranno scollegati. I gruppi di host devono avere definita solo la proprietà `groupid`.

Parameter	Type	Description
interfaces	object/array	Interfacce del host per sostituire le interfacce correnti del host. Tutte le interfacce non elencate nella richiesta verranno rimosse.
tags	object/array	Tag del host per sostituire i tag correnti del host. Tutti i tag non elencati nella richiesta verranno rimossi.
inventory	object	Proprietà dell'inventario del host.
macros	object/array	Macro utente per sostituire le macro utente correnti. Tutte le macro non elencate nella richiesta verranno rimosse.
templates	object/array	Template per sostituire i template attualmente collegati. Tutti i template non elencati nella richiesta verranno solo scollegati. I template devono avere definita solo la proprietà `templateid`.
templates_clear	object/array	Template da scollegare e rimuovere dal host. I template devono avere definita solo la proprietà `templateid`.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli host aggiornati sotto la proprietà `hostids`.

Esempi

Abilitazione di un host

Abilitare il monitoraggio dell'host, ovvero impostarne lo stato a "0".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Abilitazione di più host

Abilitare il monitoraggio di due host, cioè impostare il loro stato su "0".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
```

```
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "id": 1
}
```

Scollegamento dei template

Scollega e rimuove due template da un host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Aggiornamento delle macro dell'host

Sostituisce tutte le macro dell'host con due nuove.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
```

```
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Descrizione aggiornata"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Aggiornamento dell'inventario dell'host

Modificare la modalità dell'inventario e aggiungere la posizione

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Aggiornamento dei tag del host

Sostituisci tutti i tag del host con uno nuovo.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
```

```
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Aggiornamento delle macro del host individuato

Convertire la macro "automatic" creata dalla regola di individuazione in "manual" e modificarne il valore in "new-value".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Aggiornamento della crittografia dell'host

Aggiornare l'host "10590" in modo che utilizzi la crittografia PSK solo per le connessioni dall'host al server Zabbix, e modificare l'identità PSK e la chiave PSK. Si noti che l'host deve essere preconfigurato per usare PSK.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Vedi anche

- host.massadd
- host.massremove
- Gruppo host
- Template
- Macro utente
- Interfaccia host
- Inventario host
- Tag host
- Proxy
- Gruppo proxy

Fonte

CHost::update() in *ui/include/classes/api/services/CHost.php*.

Host rilevato

Questa classe è progettata per funzionare con gli host rilevati.

Riferimenti agli oggetti:

- Host rilevato

Metodi disponibili:

- dhost.get - recupera gli host rilevati

Oggetto host discoverato

I seguenti oggetti sono direttamente correlati all'API `dhost`.

Host rilevato

> **Note:**
> Gli host rilevati vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

L'oggetto host rilevato contiene informazioni su un host rilevato da una regola di rilevamento di rete. Ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
dhostid	ID	ID dell'host rilevato.
druleid	ID	ID della regola di rilevamento che ha individuato l'host.
lastdown	timestamp	Ora in cui l'host rilevato è andato giù l'ultima volta.
lastup	timestamp	Ora in cui l'host rilevato è tornato su l'ultima volta.
status	integer	Indica se l'host rilevato è attivo o non attivo. Un host è attivo se ha almeno un servizio rilevato attivo. Valori possibili: 0 - host attivo; 1 - host non attivo.

dhost.get

Descrizione

`integer/array dhost.get(object parameters)`

Il metodo consente di recuperare gli host scoperti in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
dhostids	ID/array	Restituisce solo gli host rilevati con gli ID specificati.
druleids	ID/array	Restituisce solo gli host rilevati che sono stati creati dalle regole di discovery specificate.
dserviceids	ID/array	Restituisce solo gli host rilevati che eseguono i servizi specificati.
selectDRules	query	Restituisce una proprietà `drules` con un array delle regole di discovery che hanno rilevato l'host.
selectDServices	query	Restituisce una proprietà `dservices` con i servizi rilevati in esecuzione sull'host. Supporta `count`.
limitSelects	integer	Limita il numero di record restituiti dalle sottoquery. Si applica alle seguenti sottoquery: `selectDServices` - i risultati saranno ordinati per `dserviceid`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `dhostid`, `druleid`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare gli host rilevati tramite regola di individuazione

Recupera tutti gli host e i servizi rilevati che stanno eseguendo e che sono stati individuati dalla regola di individuazione "4".

Richiesta:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Servizio rilevato
- Regola di discovery

Fonte

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

Immagine

Questa classe è progettata per lavorare con le immagini.

Riferimenti degli oggetti:

- Immagine

Metodi disponibili:

- image.create - crea nuove immagini
- image.delete - elimina immagini
- image.get - recupera immagini
- image.update - aggiorna immagini

Oggetto immagine

I seguenti oggetti sono direttamente correlati all'API `image`.

Immagine

L'oggetto immagine ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
imageid	ID	ID dell'immagine. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome dell'immagine. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
imagetype	integer	Tipo di immagine. Valori possibili: 1 - *(predefinito)* icona; 2 - immagine di sfondo. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
image	string	Immagine codificata in Base64. La dimensione massima dell'immagine codificata è 1 MB. La dimensione massima può essere modificata cambiando il valore della costante `ZBX_MAX_IMAGE_SIZE`. Formati immagine supportati: PNG, JPEG, GIF e WebP. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

image.create

Descrizione

`object image.create(object/array images)`

Questo metodo consente di creare nuove immagini.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Immagini da creare.

Il metodo accetta immagini con le proprietà standard delle immagini.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle immagini create nella proprietà `imageids`. L'ordine degli ID restituiti corrisponde all'ordine delle immagini passate.

Esempi

Creare un'immagine icona

Creare un'icona cloud.

Richiesta:


```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```


Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

Creare un'immagine di sfondo

Creare una piccola immagine di sfondo da utilizzare come motivo.

Richiesta:


```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```


Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

Fonte

CImage::create() in *ui/include/classes/api/services/CImage.php*.

image.delete

Descrizione

`object image.delete(array imageIds)`

Questo metodo consente di eliminare le immagini.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle immagini da eliminare.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle immagini eliminate sotto la proprietà `imageids`.

Esempi

Eliminare più immagini

Eliminare due immagini.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Fonte

CImage::delete() in *ui/include/classes/api/services/CImage.php*.

image.get

Descrizione

`integer/array image.get(object parameters)`

Il metodo consente di recuperare le immagini in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
imageids	ID/array	Restituisce solo le immagini con gli ID specificati.
sysmapids	ID/array	Restituisce le immagini utilizzate nelle mappe specificate.

Parametro	Type	Descrizione
select_image	flag	Restituisce una proprietà `image` con l'immagine codificata in Base64.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `imageid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare un'immagine

Recupera tutti i dati dell'immagine con ID "2".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

Fonte

CImage::get() in *ui/include/classes/api/services/CImage.php*.

image.update

Descrizione

`object image.update(object/array images)`

Questo metodo consente di aggiornare le immagini esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà dell'immagine da aggiornare.

La proprietà `imageid` deve essere definita per ogni immagine, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Il metodo accetta immagini con le proprietà standard dell'immagine.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle immagini aggiornate sotto la proprietà `imageids`.

Esempi

Rinomina immagine

Rinomina l'immagine in "Cloud icon".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CImage::update() in *ui/include/classes/api/services/CImage.php*.

Impostazioni

Questa classe è progettata per lavorare con le impostazioni di amministrazione comuni.

Riferimenti degli oggetti:

- Impostazioni

Metodi disponibili:

- settings.get - recupera le impostazioni
- settings.update - aggiorna le impostazioni

Oggetto Impostazioni

I seguenti oggetti sono direttamente correlati all'API `settings`.

Impostazioni

L'oggetto delle impostazioni presenta le seguenti proprietà.

Property	Type	Description
default_lang	string	Lingua di sistema predefinita. Predefinito: `en_US`.
default_timezone	string	Fuso orario di sistema predefinito. Predefinito: `system` - predefinito del sistema. Per l'elenco completo dei fusi orari supportati, consultare la documentazione PHP.
default_theme	string	Tema predefinito. Valori possibili: `blue-theme` - *(predefinito)* Blu; `blue-classic-theme` - Blu (classico); `dark-theme` - Scuro; `dark-blue-theme` - Blu scuro; `dark-classic-theme` - Scuro (classico); `hc-light` - Chiaro ad alto contrasto; `hc-dark` - Scuro ad alto contrasto.
search_limit	integer	Limite per i risultati di ricerca e filtro. Predefinito: 1000.
max_overview_table_size	integer	Numero massimo di colonne e righe nei widget dashboard Panoramica dati e Panoramica trigger. Predefinito: 50.
max_in_table	integer	Numero massimo di elementi da visualizzare all'interno di una cella della tabella. Predefinito: 50.
server_check_interval	integer	Visualizza un avviso se il server Zabbix è inattivo. Valori possibili: 0 - Non visualizzare l'avviso; 10 - *(predefinito)* Visualizzare l'avviso.
work_period	string	Orario di lavoro. Predefinito: 1-5,09:00-18:00.
show_technical_errors	integer	Visualizza gli errori tecnici (PHP/SQL) agli utenti non Super admin e agli utenti che non fanno parte di gruppi di utenti con la modalità di debug abilitata. Valori possibili: 0 - *(predefinito)* Non visualizzare gli errori tecnici; 1 - Visualizzare gli errori tecnici.
history_period	string	Periodo massimo per visualizzare i dati della cronologia nei widget dashboard Ultimi dati, Web e Panoramica dati. Accetta un numero di secondi (ad esempio 1, 60, 300) o un valore con un suffisso temporale (ad esempio `30s`, `1m`, `2h`, `1d`). Predefinito: 24h.

Property	Type	Description
period_default	string	Periodo predefinito del filtro temporale. Accetta un numero di secondi (ad esempio 1, 60, 300) o un valore con un suffisso temporale (ad esempio 30s, 1m, 2h, 1d, 1M, 1y), inclusi mesi e anni. Predefinito: 1h.
max_period	string	Periodo massimo per il filtro temporale. Accetta un numero di secondi (ad esempio 1, 60, 300) o un valore con un suffisso temporale (ad esempio 30s, 1m, 2h, 1d, 1M, 1y), inclusi mesi e anni. Predefinito: 2y.
severity_color_0	string	Colore per la gravità "Non classificato" come codice colore esadecimale. Predefinito: 97AAB3.
severity_color_1	string	Colore per la gravità "Informazioni" come codice colore esadecimale. Predefinito: 7499FF.
severity_color_2	string	Colore per la gravità "Avviso" come codice colore esadecimale. Predefinito: FFC859.
severity_color_3	string	Colore per la gravità "Media" come codice colore esadecimale. Predefinito: FFA059.
severity_color_4	string	Colore per la gravità "Alta" come codice colore esadecimale. Predefinito: E97659.
severity_color_5	string	Colore per la gravità "Disastro" come codice colore esadecimale. Predefinito: E45959.
severity_name_0	string	Nome della gravità "Non classificato". Predefinito: Non classificato.
severity_name_1	string	Nome della gravità "Informazioni". Predefinito: Informazioni.
severity_name_2	string	Nome della gravità "Avviso". Predefinito: Avviso.
severity_name_3	string	Nome della gravità "Media". Predefinito: Media.
severity_name_4	string	Nome della gravità "Alta". Predefinito: Alta.
severity_name_5	string	Nome della gravità "Disastro". Predefinito: Disastro.
custom_color	integer	Utilizza colori personalizzati per lo stato degli eventi. Valori possibili: 0 - *(predefinito)* Non utilizzare colori personalizzati per lo stato degli eventi; 1 - Utilizzare colori personalizzati per lo stato degli eventi.
ok_period	string	Visualizza il periodo dei trigger OK. Accetta un numero di secondi (ad esempio 1, 60, 300) o un valore con un suffisso temporale (ad esempio 30s, 1m, 2h). Predefinito: 5m.

Property	Type	Description
blink_period	string	Periodo di lampeggiamento al cambio dello stato del trigger. Accetta un numero di secondi (ad esempio 1, 60, 300) o un valore con un suffisso temporale (ad esempio 30s, 1m, 2h). Predefinito: 2m.
problem_unack_color	string	Colore per gli eventi PROBLEM non riconosciuti come codice colore esadecimale. Predefinito: CC0000.
problem_ack_color	string	Colore per gli eventi PROBLEM riconosciuti come codice colore esadecimale. Predefinito: CC0000.
ok_unack_color	string	Colore per gli eventi RESOLVED non riconosciuti come codice colore esadecimale. Predefinito: 009900.
ok_ack_color	string	Colore per gli eventi RESOLVED riconosciuti come codice colore esadecimale. Predefinito: 009900.
problem_unack_style	integer	Lampeggiamento per gli eventi PROBLEM non riconosciuti. Valori possibili: 0 - Non visualizzare il lampeggiamento; 1 - *(predefinito)* Visualizzare il lampeggiamento.
problem_ack_style	integer	Lampeggiamento per gli eventi PROBLEM riconosciuti. Valori possibili: 0 - Non visualizzare il lampeggiamento; 1 - *(predefinito)* Visualizzare il lampeggiamento.
ok_unack_style	integer	Lampeggiamento per gli eventi RESOLVED non riconosciuti. Valori possibili: 0 - Non visualizzare il lampeggiamento; 1 - *(predefinito)* Visualizzare il lampeggiamento.
ok_ack_style	integer	Lampeggiamento per gli eventi RESOLVED riconosciuti. Valori possibili: 0 - Non visualizzare il lampeggiamento; 1 - *(predefinito)* Visualizzare il lampeggiamento.
url	string	URL del frontend.
discovery_groupid	ID	ID del gruppo di host in cui verranno inseriti automaticamente gli host rilevati.
default_inventory_mode	integer	Modalità predefinita dell'inventario host. Valori possibili: -1 - *(predefinito)* Disabilitato; 0 - Manuale; 1 - Automatico.
alert_usrgrpid	ID	ID del gruppo di utenti a cui verrà inviato il messaggio di allarme relativo all'indisponibilità del database. Se impostato su "0", il messaggio di allarme non verrà inviato.
snmptrap_logging	integer	Registra i trap SNMP non corrispondenti. Valori possibili: 0 - Non registrare i trap SNMP non corrispondenti; 1 - *(predefinito)* Registra i trap SNMP non corrispondenti.

Property	Type	Description
login_attempts	integer	Numero di tentativi di accesso non riusciti dopo il quale il modulo di accesso verrà bloccato. Predefinito: 5.
login_block	string	Intervallo di tempo durante il quale il modulo di accesso verrà bloccato se il numero di tentativi di accesso non riusciti supera quello definito nel campo login_attempts. Accetta un numero di secondi (ad esempio 1, 60) o un valore con un suffisso temporale (ad esempio 30s, 1m). Predefinito: 30s.
validate_uri_schemes	integer	Convalida gli schemi URI. Valori possibili: 0 - Non convalidare; 1 - *(predefinito)* Convalidare.
uri_valid_schemes	string	Schemi URI validi. Predefinito: http,https,ftp,file,mailto,tel,ssh.
x_frame_options	string	Intestazione HTTP X-Frame-Options. Predefinito: SAMEORIGIN.
iframe_sandboxing_enabled	integer	Utilizza il sandboxing degli iframe. Valori possibili: 0 - Non utilizzare; 1 - *(predefinito)* Utilizzare.
iframe_sandboxing_exceptions	string	Eccezioni del sandboxing degli iframe.
connect_timeout	string	Timeout di connessione con il server Zabbix. Intervallo di valori possibili: 1-30s. Predefinito: 3s. Comportamento della proprietà: - *obbligatoria*
socket_timeout	string	Timeout di rete predefinito. Intervallo di valori possibili: 1-300s. Predefinito: 3s. Comportamento della proprietà: - *obbligatoria*
media_type_test_timeout	string	Timeout di rete per il test del tipo di supporto. Intervallo di valori possibili: 1-300s. Predefinito: 65s. Comportamento della proprietà: - *obbligatoria*
item_test_timeout	string	Timeout di rete per i test degli item. Intervallo di valori possibili: 1-600s. Predefinito: 60s. Comportamento della proprietà: - *obbligatoria*

Property	Type	Description
script_timeout	string	Timeout di rete per l'esecuzione degli script. Intervallo di valori possibili: 1-300s. Predefinito: 60s. Comportamento della proprietà: - *obbligatoria*
report_test_timeout	string	Timeout di rete per il test del report pianificato. Intervallo di valori possibili: 1-300s. Predefinito: 60s. Comportamento della proprietà: - *obbligatoria*
device_link_timeout	string	Timeout per aggiungere un dispositivo mobile da utilizzare con l'app mobile Zabbix; si applica alla comunicazione tra l'API Zabbix e l'adattatore bridge. Intervallo di valori possibili: 1-300s. Predefinito: 60s. Comportamento della proprietà: - *obbligatoria*
auditlog_enabled	integer	Indica se abilitare la registrazione degli audit. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
auditlog_mode	integer	Indica se abilitare la registrazione degli audit delle attività di low-level discovery, network discovery e autoregistrazione eseguite dal server (utente System). Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
ha_failover_delay	string	Ritardo del failover in secondi. Predefinito: 1m. Comportamento della proprietà: - *sola lettura*
geomaps_tile_provider	string	Provider delle tile della geomappa. Valori possibili: `OpenStreetMap.Mapnik` - *(predefinito)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery. Supporta una stringa vuota per specificare valori personalizzati di `geomaps_tile_url`, `geomaps_max_zoom` e `geomaps_attribution`.
geomaps_tile_url	string	URL delle tile della geomappa. Comportamento della proprietà: - *supportata* se `geomaps_tile_provider` è impostato su una stringa vuota

Property	Type	Description
geomaps_max_zoom	integer	Livello massimo di zoom della geomappa. Valori possibili: 0-30. Comportamento della proprietà: - *supportata* se `geomaps_tile_provider` è impostato su una stringa vuota
geomaps_attribution	string	Testo di attribuzione della geomappa. Comportamento della proprietà: - *supportata* se `geomaps_tile_provider` è impostato su una stringa vuota
vault_provider	integer	Provider Vault. Valori possibili: 0 - *(predefinito)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Risoluzione dei valori delle macro segrete tramite: Valori possibili: 0 - *(predefinito)* solo il server; 1 - server e proxy in modo indipendente.
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 timeout_browser timeout_telemetry_query	string	Non impiegare più di `timeout_*` secondi per l'elaborazione. Accetta un numero di secondi (ad esempio 1, 60, 300) o un valore con un suffisso temporale (ad esempio `30s`, `1m`). Accetta anche macro utente. Intervallo di valori possibili: 1-600s. Predefinito: 3s. Predefinito per `timeout_browser`: 60s. Comportamento della proprietà: - *obbligatoria*
apm_global_db	object	Configurazione della fonte dati globale.

Origine dati globale

L'oggetto dell'origine dati globale presenta le seguenti proprietà:

Property	Type	Description
status	integer	Stato dell'origine dati globale. Valori possibili: 0 - *(predefinito)* non configurato; 1 - configurato.
url	string	URL utilizzato per connettersi all'origine dati. Comportamento della proprietà: - *obbligatorio* se l'utente modifica `status` in 1 (configurato).

Property	Type	Description
authentication_type	integer	Tipo di autenticazione. Valori possibili: 0 - *(predefinito)* utilizzo di un nome utente e una password; 1 - nessuna autenticazione utilizzata. Comportamento della proprietà: - *obbligatorio* se l'utente modifica `status` in 1 (configurato).
username	string	Nome utente utilizzato per accedere all'origine dati. Comportamento della proprietà: - *supportato* se `status` è impostato su 1 (configurato) e `authentication_type` è impostato su 0 (nome utente e password). - *obbligatorio* se l'utente modifica `authentication_type` in 0 (nome utente e password).
password	string	Password utilizzata per accedere all'origine dati. Il valore di questo parametro può essere vuoto. Comportamento della proprietà: - *supportato* se `status` è impostato su 1 (configurato) e `authentication_type` è impostato su 0 (nome utente e password).
db	string	Nome del database. Il valore di questo parametro può essere vuoto. Comportamento della proprietà: - *supportato* se `status` è impostato su 1 (configurato).
ssl_verify_peer	integer	Verifica del certificato del server. Valori possibili: 0 - *(predefinito)* disabilitata; 1 - abilitata. Comportamento della proprietà: - *supportato* se `status` è impostato su 1 (configurato) e `url` utilizza lo schema `https://`.
ssl_verify_host	integer	Verifica dell'host. Valori possibili: 0 - disabilitata; 1 - *(predefinito)* abilitata. Comportamento della proprietà: - *supportato* se `status` è impostato su 1 (configurato) e `url` utilizza lo schema `https://`.

settings.get

Descrizione

`object settings.get(object parameters)`

Il metodo consente di recuperare l'oggetto settings in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere User roles per ulteriori informazioni.

Parametri

(object) Parametri che definiscono l'output desiderato.

Il metodo supporta un solo parametro.

Parametro	Type	Descrizione
output	query	Questo parametro è descritto nel commento di riferimento.

Valori di ritorno

(object) Restituisce l'oggetto impostazioni.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "alert_usrgrpid": "7",
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
```

```
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Non classificato",
        "severity_name_1": "Informazioni",
        "severity_name_2": "Avviso",
        "severity_name_3": "Media",
        "severity_name_4": "Alta",
        "severity_name_5": "Disastro",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```


Fonte

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

Descrizione

`object settings.update(object settings)`

Questo metodo consente di aggiornare le impostazioni comuni esistenti.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles .

Parametri

`(object)` Proprietà impostazioni da aggiornare.

Valori restituiti

"(array)" Restituisce un array con i nomi dei parametri aggiornati.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

Fonte

CSettings::update() in *ui/include/classes/api/services/CSettings.php*.

Informazioni sull'API

Questa classe è progettata per recuperare meta-informazioni sull'API.

Metodi disponibili:

- apiinfo.version - recupera la versione dell'API Zabbix

apiinfo.version

Descrizione

`string apiinfo.version(array)`

Questo metodo consente di recuperare la versione delle API di Zabbix.

Attention:
Questo metodo è disponibile solo per gli utenti non autenticati.

Parametri

`(array)` Il metodo accetta un array vuoto.

Valori restituiti

`(string)` Restituisce la versione dell'API Zabbix.

A partire da Zabbix 2.0.4 la versione delle API corrisponde alla versione di Zabbix.

Esempi

Recupero della versione dell'API

Recupera la versione dell'API di Zabbix.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Fonte

CAPIInfo::version() in *ui/include/classes/api/services/CAPIInfo.php*.

Informazioni sul server

Questa classe è progettata per recuperare informazioni sul server Zabbix.

Metodi disponibili:

- serverinfo.get - recupera l'ora dell'ultimo aggiornamento dal server Zabbix

serverinfo.get

Questo metodo consente di recuperare l'ora dell'ultimo aggiornamento dal server Zabbix specificato.

Il metodo è disponibile per gli utenti *User*, *Admin*, *Super admin* e per gli utenti non autenticati.

Firma del metodo: `object serverinfo.get(object parameters)`

Origine del metodo: `CServerInfo::get()` in `ui/include/classes/api/services/CServerInfo.php`

Parametri

Parameter	Type	Description
serverid	ID	ID del server Zabbix. Comportamento del parametro: - Obbligatorio
output	query	Questo parametro è descritto nella documentazione di riferimento.

Valori restituiti

Property	Type	Description
serverid	ID	ID del server Zabbix.
lastaccess	timestamp	Ora dell'ultimo aggiornamento dal server Zabbix. Per un cluster ad alta disponibilità (HA), mostra il valore `lastaccess` del nodo attivo.

Esempi

Recuperare le informazioni sul server

Recupera le informazioni sul server Zabbix specificato.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

Interfaccia host

Questa classe consente di accedere alle interfacce host.

Riferimenti agli oggetti:

- Interfaccia host
  - Dettagli

Metodo disponibile:

- hostinterface.get - recupera le interfacce host

Le interfacce host vengono create, aggiornate ed eliminate utilizzando i metodi host.

Oggetto interfaccia dell'host

I seguenti oggetti sono direttamente correlati all'API `hostinterface`.

Interfaccia del host

L'oggetto dell'interfaccia del host ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
interfaceid	ID	ID dell'interfaccia. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
available	integer	Disponibilità dell'interfaccia del host. Valori possibili: 0 - *(predefinito)* sconosciuta; 1 - disponibile; 2 - non disponibile. Comportamento della proprietà: - *sola lettura*
hostid	ID	ID del host a cui appartiene l'interfaccia. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione

Proprietà	Tipo	Descrizione
type	integer	Tipo di interfaccia. Valori possibili: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
ip	string	Indirizzo IP utilizzato dall'interfaccia. Può essere vuoto se la connessione viene effettuata tramite DNS. Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione
dns	string	Nome DNS utilizzato dall'interfaccia. Può essere vuoto se la connessione viene effettuata tramite IP. Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione
port	string	Numero di porta utilizzato dall'interfaccia. Può contenere macro utente. Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione
useip	integer	Indica se la connessione deve essere effettuata tramite IP. Valori possibili: 0 - connessione tramite il nome DNS del host; 1 - connessione tramite l'indirizzo IP del host. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
main	integer	Indica se l'interfaccia viene utilizzata come predefinita sul host. Su un host è possibile impostare come predefinita una sola interfaccia di ciascun tipo. Valori possibili: 0 - non predefinita; 1 - predefinita. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
details	object	Oggetto details aggiuntivo per l'interfaccia. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "SNMP"
disable_until	timestamp	Orario del prossimo polling di un'interfaccia del host non disponibile. Comportamento della proprietà: - *sola lettura*
error	string	Testo dell'errore se l'interfaccia del host non è disponibile. Comportamento della proprietà: - *sola lettura*
errors_from	timestamp	Orario in cui l'interfaccia del host è diventata non disponibile. Comportamento della proprietà: - *sola lettura*

Dettagli

L'oggetto details ha le seguenti proprietà.

Proprietà	Type	Descrizione
version	integer	Versione dell'interfaccia SNMP. Valori possibili: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Comportamento della proprietà: - *obbligatoria*
bulk	integer	Indica se utilizzare richieste SNMP bulk. Valori possibili: 0 - non utilizzare richieste bulk; 1 - (predefinito) - utilizzare richieste bulk.
community	string	Community SNMP. Utilizzata solo dalle interfacce SNMPv1 e SNMPv2. Comportamento della proprietà: - *obbligatoria* se `version` è impostato su "SNMPv1" o "SNMPv2c"
max_repetitions	integer	Valore massimo di ripetizione per le richieste SNMP bulk native (GetBulkRequest-PDUs). Utilizzato solo per gli item `discovery[]` e `walk[]` in SNMPv2 e v3. Predefinito: 10.
securityname	string	Nome di sicurezza SNMPv3. Utilizzato solo dalle interfacce SNMPv3.
securitylevel	integer	Livello di sicurezza SNMPv3. Utilizzato solo dalle interfacce SNMPv3. Valori possibili: 0 - (predefinito) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	Passphrase di autenticazione SNMPv3. Utilizzata solo dalle interfacce SNMPv3.
privpassphrase	string	Passphrase di privacy SNMPv3. Utilizzata solo dalle interfacce SNMPv3.
authprotocol	integer	Protocollo di autenticazione SNMPv3. Utilizzato solo dalle interfacce SNMPv3. Valori possibili: 0 - (predefinito) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	Protocollo di privacy SNMPv3. Utilizzato solo dalle interfacce SNMPv3. Valori possibili: 0 - (predefinito) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	Nome del contesto SNMPv3. Utilizzato solo dalle interfacce SNMPv3.

hostinterface.get

Descrizione

`integer/array hostinterface.get(object parameters)`

Il metodo consente di recuperare le interfacce host in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
hostids	ID/array	Restituisce solo le interfacce host utilizzate dagli host specificati.
interfaceids	ID/array	Restituisce solo le interfacce host con gli ID specificati.
itemids	ID/array	Restituisce solo le interfacce host utilizzate dagli item specificati.
triggerids	ID/array	Restituisce solo le interfacce host utilizzate dagli item nei trigger specificati.
selectItems	query	Restituisce una proprietà `items` con gli item che utilizzano l'interfaccia. Supporta `count`.
selectHosts	query	Restituisce una proprietà `hosts` con un array di host che utilizzano l'interfaccia.
limitSelects	integer	Limita il numero di record restituiti dalle sottoquery. Si applica alle seguenti sottoquery: `selectItems`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `interfaceid`, `dns`, `ip`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare le interfacce del host

Recupera tutti i dati sulle interfacce utilizzate dal host "30057".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
```

```
        "hostids": "30057"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Guarda anche

- Host
- Elemento

Fonte

CHostInterface::get() in *ui/include/classes/api/services/CHostInterface.php*.

Item

Questa classe è progettata per lavorare con gli item.

Riferimenti agli oggetti:

- Item
  - Intestazione HTTP

    - Campo query HTTP
- Tag item
- Preprocessing item

Metodi disponibili:

- item.create - crea nuovi item
- item.delete - elimina gli item
- item.get - recupera gli item
- item.update - aggiorna gli item

Oggetto elemento

I seguenti oggetti sono direttamente correlati all'API `item`.

Item

Note:
Gli item web non possono essere creati, aggiornati o eliminati direttamente tramite l'API di Zabbix.

L'oggetto item ha le seguenti proprietà.

Property	Type	Description
itemid	ID	ID dell'item. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
delay	string	Intervallo di aggiornamento dell'item. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d) oppure, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e di pianificazione. Accetta macro utente. Se utilizzato, il valore deve essere una singola macro. Non sono supportate più macro o macro combinate con testo. Gli intervalli flessibili possono essere scritti come due macro separate da una barra (ad esempio, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Esempio: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Zabbix agent" (0), "Simple check" (3), "Zabbix internal" (5), "External check" (10), "Database monitor" (11), "IPMI agent" (12), "SSH agent" (13), "TELNET agent" (14), "Calculated" (15), "JMX agent" (16), "HTTP agent" (19), "SNMP agent" (20), "Script" (21), "Browser" (22), "Telemetry query" (24), oppure se `type` è impostato su "Zabbix agent (active)" (7) e `key_` non contiene "mqtt.get"
hostid	ID	ID dell'host o del template a cui appartiene l'item. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione

Property	Type	Description
interfaceid	ID	ID dell'interfaccia host dell'item. Comportamento della proprietà: - *obbligatoria* se l'item appartiene a un host e `type` è impostato su "Zabbix agent", "IPMI agent", "JMX agent", "SNMP trap" o "SNMP agent" - *supportata* se l'item appartiene a un host e `type` è impostato su "Simple check", "External check", "SSH agent", "TELNET agent" o "HTTP agent" - *sola lettura* per gli oggetti rilevati
key_	string	Chiave dell'item. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o rilevati
name	string	Nome dell'item. Supporta macro utente. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o rilevati
name_resolved	string	Nome dell'item con le macro utente risolte. Comportamento della proprietà: - *sola lettura*
type	integer	Tipo di item. Valori possibili: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o rilevati
url	string	Stringa URL. Supporta macro utente, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati o rilevati

Property	Type	Description
value_type	integer	Tipo di informazione dell'item. Valori possibili: 0 - virgola mobile numerica; 1 - carattere; 2 - log; 3 - numerico senza segno; 4 - testo; 5 - binario; 6 - JSON. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o rilevati
allow_traps	integer	Consente di popolare il valore in modo analogo all'item trapper. 0 - *(predefinito)* Non accettare dati in ingresso; 1 - Accettare dati in ingresso. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti rilevati
authtype	integer	Metodo di autenticazione. Valori possibili se `type` è impostato su "SSH agent": 0 - *(predefinito)* password; 1 - chiave pubblica. Valori possibili se `type` è impostato su "HTTP agent": 0 - *(predefinito)* nessuno; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento della proprietà: - *supportata* se `type` è impostato su "SSH agent" o "HTTP agent" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "HTTP agent") o rilevati
description	string	Descrizione dell'item. Comportamento della proprietà: - *sola lettura* per gli oggetti rilevati
error	string	Testo dell'errore se si verificano problemi durante l'aggiornamento del valore dell'item. Comportamento della proprietà: - *sola lettura*
flags	integer	Origine dell'item. Valori possibili: 0 - item normale; 4 - item convertito da un prototipo. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
follow_redirects	integer	Segue i reindirizzamenti della risposta durante il polling dei dati. Valori possibili: 0 - Non seguire i reindirizzamenti; 1 - *(predefinito)* Seguire i reindirizzamenti. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
headers	array	Array di intestazioni da inviare durante l’esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
history	string	Unità di tempo per la quale devono essere conservati i dati della cronologia. Accetta anche macro utente. Predefinito: 31d. Comportamento della proprietà: - *sola lettura* per gli oggetti rilevati
http_proxy	string	Stringa di connessione proxy HTTP(S). Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
inventory_link	integer	ID del campo dell’inventario host popolato dall’item. Consultare la pagina dell’inventario host per un elenco dei campi dell’inventario host supportati e dei relativi ID. Predefinito: 0. Comportamento della proprietà: - *supportata* se `value_type` è impostato su ”numeric float”, ”character”, ”numeric unsigned” o ”text” - *sola lettura* per gli oggetti rilevati
ipmi_sensor	string	Sensore IPMI. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”IPMI agent” e `key_` non è impostato su ”ipmi.get” - *supportata* se `type` è impostato su ”IPMI agent” - *sola lettura* per gli oggetti ereditati o rilevati
jmx_endpoint	string	Stringa di connessione personalizzata dell’agent JMX. Valore predefinito: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento della proprietà: - *supportata* se `type` è impostato su ”JMX agent” - *sola lettura* per gli oggetti rilevati

Property	Type	Description
lastclock	timestamp	Momento dell'ultimo aggiornamento del valore dell'item. Per impostazione predefinita, vengono visualizzati solo i valori relativi alle ultime 24 ore. È possibile estendere questo periodo modificando il valore del parametro *Max history display period* nella sezione di menu *Administration > General*. Comportamento della proprietà: - *sola lettura*
lastns	integer	Nanosecondi trascorsi dall'ultimo aggiornamento del valore dell'item. Per impostazione predefinita, vengono visualizzati solo i valori relativi alle ultime 24 ore. È possibile estendere questo periodo modificando il valore del parametro *Max history display period* nella sezione di menu *Administration > General*. Comportamento della proprietà: - *sola lettura*
lastvalue	string	Ultimo valore dell'item. Per impostazione predefinita, vengono visualizzati solo i valori relativi alle ultime 24 ore. È possibile estendere questo periodo modificando il valore del parametro *Max history display period* nella sezione di menu *Administration > General*. Comportamento della proprietà: - *sola lettura*
logtimefmt	string	Formato dell'ora nelle voci di log. Comportamento della proprietà: - *supportata* se `value_type` è impostato su "log" - *sola lettura* per gli oggetti ereditati o rilevati
master_itemid	ID	ID dell'item master. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Dependent item" - *sola lettura* per gli oggetti ereditati o rilevati
output_format	integer	Indica se la risposta deve essere convertita in JSON. 0 - *(predefinito)* Memorizzazione non elaborata; 1 - Converti in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati o rilevati
params	string	Parametri aggiuntivi in base al tipo di item: - script eseguito per gli item SSH agent e TELNET agent; - query SQL per gli item database monitor; - formula per gli item calculated; - script per gli item script e browser. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Database monitor", "SSH agent", "TELNET agent", "Calculated", "Script" o "Browser" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "Script" o "Browser") o rilevati

Property	Type	Description
parameters	object/array	Parametri aggiuntivi se `type` è impostato su ”Script” o ”Browser”. Array di oggetti con le proprietà `name` e `value`, dove `name` deve essere univoco. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Script” o ”Browser” - *sola lettura* per gli oggetti ereditati o rilevati
password	string	Password per l’autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”JMX agent” e `username` è impostato - *supportata* se `type` è impostato su ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (se `type` è impostato su ”HTTP agent”) o rilevati
post_type	integer	Tipo di corpo dei dati POST memorizzato nella proprietà `posts`. Valori possibili: 0 - *(predefinito)* Dati non elaborati; 2 - Dati JSON; 3 - Dati XML. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
posts	string	Dati del corpo della richiesta HTTP(S). Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”HTTP agent” e `post_type` è impostato su ”JSON data” o ”XML data” - *supportata* se `type` è impostato su ”HTTP agent” e `post_type` è impostato su ”Raw data” - *sola lettura* per gli oggetti ereditati o rilevati
prevvalue	string	Valore precedente dell’item. Per impostazione predefinita, vengono visualizzati solo i valori relativi alle ultime 24 ore. È possibile estendere questo periodo modificando il valore del parametro *Max history display period* nella sezione di menu *Administration > General*. Comportamento della proprietà: - *sola lettura*
privatekey	string	Nome del file della chiave privata. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent” e `authtype` è impostato su ”public key” - *sola lettura* per gli oggetti rilevati
publickey	string	Nome del file della chiave pubblica. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent” e `authtype` è impostato su ”public key” - *sola lettura* per gli oggetti rilevati
query_fields	array	Array di campi query da inviare durante l’esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati

Property	Type	Description
request_method	integer	Tipo di metodo della richiesta. Valori possibili: 0 - *(predefinito)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
retrieve_mode	integer	Parte della risposta da memorizzare. Valori possibili se `request_method` è impostato su ”GET”, ”POST” o ”PUT”: 0 - *(predefinito)* Corpo; 1 - Intestazioni; 2 - Verranno memorizzati sia il corpo sia le intestazioni. Valori possibili se `request_method` è impostato su ”HEAD”: 1 - Intestazioni. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
snmp_oid	string	OID SNMP. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SNMP agent” - *sola lettura* per gli oggetti ereditati o rilevati
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
ssl_key_file	string	Percorso del file della chiave SSL privata. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
ssl_key_password	string	Password del file della chiave SSL. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati
state	integer	Stato dell’item. Valori possibili: 0 - *(predefinito)* normale; 1 - non supportato. Comportamento della proprietà: - *sola lettura*
status	integer	Stato dell’item. Valori possibili: 0 - *(predefinito)* item abilitato; 1 - item disabilitato.

Property	Type	Description
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole. Supporta anche macro utente come parte dell'elenco separato da virgole. Esempio: 200,200-{$M},{$M},200-400 Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati o rilevati
templateid	ID	ID dell'item del template padre. *Suggerimento*: utilizzare la proprietà `hostid` per specificare il template a cui appartiene l'item. Comportamento della proprietà: - *sola lettura*
timeout	string	Timeout della richiesta di polling dei dati dell'item. Accetta un numero di secondi (ad esempio, 1, 60, 300) o un valore con un suffisso temporale (ad esempio, `300s`, `1m`). Accetta anche macro utente. Intervallo di valori possibile: 1-600s. Predefinito: "" - utilizza le impostazioni del proxy/globali. Comportamento della proprietà: - *supportata* se `type` è impostato su "Zabbix agent" (0), "Simple check" (3) e `key_` non inizia con "`vmware.`" e "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) e `snmp_oid` inizia con "`walk[`" o "`get[`", "Script" (21), "Browser" (22) - *sola lettura* per gli oggetti ereditati e rilevati, - se viene applicato il valore predefinito e l'item appartiene a un host monitorato da un proxy o da un gruppo di proxy non accessibile, viene utilizzato il timeout globale.

Property	Type	Description
trapper_hosts	string	Host consentiti. Comportamento della proprietà: - *sola lettura* per gli oggetti rilevati - *supportata* se `type` è impostato su ”Zabbix trapper” oppure se `type` è impostato su ”HTTP agent” e `allow_traps` è impostato su ”Allow incoming data”. Se la proprietà viene omessa, viene impostata automaticamente sulla macro utente `{$TRAPPER.ALLOWED_HOSTS} (valore predefinito). Per impostazione predefinita,{$TRAPPER.ALLOWED_HOSTS}`è127.0.0.1,::1. Se la proprietà è impostata su una stringa vuota, le connessioni in ingresso verranno negate da tutti gli host.| |trends|string|Unità di tempo per la quale devono essere conservati i dati dei trend. Accetta anche macro utente. Predefinito: 365d. [Comportamento della proprietà](/manual/api/reference_commentary#property-behavior): - *supportata* sevalue_typeè impostato su ”numeric float” o ”numeric unsigned” - *sola lettura* per gli oggetti rilevati| |units|string|Unità dei valori. [Comportamento della proprietà](/manual/api/reference_commentary#property-behavior): - *supportata* sevalue_typeè impostato su ”numeric float” o ”numeric unsigned” - *sola lettura* per gli oggetti ereditati o rilevati| |username|string|Nome utente per l’autenticazione. [Comportamento della proprietà](/manual/api/reference_commentary#property-behavior): - *obbligatorio* setypeè `impostato su "SSH agent", "TELNET agent" oppure setypeè impostato su "JMX agent"` epasswordè impostato - *supportato* setypeè impostato su ”Simple check”, ”Database monitor” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (setypeè impostato su ”HTTP agent”) o rilevati| |uuid|string|Identificatore univoco universale, utilizzato per collegare l’item importato a quelli già esistenti. Generato automaticamente se non specificato. [Comportamento della proprietà](/manual/api/reference_commentary#property-behavior): - *supportato* se l’item appartiene a un template| |valuemapid|ID|ID della mappa dei valori associata. [Comportamento della proprietà](/manual/api/reference_commentary#property-behavior): - *supportata* sevalue_typeè impostato su ”numeric float”, ”character” o ”numeric unsigned” - *sola lettura* per gli oggetti ereditati o rilevati| |verify_host|integer|Indica se convalidare che il nome host della connessione corrisponda a quello presente nel certificato dell’host. Valori possibili: 0 - *(predefinito)* Non convalidare; 1 - Convalida. [Comportamento della proprietà](/manual/api/reference_commentary#property-behavior): - *supportata* setypeè impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o rilevati| |verify_peer|integer|Indica se convalidare l’autenticità del certificato

Property	Type	Description

Intestazione HTTP

L'oggetto header ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
name	string	Nome dell'intestazione HTTP. Comportamento della proprietà: - *obbligatorio*
value	string	Valore dell'intestazione. Comportamento della proprietà: - *obbligatorio*

Campo della query HTTP

L'oggetto del campo della query definisce un nome e un valore utilizzati per specificare un parametro URL. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
name	string	Nome del parametro. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro. Comportamento della proprietà: - *obbligatorio*

Query di telemetria

L'oggetto query di telemetria ha le seguenti proprietà.

Property	Type	Description
signal_type	integer	Tipo di segnale interrogato dall'item. Valori possibili: 0 - Tracce; 1 - Metriche; 2 - Log. Comportamento della proprietà: - *obbligatoria*

Property	Type	Description
metric_point_type	integer	Tipo di punto metrica interrogato dall'item. Valori possibili: 0 - Somma; 1 - Gauge; 2 - Istogramma; 3 - Istogramma esponenziale. Comportamento della proprietà: - *obbligatoria* se `signal_type` è impostato su "Metrics"
columns	array	Array di colonne di dati di telemetria interrogate dall'item. Può essere vuoto. Comportamento della proprietà: - *obbligatoria*
aggregated_columns	array	Array di colonne aggregate di dati di telemetria interrogate dall'item e modalità di calcolo di un valore dai record delle colonne. Comportamento della proprietà: - *obbligatoria*
filter	object	Filtro della query di telemetria per i record delle colonne nella query. Comportamento della proprietà: - *obbligatoria*

Colonne dei dati di telemetria

L'oggetto delle colonne dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
column	string	Nome della colonna dei dati di telemetria interrogata dall'item. Vedere colonne dei dati di telemetria supportate. Per più colonne a valore singolo, ogni valore `column` deve essere univoco. Per più colonne che contengono coppie chiave-valore, ogni coppia `column` e `attribute_key` deve essere univoca.
attribute_key	string	Nome della chiave interrogata dall'item all'interno della colonna che contiene coppie chiave-valore. Per più colonne che contengono coppie chiave-valore, ogni coppia `column` e `attribute_key` deve essere univoca. Comportamento della proprietà: - *required* se `column` è impostato su una colonna che contiene coppie chiave-valore

Colonne aggregate dei dati di telemetria

L'oggetto delle colonne aggregate dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
column	string	Nome della colonna dei dati di telemetria interrogata dall'item. Vedere colonne dei dati di telemetria supportate. Per più colonne, ogni valore di `column` deve essere univoco. Questa proprietà non è supportata se `function` è impostato su "Count". Comportamento della proprietà: - *obbligatoria* se `function` è impostato su "Min", "Max", "Average", "Sum" o "Percentile"
function	integer	Funzione applicata dall'item ai record della colonna. Valori possibili: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Comportamento della proprietà: - *obbligatoria*
parameters	array	Percentile calcolato dall'item a partire dai record della colonna. Valore possibile: un numero da 0 a 100, specificato come stringa. Comportamento della proprietà: - *obbligatoria* se `function` è impostato su "Percentile"
alias	string	Nome univoco del valore calcolato, utilizzato per identificarlo nel valore dell'item. Ogni valore di `alias` deve essere univoco. Comportamento della proprietà: - *obbligatoria*

Filtro dei dati di telemetria

L'oggetto filtro dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
conditions	array	Condizioni del filtro dei dati di telemetria che un record in una colonna di dati di telemetria deve soddisfare. Comportamento della proprietà: - *obbligatoria*
evaltype	integer	Metodo di valutazione delle condizioni del filtro. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Comportamento della proprietà: - *obbligatoria*

Property	Type	Description
eval_formula	string	Espressione generata utilizzata per valutare le condizioni del filtro. L'espressione contiene ID che fanno riferimento a condizioni specifiche del filtro tramite il relativo `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Comportamento della proprietà: - *sola lettura*
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a condizioni specifiche del filtro tramite il relativo `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni del filtro: nessuna condizione può rimanere inutilizzata o essere omessa. Comportamento della proprietà: - *obbligatoria* se `evaltype` è impostato su "Espressione personalizzata"

Condizioni del filtro dei dati di telemetria

L'oggetto delle condizioni del filtro dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
column	string	Nome della colonna dei dati di telemetria a cui si applica la condizione del filtro. Vedere le colonne dei dati di telemetria supportate. Comportamento della proprietà: - *obbligatoria*
attribute_key	string	Nome della chiave a cui si applica la condizione, all'interno della colonna che contiene coppie chiave-valore. Comportamento della proprietà: - *obbligatoria* se `column` è impostata su una colonna che contiene coppie chiave-valore
operator	integer	Operatore della condizione del filtro. Valori possibili per le colonne a valore singolo: 0 - Uguale; 1 - Diverso da; 2 - Contiene; 3 - Non contiene. Valori possibili per le colonne che contengono coppie chiave-valore: 0 - Uguale; 1 - Diverso da; 12 - Esiste. Comportamento della proprietà: - *obbligatoria*
value	string	Valore con cui effettuare il confronto. Comportamento della proprietà: - *obbligatoria* se `operator` è impostato su "Uguale", "Diverso da", "Contiene" o "Non contiene"

Property	Type	Description
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente quando modifica le condizioni del filtro, ma verrà generato nuovamente quando queste verranno richieste in seguito. Comportamento della proprietà: - *obbligatoria* se `evaltype` dell'oggetto Filtro della query di telemetria è impostato su "Espressione personalizzata"

Tag dell'item

L'oggetto tag dell'item ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag dell'item. Comportamento della proprietà: - *obbligatoria*
value	string	Valore del tag dell'item. Comportamento della proprietà: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template; 1 - host. Comportamento della proprietà: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.

Preelaborazione dell'item

L'oggetto di preelaborazione dell'item ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
type	integer	Il tipo di opzione di preelaborazione. Valori possibili: 1 - Moltiplicatore personalizzato; 2 - Rimozione a destra; 3 - Rimozione a sinistra; 4 - Rimozione; 5 - Espressione regolare; 6 - Booleano in decimale; 7 - Ottale in decimale; 8 - Esadecimale in decimale; 9 - Modifica semplice; 10 - Modifica al secondo; 11 - XPath XML; 12 - JSONPath; 13 - Nell'intervallo; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Verifica la presenza di errori in JSON; 17 - Verifica la presenza di errori in XML; 18 - Verifica la presenza di errori tramite espressione regolare; 19 - Ignora i valori invariati; 20 - Ignora i valori invariati con heartbeat; 21 - JavaScript; 22 - Pattern Prometheus; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Sostituzione; 26 - Verifica non supportato; 27 - XML in JSON; 28 - Valore SNMP walk; 29 - SNMP walk in JSON; 30 - Valore SNMP get. Comportamento della proprietà: - *obbligatoria*
params	string	Parametri aggiuntivi utilizzati dall'opzione di preelaborazione. I parametri multipli sono separati dal carattere di nuova riga (\n). Se `type` è impostato su "Verifica non supportato", i parametri seguono la sintassi `<scope>[\n<pattern>]`, dove *pattern* è un'espressione regolare e *scope* è uno dei seguenti: -1 - corrisponde a qualsiasi errore; 0 - verifica se il messaggio di errore corrisponde a *pattern*; 1 - verifica se il messaggio di errore non corrisponde a *pattern*. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Moltiplicatore personalizzato" (1), "Rimozione a destra" (2), "Rimozione a sinistra" (3), "Rimozione" (4), "Espressione regolare" (5), "XPath XML" (11), "JSONPath" (12), "Nell'intervallo" (13), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Verifica la presenza di errori in JSON" (16), "Verifica la presenza di errori in XML" (17), "Verifica la presenza di errori tramite espressione regolare" (18), "Ignora i valori invariati con heartbeat" (20), "JavaScript" (21), "Pattern Prometheus" (22), "Prometheus in JSON" (23), "CSV in JSON" (24), "Sostituzione" (25), Verifica non supportato (26), "Valore SNMP walk" (28), "SNMP walk in JSON" (29) o "Valore SNMP get" (30)

Proprietà	Tipo	Descrizione
error_handler	integer	Tipo di azione utilizzato in caso di errore del passaggio di preelaborazione. Valori possibili: 0 - Il messaggio di errore viene impostato dal server Zabbix; 1 - Ignora il valore; 2 - Imposta un valore personalizzato; 3 - Imposta un messaggio di errore personalizzato. Valori possibili se `type` è impostato su ”Verifica non supportato”: 1 - Ignora il valore; 2 - Imposta un valore personalizzato; 3 - Imposta un messaggio di errore personalizzato. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”Moltiplicatore personalizzato” (1), ”Espressione regolare” (5), ”Booleano in decimale” (6), ”Ottale in decimale” (7), ”Esadecimale in decimale” (8), ”Modifica semplice” (9), ”Modifica al secondo” (10), ”XPath XML” (11), ”JSONPath” (12), ”Nell’intervallo” (13), ”Corrisponde all’espressione regolare” (14), ”Non corrisponde all’espressione regolare” (15), ”Verifica la presenza di errori in JSON” (16), ”Verifica la presenza di errori in XML” (17), ”Verifica la presenza di errori tramite espressione regolare” (18), ”Pattern Prometheus” (22), ”Prometheus in JSON” (23), ”CSV in JSON” (24), ”Verifica non supportato” (26), ”XML in JSON” (27), ”Valore SNMP walk” (28), ”SNMP walk in JSON” (29) o ”Valore SNMP get” (30)
error_handler_params	string	Parametri del gestore degli errori. Comportamento della proprietà: - *obbligatoria* se `error_handler` è impostato su ”Imposta un valore personalizzato” o ”Imposta un messaggio di errore personalizzato”

I seguenti parametri e gestori degli errori sono supportati per ogni tipo di preelaborazione.

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
1	Moltiplicatore personalizzato	numero^1, 6^			0, 1, 2, 3
2	Rimozione a destra	elenco di caratteri[2]			
3	Rimozione a sinistra	elenco di caratteri[2]			
4	Rimozione	elenco di caratteri[2]			
5	Espressione regolare	pattern[3]	output[2]		0, 1, 2, 3
6	Booleano in decimale				0, 1, 2, 3
7	Ottale in decimale				0, 1, 2, 3

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
8	Esadecimale in decimale				0, 1, 2, 3
9	Modifica semplice				0, 1, 2, 3
10	Modifica al secondo				0, 1, 2, 3
11	XPath XML	percorso[4]			0, 1, 2, 3
12	JSONPath	percorso[4]			0, 1, 2, 3
13	Nell’intervallo	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Corrisponde all’espressione regolare	pattern[3]			0, 1, 2, 3
15	Non corrisponde all’espressione regolare	pattern[3]			0, 1, 2, 3
16	Verifica la presenza di errori in JSON	percorso[4]			0, 1, 2, 3
17	Verifica la presenza di errori in XML	percorso[4]			0, 1, 2, 3
18	Verifica la presenza di errori tramite espressione regolare	pattern[3]	output[2]		0, 1, 2, 3
19	Ignora i valori invariati				

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
20	Ignora i valori invariati con heartbeat	secondi^5, 6^			
21	JavaScript	script[2]			
22	Pattern Prometheus	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus in JSON	pattern^6, 7^			0, 1, 2, 3
24	CSV in JSON	carattere[2]	carattere[2]	0,1	0, 1, 2, 3
25	Sostituzione	stringa di ricerca[2]	sostituzione[2]		
26	Verifica non supportato	scope[1]	pattern^3, 6^		1, 2, 3
27	XML in JSON				0, 1, 2, 3
28	Valore SNMP walk	OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS		0, 1, 2, 3
29	SNMP walk in JSON[10]	Nome del campo[2]	Prefisso OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS	0, 1, 2, 3
30	Valore SNMP get	Formato: 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS			0, 1, 2, 3

[1] numero a virgola mobile (gli interi vengono convertiti implicitamente in valori a virgola mobile)
 [2] stringa
 [3] espressione regolare
 [4] JSONPath o XPath XML
 [5] intero positivo (con supporto dei suffissi temporali, ad esempio `30s`, `1m`, `2h`, `1d`)
 [6] macro utente
 [7] pattern Prometheus conforme alla sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ogni componente del pattern Prometheus (metrica, nome dell'etichetta, valore dell'etichetta e valore della metrica) può essere una macro utente.
 [8] output Prometheus conforme alla sintassi: `<label name>` (può essere una macro utente) se `label` è selezionato come secondo parametro.
 [9] Una delle funzioni di aggregazione: `sum`, `min`, `max`, `avg`, `count` se `function` è selezionato come secondo parametro.
 [10] Supporta più record "Nome del campo,Prefisso OID,Formato" delimitati da un carattere di nuova riga.

item.create

Descrizione

`object item.create(object/array items)`

Questo metodo consente di creare nuovi item.

> **Note:**
> Gli item web non possono essere creati tramite la Zabbix API.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Item da creare.

Oltre alle proprietà standard degli item, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
preprocessing	array	Opzioni di pre-elaborazione degli item.
tags	array	Tag degli item.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli item creati nella proprietà `itemids`. L'ordine degli ID restituiti corrisponde all'ordine degli item passati.

Esempi

Creazione di un item

Creare un item numerico di Zabbix agent con 2 tag dell'item per monitorare lo spazio libero su disco sul host con ID "30074".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Creazione di un item di inventario host

Creare un item Zabbix agent per popolare il campo di inventario "OS" dell'host.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

Creazione di un item con preprocessing

Creare un item utilizzando un moltiplicatore personalizzato.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Tempo di attività del dispositivo",
        "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

Creazione di un item dipendente

Creare un item dipendente per il master item con ID 24759. Sono consentite solo dipendenze sullo stesso host, pertanto il master e l'item dipendente devono avere lo stesso hostid.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Creare un item HTTP agent

Creare un item con metodo di richiesta POST con preprocessing della risposta JSON.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Creare un item script

Creare una semplice raccolta dati utilizzando un item script.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Esempio di script",
        "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItem::create() in *ui/include/classes/api/services/CItem.php*.

item.delete

Descrizione

`object item.delete(array itemIds)`

Questo metodo consente di eliminare gli item.

Note:
Gli item web non possono essere eliminati tramite la Zabbix API.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID degli elementi da eliminare.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID degli articoli cancellati sotto la proprietà `itemids`.

Esempi

Eliminazione di più item

Eliminare due item.
 Gli item dipendenti e i prototipi di item vengono rimossi automaticamente se il master item viene eliminato.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
```

```
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

Fonte

CItem::delete() in *ui/include/classes/api/services/CItem.php*.

item.get

Descrizione

`integer/array item.get(object parameters)`

Il metodo consente di recuperare gli item in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parameter	Type	Description
itemids	ID/array	Restituisce solo gli item con gli ID specificati.
groupids	ID/array	Restituisce solo gli item che appartengono agli host dei gruppi specificati.
templateids	ID/array	Restituisce solo gli item che appartengono ai template specificati.
hostids	ID/array	Restituisce solo gli item che appartengono agli host specificati.
proxyids	ID/array	Restituisce solo gli item monitorati dai proxy specificati.
interfaceids	ID/array	Restituisce solo gli item che utilizzano le interfacce host specificate.
graphids	ID/array	Restituisce solo gli item utilizzati nei grafici specificati.
triggerids	ID/array	Restituisce solo gli item utilizzati nei trigger specificati.
webitems	flag	Include gli item web nel risultato.
inherited	boolean	Se impostato su `true`, restituisce solo gli item ereditati da un template.
inheritedTags	boolean	Restituisce gli item che hanno i `tags` specificati anche in host/template/template collegati. Valori possibili: `true` - anche template/host/template collegati devono avere i tag specificati; `false` - *(predefinito)* i tag di template/host/template collegati vengono ignorati.
templated	boolean	Se impostato su `true`, restituisce solo gli item che appartengono ai template.
monitored	boolean	Se impostato su `true`, restituisce solo gli item abilitati che appartengono a host monitorati.
group	string	Restituisce solo gli item che appartengono a un gruppo con il nome specificato.
host	string	Restituisce solo gli item che appartengono a un host con il nome specificato.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.

Parameter	Type	Description
tags	array	Restituisce solo gli item con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti gli item. Valori possibili per operator: 0 - *(predefinito)* Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
with_triggers	boolean	Se impostato su `true`, restituisce solo gli item utilizzati nei trigger.
selectHosts	query	Restituisce una proprietà `hosts` con un array di host a cui appartiene l'item.
selectInterfaces	query	Restituisce una proprietà `interfaces` con un array di interfacce host utilizzate dall'item.
selectTriggers	query	Restituisce una proprietà `triggers` con i trigger in cui viene utilizzato l'item. Supporta `count`.
selectGraphs	query	Restituisce una proprietà `graphs` con i grafici che contengono l'item. Supporta `count`.
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery dell'item. L'oggetto di discovery dell'item collega un item rilevato a un prototipo di item da cui è stato rilevato. Ha le seguenti proprietà: `parent_itemid` - `(string)` ID del prototipo di item da cui l'item è stato creato; `key_` - `(string)` chiave del prototipo di item; `status` - `(int)` stato di discovery dell'item: 0 - *(predefinito)* l'item è rilevato, 1 - l'item non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un item non più rilevato verrà eliminato; `ts_disable` - `(timestamp)` momento in cui un item non più rilevato verrà disabilitato; `disable_source` - `(int)` indicatore che specifica se l'item è stato disabilitato da una regola LLD o manualmente: 0 - *(predefinito)* disabilitato automaticamente, 1 - disabilitato da una regola LLD.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola LLD che ha creato l'item.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag presenti su template/host/template collegati.
selectPreprocessing	query	Restituisce una proprietà `preprocessing` con le opzioni di preprocessing dell'item.
selectTags	query	Restituisce i tag dell'item nella proprietà `tags`.
selectValueMap	query	Restituisce una proprietà `valuemap` con la mappa dei valori dell'item.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dato `text` data type. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene l'item.

Parameter	Type	Description
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectGraphs` - i risultati saranno ordinati per `name`; `selectTriggers` - i risultati saranno ordinati per `description`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	Restituisce una proprietà `itemDiscovery` con l'oggetto di discovery dell'item. L'oggetto di discovery dell'item collega l'item a un prototipo di item da cui è stato creato. Questa query è **deprecata**; utilizzare invece `selectDiscoveryData`.

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Trovare item per chiave

Recupera tutti gli item utilizzati nei trigger per uno specifico ID host che hanno la parola "system.cpu" nella chiave dell'item e ordina i risultati per nome.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Trovare item dipendenti per chiave

Recupera tutti gli item dipendenti dall'host con ID "10116" che hanno la parola "apache" nella chiave.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Trovare un item HTTP agent

Trovare un item HTTP agent con tipo di corpo POST XML per uno specifico ID host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Recupero degli item con regole di preprocessing

Recupera tutti gli item e le relative regole di preprocessing per uno specifico ID host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Regola di individuazione
- Grafico
- Host
- Interfaccia host
- Trigger

Fonte

CItem::get() in *ui/include/classes/api/services/CItem.php*.

item.update

Descrizione

`object item.update(object/array items)`

Questo metodo consente di aggiornare gli item esistenti.

> **Note:**
> Gli item web non possono essere aggiornati tramite la Zabbix API.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dell'item da aggiornare.

La proprietà `itemid` deve essere definita per ogni item, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard dell'item, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
preprocessing	array	Opzioni di preprocessing dell'item per sostituire le opzioni di preprocessing correnti. Comportamento del parametro: - *sola lettura* per oggetti ereditati o oggetti individuati
tags	array	Tag dell'item. Comportamento del parametro: - *sola lettura* per oggetti individuati

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID degli articoli aggiornati sotto la proprietà `itemids`.

Esempi

Abilitazione di un item

Abilitare un item, ovvero impostarne lo stato a "0".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

Aggiornare un item dipendente

Aggiorna il nome dell'item dipendente e l'ID dell'item principale. Sono consentite solo dipendenze sullo stesso host, pertanto l'item principale e l'item dipendente devono avere lo stesso hostid.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Nome aggiornato dell'item dipendente",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

Aggiornare un item HTTP agent

Abilitare il trapping del valore dell'item.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Aggiornamento di un item con preprocessing

Aggiorna un item con la regola di preprocessing dell'item "In range".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
```

```
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Aggiornamento di un item script

Aggiornare un item script con uno script diverso e rimuovere i parametri non necessari utilizzati dallo script precedente.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItem::update() in *ui/include/classes/api/services/CItem.php*.

Log APM

Questa classe è progettata per funzionare con i log raccolti per il monitoraggio delle prestazioni delle applicazioni (APM).

Riferimenti agli oggetti:

- Log APM

Metodi disponibili:

- apmlog.get - recupera i record di log

Oggetto log APM

I seguenti oggetti sono direttamente correlati all'API `apmlog`.

> **Note:**
> Ogni proprietà corrisponde alla colonna con lo stesso nome nella tabella `otel_logs` della sorgente dati di telemetria (ad esempio, la proprietà `severity_text` corrisponde alla colonna `SeverityText`). Per l'elenco delle colonne, vedere Colonne dei dati di telemetria supportate.

Log APM

L'oggetto log APM rappresenta un record di log (un record di testo di un evento, ad esempio un pagamento non riuscito o un timeout del database). Un record di log contiene informazioni su un evento verificatosi in un momento specifico, inclusi la relativa gravità, il messaggio e gli attributi. Se un record di log è stato prodotto durante un'operazione che fa parte di una trace, contiene anche gli ID di tale trace e span.

L'oggetto log APM ha le seguenti proprietà:

Property	Type	Description
timestamp	integer	Ora Unix (in nanosecondi) in cui si è verificato l'evento. Property behavior: - *sola lettura*
traceid	string	ID, se disponibile, della trace durante la quale è stato prodotto il record di log. Property behavior: - *sola lettura*
spanid	string	ID, se disponibile, dello span durante il quale è stato prodotto il record di log. Property behavior: - *sola lettura*
trace_flags	integer	Flag della trace del contesto della trace del record di log, come maschera di bit. Se il bit meno significativo è impostato (valore 1), la trace viene campionata. Property behavior: - *sola lettura*
severity_text	string	Livello di gravità del record di log, fornito dall'applicazione (ad esempio, `INFO`, `WARN` o `ERROR`). Property behavior: - *sola lettura*
severity_number	integer	Livello di gravità del record di log, come numero. Valori possibili: 0 - UNSPECIFIED; 1-4 - TRACE; 5-8 - DEBUG; 9-12 - INFO; 13-16 - WARN; 17-20 - ERROR; 21-24 - FATAL. Property behavior: - *sola lettura*

Property	Type	Description
service_name	string	Nome del servizio che ha prodotto il record di log (ad esempio, `zabbix-frontend`), ricavato dall'attributo della risorsa OpenTelemetry `service.name`. Property behavior: - *sola lettura*
body	string	Messaggio del record di log. Property behavior: - *sola lettura*
resource_schema_url	string	URL, se disponibile, dello schema OpenTelemetry degli attributi della risorsa. Property behavior: - *sola lettura*
resource_attributes	object	Attributi della risorsa chiave-valore del record di log (ad esempio, `host.name: eabf8c2a5ddc`). Property behavior: - *sola lettura*
scope_schema_url	string	URL, se disponibile, dello schema OpenTelemetry dell'ambito di strumentazione e dei relativi attributi. Property behavior: - *sola lettura*
scope_name	string	Nome dell'ambito di strumentazione che ha prodotto il record di log. Property behavior: - *sola lettura*
scope_version	string	Versione, se disponibile, dell'ambito di strumentazione. Property behavior: - *sola lettura*
scope_attributes	object	Attributi chiave-valore dell'ambito di strumentazione. Property behavior: - *sola lettura*
log_attributes	object	Attributi del log chiave-valore del record di log (ad esempio, `log.file.path: /var/log/zabbix/zabbix_frontend.log`). Property behavior: - *sola lettura*
event_name	string	Nome, se disponibile, del tipo di evento rappresentato dal record di log. Property behavior: - *sola lettura*

apmlog.get

Descrizione

`integer/array apmlog.get(object parameters)`

Il metodo consente di recuperare i record di log in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

(`object`) Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
time_from	timestamp	Restituisce solo i record di log degli eventi verificatisi al momento specificato o successivamente. Comportamento del parametro: - *obbligatorio*
time_till	timestamp	Restituisce solo i record di log degli eventi verificatisi prima del momento specificato. Comportamento del parametro: - *obbligatorio*
traceids	string/array	Restituisce solo i record di log prodotti durante le tracce specificate.
spanids	string/array	Restituisce solo i record di log prodotti durante gli span specificati.
with_flags_on	integer	Restituisce solo i record di log che hanno impostati tutti i flag di traccia specificati. Per specificare più flag, utilizzare la somma dei relativi valori.
with_flags_off	integer	Restituisce solo i record di log che non hanno impostato nessuno dei flag di traccia specificati. Per specificare più flag, utilizzare la somma dei relativi valori.
resource_attributes	object/array	Restituisce solo i record di log con gli attributi della risorsa specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i record di log. Valori possibili per operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste.
resource_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `resource_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
scope_attributes	object/array	Restituisce solo i record di log con gli attributi dell'ambito specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i record di log. Valori possibili per operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste.
scope_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `scope_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.

Parametro	Tipo	Descrizione
log_attributes	object/array	Restituisce solo i record di log con gli attributi di log specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i record di log. Valori possibili per operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste.
log_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `log_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore o un array di valori da confrontare. Proprietà supportate: `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono specificate opzioni aggiuntive, viene eseguita una ricerca `LIKE "%...%"`. Proprietà supportate: `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
offset	integer	Numero di record da ignorare prima di restituire il risultato.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei log degli errori di una traccia

Recupera i record di log con gravità ERROR (numeri di gravità 17-20) associati alla traccia "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```

Ricerca dei messaggi di log

Recupera l'ora, il servizio e il messaggio degli ultimi 10 record di log il cui messaggio contiene "timeout".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
```

```
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```

Vedi anche

- Traccia APM
- Span APM
- Metrica APM

Origine

CApmLog::get() in *ui/include/classes/api/services/CApmLog.php*.

Log di audit

Questa classe è progettata per funzionare con il log di audit.

Riferimenti agli oggetti:

- Log di audit

Metodi disponibili:

- auditlog.get - recupera i record del log di audit

Oggetto log di audit

I seguenti oggetti sono direttamente correlati all'API `auditlog`.

Log di audit

L'oggetto log di audit contiene informazioni sulle azioni degli utenti. Ha le seguenti proprietà.

Property	Type	Description
auditid	ID	ID della voce del log di audit. Generato usando l'algoritmo CUID.
userid	ID	Userid dell'autore della voce del log di audit.

Property	Type	Description
username	string	Nome utente dell’autore della voce del log di audit.
clock	timestamp	Timestamp di creazione della voce del log di audit.
ip	string	Indirizzo IP dell’autore della voce del log di audit.
action	integer	Azione della voce del log di audit. Valori possibili: 0 - Aggiungi; 1 - Aggiorna; 2 - Elimina; 4 - Logout; 7 - Esegui; 8 - Login; 9 - Login fallito; 10 - Cancella cronologia; 11 - Aggiorna configurazione; 12 - Push.

Property	Type	Description
resourcetype	integer	Tipo di risorsa della voce del log di audit. Valori possibili: 0 - Utente; 3 - Tipo di supporto; 4 - Host; 5 - Azione; 6 - Grafico; 11 - Gruppo utenti; 13 - Trigger; 14 - Gruppo host; 15 - Item; 16 - Immagine; 17 - Mappa valori; 18 - Servizio; 19 - Mappa; 22 - Scenario web; 23 - Regola di discovery; 25 - Script; 26 - Proxy; 27 - Manutenzione; 28 - Espressione regolare; 29 - Macro; 30 - Template; 31 - Prototipo di trigger; 32 - Mappatura icone; 33 - Dashboard; 34 - Correlazione eventi; 35 - Prototipo di grafico; 36 - Prototipo di item; 37 - Prototipo di host; 38 - Registrazione automatica; 39 - Modulo; 40 - Impostazioni; 41 - Housekeeping; 42 - Autenticazione; 43 - Dashboard del template; 44 - Ruolo utente; 45 - Token API; 46 - Report pianificato; 47 - Nodo ad alta disponibilità; 48 - SLA; 49 - Directory utenti; 50 - Gruppo template; 51 - Connettore; 52 - Regola LLD; 53 - Cronologia; 54 - Autenticazione a più fattori; 55 - Gruppo proxy; 56 - Prototipo di regola LLD.
resourceid	ID	Identificatore della risorsa della voce del log di audit.
resource_cuid	ID	Identificatore univoco della risorsa della voce del log di audit generato usando l’algoritmo CUID.
resourcename	string	Nome leggibile della risorsa della voce del log di audit.
recordsetid	ID	ID del recordset della voce del log di audit. I record del log di audit creati durante la stessa operazione avranno lo stesso ID recordset. Generato usando l’algoritmo CUID.

Property	Type	Description
details	text	Dettagli della voce del log di audit. I dettagli sono memorizzati come oggetto JSON, dove ogni nome di proprietà è un percorso alla proprietà o all'oggetto nidificato in cui si è verificata la modifica, e dove ogni valore contiene i dati (in formato array) relativi alla modifica di questa proprietà o oggetto nidificato. Formati di valore possibili: ["add"] - L'oggetto nidificato è stato aggiunto; ["add", "<value>"] - La proprietà dell'oggetto aggiunto è uguale a <value>; ["update"] - L'oggetto nidificato è stato aggiornato; ["update", "<new value>", "<old value>"] - La proprietà dell'oggetto aggiornato è stata modificata da <old value> a <new value>; ["delete"] - L'oggetto nidificato è stato eliminato.

auditlog.get

Descrizione

`integer/array auditlog.get(object parameters)`

Il metodo consente di recuperare i record del registro di audit in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
auditids	ID/array	Restituisce solo il log di audit con gli ID specificati.
userids	ID/array	Restituisce solo il log di audit creato dagli utenti specificati.
time_from	timestamp	Restituisce solo le voci del log di audit create dopo o al momento specificato.
time_till	timestamp	Restituisce solo le voci del log di audit create prima o al momento specificato.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `auditid`, `userid`, `clock`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recuperare il log di audit

Recupera gli ultimi due record del log di audit.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Oggetto log di audit

Sorgente

CAuditLog::get() in *ui/include/classes/api/services/CAuditLog.php*.

Macro utente

Questa classe è progettata per funzionare con le macro utente a livello di host e globali.

Riferimenti agli oggetti:

- Macro globale
- Macro host
  - Configurazione della macro

Metodi disponibili:

- usermacro.create - crea nuove macro host
- usermacro.createglobal - crea nuove macro globali
- usermacro.delete - elimina macro host
- usermacro.deleteglobal - elimina macro globali
- usermacro.get - recupera macro host e globali
- usermacro.update - aggiorna macro host
- usermacro.updateglobal - aggiorna macro globali

Oggetto macro utente

I seguenti oggetti sono direttamente correlati all'API `usermacro`.

Macro globale

L'oggetto macro globale ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
globalmacroid	ID	ID della macro globale. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
description	string	Descrizione della macro.
macro	string	Stringa della macro. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
type	integer	Tipo di macro. Valori possibili: 0 - *(predefinito)* Macro di testo; 1 - Macro segreta; 2 - Segreto del vault.
value	string	Valore della macro. Comportamento della proprietà: - *sola scrittura* se `type` è impostato su "Macro segreta" - *obbligatoria* per le operazioni di creazione

Macro host

L'oggetto macro host definisce una macro disponibile su un host, un prototipo di host o un template. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
hostmacroid	ID	ID della macro. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento

Proprietà	Type	Descrizione
automatic	integer	Definisce se la macro è controllata dalla regola di discovery. Valori possibili: 0 - *(predefinito)* La macro è gestita dall'utente; 1 - La macro è gestita dalla regola di discovery. L'utente non può creare una macro automatica. Per aggiornare una macro automatica, deve essere convertita in manuale.
config	object/array	Configurazione della macro responsabile di come la macro viene visualizzata nel Host Wizard.
description	string	Descrizione della macro.
hostid	ID	ID dell'host, del prototipo di host o del template a cui appartiene la macro. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
macro	string	Stringa della macro. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
type	integer	Tipo di macro. Valori possibili: 0 - *(predefinito)* Macro di testo; 1 - Macro segreta; 2 - Segreto del vault.
value	string	Valore della macro. Comportamento della proprietà: - *sola scrittura* se `type` è impostato su "Macro segreta" - *obbligatoria* per le operazioni di creazione

Configurazione della macro

L'oggetto di configurazione della macro definisce come una macro viene visualizzata nel Host Wizard.

Property	Type	Description
type	integer	Tipo del campo di input della macro. Valori possibili: 0 - La macro non viene utilizzata in Host Wizard; 1 - Casella di testo; 2 - Elenco; 3 - Casella di controllo. Property behavior: - *required*
label	string	Etichetta del campo di input della macro. Property behavior: - *required* se `type` è impostato su "Textbox", "List" o "Checkbox"
description	text	Testo di aiuto visualizzato accanto al campo di input della macro. Supporta la formattazione Markdown.
priority	integer	Posizione del campo di input della macro nell'elenco delle macro. Se `priority` non è impostato, la macro viene aggiunta alla fine dell'elenco delle macro non raggruppate.

Property	Type	Description
required	integer	Indica la macro come obbligatoria. Valori possibili: 0 - Non obbligatoria; 1 - Obbligatoria. Property behavior: - *supported* se `type` è impostato su ”Textbox” o ”List”
regex	string	Espressione regolare per convalidare l’input dell’utente in un campo casella di testo. Property behavior: - *supported* se `type` è impostato su ”Textbox”
section_name	string	Etichetta della sezione comprimibile in cui la macro è raggruppata. Se `section_name` non è impostato, la macro non viene raggruppata. Le macro non raggruppate vengono visualizzate per prime; le macro raggruppate vengono visualizzate sotto, ordinate in base alla loro `priority` all’interno di ciascuna sezione comprimibile.
options	text	Stringa JSON che definisce gli elementi dell’elenco o i valori della casella di controllo. Per gli elenchi: un array di oggetti con proprietà `value` e `text`. Esempio: `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` Per le caselle di controllo: un oggetto con proprietà `checked` e `unchecked`. Esempio: `{"checked": true, "unchecked": false}` Property behavior: - *required* se `type` è impostato su ”List” o ”Checkbox”

usermacro.create

Descrizione

`object usermacro.create(object/array hostMacros)`

Questo metodo consente di creare nuovi macro dell’host.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Macro host da creare.

Il metodo accetta macro host con le proprietà standard delle macro host.

Valori restituiti

”(object)” Restituisce un oggetto contenente gli ID delle macro host create sotto la proprietà ”hostmacroids”. L’ordine degli ID restituiti corrisponde all’ordine delle macro host passate.

Esempi

Creazione di una macro host

Creare una macro host `{$SNMP_COMMUNITY}` con il valore ”public” sull’host ”10198”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::create() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

Descrizione

`object usermacro.createglobal(object/array globalMacros)`

Questo metodo consente di creare nuovi macro globali.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Macro globali da creare.

Il metodo accetta macro globali con le proprietà standard delle macro globali.

Valori restituiti

”(object)” Restituisce un oggetto contenente gli ID delle macro globali create globale sotto la proprietà ”globalmacroids”. L’ordine degli ID restituiti corrisponde all’ordine delle macro globali passate.

Esempi

Creazione di una macro globale

Creare una macro globale `{$SNMP_COMMUNITY}` con valore ”public”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::createGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

Descrizione

`object usermacro.delete(array hostMacroIds)`

Questo metodo consente di eliminare le macro dell'host.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

"(array)" ID delle macro host da eliminare.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID delle macro host eliminate sotto la proprietà "hostmacroids".

Esempi

Eliminazione di più macro host

Eliminare due macro host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::delete() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

Descrizione

`object usermacro.deleteglobal(array globalMacroIds)`

Questo metodo consente di eliminare le macro globali.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle macro globali da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle macro globali cancellate sotto la proprietà `globalmacroids`.

Esempi

Eliminazione di più macro globali

Eliminare due macro globali.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::deleteGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

Descrizione

`integer/array usermacro.get(object parameters)`

Il metodo consente di recuperare macro di host e globali in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
globalmacro	flag	Restituisce le macro globali invece delle macro host.
globalmacroids	ID/array	Restituisce solo le macro globali con gli ID specificati.
groupids	ID/array	Restituisce solo le macro host che appartengono a host o template dei gruppi host o gruppi template specificati.
hostids	ID/array	Restituisce solo le macro che appartengono agli host o template specificati.
hostmacroids	ID/array	Restituisce solo le macro host con gli ID specificati.
inherited	boolean	Se impostato su `true`, restituisce solo le macro utente dei prototipi host ereditate da un template.
selectHostGroups	query	Restituisce i gruppi host a cui appartiene la macro host nella proprietà `hostgroups`. Utilizzato solo durante il recupero delle macro host.
selectHosts	query	Restituisce gli host a cui appartiene la macro host nella proprietà `hosts`. Utilizzato solo durante il recupero delle macro host.
selectTemplateGroups	query	Restituisce i gruppi template a cui appartiene la macro template nella proprietà `templategroups`. Utilizzato solo durante il recupero delle macro template.
selectTemplates	query	Restituisce i template a cui appartiene la macro host nella proprietà `templates`. Utilizzato solo durante il recupero delle macro host.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `macro`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero delle macro host per un host

Recupera tutte le macro host definite per l'host "10198".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
```

```
        "hostids": "10198"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Recupero delle macro host per un template

Recupera le macro host definite per il template ”10265” che contengono ”STATUS” nel nome della macro.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "Il nome host o l'indirizzo IP dell'host della pagina di stato di Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
```

```
                "priority": "1",
                "section_name": "",
                "label": "Host stato Apache",
                "description": "Nome host o indirizzo IP dell'host della pagina di stato di Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "Il percorso URL della pagina di stato di Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Percorso pagina di stato Apache",
                "description": "Percorso URL della pagina di stato di Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "La porta della pagina di stato di Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "Porta pagina di stato Apache",
                "description": "Nell'intervallo da 1 a 65535 inclusi.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "Lo schema della richiesta, che può essere HTTP o HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Schema della richiesta",
                "description": "Schema della richiesta, che può essere HTTP o HTTPS.",
                "required": "0",
```

```
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Recupero delle macro globali

Recupera tutte le macro globali.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
    ],
    "id": 1
}
```

Fonte

CUserMacro::get() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

Descrizione

`object usermacro.update(object/array hostMacros)`

Questo metodo consente di aggiornare macro host esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

(object/array) Proprietà della macro del host da aggiornare.

La proprietà hostmacroid deve essere definita per ciascuna macro del host, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID delle macro host aggiornate sotto la proprietà "hostmacroids".

Esempi

Modifica del valore di una macro host

Modificare il valore di una macro host in "public".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Modificare il valore della macro creata da una regola di discovery

Convertire la macro "automatica" creata dalla regola di discovery in "manuale" e modificarne il valore in "new-value".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::update() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

Descrizione

`object usermacro.updateglobal(object/array globalMacros)`

Questo metodo consente di aggiornare i macro globali esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object/array)` Proprietà della macro globale da aggiornare.

La proprietà `globalmacroid` deve essere definita per ogni macro globale, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle macro globali aggiornate sotto la proprietà `globalmacroids`.

Esempi

Modifica del valore di una macro globale

Modificare il valore di una macro globale in ”public”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::updateGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

Manutenzione

Questa classe è progettata per funzionare con le manutenzioni.

Riferimenti agli oggetti:

- Maintenance
- Time period
- Problem tag

Metodi disponibili:

- maintenance.create - crea nuove manutenzioni
- maintenance.delete - elimina manutenzioni

- maintenance.get - recupera manutenzioni
- maintenance.update - aggiorna manutenzioni

Oggetto manutenzione

I seguenti oggetti sono direttamente correlati all'API `maintenance`.

Maintenance

L'oggetto maintenance ha le seguenti proprietà.

Property	Type	Description
maintenanceid	ID	ID della maintenance. Property behavior: - *read-only* - *required* for update operations
name	string	Nome della maintenance. Property behavior: - *required* for create operations
active_since	timestamp	Ora in cui la maintenance diventa attiva (inclusa). Il valore fornito verrà arrotondato per difetto ai minuti. Property behavior: - *required* for create operations
active_till	timestamp	Ora in cui la maintenance smette di essere attiva (esclusa). Il valore fornito verrà arrotondato per difetto ai minuti. Property behavior: - *required* for create operations
description	string	Descrizione della maintenance.
maintenance_type	integer	Tipo di maintenance. Valori possibili: 0 - *(default)* con raccolta dati; 1 - senza raccolta dati.
tags_evaltype	integer	Metodo di valutazione dei tag del problema. Valori possibili: 0 - *(default)* And/Or; 2 - Or.

Periodo di tempo

L'oggetto periodo di tempo viene utilizzato per definire i periodi in cui la manutenzione deve entrare in vigore. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
period	integer	Durata del periodo di manutenzione in secondi. Il valore specificato verrà arrotondato per difetto ai minuti. Intervallo di valori possibili: 300 - 86399940. Predefinito: 3600.

Proprietà	Type	Descrizione
timeperiod_type	integer	Tipo di periodo di tempo. Valori possibili: 0 - *(predefinito)* una sola volta; 2 - giornaliero; 3 - settimanale; 4 - mensile.
start_date	timestamp	Data in cui il periodo di manutenzione deve entrare in vigore. Il valore specificato verrà arrotondato per difetto ai minuti. Predefinito: data corrente. Comportamento della proprietà: - *supportata* se `timeperiod_type` è impostato su ”una sola volta”
start_time	integer	Ora del giorno in cui la manutenzione inizia, in secondi. Il valore specificato verrà arrotondato per difetto ai minuti. Predefinito: 0. Comportamento della proprietà: - *supportata* se `timeperiod_type` è impostato su ”giornaliero”, ”settimanale” o ”mensile”
every	integer	Per i periodi giornalieri e settimanali, `every` definisce gli intervalli di giorni o settimane ai quali la manutenzione deve entrare in vigore. Valore predefinito se `timeperiod_type` è impostato su ”giornaliero” o ”settimanale”: 1. Per i periodi mensili, quando `day` è impostato, la proprietà `every` definisce il giorno del mese in cui la manutenzione deve entrare in vigore. Valore predefinito se `timeperiod_type` è impostato su ”mensile” e `day` è impostato: 1. Per i periodi mensili, quando `dayofweek` è impostato, la proprietà `every` definisce la settimana del mese in cui la manutenzione deve entrare in vigore. Valori possibili se `timeperiod_type` è impostato su ”mensile” e `dayofweek` è impostato: 1 - *(predefinito)* prima settimana; 2 - seconda settimana; 3 - terza settimana; 4 - quarta settimana; 5 - ultima settimana. Comportamento della proprietà: - *supportata* se `timeperiod_type` è impostato su ”giornaliero”, ”settimanale” o ”mensile”

Proprietà	Type	Descrizione
dayofweek	integer	Giorni della settimana in cui la manutenzione deve entrare in vigore. Valori bitmap possibili: 1 - lunedì; 2 - martedì; 4 - mercoledì; 8 - giovedì; 16 - venerdì; 32 - sabato; 64 - domenica. Questo è un campo bitmask; è accettata qualsiasi somma dei valori bitmap possibili (ad esempio, 21 per lunedì, mercoledì e venerdì). Comportamento della proprietà: - *richiesta* se `timeperiod_type` è impostato su ”settimanale” oppure se `timeperiod_type` è impostato su ”mensile” e `day` non è impostato
day	integer	Giorno del mese in cui la manutenzione deve entrare in vigore. Comportamento della proprietà: - *richiesta* se `timeperiod_type` è impostato su ”mensile” e `dayofweek` non è impostato
month	integer	Mesi in cui la manutenzione deve entrare in vigore. Valori bitmap possibili: 1 - gennaio; 2 - febbraio; 4 - marzo; 8 - aprile; 16 - maggio; 32 - giugno; 64 - luglio; 128 - agosto; 256 - settembre; 512 - ottobre; 1024 - novembre; 2048 - dicembre. Questo è un campo bitmask; è accettata qualsiasi somma dei valori bitmap possibili (ad esempio, 585 per gennaio, aprile, luglio e ottobre). Comportamento della proprietà: - *richiesta* se `timeperiod_type` è impostato su ”mensile”

Nome dell’evento

L’oggetto del nome dell’evento viene utilizzato per definire quali problemi devono essere soppressi quando la manutenzione diventa effettiva. I nomi degli eventi possono essere specificati solo se `maintenance_type` dell’oggetto Maintenance è impostato su ”with data collection”. Presenta le seguenti proprietà.

Property	Type	Description
operator	integer	Operatore della condizione. Valori possibili: 2 - *(predefinito)* Contiene; 3 - Non contiene.

Property	Type	Description
value	string	Valore del nome dell'evento. Comportamento della proprietà: - *obbligatoria*

Tag del problema

L'oggetto tag del problema viene utilizzato per definire quali problemi devono essere soppressi quando entra in vigore la manutenzione. I tag possono essere specificati solo se `maintenance_type` dell'oggetto Maintenance è impostato su "with data collection". Presenta le seguenti proprietà.

Property	Type	Description
tag	string	Nome del tag del problema. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione. Valori possibili: 0 - Uguale; 1 - Diverso da; 2 - *(predefinito)* Contiene; 3 - Non contiene.
value	string	Valore del tag del problema.

maintenance.create

Descrizione

`object maintenance.create(object/array maintenances)`

Questo metodo consente di creare nuove manutenzioni.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Manutenzioni da creare.

Oltre alle proprietà standard delle manutenzioni, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groups	object/array	Gruppi di host da inserire nella manutenzione. I gruppi di host devono avere definita solo la proprietà `groupid`. Comportamento del parametro: - *obbligatorio* se `hosts` non è impostato
hosts	object/array	Host da inserire nella manutenzione. Gli host devono avere definita solo la proprietà `hostid`. Comportamento del parametro: - *obbligatorio* se `groups` non è impostato

Parametro	Tipo	Descrizione
triggers	object/array	trigger da inserire nella manutenzione. I trigger devono avere definita solo la proprietà `triggerid`. Comportamento del parametro: - *obbligatorio* se `groups` e `hosts` non sono impostati - *supportato* se maintenance_type=0 (con raccolta dati)
event_names	object/array	Nomi degli eventi per definire i problemi da includere nella manutenzione. Comportamento del parametro: - *supportato* se maintenance_type=0 (con raccolta dati)
tags	object/array	Tag dei problemi per definire i problemi da includere nella manutenzione. Comportamento del parametro: - *supportato* se `maintenance_type` dell'oggetto Maintenance è impostato su "con raccolta dati"
timeperiods	object/array	Periodi di tempo della manutenzione. Comportamento del parametro: - *obbligatorio*

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle maintenance create nella proprietà `maintenanceids`. L'ordine degli ID restituiti corrisponde all'ordine delle maintenance passate.

Esempi

Creazione di una manutenzione

Creare una manutenzione con raccolta dati per il gruppo di host con ID "20" e con i tag dei problemi **service:mysqld** e **error**. Deve essere attiva dal 17.03.2026 al 17.03.2027, entrare in vigore ogni domenica alle 18:00 e durare un'ora.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
```

```
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Creazione di una manutenzione ad hoc

Crea una manutenzione ad hoc della durata di un'ora con raccolta dati per il gruppo di host con ID "4", il trigger con ID "12345" e il nome dell'evento **Printer: ink low**.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Manutenzione ad hoc",
        "active_since": 1792057740,
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "4"
```

```
        ]
    },
    "id": 1
}
```

Vedi anche

- Host
- Gruppo di host
- Trigger
- Nome dell'evento
- Tag del problema
- Periodo di tempo

Fonte

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Descrizione

`object maintenance.delete(array maintenanceIds)`

Questo metodo consente di eliminare i periodi di manutenzione.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei periodi di manutenzione da eliminare.

valori restituiti

"(object)" Restituisce un oggetto contenente gli ID dell'oggetto eliminato periodi di manutenzione sotto la proprietà "maintenanceids".

Esempi

Eliminazione di più periodi di manutenzione

Eliminare due periodi di manutenzione.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CMaintenance::delete() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

Descrizione

`integer/array maintenance.get(object parameters)`

Il metodo consente di recuperare le manutenzioni in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
groupids	ID/array	Restituisce solo le manutenzioni assegnate ai gruppi di host specificati.
hostids	ID/array	Restituisce solo le manutenzioni assegnate agli host specificati.
triggerids	ID/array	Restituisce solo le manutenzioni assegnate ai trigger specificati.
maintenanceids	ID/array	Restituisce solo le manutenzioni con gli ID specificati.
selectHostGroups	query	Restituisce una proprietà `hostgroups` con i gruppi di host assegnati alla manutenzione.
selectHosts	query	Restituisce una proprietà `hosts` con gli host assegnati alla manutenzione.
selectTriggers	query	Restituisce una proprietà `triggers` con i trigger assegnati alla manutenzione.
selectEventNames	query	Restituisce una proprietà `event_names` con i nomi degli eventi assegnati alla manutenzione.
selectTags	query	Restituisce una proprietà `tags` con i tag dei problemi della manutenzione.
selectTimeperiods	query	Restituisce una proprietà `timeperiods` con i periodi di tempo della manutenzione.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero delle manutenzioni

Recupera tutte le manutenzioni configurate, i dati relativi all'ambito di manutenzione assegnato (gruppo di host, host, trigger, nome dell'evento, tag) e i periodi di tempo definiti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
      "name": "Manutenzione regolare del database",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1791936000",
      "active_till": "2107555200",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "20",
          "name": "Database",
          "flags": "0",
          "uuid": "748ad4d098d447d492bb935c907f652f"
        }
      ],
      "triggers": [],
      "event_names": [],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "4",
          "every": "1",
          "month": "4095",
          "dayofweek": "0",
          "day": "1",
          "start_time": "64800",
          "period": "3600",
          "start_date": "0"
        }
      ]
    },
    {
      "maintenanceid": "2",
      "name": "Ad hoc: Stampante: inchiostro ESAURITO",
      "maintenance_type": "0",
      "description": "",
```

```
      "active_since": "1792057740",
      "active_till": "1792061340",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "4",
          "name": "Server Zabbix",
          "flags": "0",
          "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
      ],
      "triggers": [],
      "event_names": [
        {
          "operator": "2",
          "value": "Stampante: inchiostro ESAURITO"
        }
      ],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "0",
          "every": "1",
          "month": "0",
          "dayofweek": "0",
          "day": "0",
          "start_time": "0",
          "period": "3600",
          "start_date": "1792057740"
        }
      ]
    }
  ],
  "id": 1
}
```


Vedi anche

- Host
- Gruppo di host
- Trigger
- Nome dell'evento
- Tag del problema
- Periodo di tempo

Sorgente

CMaintenance::get() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Descrizione

`object maintenance.update(object/array maintenances)`

Questo metodo consente di aggiornare le maintenances esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà della manutenzione da aggiornare.

La proprietà `maintenanceid` deve essere definita per ogni manutenzione; tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, mentre tutte le altre rimarranno invariate.

Oltre alle proprietà standard della manutenzione, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groups	object/array	Gruppi di host con cui sostituire i gruppi correnti. I gruppi di host devono avere definita solo la proprietà `groupid`. Comportamento del parametro: - *obbligatorio* se `hosts` non è impostato
hosts	object/array	Host con cui sostituire gli host correnti. Gli host devono avere definita solo la proprietà `hostid`. Comportamento del parametro: - *obbligatorio* se `groups` non è impostato
triggers	object/array	Trigger con cui sostituire i trigger correnti. I trigger devono avere definita solo la proprietà `triggerid`. Comportamento del parametro: - *obbligatorio* se `groups` e `hosts` non sono impostati - *supportato* se maintenance_type=0 (con raccolta dati)
event_names	object/array	Nomi degli eventi con cui sostituire i nomi degli eventi correnti. Comportamento del parametro: - *supportato* se maintenance_type=0 (con raccolta dati)
tags	object/array	Tag dei problemi con cui sostituire i tag correnti. Comportamento del parametro: - *supportato* se `maintenance_type` dell'oggetto Maintenance è impostato su "con raccolta dati"
timeperiods	object/array	Periodi di tempo della manutenzione con cui sostituire i periodi correnti.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID dell'oggetto aggiornato manutenzioni sotto la proprietà `maintenanceids`.

Esempi

Assegnazione di host diversi

Sostituisci gli host attualmente assegnati alla manutenzione con altri due.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Periodo di tempo

Fonte

CMaintenance::update() in *ui/include/classes/api/services/CMaintenance.php*.

Mappa

Questa classe è progettata per funzionare con le mappe.

Riferimenti agli oggetti:

- Mappa
- Elemento della mappa
  - Elemento della mappa Host
  - Elemento della mappa Gruppo host
  - Elemento della mappa Mappa
  - Elemento della mappa Trigger
  - Elemento della mappa Tag
  - Elemento della mappa URL
- Collegamento della mappa
  - Trigger del collegamento della mappa
  - Indicatori del collegamento della mappa
- URL della mappa
- Utente della mappa
- Gruppo utenti della mappa
- Forme della mappa
- Linee della mappa

Metodi disponibili:

- map.create - crea nuove mappe
- map.delete - elimina mappe
- map.get - recupera mappe
- map.update - aggiorna mappe

Oggetto mappa

I seguenti oggetti sono direttamente correlati all'API `map`.

Mappa

L'oggetto mappa ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
sysmapid	ID	ID della mappa. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
height	integer	Altezza della mappa in pixel. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Proprietà	Tipo	Descrizione
name	string	Nome della mappa. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
width	integer	Larghezza della mappa in pixel. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
backgroundid	ID	ID dell'immagine utilizzata come sfondo della mappa.
background_scale	integer	Indica se abilitare il ridimensionamento proporzionale dell'immagine di sfondo. Valori possibili: 0 - disabilita il ridimensionamento; 1 - *(predefinito)* abilita il ridimensionamento dell'immagine.
expand_macros	integer	Indica se espandere le macro nelle etichette durante la configurazione della mappa. Valori possibili: 0 - *(predefinito)* non espandere le macro; 1 - espandere le macro.
expandproblem	integer	Indica se il trigger del problema verrà visualizzato per gli elementi con un solo problema. Valori possibili: 0 - visualizza sempre il numero di problemi; 1 - *(predefinito)* visualizza il trigger del problema se c'è un solo problema.
grid_align	integer	Indica se abilitare l'allineamento alla griglia. Valori possibili: 0 - disabilita l'allineamento alla griglia; 1 - *(predefinito)* abilita l'allineamento alla griglia.
grid_show	integer	Indica se mostrare la griglia sulla mappa. Valori possibili: 0 - non mostrare la griglia; 1 - *(predefinito)* mostrare la griglia.
grid_size	integer	Dimensione della griglia della mappa in pixel. Valori supportati: 20, 40, 50, 75 e 100. Predefinito: 50.
highlight	integer	Indica se l'evidenziazione delle icone è abilitata. Valori possibili: 0 - evidenziazione disabilitata; 1 - *(predefinito)* evidenziazione abilitata.
iconmapid	ID	ID della mappa icone utilizzata nella mappa.
label_format	integer	Indica se abilitare le etichette avanzate. Valori possibili: 0 - *(predefinito)* disabilita le etichette avanzate; 1 - abilita le etichette avanzate.
label_location	integer	Posizione dell'etichetta dell'elemento della mappa. Valori possibili: 0 - *(predefinito)* in basso; 1 - a sinistra; 2 - a destra; 3 - in alto.

Proprietà	Tipo	Descrizione
label_string_host	string	Etichetta personalizzata per gli elementi host. Comportamento della proprietà: - *obbligatoria* se `label_type_host` è impostato su ”custom”
label_string_hostgroup	string	Etichetta personalizzata per gli elementi del gruppo host. Comportamento della proprietà: - *obbligatoria* se `label_type_hostgroup` è impostato su ”custom”
label_string_image	string	Etichetta personalizzata per gli elementi immagine. Comportamento della proprietà: - *obbligatoria* se `label_type_image` è impostato su ”custom”
label_string_map	string	Etichetta personalizzata per gli elementi mappa. Comportamento della proprietà: - *obbligatoria* se `label_type_map` è impostato su ”custom”
label_string_trigger	string	Etichetta personalizzata per gli elementi trigger. Comportamento della proprietà: - *obbligatoria* se `label_type_trigger` è impostato su ”custom”
label_type	integer	Tipo di etichetta dell’elemento della mappa. Valori possibili: 0 - etichetta; 1 - indirizzo IP; 2 - *(predefinito)* nome dell’elemento; 3 - solo stato; 4 - niente.
label_type_host	integer	Tipo di etichetta per gli elementi host. Valori possibili: 0 - etichetta; 1 - indirizzo IP; 2 - *(predefinito)* nome dell’elemento; 3 - solo stato; 4 - niente; 5 - personalizzata.
label_type_hostgroup	integer	Tipo di etichetta per gli elementi del gruppo host. Valori possibili: 0 - etichetta; 2 - *(predefinito)* nome dell’elemento; 3 - solo stato; 4 - niente; 5 - personalizzata.
label_type_image	integer	Tipo di etichetta per gli elementi del gruppo host. Valori possibili: 0 - etichetta; 2 - *(predefinito)* nome dell’elemento; 4 - niente; 5 - personalizzata.
label_type_map	integer	Tipo di etichetta per gli elementi mappa. Valori possibili: 0 - etichetta; 2 - *(predefinito)* nome dell’elemento; 3 - solo stato; 4 - niente; 5 - personalizzata.

Proprietà	Tipo	Descrizione
label_type_trigger	integer	Tipo di etichetta per gli elementi trigger. Valori possibili: 0 - etichetta; 2 - *(predefinito)* nome dell'elemento; 3 - solo stato; 4 - niente; 5 - personalizzata.
markelements	integer	Indica se evidenziare gli elementi della mappa che hanno cambiato di recente il proprio stato. Valori possibili: 0 - *(predefinito)* non evidenziare gli elementi; 1 - evidenziare gli elementi.
severity_min	integer	Gravità minima dei trigger che verranno visualizzati sulla mappa. Fare riferimento alla proprietà `severity` del trigger per un elenco delle gravità dei trigger supportate.
show_element_label	integer	Come visualizzare per impostazione predefinita le etichette degli elementi. Valori possibili: 0 - visualizza sempre; 1 - *(predefinito)* nascondi automaticamente.
show_link_label	integer	Come visualizzare per impostazione predefinita le etichette dei collegamenti. Valori possibili: 0 - visualizza sempre; 1 - *(predefinito)* nascondi automaticamente.
show_unack	integer	Come devono essere visualizzati i problemi. Valori possibili: 0 - *(predefinito)* visualizza il conteggio di tutti i problemi; 1 - visualizza solo il conteggio dei problemi non riconosciuti; 2 - visualizza separatamente il conteggio dei problemi riconosciuti e non riconosciuti.
userid	ID	ID dell'utente proprietario della mappa.
private	integer	Tipo di condivisione della mappa. Valori possibili: 0 - mappa pubblica; 1 - *(predefinito)* mappa privata.
show_suppressed	integer	Indica se i problemi soppressi vengono mostrati. Valori possibili: 0 - *(predefinito)* nascondi i problemi soppressi; 1 - mostra i problemi soppressi.

Elemento della mappa

L'oggetto elemento della mappa definisce un oggetto visualizzato su una mappa. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
selementid	ID	ID dell'elemento della mappa. Comportamento della proprietà: - *sola lettura*

Proprietà	Type	Descrizione
elements	array	Oggetto dati Element. Comportamento della proprietà: - *obbligatoria* se `elementtype` è impostato su ”host”, ”map”, ”trigger” o ”host group”
elementtype	integer	Tipo di elemento della mappa. Valori possibili: 0 - host; 1 - map; 2 - trigger; 3 - host group; 4 - immagine. Comportamento della proprietà: - *obbligatoria*
iconid_off	ID	ID dell’immagine utilizzata per visualizzare l’elemento nello stato predefinito. Comportamento della proprietà: - *obbligatoria*
areatype	integer	Come devono essere visualizzati i singoli host di un host group. Valori possibili: 0 - *(predefinito)* l’elemento host group occuperà l’intera mappa; 1 - l’elemento host group avrà una dimensione fissa.
elementsubtype	integer	Come un elemento host group deve essere visualizzato su una mappa. Valori possibili: 0 - *(predefinito)* visualizza l’host group come un singolo elemento; 1 - visualizza separatamente ciascun host nel gruppo.
evaltype	integer	Condizione di filtraggio dei tag dell’elemento della mappa metodo di valutazione. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
height	integer	Altezza in pixel dell’elemento host group a dimensione fissa. Predefinito: 200.
iconid_disabled	ID	ID dell’immagine utilizzata per visualizzare gli elementi della mappa disabilitati. Comportamento della proprietà: - *supportata* se `elementtype` è impostato su ”host”, ”map”, ”trigger” o ”host group”
iconid_maintenance	ID	ID dell’immagine utilizzata per visualizzare gli elementi della mappa in manutenzione. Comportamento della proprietà: - *supportata* se `elementtype` è impostato su ”host”, ”map”, ”trigger” o ”host group”
iconid_on	ID	ID dell’immagine utilizzata per visualizzare gli elementi della mappa con problemi. Comportamento della proprietà: - *supportata* se `elementtype` è impostato su ”host”, ”map”, ”trigger” o ”host group”
label	string	Etichetta dell’elemento.

Proprietà	Type	Descrizione
label_location	integer	Posizione dell'etichetta dell'elemento della mappa. Valori possibili: -1 - *(predefinito)* posizione predefinita; 0 - in basso; 1 - a sinistra; 2 - a destra; 3 - in alto.
permission	integer	Tipo di livello di autorizzazione. Valori possibili: -1 - nessuno; 2 - sola lettura; 3 - lettura-scrittura.
show_label	integer	Come visualizzare l'etichetta dell'elemento. Valori possibili: -1 - *(predefinito)* valore predefinito della mappa; 0 - visualizza sempre; 1 - nascondi automaticamente.
sysmapid	ID	ID della mappa a cui appartiene l'elemento. Comportamento della proprietà: - *sola lettura*
urls	array	URL dell'elemento della mappa. L'oggetto URL dell'elemento della mappa è descritto in dettaglio di seguito.
use_iconmap	integer	Indica se la mappatura delle icone deve essere utilizzata per gli elementi host. Valori possibili: 0 - non utilizzare la mappatura delle icone; 1 - *(predefinito)* utilizza la mappatura delle icone.
viewtype	integer	Algoritmo di posizionamento dell'elemento host group. Valori possibili: 0 - *(predefinito)* griglia.
width	integer	Larghezza in pixel dell'elemento host group a dimensione fissa. Predefinito: 200.
x	integer	Coordinate X dell'elemento in pixel. Predefinito: 0.
y	integer	Coordinate Y dell'elemento in pixel. Predefinito: 0.
zindex	integer	Valore utilizzato per ordinare gli elementi della mappa (z-index). Predefinito: 0.

Elemento mappa Host

L'oggetto elemento mappa Host definisce un elemento host.

Property	Type	Description
hostid	ID	ID dell'host.

Elemento mappa Gruppo host

L’oggetto elemento mappa Gruppo host definisce un elemento di gruppo host.

Proprietà	Type	Descrizione
groupid	ID	ID del gruppo host.

Oggetto elemento mappa Mappa

L’oggetto elemento mappa Mappa definisce un elemento della mappa.

Proprietà	Type	Descrizione
sysmapid	ID	ID della mappa.

Elemento mappa Trigger

L’oggetto elemento mappa Trigger definisce uno o più elementi trigger.

Proprietà	Type	Descrizione
triggerid	ID	ID del trigger.

Tag dell’elemento della mappa

L’oggetto Tag dell’elemento della mappa ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag dell’elemento della mappa. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione del tag dell’elemento della mappa. Valori possibili: 0 - *(predefinito)* Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
value	string	Valore del tag dell’elemento della mappa.

URL dell’elemento della mappa

L’oggetto URL dell’elemento della mappa definisce un collegamento cliccabile che sarà disponibile per uno specifico elemento della mappa. Ha le seguenti proprietà:

Property	Type	Description
sysmapelementurlid	ID	ID dell’URL dell’elemento della mappa. Property behavior: - *sola lettura*
name	string	Didascalia del collegamento. Property behavior: - *obbligatorio*
url	string	URL del collegamento. Property behavior: - *obbligatorio*
selementid	ID	ID dell’elemento della mappa a cui appartiene l’URL.

Collegamento della mappa

L'oggetto collegamento della mappa definisce un collegamento tra due elementi della mappa. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
linkid	ID	ID del collegamento della mappa. Comportamento della proprietà: - *sola lettura*
sysmapid	ID	ID della mappa a cui appartiene il collegamento.
selementid1	ID	ID del primo elemento della mappa collegato a un'estremità. Comportamento della proprietà: - *obbligatorio*
selementid2	ID	ID del primo elemento della mappa collegato all'altra estremità. Comportamento della proprietà: - *obbligatorio*
drawtype	integer	Stile di disegno della linea del collegamento. Valori possibili: 0 - *(predefinito)* linea; 2 - linea in grassetto; 3 - linea puntinata; 4 - linea tratteggiata.
color	string	Colore della linea come codice colore esadecimale. Predefinito: 000000.
label	string	Etichetta del collegamento.
show_label	integer	Come visualizzare l'etichetta del collegamento. Valori possibili: -1 - *(predefinito)* valore predefinito della mappa; 0 - visualizza sempre; 1 - nascondi automaticamente.
indicator_type	integer	Seleziona il tipo di indicatore del collegamento. Valori possibili: 0 - *(predefinito)* collegamento statico; 1 - trigger; 2 - valore dell'item.
linktriggers	array	Trigger del collegamento della mappa da utilizzare come indicatori di stato del collegamento. L'oggetto trigger del collegamento della mappa è descritto in dettaglio di seguito. Comportamento della proprietà: - *obbligatorio* se `indicator_type` è impostato su "trigger" (1).
itemid	ID	ID dell'item. Comportamento della proprietà: - *obbligatorio* se `indicator_type` è impostato su "valore dell'item" (2).
highlights	array	Evidenziazioni del collegamento della mappa da utilizzare come indicatori di stato del collegamento. L'oggetto indicatori del collegamento della mappa è descritto in dettaglio di seguito. Comportamento della proprietà: - *obbligatorio* se `indicator_type` è impostato su "valore dell'item" (2).

Proprietà	Type	Descrizione
thresholds	array	Soglie del collegamento della mappa da utilizzare come indicatori di stato del collegamento. L'oggetto indicatori del collegamento della mappa è descritto in dettaglio di seguito. Comportamento della proprietà: - *obbligatorio* se `indicator_type` è impostato su "valore dell'item" (2).
permission	integer	Tipo di livello di autorizzazione. Valori possibili: -1 - nessuno; 2 - sola lettura; 3 - lettura-scrittura.

Trigger del collegamento della mappa

L'oggetto trigger del collegamento della mappa definisce un indicatore di stato del collegamento della mappa in base allo stato di un trigger. Ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
triggerid	ID	ID del trigger utilizzato come indicatore del collegamento. Comportamento della proprietà: - *obbligatorio*
color	string	Colore dell'indicatore come codice colore esadecimale. Predefinito: `DD0000`.
drawtype	integer	Stile di disegno dell'indicatore. Valori possibili: 0 - *(predefinito)* linea; 2 - linea in grassetto; 3 - linea puntinata; 4 - linea tratteggiata.

Indicatori dei collegamenti della mappa

L'oggetto degli indicatori dei collegamenti della mappa definisce un indicatore di stato del collegamento della mappa in base al valore dell'item. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
pattern	string	Espressione regolare con cui confrontare. Disponibile solo per le evidenziazioni.
sortorder	integer	Utilizzato per definire l'ordine di ordinamento delle evidenziazioni. Disponibile solo per le evidenziazioni.
threshold	string	Valore numerico con cui confrontare. Disponibile solo per le soglie.
drawtype	integer	Stile di disegno dell'indicatore. Valori possibili: 0 - *(predefinito)* linea; 2 - linea in grassetto; 3 - linea puntinata; 4 - linea tratteggiata.

Proprietà	Type	Descrizione
color	string	Colore dell'indicatore come codice colore esadecimale. Predefinito: `DD0000`. Comportamento della proprietà: - *obbligatorio*

URL della mappa

L'oggetto URL della mappa definisce un collegamento cliccabile che sarà disponibile per tutti gli elementi di un tipo specifico sulla mappa. Ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
sysmapurlid	ID	ID dell'URL della mappa. Comportamento della proprietà: - *sola lettura*
name	string	Didascalia del collegamento. Comportamento della proprietà: - *obbligatoria*
url	string	URL del collegamento. Comportamento della proprietà: - *obbligatoria*
elementtype	integer	Tipo di elemento della mappa per il quale l'URL sarà disponibile. Fare riferimento alla proprietà `type` dell'elemento della mappa per un elenco dei tipi supportati. Predefinito: 0.
sysmapid	ID	ID della mappa a cui appartiene l'URL.

Utente della mappa

Elenco dei permessi della mappa in base agli utenti. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
sysmapuserid	ID	ID dell'utente della mappa. Comportamento della proprietà: - *sola lettura*
userid	ID	ID dell'utente. Comportamento della proprietà: - *obbligatoria*
permission	integer	Tipo di livello di permesso. Valori possibili: 2 - sola lettura; 3 - lettura-scrittura. Comportamento della proprietà: - *obbligatoria*

Gruppo utenti della mappa

Elenco dei permessi della mappa in base ai gruppi utenti. Ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
sysmapusrgrpid	ID	ID del gruppo utenti della mappa. Comportamento della proprietà: - *sola lettura*
usrgrpid	ID	ID del gruppo utenti. Comportamento della proprietà: - *obbligatoria*
permission	integer	Tipo di livello di permesso. Valori possibili: 2 - sola lettura; 3 - lettura-scrittura. Comportamento della proprietà: - *obbligatoria*

Forme della mappa

L'oggetto forma della mappa definisce una forma geometrica (con o senza testo) visualizzata su una mappa. Ha le seguenti proprietà:

Property	Type	Description
sysmap_shapeid	ID	ID dell'elemento forma della mappa. Property behavior: - *sola lettura*
type	integer	Tipo dell'elemento forma della mappa. Valori possibili: 0 - rettangolo; 1 - ellisse. La proprietà è obbligatoria quando vengono create nuove forme. Property behavior: - *obbligatoria*
x	integer	Coordinate X della forma in pixel. Predefinito: 0.
y	integer	Coordinate Y della forma in pixel. Predefinito: 0.
width	integer	Larghezza della forma in pixel. Predefinito: 200.
height	integer	Altezza della forma in pixel. Predefinito: 200.
text	string	Testo della forma.

Property	Type	Description
font	integer	Carattere del testo all'interno della forma. Valori possibili: 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 Predefinito: 9.
font_size	integer	Dimensione del carattere in pixel. Predefinito: 11.
font_color	string	Colore del carattere. Predefinito: 000000.
text_halign	integer	Allineamento orizzontale del testo. Valori possibili: 0 - centro; 1 - sinistra; 2 - destra. Predefinito: 0.
text_valign	integer	Allineamento verticale del testo. Valori possibili: 0 - centro; 1 - alto; 2 - basso. Predefinito: 0.
border_type	integer	Tipo di bordo. Valori possibili: 0 - nessuno; 1 - ——————; 2 - ┅; 3 - - - -. Predefinito: 0.
border_width	integer	Larghezza del bordo in pixel. Predefinito: 0.
border_color	string	Colore del bordo. Predefinito: 000000.
background_color	string	Colore di sfondo (colore di riempimento). Predefinito: (vuoto).
zindex	integer	Valore utilizzato per ordinare tutte le forme e le linee (z-index). Predefinito: 0.

Linee della mappa

L'oggetto linea della mappa definisce una linea visualizzata su una mappa. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
sysmap_shapeid	ID	ID dell'elemento forma della mappa. Comportamento della proprietà: *- sola lettura*
x1	integer	Coordinate X del punto 1 della linea in pixel. Predefinito: 0.
y1	integer	Coordinate Y del punto 1 della linea in pixel. Predefinito: 0.
x2	integer	Coordinate X del punto 2 della linea in pixel. Predefinito: 200.
y2	integer	Coordinate Y del punto 2 della linea in pixel. Predefinito: 200.
line_type	integer	Tipo delle linee. Valori possibili: 0 - nessuno; 1 - ————; 2 - ┅; 3 - – – –. Predefinito: 0.
line_width	integer	Larghezza delle linee in pixel. Predefinito: 0.
line_color	string	Colore della linea. Predefinito: 000000.
zindex	integer	Valore utilizzato per ordinare tutte le forme e le linee (z-index). Predefinito: 0.

map.create

Descrizione

`object map.create(object/array maps)`

Questo metodo consente di creare nuovi map.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Mappe da creare.

Oltre alle proprietà standard della mappa, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
links	array	Collegamenti della mappa da creare nella mappa.
selements	array	Elementi della mappa da creare nella mappa.
urls	array	URL della mappa da creare nella mappa.
users	array	Condivisioni utente della mappa da creare nella mappa.

Parametro	Type	Descrizione
userGroups	array	Condivisioni dei gruppi utenti della mappa da creare nella mappa.
shapes	array	Forme della mappa da creare nella mappa.
lines	array	Linee della mappa da creare nella mappa.

Note:
Per creare collegamenti della mappa, è necessario impostare `selementid` di un elemento della mappa su un valore arbitrario e quindi usare questo valore per fare riferimento a tale elemento nelle proprietà del collegamento `selementid1` o `selementid2`. Quando l'elemento viene creato, questo valore verrà sostituito con l'ID corretto generato da Zabbix. Vedi esempio.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle mappe create sotto la proprietà `sysmapids`. L'ordine degli ID restituiti corrisponde all'ordine delle mappe passati.

Esempi

Creare una mappa vuota

Creare una mappa senza elementi.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Creare una mappa di host

Creare una mappa con due elementi host e un collegamento tra di essi. Notare l'uso dei valori temporanei "selementid1" e "selementid2" nell'oggetto del collegamento della mappa per fare riferimento agli elementi della mappa.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Creare una mappa di trigger

Creare una mappa con un elemento trigger, che contiene due trigger.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Condivisione della mappa

Creare una mappa con due tipi di condivisione (utente e gruppo di utenti).

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Forme della mappa

Crea una mappa con il titolo del nome della mappa.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Linee della mappa

Crea una linea della mappa.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

Vedi anche

- Elemento della mappa
- Collegamento della mappa

- URL della mappa
- Utente della mappa
- Gruppo utenti della mappa
- Forma della mappa
- Linea della mappa

Fonte

CMap::create() in *ui/include/classes/api/services/CMap.php*.

map.delete

Descrizione

`object map.delete(array mapIds)`

Questo metodo consente di eliminare le mappe.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle mappe da eliminare.

Restituisce i valori

`(oggetto)` Restituisce un oggetto contenente gli ID delle mappe eliminate sotto la proprietà `sysmapids`.

Esempi

Eliminare più mappe

Eliminare due mappe.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

Fonte

CMap::delete() in *ui/include/classes/api/services/CMap.php*.

map.get

Descrizione

`integer/array map.get(object parameters)`

Il metodo consente di recuperare le mappe in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
sysmapids	ID/array	Restituisce solo le mappe con gli ID specificati.
userids	ID/array	Restituisce solo le mappe che appartengono agli ID utente specificati.
expandUrls	flag	Aggiunge gli URL globali della mappa ai corrispondenti elementi della mappa ed espande le macro in tutti gli URL degli elementi della mappa.
selectIconMap	query	Restituisce una proprietà `iconmap` con la mappa di icone utilizzata nella mappa.
selectLinks	query	Restituisce una proprietà `links` con i collegamenti della mappa tra gli elementi.
selectSelements	query	Restituisce una proprietà `selements` con gli elementi della mappa.
selectUrls	query	Restituisce una proprietà `urls` con gli URL della mappa.
selectUsers	query	Restituisce una proprietà `users` con gli utenti con cui la mappa è condivisa.
selectUserGroups	query	Restituisce una proprietà `userGroups` con i gruppi di utenti con cui la mappa è condivisa.
selectShapes	query	Restituisce una proprietà `shapes` con le forme della mappa.
selectLines	query	Restituisce una proprietà `lines` con le linee della mappa.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `name`, `width`, `height`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare una mappa

Recupera tutti i dati della mappa "3".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Mappa delle icone
- Elemento della mappa
- Collegamento della mappa
- URL della mappa
- Utente della mappa
- Gruppo di utenti della mappa
- Forme della mappa
- Linee della mappa

Fonte

CMap::get() in *ui/include/classes/api/services/CMap.php*.

map.update

Descrizione

`object map.update(object/array maps)`

Questo metodo consente di aggiornare mappe esistenti.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà della mappa da aggiornare.

La proprietà `mapid` deve essere definita per ogni mappa, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard della mappa, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
links	array	Collegamenti della mappa per sostituire i collegamenti esistenti.
selements	array	Elementi della mappa per sostituire gli elementi esistenti.
urls	array	URL della mappa per sostituire gli URL esistenti.
users	array	Condivisioni utente della mappa per sostituire gli elementi esistenti.
userGroups	array	Condivisioni gruppo utenti della mappa per sostituire gli elementi esistenti.
shapes	array	Forme della mappa per sostituire le forme esistenti.
lines	array	Linee della mappa per sostituire le linee esistenti.

Note:
Per creare collegamenti della mappa tra nuovi elementi della mappa, sarà necessario impostare `selementid` di un elemento su un valore arbitrario e poi usare questo valore per fare riferimento a questo elemento nelle proprietà `selementid1` o `selementid2` dei collegamenti. Quando l'elemento viene creato, questo valore verrà sostituito con l'ID corretto generato da Zabbix. Vedere l'esempio per map.create.

Restituisce i valori

(`oggetto`) Restituisce un oggetto contenente gli ID delle mappe aggiornate sotto la proprietà `sysmapids`.

Esempi

Ridimensionare una mappa

Modificare la dimensione della mappa a 1200x1200 pixel.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Cambiare il proprietario della mappa

Disponibile solo per admin e super admin.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Vedi anche

- Elemento della mappa
- Collegamento della mappa
- URL della mappa
- Utente della mappa
- Gruppo utenti della mappa
- Forme della mappa
- Linee della mappa

Fonte

CMap::update() in *ui/include/classes/api/services/CMap.php*.

Mappa dei valori

Questa classe è progettata per funzionare con le mappe dei valori.

Riferimenti agli oggetti:

- Mappa dei valori
  - Associazioni dei valori

Metodi disponibili:

- valuemap.create - crea nuove mappe dei valori
- valuemap.delete - elimina mappe dei valori
- valuemap.get - recupera mappe dei valori
- valuemap.update - aggiorna mappe dei valori

Oggetto mappa dei valori

I seguenti oggetti sono direttamente correlati all'API `valuemap`.

Mappa dei valori

L'oggetto mappa dei valori ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
valuemapid	ID	ID della mappa dei valori. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
hostid	ID	ID dell'host o del template a cui appartiene la mappa dei valori. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
name	string	Nome della mappa dei valori. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
mappings	array	Mappature dei valori per la mappa dei valori corrente. L'oggetto mapping è descritto in dettaglio di seguito. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
uuid	string	Identificatore univoco universale, utilizzato per collegare le mappe dei valori importate a quelle già esistenti. Generato automaticamente, se non specificato. Comportamento della proprietà: - *supportato* se la mappa dei valori appartiene a un template

Mappature dei valori

L'oggetto delle mappature dei valori definisce le mappature dei valori della mappa dei valori. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
type	integer	Tipo di corrispondenza della mappatura. Valori possibili: 0 - *(predefinito)* la mappatura verrà applicata se il valore è uguale; 1 - la mappatura verrà applicata se il valore è maggiore o uguale[1]; 2 - la mappatura verrà applicata se il valore è minore o uguale[1]; 3 - la mappatura verrà applicata se il valore rientra in un intervallo (gli intervalli sono inclusivi; è possibile definire più intervalli, separati da una virgola)[1]; 4 - la mappatura verrà applicata se il valore corrisponde a un'espressione regolare[2]; 5 - se non viene trovata alcuna corrispondenza, la mappatura non verrà applicata e verrà utilizzato il valore predefinito. Se `type` è impostato su "0", "1", "2", "3", "4", allora `value` non può essere vuoto. Se `type` è impostato su "5", allora `value` deve essere vuoto.
value	string	Valore originale. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "1", "2", "3", "4" - *supportato* se `type` è impostato su "5"
newvalue	string	Valore a cui viene mappato il valore originale. Comportamento della proprietà: - *obbligatorio*

[1] supportato solo per gli item con tipo di valore "numeric unsigned", "numeric float".
 [2] supportato solo per gli item con tipo di valore "character".

valuemap.create

Descrizione

`object valuemap.create(object/array valuemaps)`

Questo metodo consente di creare nuovi value map.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Mappe valori da creare.

Il metodo accetta mappe valori con le proprietà standard delle mappe valori.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle mappe di valori create nella proprietà `valuemapids`. L'ordine degli ID restituiti corrisponde all'ordine delle mappe di valori passate.

Esempi

Creazione di una mappa dei valori

Creare una mappa dei valori con due mappature.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CValueMap::create() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

Descrizione

`object valuemap.delete(array valuemapids)`

Questo metodo consente di eliminare le mappe dei valori.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

”(array)” ID delle mappe valori da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle mappe di valori eliminate nella proprietà `valuemapids`.

Esempi

Eliminazione di più mappe di valori

Elimina due mappe di valori.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
```

```
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CValueMap::delete() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

Descrizione

`integer/array valuemap.get(object parameters)`

Il metodo consente di recuperare le value map in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
valuemapids	ID/array	Restituisce solo le mappe di valori con gli ID specificati.
selectMappings	query	Restituisce le mappature dei valori per la mappa di valori corrente nella proprietà `mappings`. Supporta `count`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `valuemapid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero delle mappe dei valori

Recupera tutte le mappe dei valori configurate.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Recupera una mappa dei valori con le relative associazioni.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Fonte

CValueMap::get() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

Descrizione

`object valuemap.update(object/array valuemaps)`

Questo metodo consente di aggiornare i value map esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object/array)` Proprietà della mappa dei valori da aggiornare.

La proprietà `valuemapid` deve essere definita per ogni mappa dei valori, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Valori restituiti

(object) Restituisce un oggetto contenente gli ID delle mappe di valori aggiornate nella proprietà valuemapids.

Esempi

Modifica del nome della mappa dei valori

Modificare il nome della mappa dei valori in "Device status".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Modifica delle mappature per una mappa dei valori.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CValueMap::update() in *ui/include/classes/api/services/CValueMap.php*.

Mappa delle icone

Questa classe è progettata per funzionare con le mappe delle icone.

Riferimenti agli oggetti:

- Mappa delle icone
- Associazione delle icone

Metodi disponibili:

- iconmap.create - crea nuove mappe delle icone
- iconmap.delete - elimina mappe delle icone
- iconmap.get - recupera mappe delle icone
- iconmap.update - aggiorna mappe delle icone

Oggetto mappa delle icone

I seguenti oggetti sono direttamente correlati all'API `iconmap`.

Mappa delle icone

L'oggetto mappa delle icone ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
iconmapid	ID	ID della mappa delle icone. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
default_iconid	ID	ID dell'icona predefinita. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
name	string	Nome della mappa delle icone. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Mappatura delle icone

L'oggetto di mappatura delle icone definisce un'icona specifica da utilizzare per gli host con un determinato valore del campo di inventario. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
iconid	ID	ID dell'icona utilizzata dalla mappatura delle icone. Property behavior: - *required*
expression	string	Espressione da confrontare con il campo di inventario. Property behavior: - *required*
inventory_link	integer	ID del campo di inventario dell'host. Fare riferimento a host inventory object per un elenco dei campi di inventario supportati. Property behavior: - *required*

Proprietà	Type	Descrizione
sortorder	integer	Posizione della mappatura delle icone nella mappa delle icone. Property behavior: - *read-only*

iconmap.create

Descrizione

`object iconmap.create(object/array iconMaps)`

Questo metodo consente di creare nuovi icon map.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Mappe di icone da creare.

Oltre alle proprietà standard della mappa di icone, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
mappings	array	Associazioni di icone da creare per la mappa di icone. Comportamento del parametro: - *obbligatorio*

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle mappe di icone nella proprietà `iconmapids`. L'ordine degli ID restituiti corrisponde all'ordine delle mappe di icone passate.

Esempi

Creare una mappa di icone

Creare una mappa di icone per visualizzare host di tipi diversi.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

Vedi anche

- Mappatura delle icone

Fonte

CIconMap::create() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Descrizione

`object iconmap.delete(array iconMapIds)`

Questo metodo consente di eliminare le mappe delle icone.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID delle mappe di icone da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle mappe di icone eliminate nella proprietà `iconmapids`.

Esempi

Eliminare più mappe di icone

Eliminare due mappe di icone.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CIconMap::delete() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

Descrizione

`integer/array iconmap.get(object parameters)`

Il metodo consente di recuperare le mappe delle icone in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
iconmapids	ID/array	Restituisce solo le mappe di icone con gli ID specificati.
sysmapids	ID/array	Restituisce solo le mappe di icone utilizzate nelle mappe specificate.
selectMappings	query	Restituisce una proprietà `mappings` con le associazioni di icone utilizzate.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `iconmapid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare una mappa di icone

Recupera tutti i dati della mappa di icone "3".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "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": "Icone del tipo di host",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

Vedi anche

- Mappatura delle icone

Fonte

CIconMap::get() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

Descrizione

`object iconmap.update(object/array iconMaps)`

Questo metodo consente di aggiornare gli icon map esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà della mappa di icone da aggiornare.

La proprietà `iconmapid` deve essere definita per ogni mappa di icone, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard della mappa di icone, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
mappings	array	Mappature di icone per sostituire le mappature di icone esistenti.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle mappe di icone aggiornate nella proprietà `iconmapids`.

Esempi

Rinomina mappa di icone

Rinomina una mappa di icone in "OS icons".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

Vedi anche

- Mappatura delle icone

Fonte

CIconMap::update() in *ui/include/classes/api/services/CIconMap.php*.

Metrica APM

Questa classe è progettata per funzionare con le metriche raccolte per il monitoraggio delle prestazioni delle applicazioni (APM).

Riferimenti agli oggetti:

- Metrica APM
- Esemplare

Metodi disponibili:

- apmmetric.get - recupera i punti dati della metrica

Oggetto metrica APM

I seguenti oggetti sono direttamente correlati all'API `apmmetric`.

Note:
Ogni proprietà, ad eccezione di `type`, corrisponde a una colonna nelle tabelle delle metriche (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` e `otel_metrics_exponential_histogram`) della sorgente dati di telemetria. La maggior parte delle proprietà corrisponde alla colonna con lo stesso nome (ad esempio, la proprietà `metric_name` corrisponde alla colonna `MetricName`). Quando i nomi differiscono, le colonne della sorgente dati sono specificate nelle tabelle seguenti. Per l'elenco delle colonne, vedere Colonne di dati di telemetria supportate.

Metrica APM

L'oggetto metrica APM rappresenta un punto dati di una metrica (il valore o i valori aggregati di una misurazione in un momento specifico).

L'oggetto metrica APM ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
type	integer	Tipo del punto dati. Valori possibili: 0 - Gauge; 1 - Sum; 2 - Histogram; 3 - Exponential histogram. Il tipo è determinato dalla tabella che contiene il punto dati: `otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` o `otel_metrics_exponential_histogram`. Comportamento della proprietà: - *sola lettura*
resource_attributes	object	Attributi della risorsa chiave-valore del punto dati (ad esempio, `host.name: eabf8c2a5ddc`). Comportamento della proprietà: - *sola lettura*
resource_schema_url	string	URL, se presente, dello schema OpenTelemetry degli attributi della risorsa. Comportamento della proprietà: - *sola lettura*
scope_name	string	Nome dell'ambito di strumentazione che ha prodotto il punto dati. Comportamento della proprietà: - *sola lettura*
scope_version	string	Versione, se presente, dell'ambito di strumentazione. Comportamento della proprietà: - *sola lettura*
scope_attributes	object	Attributi chiave-valore dell'ambito di strumentazione. Comportamento della proprietà: - *sola lettura*
scope_schema_url	string	URL, se presente, dello schema OpenTelemetry dell'ambito di strumentazione e dei relativi attributi. Comportamento della proprietà: - *sola lettura*
service_name	string	Nome del servizio che ha prodotto il punto dati (ad esempio, `zabbix-frontend`), ricavato dall'attributo della risorsa OpenTelemetry `service.name`. Comportamento della proprietà: - *sola lettura*

Proprietà	Tipo	Descrizione
metric_name	string	Nome della metrica a cui appartiene il punto dati (ad esempio, `http.server.request.duration`). Comportamento della proprietà: - *sola lettura*
metric_description	string	Descrizione, se presente, della metrica. Comportamento della proprietà: - *sola lettura*
metric_unit	string	Unità, se presente, dei valori della metrica (ad esempio, `s` o `By`; vedere UCUM). Comportamento della proprietà: - *sola lettura*
attributes	object	Attributi del punto dati chiave-valore (ad esempio, `http.request.method: POST`). Comportamento della proprietà: - *sola lettura*
start_time_unix	integer	Ora Unix (in nanosecondi) di inizio del periodo di misurazione del punto dati. Comportamento della proprietà: - *sola lettura*
time_unix	integer	Ora Unix (in nanosecondi) del punto dati. Per i punti dati *Sum*, *Histogram* ed *Exponential histogram*, indica la fine del periodo di misurazione. Comportamento della proprietà: - *sola lettura*
value	float	Valore del punto dati. Si applica ai punti dati *Gauge* e *Sum*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
flags	integer	Maschera di bit dei flag del punto dati. Se il bit meno significativo è impostato (valore 1), non è stato registrato alcun valore per il punto dati. Comportamento della proprietà: - *sola lettura*
exemplars	array	Array di oggetti exemplar, ovvero singole misurazioni conservate come esempi, con il trace e lo span durante i quali sono state registrate. Comportamento della proprietà: - *sola lettura*
aggregation_temporality	integer	Modalità di aggregazione nel tempo dei valori del punto dati. Si applica ai punti dati *Sum*, *Histogram* ed *Exponential histogram*. Valori possibili: 0 - *(predefinito)* Non specificato; 1 - Delta (aggregato sul periodo trascorso dal punto dati precedente); 2 - Cumulativo (aggregato sul periodo trascorso da `start_time_unix`). Comportamento della proprietà: - *sola lettura*

Proprietà	Tipo	Descrizione
is_monotonic	boolean	Indica se la somma può solo aumentare o rimanere invariata nel tempo. Si applica ai punti dati *Sum*. Predefinito: `false`. Comportamento della proprietà: - *sola lettura*
count	integer	Numero di misurazioni nel punto dati. Si applica ai punti dati *Histogram* ed *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
sum	float	Somma di tutte le misurazioni nel punto dati. Si applica ai punti dati *Histogram* ed *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
bucket_counts	array	Array di interi contenente il numero di misurazioni in ciascun bucket dell'istogramma. I bucket sono definiti da `explicit_bounds`. Si applica ai punti dati *Histogram*. Predefinito: `[]`. Comportamento della proprietà: - *sola lettura*
explicit_bounds	array	Array di numeri a virgola mobile contenente i limiti tra i bucket dell'istogramma. Si applica ai punti dati *Histogram*. Predefinito: `[]`. Comportamento della proprietà: - *sola lettura*
min	float	Valore minimo delle misurazioni nel punto dati. Si applica ai punti dati *Histogram* ed *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
max	float	Valore massimo delle misurazioni nel punto dati. Si applica ai punti dati *Histogram* ed *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*

Proprietà	Tipo	Descrizione
scale	integer	Scala dell'istogramma esponenziale, che determina i limiti e la risoluzione dei bucket. Una scala maggiore produce bucket più stretti e una risoluzione più fine; una scala minore produce bucket più ampi e una risoluzione più grossolana. Si applica ai punti dati *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
zero_count	integer	Numero di misurazioni nel bucket zero dell'istogramma esponenziale. Si applica ai punti dati *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
positive_bucket_counts	array	Array di interi contenente il numero di misurazioni positive in ciascun bucket dell'istogramma esponenziale. I limiti dei bucket sono determinati da `scale`; l'indice del primo bucket è impostato in `positive_offset`. Si applica ai punti dati *Exponential histogram*. Predefinito: `[]`. Comportamento della proprietà: - *sola lettura*
positive_offset	integer	Indice del primo bucket in `positive_bucket_counts`. Si applica ai punti dati *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*
negative_bucket_counts	array	Array di interi contenente il numero di misurazioni negative in ciascun bucket dell'istogramma esponenziale. I limiti dei bucket sono determinati da `scale`; l'indice del primo bucket è impostato in `negative_offset`. Si applica ai punti dati *Exponential histogram*. Predefinito: `[]`. Comportamento della proprietà: - *sola lettura*
negative_offset	integer	Indice del primo bucket in `negative_bucket_counts`. Si applica ai punti dati *Exponential histogram*. Predefinito: 0. Comportamento della proprietà: - *sola lettura*

Esemplare

L'oggetto esemplare presenta le seguenti proprietà:

Proprietà	Tipo	Descrizione
traceid	string	ID, se presente, della traccia durante la quale è stata registrata la misurazione. Colonna dell'origine dati: `Exemplars.TraceId`.
spanid	string	ID, se presente, dello span durante il quale è stata registrata la misurazione. Colonna dell'origine dati: `Exemplars.SpanId`.
value	float	Valore della misurazione. Colonna dell'origine dati: `Exemplars.Value`.
time_unix	integer	Ora Unix (in nanosecondi) in cui è stata registrata la misurazione. Colonna dell'origine dati: `Exemplars.TimeUnix`.
filtered_attributes	object	Attributi chiave-valore della misurazione che non sono inclusi negli `attributes` del punto dati. Colonna dell'origine dati: `Exemplars.FilteredAttributes`.

apmmetric.get

Descrizione

`integer/array apmmetric.get(object parameters)`

Il metodo consente di recuperare i punti dati delle metriche in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
time_from	timestamp	Restituisce solo i punti dati il cui orario è uguale o successivo all'orario specificato. Comportamento del parametro: - *obbligatorio*
time_till	timestamp	Restituisce solo i punti dati il cui orario è precedente all'orario specificato. Comportamento del parametro: - *obbligatorio*
types	integer/array	Restituisce solo i punti dati dei tipi specificati. Per un elenco dei tipi di metriche supportati, consultare la pagina dell'oggetto metrica APM.
with_flags_on	integer	Restituisce solo i punti dati che hanno impostati tutti i flag specificati. Per specificare più flag, utilizzare la somma dei relativi valori.
with_flags_off	integer	Restituisce solo i punti dati che non hanno impostato nessuno dei flag specificati. Per specificare più flag, utilizzare la somma dei relativi valori.

Parametro	Tipo	Descrizione
resource_attributes	object/array	Restituisce solo i punti dati con gli attributi della risorsa specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i punti dati. Possibili valori di operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste.
resource_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `resource_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
scope_attributes	object/array	Restituisce solo i punti dati con gli attributi dell'ambito specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i punti dati. Possibili valori di operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste.
scope_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `scope_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
attributes	object/array	Restituisce solo i punti dati con gli attributi specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i punti dati. Possibili valori di operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non uguale; 4 - Esiste; 5 - Non esiste.
attributes_evaltype	integer	Metodo di valutazione per le condizioni in `attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore o un array di valori da confrontare. Proprietà supportate: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `flags`.

Parametro	Tipo	Descrizione
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono specificate opzioni aggiuntive, viene eseguita una ricerca `LIKE "%...%"`. Proprietà supportate: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_description`, `metric_unit`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `type`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `start_time_unix`, `time_unix`.
offset	integer	Numero di record da ignorare prima di restituire il risultato.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei punti dati dell'istogramma

Recupera il punto dati più recente della metrica dell'istogramma `http.server.request.duration` del servizio `zabbix-frontend`.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
                    "filtered_attributes": {}
                }
            ]
        }
    ],
    "id": 1
}
```


Vedi anche

- Traccia APM
- Span APM
- Log APM

Origine

CApmMetric::get() in *ui/include/classes/api/services/CApmMetric.php*.

MFA

Questa classe è progettata per funzionare con i metodi MFA (autenticazione a più fattori).

Riferimenti oggetto:

- MFA

Metodi disponibili:

- mfa.create - crea nuovi metodi MFA
- mfa.delete - elimina metodi MFA
- mfa.get - recupera metodi MFA
- mfa.update - aggiorna metodi MFA

Oggetto MFA

I seguenti oggetti sono direttamente correlati all'API `mfa`.

MFA

L'oggetto MFA (autenticazione a più fattori) ha le seguenti proprietà.

Proprietà	Type	Descrizione
mfaid	ID	ID del metodo MFA. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento

Proprietà	Type	Descrizione
type	integer	Tipo del metodo MFA. Valori possibili: 1 - TOTP (password monouso basate sul tempo); 2 - Duo Universal Prompt.
name	string	Nome univoco del metodo MFA. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
hash_function	integer	Tipo della funzione hash per la generazione dei codici TOTP. Valori possibili: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "TOTP"
code_length	integer	Lunghezza del codice di verifica. Valori possibili: 6 - di 6 cifre; 8 - di 8 cifre. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "TOTP"
api_hostname	string	Nome host API fornito dal servizio di autenticazione Duo. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Duo Universal Prompt"
clientid	string	ID client fornito dal servizio di autenticazione Duo. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Duo Universal Prompt"
client_secret	string	Segreto client fornito dal servizio di autenticazione Duo. Comportamento della proprietà: - *sola scrittura* - *obbligatoria* se `type` è impostato su "Duo Universal Prompt"

mfa.create

Descrizione

`object mfa.create(object/array MFA methods)`

Questo metodo consente di creare nuovi metodi MFA.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Metodi MFA da creare.

Il metodo accetta metodi MFA con le proprietà standard del metodo MFA.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei metodi MFA creati nella proprietà `mfaids`. L'ordine degli ID restituiti corrisponde all'ordine degli elementi passati.

Esempi

Creazione di metodi MFA

Creare un metodo MFA "Zabbix TOTP" che utilizzi password monouso temporizzate (TOTP), con la funzione hash per la generazione di codici TOTP impostata su SHA-1 e la lunghezza del codice di verifica impostata su 6 cifre.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "method": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Sorgente

CMfa::create() in *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Descrizione

`object mfa.delete(array mfaids)`

Questo metodo consente di eliminare i metodi MFA.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei metodi MFA da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei metodi MFA eliminati nella proprietà `mfaids`.

Esempi

Eliminare metodi MFA

Elimina un metodo MFA.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "mfa.delete",
"params": [
"2"
],
```

```
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"mfaids": [
"2"
]
},
"id": 1
}
```

Sorgente

CMfa::delete() in *ui/include/classes/api/services/CMfa.php*.

mfa.get

Descrizione

`integer/array mfa.get(object parameters)`

Il metodo consente di recuperare i metodi MFA in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
mfaids	ID/array	Restituisce solo i metodi MFA con gli ID specificati.
selectUsrgrps	query	Restituisce una proprietà `usrgrps` con i gruppi utenti associati ai metodi MFA. Supporta `count`.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Supporta le proprietà: `mfaid` - ID del metodo MFA; `type` - tipo del metodo MFA.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `name`.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Proprietà supportate: `name`.
countOutput excludeSearch	boolean boolean	Questi parametri sono descritti nel commento di riferimento.

Parametro	Tipo	Descrizione
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(intero/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Ricercare metodi MFA per nome

Recupera tutti i metodi MFA con ”Zabbix” nel nome.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "mfa.get",
"params": {
"output": "extend",
"search": {
"name": "Zabbix"
}
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"risultato": [
{
"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
}
```

Sorgente

CMfa::get() in *ui/include/classes/api/services/CMfa.php*.

mfa.update

Descrizione

`object mfa.update(object/array MFA methods)`

Questo metodo consente di aggiornare i metodi MFA esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

Proprietà del metodo MFA (`object/array`) da aggiornare.

La proprietà `mfaid` deve essere definita per ogni elemento, tutte le altre proprietà sono facoltative. Solo le proprietà passate verranno aggiornate, tutte le altre rimarranno invariate.

Il metodo accetta metodi MFA con le proprietà standard del metodo MFA.

Valori restituiti

(`object`) Restituisce un oggetto contenente gli ID dei metodi MFA aggiornati nella proprietà `mfaids`.

Esempi

Aggiornare le proprietà del metodo

Aggiornare la funzione hash per la generazione di codici TOTP e la lunghezza del codice di verifica per il metodo MFA "Zabbix TOTP" che utilizza password monouso basate sul tempo (TOTP).

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "mfa.update",
"params": {
"mfaid": "1",
"hash_function": 3,
"code_length": 8
},
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"mfaids": [
"1"
]
},
"id": 1
}
```

Sorgente

CMfa::update() in *ui/include/classes/api/services/CMfa.php*.

Moduli

Questa classe è progettata per funzionare con i moduli del frontend.

Riferimenti agli oggetti:

- Module

Metodi disponibili:

- module.create - installa nuovi moduli
- module.delete - disinstalla i moduli
- module.get - recupera i moduli
- module.update - aggiorna i moduli

Oggetto modulo

I seguenti oggetti sono direttamente correlati all'API `module`.

Module

L'oggetto module ha le seguenti proprietà.

Property	Type	Description
moduleid	ID	ID del module come memorizzato nel database. Property behavior: - *sola lettura* - *obbligatorio* per le operazioni di aggiornamento
id	string	ID univoco del module, definito da uno sviluppatore nel file `manifest.json` del module. Valori possibili per i module integrati: vedere la descrizione della proprietà "type" in Dashboard widget. Property behavior: - *obbligatorio* per le operazioni di creazione
relative_path	string	Percorso della directory del module relativo alla directory del frontend di Zabbix. Valori possibili: `widgets/*` - per i module widget integrati; `modules/*` - per i module di terze parti. Property behavior: - *obbligatorio* per le operazioni di creazione
status	integer	Indica se il module è abilitato o disabilitato. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato.
config	object	Configurazione del module.

module.create

Descrizione

`object module.create(object/array modules)`

Questo metodo consente di installare nuovi moduli frontend.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` è impostato su `false` nel file di configurazione del frontend, all'utente non sarà consentito chiamare questo metodo.

Attention:
I file dei moduli devono essere decompressi manualmente nelle sottodirectory corrette, in base alla proprietà `relative_path` dei moduli.

Parametri

(object/array) Moduli da creare.

Il metodo accetta moduli con le proprietà standard del modulo.

Valori restituiti

(object) Restituisce un oggetto contenente gli ID dei moduli installati nella proprietà moduleids. L'ordine degli ID restituiti corrisponde all'ordine dei moduli passati.

Esempi

Installazione di un modulo

Installare un modulo con lo stato "Abilitato".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Guarda anche

- Modulo
- Moduli frontend

Fonte

CModule::create() in *ui/include/classes/api/services/CModule.php*.

module.delete

Descrizione

`object module.delete(array moduleids)`

Questo metodo consente di disinstallare i moduli.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se $ZBX_FEATURE_FLAGS['modules_config_enabled'] è impostato su false nel file di configurazione del frontend, all'utente non sarà consentito chiamare questo metodo.

Attention:
I file dei moduli devono essere rimossi manualmente.

Parametri

"(array)" ID dei moduli da disinstallare.

Restituisce i valori

”(object)” Restituisce un oggetto contenente gli ID dei moduli disinstallati nella proprietà ”moduleids”.

Esempi

Disinstallazione di più moduli

Disinstallare i moduli ”27” e ”28”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

Fonte

CModule::delete() in *ui/include/classes/api/services/CModule.php*.

module.get

Descrizione

`integer/array module.get(object parameters)`

Il metodo consente di recuperare i moduli in base ai parametri specificati.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` è impostato su `false` nel file di configurazione del frontend, all'utente non sarà consentito chiamare questo metodo.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
moduleids	ID/array	Restituisce solo i moduli con gli ID specificati.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `moduleid`, `relative_path`.
countOutput	boolean	Questi parametri sono descritti nel Commento di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	

Parametro	Tipo	Descrizione
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero di un modulo tramite ID

Recupera tutti i dati dei moduli "1", "2" e "25".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Guarda anche

- Modulo
- Widget dashboard
- Moduli frontend

Fonte

CModule::get() in *ui/include/classes/api/services/CModule.php*.

module.update

Descrizione

`object module.update(object/array modules)`

Questo metodo consente di aggiornare i moduli esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` è impostato su `false` nel file di configurazione del frontend, all'utente non sarà consentito chiamare questo metodo.

Parametri

`(object/array)` Proprietà del modulo da aggiornare.

La proprietà `moduleid` deve essere definita per ciascun modulo, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate.

Il metodo accetta moduli con le proprietà standard del modulo.

Valori restituiti

"(object)" Restituisce un oggetto contenente gli ID dei moduli aggiornati nella proprietà "moduleids".

Esempi

Disabilitazione di un modulo

Disabilita il modulo "25".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Guarda anche

- Modulo
- Moduli frontend

Fonte

CModule::update() in *ui/include/classes/api/services/CModule.php*.

Nodo ad alta disponibilità

Questa classe è progettata per funzionare con i nodi server che fanno parte di un cluster ad alta disponibilità o di un'istanza server standalone.

Riferimenti agli oggetti:

- Nodo ad alta disponibilità

Metodi disponibili:

- hanode.get - recupera i nodi

Oggetto nodo ad alta disponibilità

Il seguente oggetto è correlato al funzionamento di un cluster ad alta disponibilità di server Zabbix.

Nodo ad alta disponibilità

> **Note:**
> I nodi vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

L'oggetto Nodo ad alta disponibilità ha le seguenti proprietà.

Property	Type	Description
ha_nodeid	ID	ID del nodo.
name	string	Nome assegnato al nodo, usando la voce di configurazione HANodeName di `zabbix_server.conf`. Vuoto per un server in esecuzione in modalità standalone.
address	string	Indirizzo IP o nome DNS da cui il nodo si connette.
port	integer	Porta su cui il nodo è in esecuzione.
lastaccess	integer	Tempo dell'heartbeat, cioè il momento dell'ultimo aggiornamento dal nodo. Timestamp UTC.
status	integer	Stato del nodo. Valori possibili: 0 - standby; 1 - arrestato manualmente; 2 - non disponibile; 3 - attivo.

hanode.get

Descrizione

`integer/array hanode.get(object parameters)`

Il metodo consente di recuperare un elenco di nodi del cluster ad alta disponibilità in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Super admin*. Per ulteriori informazioni, vedere User roles.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
ha_nodeids	ID/array	Restituisce solo i nodi con gli ID nodo specificati.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Supporta le proprietà: `name`, `address`, `status`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `name`, `lastaccess`, `status`.
countOutput	flag	Questi parametri sono descritti nel commento di riferimento.
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Ottenere un elenco di nodi ordinati per stato

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Risposta:

```
{
  "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
}
```

Ottenere un elenco di nodi specifici in base ai loro ID

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Ottenere un elenco dei nodi arrestati

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
```

```
            "status": 1
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Ottenere un conteggio dei nodi in standby

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

Verificare lo stato dei nodi a indirizzi IP specifici

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

Fonte

CHaNode::get() in *ui/include/classes/api/services/CHaNode.php*.

Problema

Questa classe è progettata per funzionare con i problemi.

Riferimenti agli oggetti:

- Problem
  - URL del tipo di supporto
- Tag del problema

Metodi disponibili:

- problem.get - recupera i problemi

Oggetto problema

I seguenti oggetti sono direttamente correlati all'API `problem`.

Problema

Note:
I problemi vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

L'oggetto problema ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
eventid	ID	ID dell'evento problema.
source	integer	Tipo dell'evento problema. Valori possibili: 0 - evento creato da un trigger; 3 - evento interno; 4 - evento creato durante l'aggiornamento dello stato del servizio.

Proprietà	Tipo	Descrizione
object	integer	Tipo di oggetto correlato all'evento problema. Valori possibili se `source` è impostato su "evento creato da un trigger": 0 - trigger. Valori possibili se `source` è impostato su "evento interno": 0 - trigger; 4 - item; 5 - regola LLD. Valori possibili se `source` è impostato su "evento creato durante l'aggiornamento dello stato del servizio": 6 - servizio.
objectid	ID	ID dell'oggetto correlato.
clock	timestamp	Ora in cui è stato creato l'evento problema.
ns	integer	Nanosecondi al momento della creazione dell'evento problema.
r_eventid	ID	ID dell'evento di ripristino.
r_clock	timestamp	Ora in cui è stato creato l'evento di ripristino.
r_ns	integer	Nanosecondi al momento della creazione dell'evento di ripristino.
cause_eventid	ID	ID dell'evento causa.
correlationid	ID	ID della regola di correlazione se questo evento è stato ripristinato da una regola di correlazione globale.
userid	ID	ID dell'utente che ha chiuso il problema (se il problema è stato chiuso manualmente).
name	string	Nome del problema risolto.
acknowledged	integer	Stato di conferma del problema. Valori possibili: 0 - non confermato; 1 - confermato.
severity	integer	Gravità corrente del problema. Valori possibili: 0 - non classificato; 1 - informazione; 2 - avviso; 3 - media; 4 - alta; 5 - disastro.
suppressed	integer	Indica se il problema è soppresso. Valori possibili: 0 - il problema è nello stato normale; 1 - il problema è soppresso.
opdata	string	Dati operativi con macro espanse.
urls	array	URL del tipo di supporto attivi.

URL del tipo di supporto

L'oggetto URL del tipo di supporto ha le seguenti proprietà.

Proprietà	Type	Descrizione
name	string	Nome dell'URL definito del tipo di supporto.
url	string	Valore dell'URL definito del tipo di supporto.

I risultati conterranno voci solo per i tipi di supporto attivi con voce del menu eventi abilitata. Le macro utilizzate nelle proprietà verranno espanse, ma se una delle proprietà contiene una macro non espansa, entrambe le proprietà verranno escluse dai risultati. Per le macro supportate, vedere *Macro supportate*.

Tag del problema

L'oggetto tag del problema ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag del problema.
value	string	Valore del tag del problema.

problem.get

Descrizione

`integer/array problem.get(object parameters)`

Il metodo consente di recuperare i problemi in base ai parametri specificati.

Questo metodo serve a recuperare i problemi non risolti. Se specificato, è anche possibile recuperare ulteriormente i problemi risolti di recente. Il periodo che determina quanto sia "recente" è definito in *Amministrazione* > *Generale*. I problemi risolti prima di tale periodo non vengono conservati nella tabella dei problemi. Per recuperare i problemi risolti più indietro nel passato, usare il metodo `event.get`.

Attention:
Questo metodo può restituire problemi di un'entità eliminata se tali problemi non sono ancora stati rimossi dall'housekeeper.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parameter	Type	Description
eventids	ID/array	Restituisce solo i problemi con gli ID specificati.
groupids	ID/array	Restituisce solo i problemi creati da oggetti che appartengono ai gruppi host specificati.
hostids	ID/array	Restituisce solo i problemi creati da oggetti che appartengono agli host specificati.
objectids	ID/array	Restituisce solo i problemi creati dagli oggetti specificati.
source	integer	Restituisce solo i problemi con il tipo specificato. Per l'elenco dei tipi di evento supportati, fare riferimento alla pagina dell'oggetto evento problema. Predefinito: 0 - problema creato da un trigger.
object	integer	Restituisce solo i problemi creati da oggetti del tipo specificato. Per l'elenco dei tipi di oggetto supportati, fare riferimento alla pagina dell'oggetto evento problema. Predefinito: 0 - trigger.
acknowledged	boolean	Se impostato su `true`, restituisce solo i problemi riconosciuti.
action	integer	Restituisce solo i problemi per i quali sono state eseguite le azioni di aggiornamento evento specificate. Per più azioni, usare la somma di eventuali valori bitmap accettabili come bitmask (ad esempio, 34 per acknowledge e soppressione dell'evento).
action_userids	ID/array	Restituisce solo i problemi con gli ID specificati degli utenti che hanno eseguito le azioni di aggiornamento dell'evento problema.
suppressed	boolean	Se impostato su `true`, restituisce solo i problemi soppressi.

Parameter	Type	Description
symptom	boolean	Se impostato su `true`, restituisce solo gli eventi problema sintomo.
severities	integer/array	Restituisce solo i problemi con le severità evento specificate. Si applica solo se object è trigger.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
tags	array	Restituisce solo i problemi con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i problemi. Valori possibili di operator: 0 - *(predefinito)* Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
recent	boolean	Se impostato su `true`, restituisce i problemi recenti, inclusi quelli risolti entro il `ok_period` e tutti i problemi non risolti. Se impostato su `false`, restituisce solo i problemi non risolti.
eventid_from	string	Restituisce solo i problemi con ID maggiori o uguali all'ID specificato.
eventid_till	string	Restituisce solo i problemi con ID minori o uguali all'ID specificato.
time_from	timestamp	Restituisce solo i problemi creati dopo o al momento specificato.
time_till	timestamp	Restituisce solo i problemi creati prima o al momento specificato.
selectAcknowledges	query	Restituisce una proprietà `acknowledges` con gli aggiornamenti del problema. Gli aggiornamenti del problema sono ordinati in ordine cronologico inverso. L'oggetto di aggiornamento del problema ha le seguenti proprietà: `acknowledgeid` - `(ID)` ID dell'aggiornamento; `userid` - `(ID)` ID dell'utente che ha aggiornato l'evento; `eventid` - `(ID)` ID dell'evento aggiornato; `clock` - `(timestamp)` ora in cui l'evento è stato aggiornato; `message` - `(string)` testo del messaggio; `action` - `(integer)` tipo di azione di aggiornamento (vedere `event.acknowledge`); `old_severity` - `(integer)` severità dell'evento prima di questa azione di aggiornamento; `new_severity` - `(integer)` severità dell'evento dopo questa azione di aggiornamento; `suppress_until` - `(timestamp)` momento fino al quale l'evento sarà soppresso; `taskid` - `(ID)` ID del task se l'evento corrente è in fase di cambio di stato; `maintenanceid` - `(ID)` ID della maintenance. Supporta `count`.
selectTags	query	Restituisce una proprietà `tags` con i tag del problema. Formato di output: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Restituisce una proprietà `suppression_data` con l'elenco delle maintenance attive e delle soppressioni manuali: `maintenanceid` - `(ID)` ID della maintenance; `userid` - `(ID)` ID dell'utente che ha soppresso il problema; `suppress_until` - `(integer)` tempo fino al quale il problema è soppresso.

Parameter	Type	Description
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore o un array di valori da confrontare. Non supporta le proprietà di tipo data type `text`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `eventid`.
countOutput	boolean	Questi parametri sono descritti nella reference commentary.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero degli eventi di problema del trigger

Recupera gli eventi recenti dal trigger ”15112.”

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
```

```
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "1",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0",
                    "maintenanceid": "0"
                }
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


Recupero dei problemi riconosciuti da un utente specificato

Recupero dei problemi riconosciuti dall'utente con ID=10

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Risposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```


Vedi anche

- Avviso
- Item
- Host
- Regola LLD
- Trigger

Fonte

CEvent::get() in *ui/include/classes/api/services/CProblem.php*.

Prototipo di grafico

Questa classe è progettata per funzionare con i prototipi di grafico.

Riferimenti agli oggetti:

- Prototipo di grafico

Metodi disponibili:

- graphprototype.create - crea nuovi prototipi di grafico
- graphprototype.delete - elimina prototipi di grafico
- graphprototype.get - recupera prototipi di grafico
- graphprototype.update - aggiorna prototipi di grafico

Oggetto prototipo di grafico

I seguenti oggetti sono direttamente correlati all'API `graphprototype`.

Prototipo di grafico

L’oggetto prototipo di grafico ha le seguenti proprietà.

Proprietà	Type	Descrizione
graphid	ID	ID del prototipo di grafico. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
height	integer	Altezza del prototipo di grafico in pixel. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
name	string	Nome del prototipo di grafico. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
width	integer	Larghezza del prototipo di grafico in pixel. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
flags	integer	Origine del prototipo di grafico. Valori possibili: 2 - un prototipo di grafico; 6 - un prototipo di grafico individuato automaticamente Comportamento della proprietà: - *sola lettura*
graphtype	integer	Tipo di layout del prototipo di grafico. Valori possibili: 0 - *(predefinito)* normale; 1 - in pila; 2 - a torta; 3 - esploso.
percent_left	float	Percentile sinistro. Predefinito: 0.
percent_right	float	Percentile destro. Predefinito: 0.
show_3d	integer	Indica se mostrare in 3D i grafici a torta ed esplosi individuati automaticamente. Valori possibili: 0 - *(predefinito)* mostra in 2D; 1 - mostra in 3D.
show_legend	integer	Indica se mostrare la legenda nel grafico individuato automaticamente. Valori possibili: 0 - nascondi; 1 - *(predefinito)* mostra.
show_work_period	integer	Indica se mostrare l’orario di lavoro nel grafico individuato automaticamente. Valori possibili: 0 - nascondi; 1 - *(predefinito)* mostra.

Proprietà	Type	Descrizione
templateid	ID	ID del prototipo di grafico del template padre. Comportamento della proprietà: - *sola lettura*
yaxismax	float	Valore massimo fisso per l'asse Y.
yaxismin	float	Valore minimo fisso per l'asse Y.
ymax_itemid	ID	ID dell'item utilizzato come valore massimo per l'asse Y. Se un utente non ha accesso all'item specificato, il grafico viene visualizzato come se `ymax_type` fosse impostato su "calculated".
ymax_type	integer	Metodo di calcolo del valore massimo per l'asse Y. Valori possibili: 0 - *(predefinito)* calculated; 1 - fisso; 2 - item.
ymin_itemid	ID	ID dell'item utilizzato come valore minimo per l'asse Y. Se un utente non ha accesso all'item specificato, il grafico viene visualizzato come se `ymin_type` fosse impostato su "calculated".
ymin_type	integer	Metodo di calcolo del valore minimo per l'asse Y. Valori possibili: 0 - *(predefinito)* calculated; 1 - fisso; 2 - item.
discover	integer	Stato di individuazione del prototipo di grafico. Valori possibili: 0 - *(predefinito)* i nuovi grafici verranno individuati automaticamente; 1 - i nuovi grafici non verranno individuati automaticamente e i grafici esistenti saranno contrassegnati come persi.
uuid	string	Identificatore univoco universale, utilizzato per collegare i prototipi di grafico importati a quelli già esistenti. Generato automaticamente, se non fornito. Comportamento della proprietà: - *supportata* se il prototipo di grafico appartiene a un template

graphprototype.create

Descrizione

`object graphprototype.create(object/array graphPrototypes)`

Questo metodo consente di creare nuovi prototipi di grafico.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Prototipi di grafico da creare.

Oltre alle proprietà standard dei prototipi di grafico, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
gitems	array	Elementi del grafico da creare per i prototipi di grafico. Gli elementi del grafico possono fare riferimento sia a item che a prototipi di item, ma deve essere presente almeno un prototipo di item. Comportamento del parametro: - *obbligatorio*

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi grafici creati sotto la proprietà `graphids`. L'ordine degli ID restituiti corrisponde all'ordine dei prototipi grafici passati.

Esempi

Creazione di un prototipo di grafico

Creare un prototipo di grafico con due item.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Disk space usage {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Vedi anche

- Item del grafico

Fonte

CGraphPrototype::create() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Descrizione

`object graphprototype.delete(array graphPrototypeIds)`

Questo metodo consente di eliminare i prototipi di grafico.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei prototipi grafici da eliminare.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID dei prototipi grafici cancellati sotto la proprietà `graphids`.

Esempi

Eliminazione di più prototipi di grafico

Elimina due prototipi di grafico.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Fonte

CGraphPrototype::delete() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

Descrizione

`integer/array graphprototype.get(object parameters)`

Il metodo consente di recuperare i prototipi di grafico in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
discoveryids	ID/array	Restituisce solo i prototipi di grafico che appartengono alle regole di discovery specificate.
graphids	ID/array	Restituisce solo i prototipi di grafico con gli ID specificati.
groupids	ID/array	Restituisce solo i prototipi di grafico che appartengono a host o template nei gruppi di host o gruppi di template specificati.
hostids	ID/array	Restituisce solo i prototipi di grafico che appartengono agli host specificati.
inherited	boolean	Se impostato su `true`, restituisce solo i prototipi di grafico ereditati da un template.
itemids	ID/array	Restituisce solo i prototipi di grafico che contengono i prototipi di item specificati.
templated	boolean	Se impostato su `true`, restituisce solo i prototipi di grafico che appartengono ai template.
templateids	ID/array	Restituisce solo i prototipi di grafico che appartengono ai template specificati.
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery del prototipo di grafico. L'oggetto di discovery del prototipo di grafico collega un prototipo di grafico rilevato a un prototipo di grafico da cui è stato rilevato. Ha le seguenti proprietà: `parent_graphid` - `(ID)` ID del prototipo di grafico da cui è stato creato il grafico; `status` - `(int)` stato della discovery del grafico: 0 - *(predefinito)* il prototipo di grafico è rilevato, 1 - il prototipo di grafico non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un prototipo di grafico non più rilevato verrà eliminato.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola LLD a cui appartiene il prototipo di grafico.
selectDiscoveryRulePrototype	query	Restituisce una proprietà `discoveryRulePrototype` con il prototipo della regola LLD padre a cui appartiene il prototipo di grafico.
selectGraphItems	query	Restituisce una proprietà `gitems` con gli elementi del grafico utilizzati nel prototipo di grafico.
selectHostGroups	query	Restituisce una proprietà `hostgroups` con i gruppi di host a cui appartiene il prototipo di grafico.
selectHosts	query	Restituisce una proprietà `hosts` con gli host a cui appartiene il prototipo di grafico.
selectItems	query	Restituisce una proprietà `items` con gli item e i prototipi di item utilizzati nel prototipo di grafico.
selectTemplateGroups	query	Restituisce una proprietà `templategroups` con i gruppi di template a cui appartiene il prototipo di grafico.
selectTemplates	query	Restituisce una proprietà `templates` con i template a cui appartiene il prototipo di grafico.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text` data type. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene il prototipo di grafico; `hostid` - ID dell'host a cui appartiene il prototipo di grafico.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	

Parametro	Type	Descrizione
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei prototipi di grafico da una regola LLD

Recupera tutti i prototipi di grafico da una regola LLD.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "2"
            "discover": "0"
        }
    ],
    "id": 1
}
```

Vedi anche

- Regola di individuazione
- Item del grafico
- Item
- Host
- Gruppo di host
- Template
- Gruppo di template

Fonte

CGraphPrototype::get() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

Descrizione

`object graphprototype.update(object/array graphPrototypes)`

Questo metodo consente di aggiornare i graph prototype esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dei prototipi di grafico da aggiornare.

La proprietà `graphid` deve essere definita per ciascun prototipo di grafico, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del prototipo di grafico, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
gitems	array	Elementi del grafico per sostituire gli elementi del grafico esistenti. Se un elemento del grafico ha la proprietà `gitemid` definita, verrà aggiornato; in caso contrario, verrà creato un nuovo elemento del grafico.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID dei prototipi grafici aggiornati sotto la proprietà `graphids`.

Esempi

Modifica della dimensione di un prototipo di grafico

Modificare la dimensione di un prototipo di grafico a 1100 per 400 pixel.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Fonte

CGraphPrototype::update() in *ui/include/classes/api/services/CGraphPrototype.php*.

Prototipo di host

Questa classe è progettata per funzionare con i prototipi di host.

Riferimenti agli oggetti:

- Prototipo di host
- Collegamento al gruppo
- Prototipo di gruppo
- Tag del prototipo di host
- Interfaccia personalizzata
  - Dettagli dell'interfaccia personalizzata

Metodi disponibili:

- hostprototype.create - crea nuovi prototipi di host
- hostprototype.delete - elimina prototipi di host
- hostprototype.get - recupera prototipi di host
- hostprototype.update - aggiorna prototipi di host

Oggetto prototipo di host

I seguenti oggetti sono direttamente correlati all'API `hostprototype`.

Prototipo di host

L'oggetto prototipo di host ha le seguenti proprietà.

Property	Type	Description
hostid	ID	ID del prototipo di host. Property behavior: - *sola lettura* - *obbligatorio* per le operazioni di aggiornamento
host	string	Nome tecnico del prototipo di host. Property behavior: - *obbligatorio* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
name	string	Nome visibile del prototipo di host. Predefinito: valore della proprietà `host`. Property behavior: - *sola lettura* per gli oggetti ereditati
status	integer	Stato del prototipo di host. Valori possibili: 0 - *(predefinito)* host monitorato; 1 - host non monitorato.

Property	Type	Description
flags	integer	Origine del prototipo di host. Valori possibili: 2 - un prototipo di host; 6 - un prototipo di host individuato Property behavior: - *sola lettura*
inventory_mode	integer	Modalità di popolamento dell'inventario host. Valori possibili: -1 - *(predefinito)* disabilitato; 0 - manuale; 1 - automatico.
templateid	ID	ID del prototipo di host del template padre. Property behavior: - *sola lettura*
discover	integer	Stato di individuazione del prototipo di host. Valori possibili: 0 - *(predefinito)* i nuovi host verranno individuati; 1 - i nuovi host non verranno individuati e gli host esistenti verranno contrassegnati come persi.
custom_interfaces	integer	Origine delle interfacce personalizzate per gli host creati dal prototipo di host. Valori possibili: 0 - *(predefinito)* eredita le interfacce dall'host padre; 1 - usa le interfacce personalizzate dei prototipi di host. Property behavior: - *sola lettura* per gli oggetti ereditati
uuid	string	Identificatore univoco universale, utilizzato per collegare i prototipi di host importati a quelli già esistenti. Generato automaticamente, se non specificato. Property behavior: - *supportato* se il prototipo di host appartiene a un template

Collegamento del gruppo

L'oggetto collegamento del gruppo collega un prototipo di host con un gruppo di host. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
groupid	ID	ID del gruppo di host. Comportamento della proprietà: - *obbligatorio*

Prototipo di gruppo

L'oggetto prototipo di gruppo definisce un gruppo che verrà creato per un host individuato. Ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
group_prototypeid	ID	ID del prototipo di gruppo. Comportamento della proprietà: - *sola lettura*

Proprietà	Tipo	Descrizione
name	string	Nome del prototipo di gruppo. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Tag del prototipo di host

L'oggetto tag del prototipo di host ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag del prototipo di host. Comportamento della proprietà: - *obbligatoria*
value	string	Valore del tag del prototipo di host. Comportamento della proprietà: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template. Comportamento della proprietà: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.

Interfaccia personalizzata

Le interfacce personalizzate sono supportate se `custom_interfaces` dell'oggetto prototipo host è impostato su "usa le interfacce personalizzate dei prototipi host". L'oggetto interfaccia personalizzata ha le seguenti proprietà.

Proprietà	Type	Descrizione
type	integer	Tipo di interfaccia. Valori possibili: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Comportamento della proprietà: - *obbligatoria*
useip	integer	Indica se la connessione deve essere effettuata tramite IP. Valori possibili: 0 - connetti usando il nome DNS dell'host; 1 - connetti usando l'indirizzo IP dell'host. Comportamento della proprietà: - *obbligatoria*

Proprietà	Type	Descrizione
ip	string	Indirizzo IP usato dall'interfaccia. Può contenere macro. Comportamento della proprietà: - *obbligatoria* se `useip` è impostato su "connetti usando l'indirizzo IP dell'host"
dns	string	Nome DNS usato dall'interfaccia. Può contenere macro. Comportamento della proprietà: - *obbligatoria* se `useip` è impostato su "connetti usando il nome DNS dell'host"
port	string	Numero di porta usato dall'interfaccia. Può contenere macro utente e LLD. Comportamento della proprietà: - *obbligatoria*
main	integer	Indica se l'interfaccia è usata come predefinita sull'host. Solo un'interfaccia di un determinato tipo può essere impostata come predefinita su un host. Valori possibili: 0 - non predefinita; 1 - predefinita. Comportamento della proprietà: - *obbligatoria*
details	object	Oggetto aggiuntivo per i dettagli dell'interfaccia personalizzata. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SNMP"

Dettagli dell'interfaccia personalizzata

L'oggetto details ha le seguenti proprietà.

Property	Type	Description
version	integer	Versione dell'interfaccia SNMP. Valori possibili: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Comportamento della proprietà: - *obbligatoria*
bulk	integer	Indica se utilizzare richieste SNMP bulk. Valori possibili: 0 - non utilizzare richieste bulk; 1 - *(predefinito)* - utilizzare richieste bulk.
community	string	Community SNMP. Comportamento della proprietà: - *obbligatoria* se `version` è impostato su "SNMPv1" o "SNMPv2c"
max_repetitions	integer	Valore massimo di ripetizione per le richieste SNMP bulk native (GetBulkRequest-PDUs). Utilizzato solo per gli item `discovery[]` e `walk[]` in SNMPv2 e v3. Predefinito: 10.

Property	Type	Description
securityname	string	Nome di sicurezza SNMPv3. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3”
securitylevel	integer	Livello di sicurezza SNMPv3. Valori possibili: 0 - *(predefinito)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3”
authpassphrase	string	Passphrase di autenticazione SNMPv3. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3” e `securitylevel` è impostato su ”authNoPriv” o ”authPriv”
privpassphrase	string	Passphrase di privacy SNMPv3. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3” e `securitylevel` è impostato su ”authPriv”
authprotocol	integer	Protocollo di autenticazione SNMPv3. Valori possibili: 0 - *(predefinito)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3” e `securitylevel` è impostato su ”authNoPriv” o ”authPriv”
privprotocol	integer	Protocollo di privacy SNMPv3. Utilizzato solo dalle interfacce SNMPv3. Valori possibili: 0 - *(predefinito)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3” e `securitylevel` è impostato su ”authPriv”
contextname	string	Nome del contesto SNMPv3. Comportamento della proprietà: - *supportata* se `version` è impostato su ”SNMPv3”

hostprototype.create

Descrizione

`object hostprototype.create(object/array hostPrototypes)`

Questo metodo consente di creare nuovi host prototype.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Prototipi di host da creare.

Oltre alle proprietà standard del prototipo di host, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groupLinks	array	Collegamenti ai gruppi da creare per il prototipo di host. Comportamento del parametro: - *obbligatorio*
ruleid	ID	ID della regola LLD a cui appartiene il prototipo di host. Comportamento del parametro: - *obbligatorio*
groupPrototypes	array	Prototipi di gruppo da creare per il prototipo di host.
macros	object/array	Macro utente da creare per il prototipo di host.
tags	object/array	Tag del prototipo di host.
interfaces	object/array	Interfacce personalizzate del prototipo di host.
templates	object/array	Template da collegare al prototipo di host. I template devono avere definita solo la proprietà `templateid`.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di host creati nella proprietà `hostids`. L'ordine degli ID restituiti corrisponde all'ordine dei prototipi di host passati.

Esempi

Creazione di un prototipo di host

Creare un prototipo di host "{#VM.NAME}" nella regola LLD "23542" con un prototipo di gruppo "{#HV.NAME}", una coppia tag "datacenter": "{#DATACENTER.NAME}" e un'interfaccia SNMPv2 personalizzata 127.0.0.1:161 con community {$SNMP_COMMUNITY}. Collegarlo al gruppo host "2".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

Vedi anche

- Collegamento del gruppo
- Prototipo di gruppo
- Tag del prototipo di host
- Interfaccia personalizzata
- Macro utente

Fonte

CHostPrototype::create() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

Descrizione

`object hostprototype.delete(array hostPrototypeIds)`

Questo metodo consente di eliminare i prototipi di host.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei prototipi host da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di host eliminati nella proprietà `hostids`.

Esempi

Eliminazione di più prototipi di host

Elimina due prototipi di host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

Fonte

CHostPrototype::delete() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

Descrizione

`integer/array hostprototype.get(object parameters)`

Il metodo consente di recuperare i prototipi di host in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parameter	Type	Description
hostids	ID/array	Restituisce solo i prototipi di host con gli ID specificati.
discoveryids	ID/array	Restituisce solo i prototipi di host che appartengono alle regole LLD specificate.
inherited	boolean	Se impostato su `true`, restituisce solo gli item ereditati da un template.

Parameter	Type	Description
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery del prototipo di host. L'oggetto di discovery del prototipo di host collega un prototipo di host rilevato a un prototipo di host da cui è stato rilevato. Ha le seguenti proprietà: `host` - `(string)` ID dell'host; `parent_hostid` - `(string)` ID del prototipo di host da cui è stato creato il prototipo di host; `status` - `(int)` stato di discovery del prototipo di host: 0 - *(predefinito)* il prototipo di host è rilevato, 1 - il prototipo di host non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un prototipo di host che non è più rilevato verrà eliminato; `ts_disable` - `(timestamp)` momento in cui un prototipo di host che non è più rilevato verrà disabilitato; `disable_source` - `(int)` indicatore che specifica se il prototipo di host è stato disabilitato da una regola LLD o manualmente: 0 - *(predefinito)* disabilitato automaticamente, 1 - disabilitato da una regola LLD.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola LLD a cui appartiene il prototipo di host.
selectDiscoveryRulePrototype	query	Restituisce una proprietà `discoveryRulePrototype` con il prototipo della regola LLD padre a cui appartiene il prototipo di host.
selectInterfaces	query	Restituisce una proprietà `interfaces` con le interfacce personalizzate del prototipo di host.
selectGroupLinks	query	Restituisce una proprietà `groupLinks` con i collegamenti ai gruppi del prototipo di host.
selectGroupPrototypes	query	Restituisce una proprietà `groupPrototypes` con i prototipi di gruppo del prototipo di host.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag presenti nei template collegati.
selectMacros	query	Restituisce una proprietà `macros` con le macro del prototipo di host.
selectParentHost	query	Restituisce una proprietà `parentHost` con l'host a cui appartiene il prototipo di host.
selectTags	query	Restituisce una proprietà `tags` con i tag del prototipo di host.
selectTemplates	query	Restituisce una proprietà `templates` con i template collegati al prototipo di host. Supporta `count`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `hostid`, `host`, `name`, `status`, `discovered`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei prototipi di host da una regola LLD

Recupera tutti i prototipi di host, i loro collegamenti ai gruppi, i prototipi di gruppo e i tag da una regola LLD.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Collegamento del gruppo
- Prototipo di gruppo
- Macro utente

Sorgente

CHostPrototype::get() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Descrizione

`object hostprototype.update(object/array hostPrototypes)`

Questo metodo consente di aggiornare host prototype esistenti.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del prototipo di host da aggiornare.

La proprietà `hostid` deve essere definita per ciascun prototipo di host, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del prototipo di host, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groupLinks	array	Collegamenti ai gruppi per sostituire i collegamenti ai gruppi correnti nel prototipo di host. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati

Parametro	Tipo	Descrizione
groupPrototypes	array	Prototipi di gruppo per sostituire i prototipi di gruppo esistenti nel prototipo di host. Tutti i prototipi di gruppo non elencati nella richiesta verranno rimossi. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
macros	object/array	Macro utente per sostituire le macro utente correnti. Tutte le macro non elencate nella richiesta verranno rimosse.
tags	object/array	Tag del prototipo di host per sostituire i tag correnti. Tutti i tag non elencati nella richiesta verranno rimossi. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
interfaces	object/array	Interfacce personalizzate del prototipo di host per sostituire le interfacce correnti. L'oggetto interfaccia personalizzata deve contenere tutti i suoi parametri. Tutte le interfacce non elencate nella richiesta verranno rimosse. Comportamento del parametro: - *supportato* se `custom_interfaces` di Host prototype object è impostato su "use host prototypes custom interfaces" - *sola lettura* per gli oggetti ereditati
templates	object/array	Template per sostituire i template attualmente collegati. I template devono avere definita solo la proprietà `templateid`.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei prototipi host aggiornati sotto la proprietà `hostids`.

Esempi

Disabilitazione di un prototipo di host

Disabilita un prototipo di host, cioè imposta il suo stato su "1".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Aggiornamento dei tag del prototipo di host

Sostituisci i tag del prototipo di host con nuovi tag.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Aggiornamento delle interfacce personalizzate del prototipo di host

Sostituisce le interfacce ereditate con le interfacce personalizzate del prototipo di host.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Vedi anche

- Collegamento del gruppo
- Prototipo di gruppo
- Tag del prototipo di host
- Interfaccia personalizzata
- Macro utente

Fonte

CHostPrototype::update() in *ui/include/classes/api/services/CHostPrototype.php*.

Prototipo di item

Questa classe è progettata per funzionare con i prototipi di item.

Riferimenti agli oggetti:

- Prototipo di item
  - Header HTTP
  - Campo query HTTP
- Tag del prototipo di item
- Preprocessing del prototipo di item

Metodi disponibili:

- itemprototype.create - crea nuovi prototipi di item
- itemprototype.delete - elimina prototipi di item
- itemprototype.get - recupera prototipi di item
- itemprototype.update - aggiorna prototipi di item

Oggetto prototipo di elemento

I seguenti oggetti sono direttamente correlati all'API `itemprototype`.

Prototipo di item

L'oggetto prototipo di item presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
itemid	ID	ID del prototipo di item. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento

Proprietà	Tipo	Descrizione
delay	string	Intervallo di aggiornamento del prototipo di item. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d) oppure, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e di pianificazione. Accetta macro utente e macro LLD. Se utilizzato, il valore deve essere una singola macro. Non sono supportate più macro o macro combinate con testo. Gli intervalli flessibili possono essere scritti come due macro separate da una barra (ad esempio, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Esempio: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Comportamento della proprietà: - *obbligatoria* se type è impostato su "Zabbix agent" (0), "Simple check" (3), "Zabbix internal" (5), "External check" (10), "Database monitor" (11), "IPMI agent" (12), "SSH agent" (13), "TELNET agent" (14), "Calculated" (15), "JMX agent" (16), "HTTP agent" (19), "SNMP agent" (20), "Script" (21), "Browser" (22), "Telemetry query" (24), oppure se type è impostato su "Zabbix agent (active)" (7) e key_ non contiene "mqtt.get"
hostid	ID	ID dell'host a cui appartiene il prototipo di item. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
interfaceid	ID	ID dell'interfaccia host del prototipo di item. Comportamento della proprietà: - *obbligatoria* se il prototipo di item appartiene a un host e type è impostato su "Zabbix agent", "IPMI agent", "JMX agent", "SNMP trap" o "SNMP agent" - *supportata* se il prototipo di item appartiene a un host e type è impostato su "Simple check", "External check", "SSH agent", "TELNET agent" o "HTTP agent"
key_	string	Chiave del prototipo di item. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
name	string	Nome del prototipo di item. Supporta macro utente. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
type	integer	Tipo del prototipo di item. Valori possibili: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
url	string	Stringa URL. Supporta macro LLD, macro utente, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
value_type	integer	Tipo di informazione del prototipo di item. Valori possibili: 0 - numerico con virgola mobile; 1 - carattere; 2 - log; 3 - numerico senza segno; 4 - testo; 5 - binario; 6 - JSON. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
allow_traps	integer	Consente di popolare il valore in modo analogo all’item trapper. 0 - *(predefinito)* Non accettare dati in ingresso; 1 - Accettare dati in ingresso. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent”

Proprietà	Tipo	Descrizione
authtype	integer	Metodo di autenticazione. Valori possibili se `type` è impostato su ”SSH agent”: 0 - *(predefinito)* password; 1 - chiave pubblica. Valori possibili se `type` è impostato su ”HTTP agent”: 0 - *(predefinito)* nessuno; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SSH agent” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (se `type` è impostato su ”HTTP agent”)
description	string	Descrizione del prototipo di item.
flags	integer	Origine del prototipo di item. Valori possibili: 2 - un prototipo di item; 6 - un prototipo di item rilevato Comportamento della proprietà: - *sola lettura*
follow_redirects	integer	Segue i reindirizzamenti della risposta durante il polling dei dati. Valori possibili: 0 - Non seguire i reindirizzamenti; 1 - *(predefinito)* Seguire i reindirizzamenti. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
headers	array	Array di intestazioni che verranno inviate durante l’esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
history	string	Unità di tempo per la quale devono essere conservati i dati della cronologia. Accetta anche macro utente e macro LLD. Predefinito: 31d.
http_proxy	string	Stringa di connessione al proxy HTTP(S). Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ipmi_sensor	string	Sensore IPMI. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”IPMI agent” e `key_` non è impostato su ”ipmi.get” - *supportata* se `type` è impostato su ”IPMI agent” - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
jmx_endpoint	string	Stringa di connessione personalizzata dell'agent JMX. Predefinito: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento della proprietà: - *supportata* se `type` è impostato su "JMX agent"
logtimefmt	string	Formato dell'ora nelle voci di log. Comportamento della proprietà: - *supportata* se `value_type` è impostato su "log" - *sola lettura* per gli oggetti ereditati
master_itemid	ID	ID dell'item master. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Dependent item" - *sola lettura* per gli oggetti ereditati
output_format	integer	Indica se la risposta deve essere convertita in JSON. Valori possibili: 0 - *(predefinito)* Memorizzare i dati non elaborati; 1 - Convertire in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
params	string	Parametri aggiuntivi in base al tipo di prototipo di item: - script eseguito per i prototipi di item SSH agent e TELNET agent; - query SQL per i prototipi di item Database monitor; - formula per i prototipi di item Calculated; - script per i prototipi di item Script e Browser. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Database monitor", "SSH agent", "TELNET agent", "Calculated", "Script" o "Browser" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "Script" o "Browser")
parameters	object/array	Parametri aggiuntivi se `type` è impostato su "Script" o "Browser". Array di oggetti con proprietà `name` e `value`, dove `name` deve essere univoco. Comportamento della proprietà: - *supportata* se `type` è impostato su "Script" o "Browser" - *sola lettura* per gli oggetti ereditati
password	string	Password per l'autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "JMX agent" e `username` è impostato - *supportata* se `type` è impostato su "Simple check", "SSH agent", "TELNET agent", "Database monitor" o "HTTP agent" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "HTTP agent")

Proprietà	Tipo	Descrizione
post_type	integer	Tipo del corpo dei dati POST memorizzato nella proprietà `posts`. Valori possibili: 0 - *(predefinito)* Dati non elaborati. 2 - Dati JSON. 3 - Dati XML. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
posts	string	Dati del corpo della richiesta HTTP(S). Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”HTTP agent” e `post_type` è impostato su ”Dati JSON” o ”Dati XML” - *supportata* se `type` è impostato su ”HTTP agent” e `post_type` è impostato su ”Dati non elaborati” - *sola lettura* per gli oggetti ereditati
privatekey	string	Nome del file della chiave privata. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent” e `authtype` è impostato su ”chiave pubblica”
publickey	string	Nome del file della chiave pubblica. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent” e `authtype` è impostato su ”chiave pubblica”
query_fields	array	Array di campi di query che verranno inviati durante l’esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
request_method	integer	Tipo di metodo della richiesta. Valori possibili: 0 - *(predefinito)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
retrieve_mode	integer	Parte della risposta da memorizzare. Valori possibili se `request_method` è impostato su ”GET”, ”POST” o ”PUT”: 0 - *(predefinito)* Corpo; 1 - Intestazioni; 2 - Verranno memorizzati sia il corpo sia le intestazioni. Valori possibili se `request_method` è impostato su ”HEAD”: 1 - Intestazioni. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
snmp_oid	string	OID SNMP. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SNMP agent” - *sola lettura* per gli oggetti ereditati
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ssl_key_file	string	Percorso del file della chiave SSL privata. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ssl_key_password	string	Password del file della chiave SSL. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
status	integer	Stato del prototipo di item. Valori possibili: 0 - *(predefinito)* prototipo di item abilitato; 1 - prototipo di item disabilitato; 3 - prototipo di item non supportato.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole. Supporta anche macro utente o macro LLD come parte dell’elenco separato da virgole. Esempio: 200,200-{$M},{$M},200-400 Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
templateid	ID	ID del prototipo di item del template padre. Comportamento della proprietà: - *sola lettura*
timeout	string	Timeout della richiesta di polling dei dati dell’item. Accetta un numero di secondi (ad esempio, 1, 60, 300) o un valore con un suffisso temporale (ad esempio, 30s, 1m). Accetta anche macro utente e macro LLD. Intervallo dei valori possibili: 1-600s. Predefinito: ”” - utilizza le impostazioni del proxy/globali. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Zabbix agent” (0), ”Simple check” (3) e `key_` non inizia con ”`vmware.`” e ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) e `snmp_oid` inizia con ”`walk[`” o ”`get[`”, ”Script” (21), ”Browser” (22) - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
trapper_hosts	string	Host consentiti. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Zabbix trapper”, oppure se `type` è impostato su ”HTTP agent” e `allow_traps` è impostato su ”Allow incoming data”. Se la proprietà viene omessa, viene impostata automaticamente sulla macro utente `{$TRAPPER.ALLOWED_HOSTS}` (valore predefinito). Per impostazione predefinita, `{$TRAPPER.ALLOWED_HOSTS}` è `127.0.0.1,::1`. Se la proprietà è impostata su una stringa vuota, le connessioni in ingresso verranno negate da tutti gli host.
trends	string	Unità di tempo per la quale devono essere conservati i dati dei trend. Accetta anche macro utente e macro LLD. Predefinito: 365d. Comportamento della proprietà: - *supportata* se `value_type` è impostato su ”numeric float” o ”numeric unsigned”
units	string	Unità dei valori. Comportamento della proprietà: - *supportata* se `value_type` è impostato su ”numeric float” o ”numeric unsigned” - *sola lettura* per gli oggetti ereditati
username	string	Nome utente per l’autenticazione. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su ”SSH agent” o ”TELNET agent”, oppure se `type` è impostato su ”JMX agent” e `password` è impostato - *supportato* se `type` è impostato su ”Simple check”, ”Database monitor” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (se `type` è impostato su ”HTTP agent”)
uuid	string	Identificatore univoco universale, utilizzato per collegare i prototipi di item importati a quelli già esistenti. Generato automaticamente se non specificato. Comportamento della proprietà: - *supportato* se il prototipo di item appartiene a un template
valuemapid	ID	ID della mappa dei valori associata. Comportamento della proprietà: - *supportata* se `value_type` è impostato su ”numeric float”, ”character” o ”numeric unsigned” - *sola lettura* per gli oggetti ereditati
verify_host	integer	Indica se verificare che il nome host della connessione corrisponda a quello presente nel certificato dell’host. Valori possibili: 0 - *(predefinito)* Non verificare; 1 - Verificare. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
verify_peer	integer	Indica se verificare che il certificato dell'host sia autentico. Valori possibili: 0 - *(predefinito)* Non verificare; 1 - Verificare. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
discover	integer	Stato di rilevamento del prototipo di item. Valori possibili: 0 - *(predefinito)* i nuovi item verranno rilevati; 1 - i nuovi item non verranno rilevati e quelli esistenti verranno contrassegnati come persi.
query	object	Configurazione della query di telemetria. Definisce cosa interroga l'item dall'origine dati di telemetria, come filtra i dati e come aggrega il risultato. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Telemetry query" - *sola lettura* per gli oggetti ereditati o rilevati
time_shift	string	Quanto indietro nel tempo, prima dell'ora corrente, termina l'intervallo temporale interrogato dall'item. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`) o un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`). Accetta anche macro utente e macro LLD. Intervallo dei valori possibili: 0-86400s. Predefinito: 15s. Comportamento della proprietà: - *supportata* se `type` è impostato su "Telemetry query"
lookback_limit	string	Quanto indietro nel tempo, prima dell'ora corrente, l'item interroga i dati quando non esiste una query precedente da cui continuare (ad esempio, alla prima esecuzione dell'item) oppure quando la query precedente è più vecchia di `delay` (ad esempio, perché Zabbix server era inattivo). Accetta un numero di secondi (ad esempio, `1`, `60`, `300`) o un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`). Accetta anche macro utente e macro LLD. Intervallo dei valori possibili: 1-259200s. Predefinito: 10m. Comportamento della proprietà: - *supportata* se `type` è impostato su "Telemetry query"

Proprietà	Tipo	Descrizione
granularity	string	Durata di una porzione in cui viene suddiviso l'intero intervallo temporale della query. Accetta un numero di secondi (ad esempio, 1, 60, 300) o un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d). Accetta anche macro utente e macro LLD. Intervallo dei valori possibili: 1-86400s. Predefinito: 15s. Comportamento della proprietà: - *supportata* se type è impostato su "Telemetry query"

Intestazione HTTP

L'oggetto header ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
name	string	Nome dell'intestazione HTTP. Comportamento della proprietà: - *obbligatorio*
value	string	Valore dell'intestazione. Comportamento della proprietà: - *obbligatorio*

Campo di query HTTP

L'oggetto campo di query definisce un nome e un valore utilizzati per specificare un parametro URL. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
name	string	Nome del parametro. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro. Comportamento della proprietà: - *obbligatorio*

Query di telemetria

L'oggetto query di telemetria ha le seguenti proprietà.

Property	Type	Description
signal_type	integer	Tipo di segnale interrogato dall'item. Valori possibili: 0 - Tracce; 1 - Metriche; 2 - Log. Comportamento della proprietà: - *obbligatoria*
metric_point_type	integer	Tipo di punto metrica interrogato dall'item. Valori possibili: 0 - Somma; 1 - Gauge; 2 - Istogramma; 3 - Istogramma esponenziale. Comportamento della proprietà: - *obbligatoria* se `signal_type` è impostato su "Metriche"
columns	array	Array di colonne di dati di telemetria interrogate dall'item. Può essere vuoto. Comportamento della proprietà: - *obbligatoria*
aggregated_columns	array	Array di colonne aggregate di dati di telemetria interrogate dall'item e modalità di calcolo di un valore dai record delle colonne. Comportamento della proprietà: - *obbligatoria*
filter	object	Filtro della query di telemetria per i record delle colonne nella query. Comportamento della proprietà: - *obbligatoria*

Colonne dei dati di telemetria

L'oggetto delle colonne dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
column	string	Nome della colonna dei dati di telemetria interrogata dall'item. Vedere colonne dei dati di telemetria supportate. Per più colonne a valore singolo, ogni valore `column` deve essere univoco. Per più colonne che contengono coppie chiave-valore, ogni coppia `column` e `attribute_key` deve essere univoca.
attribute_key	string	Nome della chiave interrogata dall'item all'interno della colonna che contiene coppie chiave-valore. Per più colonne che contengono coppie chiave-valore, ogni coppia `column` e `attribute_key` deve essere univoca. Comportamento della proprietà: - *required* se `column` è impostato su una colonna che contiene coppie chiave-valore

Colonne aggregate dei dati di telemetria

L'oggetto delle colonne aggregate dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
column	string	Nome della colonna dei dati di telemetria interrogata dall'item. Vedere colonne dei dati di telemetria supportate. Per più colonne, ogni valore di `column` deve essere univoco. Questa proprietà non è supportata se `function` è impostato su "Count". Comportamento della proprietà: - *obbligatoria* se `function` è impostato su "Min", "Max", "Average", "Sum" o "Percentile"
function	integer	Funzione applicata dall'item ai record della colonna. Valori possibili: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Comportamento della proprietà: - *obbligatoria*
parameters	array	Percentile calcolato dall'item a partire dai record della colonna. Valore possibile: un numero da 0 a 100, specificato come stringa. Comportamento della proprietà: - *obbligatoria* se `function` è impostato su "Percentile"
alias	string	Nome univoco del valore calcolato, utilizzato per identificarlo nel valore dell'item. Ogni valore di `alias` deve essere univoco. Comportamento della proprietà: - *obbligatoria*

Filtro dei dati di telemetria

L'oggetto filtro dei dati di telemetria ha le seguenti proprietà.

Property	Type	Description
conditions	array	Condizioni del filtro dei dati di telemetria che un record in una colonna di dati di telemetria deve soddisfare. Comportamento della proprietà: - *required*
evaltype	integer	Metodo di valutazione delle condizioni del filtro. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Comportamento della proprietà: - *required*

Property	Type	Description
eval_formula	string	Espressione generata utilizzata per valutare le condizioni del filtro. L'espressione contiene ID che fanno riferimento a condizioni specifiche del filtro tramite il relativo `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Comportamento della proprietà: - *read-only*
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a condizioni specifiche del filtro tramite il relativo `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni del filtro: nessuna condizione può rimanere inutilizzata o essere omessa. Comportamento della proprietà: - *required* se `evaltype` è impostato su "Espressione personalizzata"

Condizioni del filtro dei dati di telemetria

L'oggetto delle condizioni del filtro dei dati di telemetria presenta le seguenti proprietà.

Property	Type	Description
column	string	Nome della colonna dei dati di telemetria a cui si applica la condizione del filtro. Vedere le colonne dei dati di telemetria supportate. Comportamento della proprietà: - *obbligatoria*
attribute_key	string	Nome della chiave a cui si applica la condizione, all'interno della colonna che contiene coppie chiave-valore. Comportamento della proprietà: - *obbligatoria* se `column` è impostata su una colonna che contiene coppie chiave-valore
operator	integer	Operatore della condizione del filtro. Valori possibili per le colonne a valore singolo: 0 - Uguale; 1 - Diverso da; 2 - Contiene; 3 - Non contiene. Valori possibili per le colonne che contengono coppie chiave-valore: 0 - Uguale; 1 - Diverso da; 12 - Esiste. Comportamento della proprietà: - *obbligatoria*
value	string	Valore con cui effettuare il confronto. Comportamento della proprietà: - *obbligatoria* se `operator` è impostato su "Uguale", "Diverso da", "Contiene" o "Non contiene"

Property	Type	Description
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente quando modifica le condizioni del filtro, ma verrà generato nuovamente quando queste verranno richieste in seguito. Comportamento della proprietà: - *obbligatoria* se `evaltype` dell'oggetto Filtro della query di telemetria è impostato su "Espressione personalizzata"

Tag del prototipo di item

L'oggetto tag del prototipo di item ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tag	string	Nome del tag del prototipo di item. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del tag del prototipo di item. Comportamento della proprietà: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template; 1 - host. Comportamento della proprietà: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.

Pre-elaborazione del prototipo di item

L'oggetto di pre-elaborazione del prototipo di item presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
type	integer	Il tipo di opzione di pre-elaborazione. Valori possibili: 1 - Moltiplicatore personalizzato; 2 - Rimozione a destra; 3 - Rimozione a sinistra; 4 - Rimozione; 5 - Espressione regolare; 6 - Booleano in decimale; 7 - Ottale in decimale; 8 - Esadecimale in decimale; 9 - Modifica semplice; 10 - Modifica al secondo; 11 - XPath XML; 12 - JSONPath; 13 - Nell'intervallo; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Verifica la presenza di errori in JSON; 17 - Verifica la presenza di errori in XML; 18 - Verifica la presenza di errori usando un'espressione regolare; 19 - Ignora se invariato; 20 - Ignora se invariato con heartbeat; 21 - JavaScript; 22 - Pattern Prometheus; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Sostituzione; 26 - Verifica non supportato; 27 - XML in JSON; 28 - Valore SNMP walk; 29 - SNMP walk in JSON; 30 - Valore SNMP get. Comportamento della proprietà: - *required*
params	string	Parametri aggiuntivi utilizzati dall'opzione di pre-elaborazione. I parametri multipli sono separati dal carattere di nuova riga (\n). Se `type` è impostato su "Verifica non supportato", i parametri seguono la sintassi `<scope>[\n<pattern>]`, dove *pattern* è un'espressione regolare e *scope* è uno dei seguenti: -1 - corrisponde a qualsiasi errore; 0 - verifica se il messaggio di errore corrisponde a *pattern*; 1 - verifica se il messaggio di errore non corrisponde a *pattern*. Comportamento della proprietà: - *required* se `type` è impostato su "Moltiplicatore personalizzato" (1), "Rimozione a destra" (2), "Rimozione a sinistra" (3), "Rimozione" (4), "Espressione regolare" (5), "XPath XML" (11), "JSONPath" (12), "Nell'intervallo" (13), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Verifica la presenza di errori in JSON" (16), "Verifica la presenza di errori in XML" (17), "Verifica la presenza di errori usando un'espressione regolare" (18), "Ignora se invariato con heartbeat" (20), "JavaScript" (21), "Pattern Prometheus" (22), "Prometheus in JSON" (23), "CSV in JSON" (24), "Sostituzione" (25), "Verifica non supportato" (26), "Valore SNMP walk" (28), "SNMP walk in JSON" (29) o "Valore SNMP get" (30)

Proprietà	Tipo	Descrizione
error_handler	integer	Tipo di azione utilizzato in caso di errore del passaggio di pre-elaborazione. Valori possibili: 0 - Il messaggio di errore viene impostato dal server Zabbix; 1 - Ignora il valore; 2 - Imposta un valore personalizzato; 3 - Imposta un messaggio di errore personalizzato. Valori possibili se `type` è impostato su ”Verifica non supportato”: 1 - Ignora il valore; 2 - Imposta un valore personalizzato; 3 - Imposta un messaggio di errore personalizzato. Comportamento della proprietà: - *required* se `type` è impostato su ”Moltiplicatore personalizzato” (1), ”Espressione regolare” (5), ”Booleano in decimale” (6), ”Ottale in decimale” (7), ”Esadecimale in decimale” (8), ”Modifica semplice” (9), ”Modifica al secondo” (10), ”XPath XML” (11), ”JSONPath” (12), ”Nell’intervallo” (13), ”Corrisponde all’espressione regolare” (14), ”Non corrisponde all’espressione regolare” (15), ”Verifica la presenza di errori in JSON” (16), ”Verifica la presenza di errori in XML” (17), ”Verifica la presenza di errori usando un’espressione regolare” (18), ”Pattern Prometheus” (22), ”Prometheus in JSON” (23), ”CSV in JSON” (24), ”Verifica non supportato” (26), ”XML in JSON” (27), ”Valore SNMP walk” (28), ”SNMP walk in JSON” (29) o ”Valore SNMP get” (30)
error_handler_params	string	Parametri del gestore degli errori. Comportamento della proprietà: - *required* se `error_handler` è impostato su ”Imposta un valore personalizzato” o ”Imposta un messaggio di errore personalizzato”

I seguenti parametri e gestori degli errori sono supportati per ogni tipo di pre-elaborazione.

Tipo di pre-elaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
1	Moltiplicatore personalizzato	numero^1, 6^			0, 1, 2, 3
2	Rimozione a destra	elenco di caratteri[2]			
3	Rimozione a sinistra	elenco di caratteri[2]			
4	Rimozione	elenco di caratteri[2]			
5	Espressione regolare	pattern[3]	output[2]		0, 1, 2, 3
6	Booleano in decimale				0, 1, 2, 3
7	Ottale in decimale				0, 1, 2, 3

Tipo di pre-elaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
8	Esadecimale in decimale				0, 1, 2, 3
9	Modifica semplice				0, 1, 2, 3
10	Modifica al secondo				0, 1, 2, 3
11	XPath XML	percorso[4]			0, 1, 2, 3
12	JSONPath	percorso[4]			0, 1, 2, 3
13	Nell’intervallo	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Corrisponde all’espressione regolare	pattern[3]			0, 1, 2, 3
15	Non corrisponde all’espressione regolare	pattern[3]			0, 1, 2, 3
16	Verifica la presenza di errori in JSON	percorso[4]			0, 1, 2, 3
17	Verifica la presenza di errori in XML	percorso[4]			0, 1, 2, 3
18	Verifica la presenza di errori usando un’espressione regolare	pattern[3]	output[2]		0, 1, 2, 3
19	Ignora se invariato				

Tipo di pre-elaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
20	Ignora se invariato con heartbeat	secondi^5, 6^			
21	JavaScript	script[2]			
22	Pattern Prometheus	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus in JSON	pattern^6, 7^			0, 1, 2, 3
24	CSV in JSON	carattere[2]	carattere[2]	0,1	0, 1, 2, 3
25	Sostituzione	stringa di ricerca[2]	sostituzione[2]		
26	Verifica non supportato	scope[1]	pattern^3, 6^		1, 2, 3
27	XML in JSON				0, 1, 2, 3
28	Valore SNMP walk	OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS		0, 1, 2, 3
29	SNMP walk in JSON[10]	Nome del campo[2]	Prefisso OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS	0, 1, 2, 3
30	Valore SNMP get	Formato: 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS			0, 1, 2, 3

[1] numero intero o a virgola mobile
 [2] stringa
 [3] espressione regolare
 [4] JSONPath o XPath XML
 [5] intero positivo (con supporto dei suffissi temporali, ad esempio `30s`, `1m`, `2h`, `1d`)
 [6] macro utente, macro LLD
 [7] pattern Prometheus conforme alla sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ogni componente del pattern Prometheus (metrica, nome dell'etichetta, valore dell'etichetta e valore della metrica) può essere una macro utente o una macro LLD.
 [8] output Prometheus conforme alla sintassi: `<label name>` (può essere una macro utente o una macro LLD) se `label` è selezionato come secondo parametro.
 [9] Una delle funzioni di aggregazione: `sum`, `min`, `max`, `avg`, `count` se `function` è selezionato come secondo parametro.
 [10] Supporta più record "Nome del campo,Prefisso OID,Formato" delimitati da un carattere di nuova riga.

itemprototype.create

Descrizione

`object itemprototype.create(object/array itemPrototypes)`

Questo metodo consente di creare nuovi item prototype.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Prototipo di item da creare.

Oltre alle proprietà standard del prototipo di item, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
ruleid	ID	ID della regola LLD a cui appartiene l'item. Comportamento del parametro: - *obbligatorio*
preprocessing	array	Opzioni di preprocessing del prototipo di item.
tags	array	Tag del prototipo di item.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli item prototype creati nella proprietà `itemids`. L'ordine degli ID restituiti corrisponde all'ordine degli item prototype passati.

Esempi

Creazione di un prototipo di item

Creare un prototipo di item per monitorare lo spazio libero su disco in un file system individuato. Gli item individuati devono essere item numerici di Zabbix agent aggiornati ogni 30 secondi.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Free disk space on {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

Creazione di un prototipo di item con preprocessing

Creare un item utilizzando la variazione al secondo e un moltiplicatore personalizzato come secondo passaggio.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Traffico di rete in ingresso su {#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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Creazione di un prototipo di item dipendente

Creare un prototipo di item dipendente per il prototipo di item master con ID 44211. Sono consentite solo dipendenze sullo stesso host (template/regola di discovery), pertanto il prototipo di item master e quello dipendente devono avere lo stesso hostid e ruleid.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Dependent test item prototype",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
```

```
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

Creare un prototipo di item HTTP agent

Creare un prototipo di item con URL che utilizza una macro utente, campi di query e intestazioni personalizzate.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Creare un prototipo di item script

Creare una semplice raccolta dati utilizzando un prototipo di item script.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::create() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

Descrizione

`object itemprototype.delete(array itemPrototypeIds)`

Questo metodo consente di eliminare i prototipi di item.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei prototipi elemento da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di item eliminati nella proprietà `prototypeids`.

Esempi

Eliminazione di più prototipi di item

Eliminare due prototipi di item.
 I prototipi di item dipendenti vengono rimossi automaticamente se il master item o il prototipo di item viene eliminato.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::delete() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

Descrizione

`integer/array itemprototype.get(object parameters)`

Il metodo consente di recuperare i prototipi di item in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parameter	Type	Description
discoveryids	ID/array	Restituisce solo i prototipi di item che appartengono alle regole LLD specificate.
graphids	ID/array	Restituisce solo i prototipi di item utilizzati nei prototipi di grafico specificati.
hostids	ID/array	Restituisce solo i prototipi di item che appartengono agli host specificati.
inherited	boolean	Se impostato su `true`, restituisce solo i prototipi di item ereditati da un template.
itemids	ID/array	Restituisce solo i prototipi di item con gli ID specificati.
monitored	boolean	Se impostato su `true`, restituisce solo i prototipi di item abilitati che appartengono a host monitorati.
templated	boolean	Se impostato su `true`, restituisce solo i prototipi di item che appartengono ai template.
templateids	ID/array	Restituisce solo i prototipi di item che appartengono ai template specificati.
triggerids	ID/array	Restituisce solo i prototipi di item utilizzati nei prototipi di trigger specificati.

Parameter	Type	Description
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery del prototipo di item. L'oggetto di discovery del prototipo di item collega un prototipo di item rilevato a un prototipo di item da cui è stato rilevato. Ha le seguenti proprietà: `parent_itemid` - `(string)` ID del prototipo di item da cui è stato creato il prototipo di item; `key_` - `(string)` chiave del prototipo di item; `status` - `(int)` stato di discovery del prototipo di item: 0 - *(predefinito)* il prototipo di item è rilevato, 1 - il prototipo di item non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un prototipo di item non più rilevato verrà eliminato; `ts_disable` - `(timestamp)` momento in cui un prototipo di item non più rilevato verrà disabilitato; `disable_source` - `(int)` indicatore che specifica se il prototipo di item è stato disabilitato da una regola LLD o manualmente: 0 - *(predefinito)* disabilitato automaticamente, 1 - disabilitato da una regola LLD.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola di low-level discovery a cui appartiene il prototipo di item.
selectDiscoveryRulePrototype	query	Restituisce una proprietà `discoveryRulePrototype` con il prototipo della regola LLD padre a cui appartiene il prototipo di item.
selectGraphs	query	Restituisce una proprietà `graphs` con i prototipi di grafico in cui viene utilizzato il prototipo di item. Supporta `count`.
selectHosts	query	Restituisce una proprietà `hosts` con un array di host a cui appartiene il prototipo di item.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag presenti su template/host/template collegati.
selectTags	query	Restituisce i tag del prototipo di item nella proprietà `tags`.
selectTriggers	query	Restituisce una proprietà `triggers` con i prototipi di trigger in cui viene utilizzato il prototipo di item. Supporta `count`.
selectPreprocessing	query	Restituisce una proprietà `preprocessing` con le opzioni di preprocessing del prototipo di item.
selectValueMap	query	Restituisce una proprietà `valuemap` con la mappa dei valori del prototipo di item.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text`. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene il prototipo di item.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectGraphs` - i risultati saranno ordinati per `name`; `selectTriggers` - i risultati saranno ordinati per `description`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.

Parameter	Type	Description
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei prototipi di item da una regola LLD

Recupera tutti i prototipi di item per uno specifico ID di regola LLD.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Risposta:

```
{
    "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": "Il carico del processore è calcolato come carico della CPU del sistema diviso
```

```
            "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
}
```


Trovare un item dipendente

Trova un item dipendente per uno specifico ID item.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

Risposta:


```
{
    "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
}
```


Trovare il prototipo di item HTTP agent

Trovare il prototipo di item HTTP agent con metodo di richiesta HEAD per uno specifico ID host.

Richiesta:


```
{
    "jsonrpc": "2.0",
```

```
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Host
- Prototipo di grafico
- Prototipo di trigger

Fonte

CItemPrototype::get() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

Descrizione

`object itemprototype.update(object/array itemPrototypes)`

Questo metodo consente di aggiornare gli item prototype esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dei prototipi di item da aggiornare.

La proprietà `itemid` deve essere definita per ciascun prototipo di item, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del prototipo di item, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
preprocessing	array	Opzioni di preprocessing del prototipo di item per sostituire le opzioni di preprocessing correnti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
tags	array	Tag del prototipo di item.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei prototipi elemento aggiornati sotto la proprietà `itemids`.

Esempi

Modifica dell'interfaccia di un prototipo di item

Modificare l'interfaccia del host che verrà utilizzata dagli item rilevati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

Aggiornare il prototipo di item dipendente

Aggiorna il prototipo di item dipendente con un nuovo ID del prototipo di item master. Sono consentite solo dipendenze sullo stesso host (template/regola di discovery), pertanto il master e l'item dipendente devono avere lo stesso hostid e ruleid.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

Aggiornare il prototipo di item HTTP agent

Modificare i campi della query e rimuovere tutte le intestazioni personalizzate.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
```

```
        "headers": []
    }
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Aggiornamento delle opzioni di preprocessing degli item

Aggiorna un prototipo di item con la regola di preprocessing dell'item "Moltiplicatore personalizzato".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Aggiornamento di un prototipo di item script

Aggiorna un prototipo di item script con uno script diverso e rimuove i parametri non necessari che erano utilizzati dallo script precedente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::update() in *ui/include/classes/api/services/CItemPrototype.php*.

Prototipo di regola LLD

Questa classe è progettata per funzionare con i prototipi di regole di low-level discovery.

Riferimenti agli oggetti:

- Prototipo di regola LLD
  - Header HTTP
  - Campo query HTTP
- Filtro del prototipo di regola LLD
  - Condizione del filtro del prototipo di regola LLD
- Percorso macro del prototipo LLD
- Preprocessing del prototipo di regola LLD
- Override del prototipo di regola LLD
  - Filtro di override del prototipo di regola LLD
    - Condizione del filtro di override del prototipo di regola LLD
  - Operazione di override del prototipo di regola LLD
    - Stato dell'operazione di override del prototipo di regola LLD
    - Rilevamento dell'operazione di override del prototipo di regola LLD
    - Periodo dell'operazione di override del prototipo di regola LLD
    - Storico dell'operazione di override del prototipo di regola LLD
    - Trend dell'operazione di override del prototipo di regola LLD
    - Gravità dell'operazione di override del prototipo di regola LLD
    - Tag dell'operazione di override del prototipo di regola LLD
    - Template dell'operazione di override del prototipo di regola LLD
    - Inventario dell'operazione di override del prototipo di regola LLD

Metodi disponibili:

- discoveryruleprototype.create - crea nuovi prototipi di regole LLD
- discoveryruleprototype.delete - elimina prototipi di regole LLD
- discoveryruleprototype.get - recupera prototipi di regole LLD
- discoveryruleprototype.update - aggiorna prototipi di regole LLD

Oggetto prototipo della regola LLD

I seguenti oggetti sono direttamente correlati all'API `discoveryruleprototype`.

Prototipo di regola LLD

L'oggetto prototipo della regola di discovery di basso livello presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
itemid	ID	ID del prototipo della regola LLD. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
ruleid	ID	ID della regola LLD/regola LLD prototipo padre. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di creazione
delay	string	Intervallo di aggiornamento del prototipo della regola LLD. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d) oppure, facoltativamente, uno o più intervalli personalizzati, separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e di pianificazione. Accetta macro utente. Se utilizzato, il valore deve essere una singola macro. Non sono supportate più macro o macro combinate con testo. Gli intervalli flessibili possono essere scritti come due macro separate da una barra (ad esempio, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Esempio: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Comportamento della proprietà: - *obbligatoria* se type è impostato su ”Zabbix agent”, ”Simple check”, ”Zabbix internal”, ”External check”, ”Database monitor”, ”IPMI agent”, ”SSH agent”, ”TELNET agent”, ”JMX agent”, ”HTTP agent”, ”SNMP agent”, ”Script”, ”Browser” oppure se type è impostato su ”Zabbix agent (active)” e key_ non contiene ”mqtt.get”
hostid	ID	ID dell’host a cui appartiene il prototipo della regola LLD. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
flags	integer	Origine del prototipo della regola LLD. Valori possibili: 3 - prototipo di una regola di discovery di basso livello; 7 - prototipo di una regola di discovery di basso livello rilevata. Comportamento della proprietà: - *sola lettura*
interfaceid	ID	ID dell’interfaccia dell’host del prototipo della regola LLD. Comportamento della proprietà: - *obbligatoria* se il prototipo della regola LLD appartiene a un host e type è impostato su ”Zabbix agent”, ”IPMI agent”, ”JMX agent” o ”SNMP agent” - *supportata* se il prototipo della regola LLD appartiene a un host e type è impostato su ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” o ”HTTP agent”
key_	string	Chiave del prototipo della regola LLD. È richiesta almeno una macro LLD. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
name	string	Nome del prototipo della regola LLD. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
type	integer	Tipo del prototipo della regola LLD. Valori possibili: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
url	string	Stringa URL. Supporta macro utente, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
allow_traps	integer	Consente di popolare il valore in modo analogo all’item trapper. Valori possibili: 0 - *(predefinito)* Non accettare dati in ingresso; 1 - Accettare dati in ingresso. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent”
authtype	integer	Metodo di autenticazione. Valori possibili se `type` è impostato su ”SSH agent”: 0 - *(predefinito)* password; 1 - chiave pubblica. Valori possibili se `type` è impostato su ”HTTP agent”: 0 - *(predefinito)* nessuno; 1 - basic; 2 - NTLM. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SSH agent” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (se `type` è impostato su ”HTTP agent”)
description	string	Descrizione del prototipo della regola LLD.

Proprietà	Tipo	Descrizione
error	string	Testo dell'errore se si verificano problemi durante l'aggiornamento del valore del prototipo della regola LLD. Comportamento della proprietà: - *sola lettura*
follow_redirects	integer	Segue i reindirizzamenti della risposta durante il polling dei dati. Valori possibili: 0 - Non seguire i reindirizzamenti; 1 - *(predefinito)* Seguire i reindirizzamenti. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
headers	array	Array di intestazioni da inviare durante l'esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
http_proxy	string	Stringa di connessione al proxy HTTP(S). Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
ipmi_sensor	string	Sensore IPMI. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "IPMI agent" e `key_` non è impostato su "ipmi.get" - *supportata* se `type` è impostato su "IPMI agent" - *sola lettura* per gli oggetti ereditati
jmx_endpoint	string	Stringa di connessione personalizzata dell'agent JMX. Predefinito: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento della proprietà: - *supportata* se `type` è impostato su "JMX agent"
lifetime	string	Periodo di tempo dopo il quale gli item non più rilevati verranno eliminati. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`), un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`) oppure una macro utente. Predefinito: `7d`.
lifetime_type	integer	Scenario per l'eliminazione delle risorse LLD perse. Valori possibili: 0 - *(predefinito)* Eliminare dopo il raggiungimento della soglia di durata; 1 - Non eliminare; 2 - Eliminare immediatamente.
enabled_lifetime	string	Periodo di tempo dopo il quale gli item non più rilevati verranno disabilitati. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`), un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`) oppure una macro utente. Predefinito: `0`.

Proprietà	Tipo	Descrizione
enabled_lifetime_type	integer	Scenario per la disabilitazione delle risorse LLD perse. Valori possibili: 0 - Disabilitare dopo il raggiungimento della soglia di durata; 1 - Non disabilitare; 2 - *(predefinito)* Disabilitare immediatamente.
master_itemid	ID	ID dell'item master. La regola di discovery non può essere l'item master di un'altra regola di discovery. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Dependent item" - *sola lettura* per gli oggetti ereditati
output_format	integer	Indica se la risposta deve essere convertita in JSON. Valori possibili: 0 - *(predefinito)* Memorizzare i dati non elaborati; 1 - Convertire in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
params	string	Parametri aggiuntivi in base al tipo di prototipo della regola LLD: - script eseguito per le regole LLD SSH e Telnet; - query SQL per le regole LLD del monitor database; - formula per le regole LLD calcolate; - script per le regole LLD Script e Browser. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Database monitor", "SSH agent", "TELNET agent", "Script" o "Browser" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "Script" o "Browser")
parameters	object/array	Parametri aggiuntivi se `type` è impostato su "Script" o "Browser". Array di oggetti con le proprietà `name` e `value`, dove `name` deve essere univoco. Comportamento della proprietà: - *supportata* se `type` è impostato su "Script" o "Browser" - *sola lettura* per gli oggetti ereditati
password	string	Password per l'autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "JMX agent" e `username` è impostato - *supportata* se `type` è impostato su "Simple check", "Database monitor", "SSH agent", "TELNET agent" o "HTTP agent" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "HTTP agent")
post_type	integer	Tipo del corpo dei dati POST memorizzato nella proprietà `posts`. Valori possibili: 0 - *(predefinito)* Dati non elaborati; 2 - Dati JSON; 3 - Dati XML. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
posts	string	Dati del corpo della richiesta HTTP(S). Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”HTTP agent” e `post_type` è impostato su ”Dati JSON” o ”Dati XML” - *supportata* se `type` è impostato su ”HTTP agent” e `post_type` è impostato su ”Dati non elaborati” - *sola lettura* per gli oggetti ereditati
privatekey	string	Nome del file della chiave privata. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent” e `authtype` è impostato su ”chiave pubblica”
publickey	string	Nome del file della chiave pubblica. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent” e `authtype` è impostato su ”chiave pubblica”
query_fields	array	Array di campi query da inviare durante l’esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
request_method	integer	Tipo di metodo della richiesta. Valori possibili: 0 - *(predefinito)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
retrieve_mode	integer	Parte della risposta da memorizzare. Valori possibili se `request_method` è impostato su ”GET”, ”POST” o ”PUT”: 0 - *(predefinito)* Corpo; 1 - Intestazioni; 2 - Verranno memorizzati sia il corpo sia le intestazioni. Valori possibili se `request_method` è impostato su ”HEAD”: 1 - Intestazioni. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
snmp_oid	string	OID SNMP. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SNMP agent” - *sola lettura* per gli oggetti ereditati
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
ssl_key_file	string	Percorso del file della chiave SSL privata. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ssl_key_password	string	Password del file della chiave SSL. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
state	integer	Stato del prototipo della regola LLD. Valori possibili: 0 - *(predefinito)* normale; 1 - non supportato. Comportamento della proprietà: - *sola lettura*
status	integer	Stato del prototipo della regola LLD. Valori possibili: 0 - *(predefinito)* prototipo della regola LLD abilitato; 1 - prototipo della regola LLD disabilitato.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole. Supporta anche macro utente come parte di un elenco separato da virgole. Esempio: 200,200-{$M},{$M},200-400 Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
templateid	ID	ID del prototipo della regola LLD padre del template. Comportamento della proprietà: - *sola lettura*
timeout	string	Timeout della richiesta di polling dei dati dell’item. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`) oppure un valore con un suffisso temporale (ad esempio, `30s`, `1m`). Accetta anche macro utente. L’intervallo dei valori possibili è: 1-600s. Predefinito: ”” - utilizza le impostazioni del proxy/globali. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Zabbix agent” (0), ”Simple check” (3) e `key_` non inizia con ”`vmware.`” e ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) e `snmp_oid` inizia con ”`walk[`” o ”`get[`”, ”Script” (21), ”Browser” (22) - *sola lettura* per gli oggetti ereditati

Proprietà	Tipo	Descrizione
trapper_hosts	string	Host consentiti. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Zabbix trapper” oppure se `type` è impostato su ”HTTP agent” e `allow_traps` è impostato su ”Accettare dati in ingresso”. Se la proprietà viene omessa, viene impostata automaticamente sulla macro utente `{$TRAPPER.ALLOWED_HOSTS}` (valore predefinito). Per impostazione predefinita, `{$TRAPPER.ALLOWED_HOSTS}` è `127.0.0.1,::1`. Se la proprietà è impostata su una stringa vuota, le connessioni in ingresso verranno negate da tutti gli host.
username	string	Nome utente per l’autenticazione. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su ”SSH agent”, ”TELNET agent” oppure se `type` è impostato su ”JMX agent” e `password` è impostato - *supportato* se `type` è impostato su ”Simple check”, ”Database monitor” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (se `type` è impostato su ”HTTP agent”)
uuid	string	Identificatore univoco universale, utilizzato per collegare i prototipi delle regole LLD importati a quelli già esistenti. Generato automaticamente se non specificato. Comportamento della proprietà: - *supportato* se il prototipo della regola LLD appartiene a un template
verify_host	integer	Indica se convalidare che il nome host della connessione corrisponda a quello presente nel certificato dell’host. Valori possibili: 0 - *(predefinito)* Non convalidare; 1 - Convalidare. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
verify_peer	integer	Indica se convalidare l’autenticità del certificato dell’host. Valori possibili: 0 - *(predefinito)* Non convalidare; 1 - Convalidare. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati

Intestazione HTTP

L’oggetto header ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
name	string	Nome dell’intestazione HTTP. Comportamento della proprietà: - *obbligatorio*

Proprietà	Tipo	Descrizione
value	string	Valore dell'intestazione. Comportamento della proprietà: - *obbligatorio*

Campo di query HTTP

L'oggetto campo di query definisce un nome e un valore utilizzati per specificare un parametro URL. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
name	string	Nome del parametro. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro. Comportamento della proprietà: - *obbligatorio*

Filtro del prototipo di regola LLD

L'oggetto filtro del prototipo di regola LLD definisce un insieme di condizioni che possono essere utilizzate per filtrare gli oggetti rilevati. Ha le seguenti proprietà:

Property	Type	Description
conditions	object/array	Insieme di condizioni di filtro da utilizzare per filtrare i risultati. Le condizioni saranno ordinate nell'ordine del loro posizionamento nella formula. Property behavior: - *obbligatorio*
evaltype	integer	Metodo di valutazione della condizione di filtro. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Property behavior: - *obbligatorio*
eval_formula	string	Espressione generata che verrà utilizzata per valutare le condizioni di filtro. L'espressione contiene ID che fanno riferimento a specifiche condizioni di filtro tramite il loro `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Property behavior: - *sola lettura*

Property	Type	Description
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a specifiche condizioni di filtro tramite il loro `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni di filtro: nessuna condizione può rimanere inutilizzata o omessa. Property behavior: - *obbligatorio* se `evaltype` è impostato su "custom expression"

Condizione del filtro del prototipo di regola LLD

L'oggetto condizione del filtro del prototipo di regola LLD definisce un controllo separato da eseguire sul valore di una macro LLD. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
macro	string	Macro LLD su cui eseguire il controllo. Comportamento della proprietà: - *required*
value	string	Valore con cui confrontare. Comportamento della proprietà: - *required* se `operator` è impostato su "corrisponde all'espressione regolare" o "non corrisponde all'espressione regolare"
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente durante la modifica delle condizioni del filtro, ma verrà generato nuovamente quando verranno richieste in seguito. Comportamento della proprietà: - *required* se `evaltype` di oggetto filtro del prototipo di regola LLD è impostato su "espressione personalizzata"
operator	integer	Operatore della condizione. Valori possibili: 8 - *(predefinito)* corrisponde all'espressione regolare; 9 - non corrisponde all'espressione regolare; 12 - esiste; 13 - non esiste.

Note:
Per comprendere meglio come utilizzare i filtri con vari tipi di espressioni, vedere gli esempi nelle pagine dei metodi discoveryruleprototype.get e discoveryruleprototype.create.

Percorso della macro LLD

Il percorso della macro LLD ha le seguenti proprietà:

Proprietà	Type	Descrizione
lld_macro	string	Macro LLD. Comportamento della proprietà: - *obbligatorio*

Proprietà	Type	Descrizione
path	string	Selettore del valore che verrà assegnato alla macro corrispondente. Comportamento della proprietà: - *obbligatorio*

Preelaborazione del prototipo della regola LLD

L'oggetto di preelaborazione del prototipo della regola LLD presenta le seguenti proprietà.

Property	Type	Description
type	integer	Il tipo di opzione di preelaborazione. Valori possibili: 5 - Espressione regolare; 11 - XPath XML; 12 - JSONPath; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Verifica la presenza di errori in JSON; 17 - Verifica la presenza di errori in XML; 20 - Ignora i valori invariati con heartbeat; 21 - JavaScript; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Sostituisci; 27 - XML in JSON; 28 - Valore SNMP walk; 29 - SNMP walk in JSON; 30 - Valore SNMP get. Property behavior: - *required*
params	string	Parametri aggiuntivi utilizzati dall'opzione di preelaborazione. Più parametri sono separati dal carattere di nuova riga (\n). Property behavior: - *required* se `type` è impostato su "Espressione regolare" (5), "XPath XML" (11), "JSONPath" (12), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Verifica la presenza di errori in JSON" (16), "Verifica la presenza di errori in XML" (17), "Ignora i valori invariati con heartbeat" (20), "JavaScript" (21), "Prometheus in JSON" (23), "CSV in JSON" (24), "Sostituisci" (25), "Valore SNMP walk" (28), "SNMP walk in JSON" (29) o "Valore SNMP get" (30)

Property	Type	Description
error_handler	integer	Tipo di azione utilizzato in caso di errore in un passaggio di preelaborazione. Valori possibili: 0 - Il messaggio di errore viene impostato dal server Zabbix; 1 - Ignora il valore; 2 - Imposta un valore personalizzato; 3 - Imposta un messaggio di errore personalizzato. Property behavior: - *required* se `type` è impostato su ”Espressione regolare” (5), ”XPath XML” (11), ”JSONPath” (12), ”Corrisponde all’espressione regolare” (14), ”Non corrisponde all’espressione regolare” (15), ”Verifica la presenza di errori in JSON” (16), ”Verifica la presenza di errori in XML” (17), ”Prometheus in JSON” (23), ”CSV in JSON” (24), ”XML in JSON” (27), ”Valore SNMP walk” (28), ”SNMP walk in JSON” (29) o ”Valore SNMP get” (30)
error_handler_params	string	Parametri del gestore degli errori. Property behavior: - *required* se `error_handler` è impostato su ”Imposta un valore personalizzato” o ”Imposta un messaggio di errore personalizzato”

I seguenti parametri e gestori degli errori sono supportati per ciascun tipo di preelaborazione.

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
5	Espressione regolare	pattern[1]	output[2]		0, 1, 2, 3
11	XPath XML	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Corrisponde all’espressione regolare	pattern[1]			0, 1, 2, 3
15	Non corrisponde all’espressione regolare	pattern[1]			0, 1, 2, 3
16	Verifica la presenza di errori in JSON	path[3]			0, 1, 2, 3
17	Verifica la presenza di errori in XML	path[3]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
20	Ignora i valori invariati con heartbeat	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus pattern in JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Sostituisci	search string[2]	replacement[2]		
27	XML in JSON				0, 1, 2, 3
28	Valore SNMP walk	OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS		0, 1, 2, 3
29	SNMP walk in JSON[7]	Field name[2]	OID prefix[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS	0, 1, 2, 3
30	Valore SNMP get	Formato: 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS			0, 1, 2, 3

[1] espressione regolare
 [2] stringa
 [3] JSONPath o XPath XML
 [4] intero positivo (con supporto per i suffissi temporali, ad esempio 30s, 1m, 2h, 1d)
 [5] macro utente
 [6] pattern Prometheus conforme alla sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ogni componente del pattern Prometheus (metrica, nome dell'etichetta, valore dell'etichetta e valore della metrica) può essere una macro utente.
 [7] Supporta più record "Field name,OID prefix,Format" delimitati da un carattere di nuova riga.

Override dei prototipi di regola LLD

L'oggetto override dei prototipi di regola LLD definisce un insieme di regole (filtri, condizioni e operazioni) utilizzate per sovrascrivere le proprietà di diversi oggetti prototipo. Ha le seguenti proprietà:

Property	Type	Description
name	string	Nome univoco dell'override. Property behavior: - *obbligatorio*

Property	Type	Description
step	integer	Numero d'ordine univoco dell'override. Property behavior: - *obbligatorio*
stop	integer	Interrompe l'elaborazione degli override successivi in caso di corrispondenza. Valori possibili: 0 - *(predefinito)* non interrompere l'elaborazione degli override; 1 - interrompere l'elaborazione degli override se il filtro corrisponde.
filter	object	Filtro dell'override.
operations	object/array	Operazioni dell'override.

Filtro di override del prototipo di regola LLD

L'oggetto filtro di override del prototipo di regola LLD definisce un insieme di condizioni che, se corrispondono all'oggetto individuato, applicano l'override. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
conditions	object/array	Insieme di condizioni del filtro di override da utilizzare per la corrispondenza con gli oggetti individuati. Le condizioni saranno ordinate in base al loro posizionamento nella formula. Property behavior: - *required*
evaltype	integer	Metodo di valutazione della condizione del filtro di override. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Property behavior: - *required*
eval_formula	string	Espressione generata che verrà utilizzata per valutare le condizioni del filtro di override. L'espressione contiene ID che fanno riferimento a specifiche condizioni del filtro di override tramite il relativo `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Property behavior: - *read-only*
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri di override con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a specifiche condizioni del filtro di override tramite il relativo `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni del filtro di override: nessuna condizione può rimanere inutilizzata o omessa. Property behavior: - *required* se `evaltype` è impostato su "custom expression"

Condizione di filtro dell'override del prototipo di regola LLD

L'oggetto condizione di filtro dell'override del prototipo di regola LLD definisce un controllo separato da eseguire sul valore di una macro LLD. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
macro	string	Macro LLD su cui eseguire il controllo. Property behavior: - *obbligatoria*
value	string	Valore con cui confrontare. Property behavior: - *obbligatoria* se `operator` è impostato su ”corrisponde all’espressione regolare” o ”non corrisponde all’espressione regolare”
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un’espressione personalizzata. Può contenere solo lettere maiuscole. L’ID deve essere definito dall’utente durante la modifica delle condizioni di filtro, ma verrà generato nuovamente quando verranno richieste in seguito. Property behavior: - *obbligatoria* se `evaltype` dell’oggetto filtro dell’override del prototipo di regola LLD è impostato su ”espressione personalizzata”
operator	integer	Operatore della condizione. Valori possibili: 8 - *(predefinito)* corrisponde all’espressione regolare; 9 - non corrisponde all’espressione regolare; 12 - esiste; 13 - non esiste.

Operazione di override del prototipo della regola LLD

L’operazione di override del prototipo della regola LLD è una combinazione di condizioni e azioni da eseguire sull’oggetto prototipo. Ha le seguenti proprietà:

Property	Type	Description
operationobject	integer	Tipo di oggetto rilevato su cui eseguire l’azione. Valori possibili: 0 - Prototipo di item; 1 - Prototipo di trigger; 2 - Prototipo di grafico; 3 - Prototipo di host. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione di override. Valori possibili: 0 - *(predefinito)* uguale; 1 - non uguale; 2 - contiene; 3 - non contiene; 8 - corrisponde; 9 - non corrisponde.
value	string	Modello da confrontare con il nome del prototipo di item, trigger, grafico o host, a seconda dell’oggetto selezionato.
opstatus	object	Oggetto stato dell’operazione di override per gli oggetti prototipo di item, trigger e host.
opdiscover	object	Oggetto stato di rilevamento dell’operazione di override (tutti i tipi di oggetto).
opperiod	object	Oggetto periodo dell’operazione di override (intervallo di aggiornamento) per l’oggetto prototipo di item.

Property	Type	Description
ophistory	object	Oggetto cronologia dell'operazione di override per l'oggetto prototipo di item.
optrends	object	Oggetto trend dell'operazione di override per l'oggetto prototipo di item.
opseverity	object	Oggetto gravità dell'operazione di override per l'oggetto prototipo di trigger.
optag	object/array	Oggetto tag dell'operazione di override per gli oggetti prototipo di trigger e host.
optemplate	object/array	Oggetto template dell'operazione di override per l'oggetto prototipo di host.
opinventory	object	Oggetto inventario dell'operazione di override per l'oggetto prototipo di host.

Stato dell'operazione di override del prototipo della regola LLD

Stato dell'operazione di override del prototipo della regola LLD impostato sull'oggetto rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
status	integer	Esegue l'override dello stato per l'oggetto selezionato. Valori possibili: 0 - Crea abilitato; 1 - Crea disabilitato. Comportamento della proprietà: - *obbligatorio*

Operazione di override del prototipo di regola LLD: discover

Lo stato discover dell'operazione di override del prototipo di regola LLD è impostato sull'oggetto rilevato. Ha le seguenti proprietà:

Property	Type	Description
discover	integer	Esegue l'override dello stato discover per l'oggetto selezionato. Valori possibili: 0 - Sì, continua a rilevare gli oggetti; 1 - No, i nuovi oggetti non verranno rilevati e quelli esistenti verranno contrassegnati come persi. Comportamento della proprietà: - *obbligatorio*

Periodo dell'operazione di override del prototipo della regola LLD

Il periodo dell'operazione di override del prototipo della regola LLD è un valore dell'intervallo di aggiornamento impostato per l'item rilevato. Presenta le seguenti proprietà:

Property	Type	Description
delay	string	Esegue l'override dell'intervallo di aggiornamento del prototipo dell'item. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d) oppure, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e intervalli di pianificazione. Accetta macro utente o macro LLD. Se utilizzato, il valore deve essere una singola macro. Non sono supportate più macro o macro combinate con testo. Gli intervalli flessibili possono essere scritti come due macro separate da una barra (ad esempio, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Esempio: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Comportamento della proprietà: - *required*

Cronologia dell'operazione di override del prototipo della regola LLD

Valore della cronologia dell'operazione di override del prototipo della regola LLD impostato sull'item rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
history	string	Esegue l'override della cronologia del prototipo di item, che è un'unità di tempo che definisce per quanto tempo i dati della cronologia devono essere conservati. Accetta anche macro utente e macro LLD. Comportamento della proprietà: - *required*

Tendenze delle operazioni di override del prototipo della regola LLD

Valore delle tendenze delle operazioni di override del prototipo della regola LLD impostato sull'item rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
trends	string	Esegue l'override delle tendenze del prototipo di item, che rappresentano un'unità di tempo che definisce per quanto tempo i dati di tendenza devono essere conservati. Accetta anche macro utente e macro LLD. Comportamento della proprietà: - *obbligatorio*

Gravità dell'operazione di override del prototipo della regola LLD

Valore della gravità dell'operazione di override del prototipo della regola LLD impostato sul trigger rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
severity	integer	Sovrascrive la gravità del prototipo di trigger. Valori possibili: 0 - *(predefinito)* non classificato; 1 - informazione; 2 - avviso; 3 - medio; 4 - alto; 5 - disastro. Comportamento della proprietà: - *obbligatorio*

Tag dell'operazione di override del prototipo di regola LLD

L'oggetto tag dell'operazione di override del prototipo di regola LLD contiene il nome e il valore del tag che vengono impostati sull'oggetto individuato. Ha le seguenti proprietà:

Property	Type	Description
tag	string	Nuovo nome del tag. Comportamento della proprietà: - *obbligatoria*
value	string	Nuovo valore del tag.

Template dell'operazione di override del prototipo della regola LLD

Oggetto template dell'operazione di override del prototipo della regola LLD collegato al host rilevato. Ha le seguenti proprietà:

Property	Type	Description
templateid	ID	Esegue l'override del template dei template collegati del prototipo di host. Comportamento della proprietà: - *obbligatorio*

Valore della modalità dell'inventario dell'operazione di override del prototipo di regola LLD

Valore della modalità dell'inventario dell'operazione di override del prototipo di regola LLD impostato sul host rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
inventory_mode	integer	Sovrascrive la modalità dell'inventario del prototipo di host. Valori possibili: -1 - disabilitato; 0 - *(predefinito)* manuale; 1 - automatico. Comportamento della proprietà: - *obbligatorio*

discoveryruleprototype.create

Descrizione

```
object discoveryruleprototype.create(object/array lldRules)
```

Questo metodo consente di creare nuovi prototipi di regole LLD.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Prototipi di regole LLD da creare.

Oltre alle proprietà standard dei prototipi di regole LLD, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
filter	object	Filtro del prototipo di regola LLD per la regola LLD.
preprocessing	object/array	Opzioni di preprocessing del prototipo di regola LLD.
lld_macro_paths	object/array	Opzioni lld_macro_path del prototipo di regola LLD.
overrides	object/array	Opzioni di override del prototipo di regola LLD.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di regola LLD creati nella proprietà `itemids`. L'ordine degli ID restituiti corrisponde all'ordine dei prototipi di regola LLD passati.

Esempi

Creazione di un prototipo di regola LLD

Creare un prototipo di regola LLD (Tipo: Nested) per individuare i tablespace nell'istanza del database.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

Vedi anche

- Filtro del prototipo di regola LLD
- Percorsi macro LLD
- Preelaborazione del prototipo di regola LLD

Sorgente

CDiscoveryRulePrototype::create() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

Descrizione

`object discoveryruleprototype.delete(array lldRuleIds)`

Questo metodo consente di eliminare i prototipi delle regole LLD.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei prototipi di regola LLD da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di regole LLD eliminati nella proprietà `ruleids`.

Esempi

Eliminazione di più prototipi di regola LLD

Eliminare due prototipi di regola LLD.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

Sorgente

CDiscoveryRulePrototype::delete() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

Descrizione

`integer/array discoveryruleprototype.get(object parameters)`

Il metodo consente di recuperare i prototipi delle regole LLD in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

(`object`) Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
discoveryids	ID/array	Restituisce solo i prototipi di regole LLD che appartengono alle regole LLD o ai prototipi di regole LLD specificati.
itemids	ID/array	Restituisce solo i prototipi di regole LLD con gli ID specificati.
groupids	ID/array	Restituisce solo i prototipi di regole LLD che appartengono agli host dei gruppi specificati.
hostids	ID/array	Restituisce solo i prototipi di regole LLD che appartengono agli host specificati.
inherited	boolean	Se impostato su `true`, restituisce solo i prototipi di regole LLD ereditati da un template.
interfaceids	ID/array	Restituisce solo i prototipi di regole LLD che utilizzano le interfacce host specificate.
monitored	boolean	Se impostato su `true`, restituisce solo i prototipi di regole LLD abilitati che appartengono a host monitorati.
templated	boolean	Se impostato su `true`, restituisce solo i prototipi di regole LLD che appartengono ai template.
templateids	ID/array	Restituisce solo i prototipi di regole LLD che appartengono ai template specificati.
selectDiscoveryRule	query	Restituisce una proprietà `discoveryRule` con la regola LLD padre del prototipo di regola LLD. Restituirà una risposta vuota se il padre è un prototipo di regola di discovery.
selectDiscoveryRulePrototype	query	Restituisce una proprietà `discoveryRulePrototype` con il prototipo di regola LLD padre del prototipo di regola LLD. Restituirà una risposta vuota se il padre è una regola di discovery (non un prototipo).
selectDiscoveryRulePrototypes	query	Restituisce una proprietà `discoveryRulePrototypes` con i prototipi di regole LLD figli che appartengono al prototipo di regola LLD. Supporta `count`.
selectFilter	query	Restituisce una proprietà `filter` con i dati del filtro utilizzato dal prototipo di regola LLD.
selectGraphs	query	Restituisce una proprietà `graphs` con i prototipi di grafico che appartengono al prototipo di regola LLD. Supporta `count`.
selectHostPrototypes	query	Restituisce una proprietà `hostPrototypes` con i prototipi di host che appartengono al prototipo di regola LLD. Supporta `count`.
selectHosts	query	Restituisce una proprietà `hosts` con un array di host a cui appartiene il prototipo di regola LLD.
selectItems	query	Restituisce una proprietà `items` con i prototipi di item che appartengono al prototipo di regola LLD. Supporta `count`.
selectTriggers	query	Restituisce una proprietà `triggers` con i prototipi di trigger che appartengono al prototipo di regola LLD. Supporta `count`.
selectLLDMacroPaths	query	Restituisce una proprietà `lld_macro_paths` con un elenco di macro LLD e dei percorsi ai valori assegnati a ciascuna macro corrispondente.
selectPreprocessing	query	Restituisce una proprietà `preprocessing` con le opzioni di preprocessing del prototipo di regola LLD.
selectOverrides	query	Restituisce una proprietà `lld_rule_overrides` con un elenco di filtri di override, condizioni e operazioni eseguite sugli oggetti prototipo.

Parametro	Type	Descrizione
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text`. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene il prototipo di regola LLD.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` restituisce uno dei seguenti valori:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei prototipi di regola di discovery da un host

Recupera tutti i prototipi di regola di discovery per uno specifico ID host.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Individua i tablespace per {#DB}",
```

```
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Vedi anche

- Prototipo di grafico
- Host
- Prototipo di item
- Filtro del prototipo di regola LLD
- Prototipo di trigger

Sorgente

CDiscoveryRulePrototype::get() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

Descrizione

`object discoveryruleprototype.update(object/array lldRules)`

Questo metodo consente di aggiornare i prototipi di regola LLD esistenti.

Si noti che l'aggiornamento dei prototipi già scoperti è limitato.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object/array)` Proprietà dei prototipi di regola LLD da aggiornare.

La proprietà `itemid` deve essere definita per ciascun prototipo di regola LLD, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del prototipo di regola LLD, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
filter	object	Filtro del prototipo di regola LLD per sostituire il filtro esistente.
preprocessing	object/array	Opzioni di preprocessing del prototipo di regola LLD per sostituire le opzioni di preprocessing esistenti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
lld_macro_paths	object/array	Opzioni lld_macro_path del prototipo di regola LLD per sostituire le opzioni lld_macro_path esistenti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
overrides	object/array	Opzioni di override del prototipo di regola LLD per sostituire le opzioni di override esistenti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di regola LLD aggiornati nella proprietà `itemids`.

Esempi

Aggiornamento delle opzioni di preprocessing del prototipo di regola LLD

Aggiornare un prototipo di regola LLD con una regola di preprocessing JSONPath. Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

Sorgente

CDiscoveryRulePrototype::update() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

Prototipo di trigger

Questa classe è progettata per funzionare con i prototipi di trigger.

Riferimenti agli oggetti:

- Prototipo di trigger
- Tag del prototipo di trigger

Metodi disponibili:

- triggerprototype.create - crea nuovi prototipi di trigger
- triggerprototype.delete - elimina prototipi di trigger
- triggerprototype.get - recupera prototipi di trigger
- triggerprototype.update - aggiorna prototipi di trigger

Oggetto prototipo di trigger

I seguenti oggetti sono direttamente correlati all'API `triggerprototype`.

Prototipo di trigger

L'oggetto prototipo di trigger ha le seguenti proprietà.

Property	Type	Description
triggerid	ID	ID del prototipo di trigger. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
description	string	Nome del prototipo di trigger. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
expression	string	Espressione del trigger. Deve contenere almeno un prototipo di item. I prototipi di item con `value_type` impostato a 5 (binario) o 6 (JSON) non possono essere utilizzati nelle espressioni del trigger. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
event_name	string	Nome dell'evento generato dal trigger.
opdata	string	Dati operativi.
comments	string	Commenti aggiuntivi al prototipo di trigger.

Property	Type	Description
priority	integer	Gravità del prototipo di trigger. Valori possibili: 0 - *(predefinito)* non classificato; 1 - informazione; 2 - avviso; 3 - media; 4 - alta; 5 - disastro.
status	integer	Indica se il prototipo di trigger è abilitato o disabilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.
flags	integer	Origine del prototipo di trigger. Valori possibili: 2 - un prototipo di trigger; 6 - un prototipo di trigger rilevato automaticamente. Comportamento della proprietà: - *sola lettura*
templateid	ID	ID del prototipo di trigger del template padre. Comportamento della proprietà: - *sola lettura*
type	integer	Indica se il prototipo di trigger può generare più eventi di problema. Valori possibili: 0 - *(predefinito)* non generare più eventi; 1 - generare più eventi.
url	string	URL associato al prototipo di trigger.
url_name	string	Etichetta per l'URL associato al prototipo di trigger.
recovery_mode	integer	Modalità di generazione dell'evento OK. Valori possibili: 0 - *(predefinito)* Espressione; 1 - Espressione di ripristino; 2 - Nessuno.
recovery_expression	string	Espressione di ripristino del trigger. Deve contenere almeno un prototipo di item. I prototipi di item con `value_type` impostato a 5 (binario) o 6 (JSON) non possono essere utilizzati nelle espressioni di ripristino del trigger.
correlation_mode	integer	Chiusura dell'evento OK. Valori possibili: 0 - *(predefinito)* Tutti i problemi; 1 - Tutti i problemi se i valori dei tag corrispondono.
correlation_tag	string	Tag per la corrispondenza.
manual_close	integer	Consenti chiusura manuale. Valori possibili: 0 - *(predefinito)* No; 1 - Sì.
discover	integer	Stato di individuazione del prototipo di trigger. Valori possibili: 0 - *(predefinito)* verranno individuati nuovi trigger; 1 - non verranno individuati nuovi trigger e i trigger esistenti verranno contrassegnati come persi.

Property	Type	Description
uuid	string	Identificatore univoco universale, utilizzato per collegare i prototipi di trigger importati a quelli già esistenti. Generato automaticamente, se non specificato. Comportamento della proprietà: - *supportata* se il prototipo di trigger appartiene a un template

Tag del prototipo di trigger

L'oggetto tag del prototipo di trigger ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tag	string	Nome del tag del prototipo di trigger. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del tag del prototipo di trigger. Comportamento della proprietà: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template; 1 - host; 3 - item; 4 - prototipo di item. Comportamento della proprietà: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.

triggerprototype.create

Descrizione

`object triggerprototype.create(object/array triggerPrototypes)`

Questo metodo consente di creare nuovi trigger prototype.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Prototipi di trigger da creare.

Oltre alle proprietà standard del prototipo di trigger, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
dependencies	array	Trigger e prototipi di trigger da cui dipende il prototipo di trigger. I trigger devono avere definita solo la proprietà `triggerid`.
tags	array	Tag del prototipo di trigger.

Valori di ritorno

(object) Restituisce un oggetto contenente gli ID dei prototipi di trigger creati sotto la proprietà triggerids. L'ordine degli ID restituiti corrisponde all'ordine dei prototipi di trigger passati.

Esempi

Creazione di un trigger prototype

Creare un trigger prototype per rilevare quando un file system ha meno del 20% di spazio libero su disco.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "Lo spazio libero su disco è inferiore al 20% sul volume {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

Fonte

CTriggerPrototype::create() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

Descrizione

`object triggerprototype.delete(array triggerPrototypeIds)`

Questo metodo consente di eliminare i trigger prototype.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

ID (array) dei prototipi di trigger da eliminare.

Valori di ritorno

(object) Restituisce un oggetto contenente gli ID dei prototipi di trigger eliminati sotto la proprietà triggerids.

Esempi

Eliminazione di più prototipi di trigger

Eliminare due prototipi di trigger.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Fonte

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

Descrizione

`integer/array triggerprototype.get(object parameters)`

Il metodo consente di recuperare i trigger prototype in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
active	flag	Restituisce solo i prototipi di trigger abilitati che appartengono a host monitorati.
discoveryids	ID/array	Restituisce solo i prototipi di trigger che appartengono alle regole LLD specificate.
functions	string/array	Restituisce solo i trigger che utilizzano le funzioni specificate. Fare riferimento alla pagina Funzioni supportate per un elenco delle funzioni supportate.
group	string	Restituisce solo i prototipi di trigger che appartengono a host o template dei gruppi host o gruppi di template con il nome specificato.
groupids	ID/array	Restituisce solo i prototipi di trigger che appartengono a host o template dei gruppi host o gruppi di template specificati.
host	string	Restituisce solo i prototipi di trigger che appartengono a host con il nome specificato.
hostids	ID/array	Restituisce solo i prototipi di trigger che appartengono agli host specificati.

Parametro	Tipo	Descrizione
inherited	boolean	Se impostato su `true`, restituisce solo i prototipi di trigger ereditati da un template.
maintenance	boolean	Se impostato su `true`, restituisce solo i prototipi di trigger abilitati che appartengono a host in manutenzione.
min_severity	integer	Restituisce solo i prototipi di trigger con gravità maggiore o uguale a quella specificata.
monitored	flag	Restituisce solo i prototipi di trigger abilitati che appartengono a host monitorati e contengono solo item abilitati.
templated	boolean	Se impostato su `true`, restituisce solo i prototipi di trigger che appartengono a template.
templateids	ID/array	Restituisce solo i prototipi di trigger che appartengono ai template specificati.
triggerids	ID/array	Restituisce solo i prototipi di trigger con gli ID specificati.
expandExpression	flag	Espande funzioni e macro nell'espressione del trigger.
selectDependencies	query	Restituisce i prototipi di trigger e i trigger da cui il prototipo di trigger dipende nella proprietà `dependencies`.
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery del prototipo di trigger. L'oggetto di discovery del prototipo di trigger collega un prototipo di trigger rilevato a un prototipo di trigger da cui è stato rilevato. Ha le seguenti proprietà: `parent_triggerid` - `(ID)` ID del prototipo di trigger da cui è stato creato il prototipo di trigger; `status` - `(int)` stato di discovery del prototipo di trigger: 0 - *(predefinito)* il prototipo di trigger è rilevato, 1 - il prototipo di trigger non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un prototipo di trigger non più rilevato verrà eliminato; `ts_disable` - `(timestamp)` momento in cui un prototipo di trigger non più rilevato verrà disabilitato; `disable_source` - `(int)` indicatore che specifica se il prototipo di trigger è stato disabilitato da una regola LLD o manualmente: 0 - *(predefinito)* disabilitato automaticamente, 1 - disabilitato da una regola LLD.
selectDiscoveryRule	query	Restituisce la regola LLD a cui appartiene il prototipo di trigger nella proprietà `discoveryRule`.
selectDiscoveryRulePrototype	query	Restituisce una proprietà `discoveryRulePrototype` con il prototipo di regola LLD padre a cui appartiene il prototipo di trigger.
selectFunctions	query	Restituisce le funzioni utilizzate nel prototipo di trigger nella proprietà `functions`. Gli oggetti funzione rappresentano le funzioni utilizzate nell'espressione del trigger e hanno le seguenti proprietà: `functionid` - `(ID)` ID della funzione; `itemid` - `(ID)` ID dell'item utilizzato nella funzione; `function` - `(string)` nome della funzione; `parameter` - `(string)` parametro passato alla funzione. Il parametro della query viene sostituito dal simbolo $ nella stringa restituita.
selectHostGroups	query	Restituisce i gruppi host a cui appartiene il prototipo di trigger nella proprietà `hostgroups`.
selectHosts	query	Restituisce gli host a cui appartiene il prototipo di trigger nella proprietà `hosts`.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag ereditati da template, template collegati e host, e dai prototipi di item referenziati nelle espressioni di trigger o di ripristino.
selectItems	query	Restituisce gli item e i prototipi di item utilizzati dal prototipo di trigger nella proprietà `items`.
selectTags	query	Restituisce i tag del prototipo di trigger nella proprietà `tags`.
selectTemplateGroups	query	Restituisce i gruppi di template a cui appartiene il prototipo di trigger nella proprietà `templategroups`.

Parametro	Tipo	Descrizione
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text`. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene il prototipo di trigger; `hostid` - ID dell'host a cui appartiene il prototipo di trigger.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectHosts` - i risultati saranno ordinati per `host`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti valori:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recuperare i prototipi di trigger da una regola LLD

Recupera tutti i prototipi di trigger e le relative funzioni da una regola LLD.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
```

```
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "triggerid": "13272",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
```

```
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Recupero di uno specifico prototipo di trigger con tag

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Lo spazio libero su disco è inferiore al 20% sul volume {#FSNAME}",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

Vedi anche

- Regola di individuazione
- Item
- Host
- Gruppo di host
- Gruppo di template

Fonte

CTriggerPrototype::get() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.update

Descrizione

`object triggerprototype.update(object/array triggerPrototypes)`

Questo metodo consente di aggiornare i trigger prototype esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dei prototipi di trigger da aggiornare.

La proprietà `triggerid` deve essere definita per ogni prototipo di trigger, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del prototipo di trigger, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
dependencies	array	Trigger e prototipi di trigger da cui dipende il prototipo di trigger. I trigger devono avere definita solo la proprietà `triggerid`.
tags	array	Tag del prototipo di trigger.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei prototipi di trigger aggiornati sotto la proprietà `triggerids`.

Esempi

Abilitazione di un trigger prototype

Abilitare un trigger prototype, cioè impostare il suo stato su "0".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Sostituzione dei tag del prototipo di trigger

Sostituisce i tag per un prototipo di trigger.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
```

```
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Fonte

CTriggerPrototype::update() in *ui/include/classes/api/services/CTriggerPrototype.php*.

Proxy

Questa classe è progettata per lavorare con i proxy.

Riferimenti oggetto:

- Proxy

Metodi disponibili:

- proxy.create - crea nuovi proxy
- proxy.delete - elimina i proxy
- proxy.get - recupera i proxy
- proxy.update - aggiorna i proxy

Oggetto proxy

I seguenti oggetti sono direttamente correlati all'API `proxy`.

Proxy

L'oggetto proxy presenta le seguenti proprietà.

Property	Type	Description
proxyid	ID	ID del proxy. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del proxy. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Property	Type	Description
proxy_groupid	ID	ID del gruppo di proxy. 0 - il proxy non è assegnato ad alcun gruppo di proxy.
local_address	string	Indirizzo per gli agent attivi. Indirizzo IP o nome DNS a cui connettersi. Comportamento della proprietà: - *obbligatorio* se `proxy_groupid` non è 0
local_port	string	Numero della porta locale del proxy a cui connettersi. Le macro utente sono supportate. Predefinito: 10051. Comportamento della proprietà: - *supportato* se `proxy_groupid` non è 0
operating_mode	integer	Tipo di proxy. Valori possibili: 0 - proxy attivo; 1 - proxy passivo. Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione
description	text	Descrizione del proxy.
lastaccess	timestamp	Ora dell'ultima connessione del proxy al server. Comportamento della proprietà: - *sola lettura*
address	string	Indirizzo IP o nome DNS a cui connettersi. Le macro utente sono supportate. Comportamento della proprietà: - *obbligatorio* se la modalità operativa del proxy Zabbix è passiva
port	string	Numero della porta a cui connettersi. Le macro utente sono supportate. Predefinito: 10051. Comportamento della proprietà: - *supportato* se la modalità operativa del proxy Zabbix è passiva
allowed_addresses	string	Indirizzi IP o nomi DNS del proxy Zabbix attivo, delimitati da virgole.
tls_connect	integer	Connessioni al proxy. Valori possibili: 1 - *(predefinito)* Nessuna crittografia; 2 - PSK; 4 - certificato.
tls_accept	integer	Connessioni dal proxy. Valori bitmap possibili: 1 - *(predefinito)* Nessuna crittografia; 2 - PSK; 4 - certificato. Questo è un campo con maschera di bit; è accettata qualsiasi somma dei valori bitmap possibili (ad esempio, 6 per PSK e certificato).
tls_issuer	string	Emittente del certificato.
tls_subject	string	Oggetto del certificato.

Property	Type	Description
tls_psk_identity	string	Identità PSK; deve essere associata a una sola PSK (tra autoregistrazione, host e proxy). Non includere informazioni sensibili nell'identità PSK, poiché viene inviata non crittografata sulla rete per informare il destinatario quale PSK utilizzare. Comportamento della proprietà: - *sola scrittura* - *obbligatoria* se `tls_connect` è impostato su "PSK" oppure se `tls_accept` contiene il bit "PSK"
tls_psk	string	Chiave precondivisa (PSK); deve contenere almeno 32 cifre esadecimali. Comportamento della proprietà: - *sola scrittura* - *obbligatoria* se `tls_connect` è impostato su "PSK" oppure se `tls_accept` contiene il bit "PSK"
custom_timeouts	integer	Indica se sovrascrivere i timeout globali degli item a livello di proxy. Valori possibili: 0 - *(predefinito)* usa le impostazioni globali; 1 - sovrascrive i timeout.
timeout_zabbix_agent	string	Non dedicare più di `timeout_*` secondi all'elaborazione. Accetta un numero di secondi (ad esempio 1, 60, 300) oppure un valore con un suffisso temporale (ad esempio `30s`, `1m`). Accetta anche macro utente. Utilizzato solo se `custom_timeouts` è impostato su "override timeouts"; in caso contrario, il valore viene ignorato. Intervallo di valori possibile: 1-600s. Comportamento della proprietà: - *obbligatorio* se `custom_timeouts` è impostato su "override timeouts"
timeout_simple_check timeout_snmp_agent timeout_external_check timeout_db_monitor timeout_http_agent timeout_ssh_agent timeout_telnet_agent timeout_script timeout_browser timeout_telemetry_query		
version	integer	Versione del proxy. Numero di versione Zabbix composto da tre parti, in cui per ciascuna parte vengono utilizzate due cifre decimali, ad esempio 60401 per la versione 6.4.1, 70002 per la versione 7.0.2 e così via. 0 - Versione del proxy sconosciuta. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
compatibility	integer	Versione del proxy rispetto alla versione del server Zabbix. Valori possibili: 0 - Non definita; 1 - Versione corrente (proxy e server hanno la stessa versione principale); 2 - Versione obsoleta (la versione del proxy è precedente a quella del server, ma è parzialmente supportata); 3 - Versione non supportata (la versione del proxy è precedente alla versione della release LTS precedente del server oppure la versione principale del server è precedente a quella del proxy). Comportamento della proprietà: - *sola lettura*
state	integer	Stato del proxy. Valori possibili: 0 - Sconosciuto; 1 - Offline; 2 - Online. Comportamento della proprietà: - *sola lettura*

Monitoraggio delle prestazioni delle applicazioni (APM)

L'oggetto APM ha le seguenti proprietà.

Property	Type	Description
data_collection_status	integer	Indica se la raccolta dei dati di telemetria è abilitata sul proxy. Valori possibili: 0 - *(predefinito)* Disabilitata; 1 - Abilitata.
max_messages_per_second	integer	Numero massimo di messaggi al secondo accettati dal proxy. Intervallo di valori possibili: 0 (illimitato) – 2147483647. Predefinito: 0. Comportamento della proprietà: - *supportata* se `data_collection_status` è impostato su "Enabled"
additional_resource_attributes	array	Array di attributi delle risorse da aggiungere ai dati di telemetria in ingresso. Se i dati in ingresso contengono già un attributo della risorsa con lo stesso nome, viene mantenuto l'attributo in ingresso invece di quello specificato qui. Comportamento della proprietà: - *supportata* se `data_collection_status` è impostato su "Enabled"

Attributi della risorsa

L'oggetto degli attributi della risorsa ha le seguenti proprietà.

Property	Type	Description
key	string	Nome dell'attributo della risorsa. Deve essere univoco per il `signal_type` specificato. Comportamento della proprietà: - *obbligatorio*
value	string	Valore dell'attributo della risorsa. Può essere vuoto.
signal_type	integer	Tipo di dati di telemetria a cui si applica l'attributo. Valori possibili: 0 - Tracce; 1 - Metriche; 2 - Log. Comportamento della proprietà: - *obbligatorio*

proxy.create

Descrizione

`object proxy.create(object/array proxies)`

Questo metodo consente di creare nuovi proxy.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proxy da creare.

Oltre alle proprietà standard del proxy, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
hosts	array	Host da monitorare tramite il proxy. Se un host è già monitorato da un proxy diverso, verrà riassegnato al proxy corrente. Gli host devono avere definita solo la proprietà `hostid`.
apm	object	Impostazioni di Application performance monitoring (APM) del proxy.

Restituisce i valori

"(object)" Restituisce un oggetto contenente gli ID dei proxy creati nella proprietà "proxyids". L'ordine del gli ID restituiti corrispondono all'ordine dei proxy passati.

Esempi

Creare un proxy attivo

Creare un proxy di azione "Active proxy" e assegnargli un host da monitorare.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
```

```
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

Creare un proxy passivo

Creare un proxy passivo ”Passive proxy” e assegnargli due host da monitorare.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
        ]
    },
    "id": 1
}
```

Creazione di un proxy e aggiunta a un gruppo di proxy

Creare un proxy attivo ”Active proxy” e aggiungerlo al gruppo di proxy con ID ”1”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
```

```
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Creazione di un proxy con crittografia basata su certificato

Creare un proxy attivo con crittografia basata su certificato. Il proxy è configurato per monitorare l'host "10785".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
    },
    "id": 1
}
```

Creazione di un proxy con APM abilitato

Creare un proxy attivo "Active proxy" con APM abilitato.
Il proxy accetta fino a 1000 messaggi al secondo, aggiunge l'attributo della risorsa `deployment.environment` alle tracce e aggiunge l'attributo della risorsa `cloud.region` alle metriche.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
```

```
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
            "max_messages_per_second": "1000",
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Host
- Gruppo di proxy

Fonte

CProxy::create() in *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Descrizione

`object proxy.delete(array proxies)`

Questo metodo consente di eliminare i proxy.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei proxy da eliminare.

Valori restituiti

”(object)” Restituisce un oggetto contenente gli ID dei proxy eliminati nella proprietà ”proxyids”.

Esempi

Eliminare più proxy

Eliminare due proxy.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

Fonte

CProxy::delete() in *ui/include/classes/api/services/CProxy.php*.

proxy.get

Descrizione

`integer/array proxy.get(object parameters)`

Il metodo consente di recuperare i proxy in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.
 Vengono restituiti solo i proxy consentiti dall'*Elenco di accesso ai proxy* dei gruppi di utenti dell'utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
proxyids	ID/array	Restituisce solo i proxy con gli ID specificati.
proxy_groupids	ID/array	Restituisce solo i proxy che appartengono ai proxy group specificati.
selectApm	query	Restituisce una proprietà `apm` con le impostazioni di application performance monitoring (APM) del proxy.
selectAssignedHosts	query	Restituisce una proprietà `assignedHosts` con gli host assegnati al proxy. Supporta `count`.
selectHosts	query	Restituisce una proprietà `hosts` con gli host monitorati dal proxy. Supporta `count`.
selectProxyGroup	query	Restituisce una proprietà `proxyGroup` con l'oggetto proxy group.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Questi parametri sono descritti nella reference commentary.
editable	boolean	
excludeSearch	boolean	

Parametro	Tipo	Descrizione
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recuperare tutti i proxy

Recupera tutti i proxy configurati e le relative interfacce.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Host
- Gruppo di proxy

Fonte

CProxy::get() in *ui/include/classes/api/services/CProxy.php*.

proxy.update

Descrizione

`object proxy.update(object/array proxies)`

Questo metodo consente di aggiornare proxy esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà del proxy da aggiornare.

La proprietà `proxyid` deve essere definita per ogni proxy; tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, mentre tutte le altre rimarranno invariate.

Oltre alle proprietà standard del proxy, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
hosts	array	Host da monitorare tramite il proxy. Se un host è già monitorato da un proxy diverso, verrà riassegnato al proxy corrente. Gli host devono avere definita solo la proprietà `hostid`.
apm	object	Impostazioni di Application performance monitoring (APM) del proxy.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei proxy aggiornati nella proprietà `proxyids`.

Esempi

Modificare gli host monitorati da un proxy

Aggiorna il proxy affinché monitori i due host specificati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Modificare lo stato del proxy

Modificare il proxy in un proxy attivo e rinominarlo in ”Active proxy”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Aggiungere un proxy a un gruppo di proxy

Aggiornare il proxy con ID ”5” e aggiungerlo al gruppo di proxy con ID ”1”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Vedi anche

- Host
- Gruppo di proxy

Fonte

CProxy::update() in *ui/include/classes/api/services/CProxy.php*.

Registrazione automatica

Questa classe è progettata per lavorare con la registrazione automatica.

Riferimenti oggetto:

- Registrazione automatica

Metodi disponibili:

- autoregistration.get - recupera la registrazione automatica
- autoregistration.update - aggiorna la registrazione automatica

Oggetto di registrazione automatica

I seguenti oggetti sono direttamente correlati all’API `autoregistration`.

Autoregistrazione

L’oggetto di autoregistrazione ha le seguenti proprietà.

Proprietà	Type	Descrizione
tls_accept	integer	Tipo di connessioni in ingresso consentite per l'autoregistrazione. Valori possibili: 1 - consenti connessioni non crittografate; 2 - consenti TLS con PSK; 3 - consenti sia connessioni non crittografate sia connessioni TLS con PSK.
tls_psk_identity	string	Identità PSK; deve essere associata a un solo PSK (tra autoregistration, hosts e proxies). Non includere informazioni sensibili nell'identità PSK, poiché viene inviata in chiaro sulla rete per informare il destinatario su quale PSK utilizzare. Comportamento della proprietà: - *sola scrittura*
tls_psk	string	Chiave precondivisa (PSK); deve contenere almeno 32 cifre esadecimali. Comportamento della proprietà: - *sola scrittura*

autoregistration.get

Descrizione

`object autoregistration.get(object parameters)`

Il metodo consente di recuperare un oggetto di autoregistrazione in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta un solo parametro.

Parametro	Type	Descrizione
output	query	Questo parametro è descritto nel commento di riferimento.

Valori restituiti

`(object)` Restituisce l'oggetto di registrazione automatica.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tls_accept": "3"
    },
    "id": 1
}
```

Sorgente

CAutoregistration::get() in *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

Descrizione

`object autoregistration.update(object autoregistration)`

Questo metodo consente di aggiornare un'autoregistrazione esistente.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Proprietà della registrazione automatica da aggiornare.

Valori restituiti

`(boolean )` Restituisce il valore booleano vero come risultato in caso di aggiornamento riuscito.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Sorgente

CAutoregistration::update() in *ui/include/classes/api/services/CAutoregistration.php*.

Regola di individuazione

Questa classe è progettata per funzionare con le regole di individuazione di rete.

> **Note:**
> Questa API è pensata per funzionare con le regole di individuazione di rete. Per le regole di individuazione di basso livello, vedere API delle regole LLD.

Riferimenti agli oggetti:

- Regola di individuazione

Metodi disponibili:

- drule.create - crea nuove regole di individuazione di rete
- drule.delete - elimina regole di individuazione di rete
- drule.get - recupera regole di individuazione di rete
- drule.update - aggiorna regole di individuazione di rete

Oggetto regola di discovery

I seguenti oggetti sono direttamente correlati all'API `drule`.

Regola di discovery

L'oggetto della regola di discovery definisce una regola di discovery di rete. Presenta le seguenti proprietà.

Property	Type	Description
druleid	ID	ID della regola di discovery. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
iprange	string	Uno o più intervalli IP da controllare, separati da virgole. Per ulteriori informazioni sui formati supportati degli intervalli IP, consultare la sezione configurazione della discovery di rete. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
name	string	Nome della regola di discovery. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
delay	string	Intervallo di esecuzione della regola di discovery. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`), un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`) o una macro utente. Predefinito: `1h`.
proxyid	ID	ID del proxy utilizzato per la discovery.
status	integer	Indica se la regola di discovery è abilitata. Valori possibili: 0 - *(predefinito)* abilitata; 1 - disabilitata.
concurrency_max	integer	Numero massimo di controlli simultanei per la regola di discovery. Valori possibili: 0 - *(predefinito)* numero illimitato di controlli; 1 - un controllo; 2-999 - numero personalizzato di controlli.
error	string	Testo dell'errore se si sono verificati problemi durante l'esecuzione della regola di discovery. Comportamento della proprietà: - *sola lettura*

drule.create

Descrizione

`object drule.create(object/array discoveryRules)`

Questo metodo consente di creare nuove regole di discovery.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Regole di discovery da creare.

Oltre alle proprietà standard delle regole di discovery, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
dchecks	array	Controlli di discovery da creare per la regola di discovery. Comportamento del parametro: - *obbligatorio*

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle regole di scoperta create sotto la proprietà `druleids`. L'ordine degli ID restituiti corrispondono all'ordine delle regole di rilevamento passate.

Esempi

Creare una regola di individuazione

Creare una regola di individuazione per trovare le macchine che eseguono lo Zabbix agent nella rete locale. La regola deve utilizzare un singolo controllo Zabbix agent sulla porta 10050.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Vedi anche

- Controllo di discovery

Fonte

CDRule::create() in *ui/include/classes/api/services/CDRule.php*.

drule.delete

Descrizione

`object drule.delete(array discoveryRuleIds)`

Questo metodo consente di eliminare le regole di discovery.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID delle regole di rilevamento da eliminare.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle regole di discovery eliminata sotto la proprietà `druleids`.

Esempi

Eliminare più regole di individuazione

Eliminare due regole di individuazione.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CDRule::delete() in *ui/include/classes/api/services/CDRule.php*.

drule.get

Descrizione

`integer/array drule.get(object parameters)`

Il metodo consente di recuperare le discovery rule in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
dhostids	ID/array	Restituisce solo le regole di discovery che hanno creato i discovered host specificati.
druleids	ID/array	Restituisce solo le regole di discovery con gli ID specificati.
dserviceids	ID/array	Restituisce solo le regole di discovery che hanno creato i discovered service specificati.
selectDChecks	query	Restituisce una proprietà `dchecks` con i controlli di discovery utilizzati dalla regola di discovery. Supporta `count`.
selectDHosts	query	Restituisce una proprietà `dhosts` con i discovered host creati dalla regola di discovery. Supporta `count`.
limitSelects	integer	Limita il numero di record restituiti dalle sottoquery. Si applica alle seguenti sottoquery: `selectDChecks` - i risultati saranno ordinati per `dcheckid`; `selectDHosts` - i risultati saranno ordinati per `dhostsid`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `druleid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare tutte le regole di discovery

Recupera tutte le regole di discovery configurate e i controlli di discovery che utilizzano.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
```

```
        "selectDChecks": "extend"
    },
    "id": 1
}
```

Risposta:


```
{
    "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
}
```


Vedi anche

- Host rilevato
- Controllo di discovery

Fonte

CDRule::get() in *ui/include/classes/api/services/CDRule.php*.

drule.update

Descrizione

`object drule.update(object/array discoveryRules)`

Questo metodo consente di aggiornare le regole di discovery esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà della regola di discovery da aggiornare.

La proprietà `druleid` deve essere definita per ogni regola di discovery, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard della regola di discovery, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
dchecks	array	Controlli di discovery per sostituire i controlli esistenti.

Restituisce i valori

(object) Restituisce un oggetto contenente gli ID delle regole di rilevamento aggiornate sotto la proprietà druleids.

Esempi

Modificare l'intervallo IP di una regola di discovery

Modificare l'intervallo IP di una regola di discovery in "192.168.2.1-255".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Vedi anche

- Controllo di discovery

Fonte

CDRule::update() in *ui/include/classes/api/services/CDRule.php*.

Regola LLD

Questa classe è progettata per funzionare con le regole di low-level discovery.

Riferimenti agli oggetti:

- Regola LLD
  - Header HTTP
  - Campo query HTTP
- Filtro regola LLD
  - Condizione del filtro della regola LLD
- Percorso macro LLD
- Preelaborazione della regola LLD
- Override della regola LLD
  - Filtro override della regola LLD
    - Condizione del filtro override della regola LLD
  - Operazione override della regola LLD
    - Stato dell'operazione override della regola LLD
    - Rilevamento dell'operazione override della regola LLD
    - Periodo dell'operazione override della regola LLD
    - Storico dell'operazione override della regola LLD
    - Trend dell'operazione override della regola LLD
    - Gravità dell'operazione override della regola LLD
    - Tag dell'operazione override della regola LLD
    - template dell'operazione override della regola LLD
    - Inventario dell'operazione override della regola LLD

Metodi disponibili:

- discoveryrule.create - crea nuove regole LLD
- discoveryrule.delete - elimina regole LLD
- discoveryrule.get - recupera regole LLD
- discoveryrule.update - aggiorna regole LLD

Oggetto regola LLD

I seguenti oggetti sono direttamente correlati all'API `discoveryrule`.

Regola LLD

L'oggetto regola di discovery di basso livello presenta le seguenti proprietà.

Property	Type	Description
itemid	ID	ID della regola LLD. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
delay	string	Intervallo di aggiornamento della regola LLD. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`), un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`) oppure, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e di pianificazione. Accetta macro utente. Se utilizzato, il valore deve essere una singola macro. Non sono supportate più macro o macro combinate con testo. Gli intervalli flessibili possono essere scritti come due macro separate da una barra (ad esempio, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Esempio: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "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), oppure se `type` è impostato su "Zabbix agent (active)" (7) e `key_` non contiene "mqtt.get"
hostid	ID	ID dell'host a cui appartiene la regola LLD. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
flags	integer	Origine della regola di discovery. Valori possibili: 1 - una regola di discovery di basso livello; 5 - una regola di discovery di basso livello convertita da un prototipo. Comportamento della proprietà: - *sola lettura*
interfaceid	ID	ID dell'interfaccia dell'host della regola LLD. Comportamento della proprietà: - *obbligatoria* se la regola LLD appartiene a un host e `type` è impostato su "Zabbix agent", "IPMI agent", "JMX agent" o "SNMP agent" - *supportata* se la regola LLD appartiene a un host e `type` è impostato su "Simple check", "External check", "SSH agent", "TELNET agent" o "HTTP agent"

Property	Type	Description
key_	string	Chiave della regola LLD. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
name	string	Nome della regola LLD. Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
type	integer	Tipo della regola LLD. Valori possibili: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Questo tipo è consentito solo se il parent della regola è un template (che si presume sia collegato a un prototipo di host durante la discovery) o un host rilevato (che ha una regola LLD alla propria radice). Comportamento della proprietà: - *obbligatorio* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
url	string	Stringa URL. Supporta macro utente, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
allow_traps	integer	Consente di popolare il valore in modo analogo all’item trapper. Valori possibili: 0 - *(predefinito)* Non accettare dati in ingresso; 1 - Accettare dati in ingresso. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent”

Property	Type	Description
authtype	integer	Metodo di autenticazione. Valori possibili se `type` è impostato su ”SSH agent”: 0 - *(predefinito)* password; 1 - chiave pubblica. Valori possibili se `type` è impostato su ”HTTP agent”: 0 - *(predefinito)* nessuno; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SSH agent” o ”HTTP agent” - *sola lettura* per gli oggetti ereditati (se `type` è impostato su ”HTTP agent”)
description	string	Descrizione della regola LLD.
error	string	Testo dell’errore in caso di problemi durante l’aggiornamento del valore della regola LLD. Comportamento della proprietà: - *sola lettura*
follow_redirects	integer	Segue i reindirizzamenti della risposta durante il polling dei dati. Valori possibili: 0 - Non seguire i reindirizzamenti; 1 - *(predefinito)* Seguire i reindirizzamenti. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
headers	array	Array di intestazioni da inviare durante l’esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
http_proxy	string	Stringa di connessione proxy HTTP(S). Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ipmi_sensor	string	Sensore IPMI. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”IPMI agent” e `key_` non è impostato su ”ipmi.get” - *supportata* se `type` è impostato su ”IPMI agent” - *sola lettura* per gli oggetti ereditati
jmx_endpoint	string	Stringa di connessione personalizzata dell’agent JMX. Predefinito: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento della proprietà: - *supportata* se `type` è impostato su ”JMX agent”

Property	Type	Description
lifetime	string	Periodo di tempo dopo il quale gli item non più rilevati verranno eliminati. Accetta un numero di secondi (ad esempio, 3600), un valore con un suffisso temporale (ad esempio, 3600s, 180m, 24h, 1d) oppure una macro utente. Predefinito: 7d.
lifetime_type	integer	Scenario per eliminare le risorse LLD perse. Valori possibili: 0 - *(predefinito)* Eliminare dopo il raggiungimento della soglia di durata; 1 - Non eliminare; 2 - Eliminare immediatamente.
enabled_lifetime	string	Periodo di tempo dopo il quale gli item non più rilevati verranno disabilitati. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d) oppure una macro utente. Predefinito: 0.
enabled_lifetime_type	integer	Scenario per disabilitare le risorse LLD perse. Valori possibili: 0 - Disabilitare dopo il raggiungimento della soglia di durata; 1 - Non disabilitare; 2 - *(predefinito)* Disabilitare immediatamente.
master_itemid	ID	ID dell'item master. La regola di discovery non può essere l'item master di un'altra regola di discovery. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Dependent item" - *sola lettura* per gli oggetti ereditati
output_format	integer	Indica se la risposta deve essere convertita in JSON. Valori possibili: 0 - *(predefinito)* Memorizzare i dati non elaborati; 1 - Convertire in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
params	string	Parametri aggiuntivi in base al tipo di regola LLD: - script eseguito per le regole LLD SSH e Telnet; - query SQL per le regole LLD del monitor database; - formula per le regole LLD calcolate; - script per le regole LLD script e browser. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Database monitor", "SSH agent", "TELNET agent", "Script" o "Browser" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "Script" o "Browser")
parameters	object/array	Parametri aggiuntivi se `type` è impostato su "Script" o "Browser". Array di oggetti con le proprietà `name` e `value`, dove `name` deve essere univoco. Comportamento della proprietà: - *supportata* se `type` è impostato su "Script" o "Browser" - *sola lettura* per gli oggetti ereditati

Property	Type	Description
password	string	Password per l'autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "JMX agent" e `username` è impostato - *supportata* se `type` è impostato su "Simple check", "Database monitor", "SSH agent", "TELNET agent" o "HTTP agent" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "HTTP agent")
post_type	integer	Tipo del corpo dei dati POST memorizzato nella proprietà `posts`. Valori possibili: 0 - *(predefinito)* Dati non elaborati; 2 - Dati JSON; 3 - Dati XML. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
posts	string	Dati del corpo della richiesta HTTP(S). Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "HTTP agent" e `post_type` è impostato su "Dati JSON" o "Dati XML" - *supportata* se `type` è impostato su "HTTP agent" e `post_type` è impostato su "Dati non elaborati" - *sola lettura* per gli oggetti ereditati
privatekey	string	Nome del file della chiave privata. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "SSH agent" e `authtype` è impostato su "chiave pubblica"
publickey	string	Nome del file della chiave pubblica. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "SSH agent" e `authtype` è impostato su "chiave pubblica"
query_fields	array	Array di campi query da inviare durante l'esecuzione di una richiesta HTTP. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
request_method	integer	Tipo di metodo della richiesta. Valori possibili: 0 - *(predefinito)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati

Property	Type	Description
retrieve_mode	integer	Parte della risposta da memorizzare. Valori possibili se `request_method` è impostato su ”GET”, ”POST” o ”PUT”: 0 - *(predefinito)* Corpo; 1 - Intestazioni; 2 - Verranno memorizzati sia il corpo sia le intestazioni. Valori possibili se `request_method` è impostato su ”HEAD”: 1 - Intestazioni. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
snmp_oid	string	OID SNMP. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su ”SNMP agent” - *sola lettura* per gli oggetti ereditati
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ssl_key_file	string	Percorso del file della chiave SSL privata. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
ssl_key_password	string	Password del file della chiave SSL. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
state	integer	Stato della regola LLD. Valori possibili: 0 - *(predefinito)* normale; 1 - non supportata. Comportamento della proprietà: - *sola lettura*
status	integer	Stato della regola LLD. Valori possibili: 0 - *(predefinito)* regola LLD abilitata; 1 - regola LLD disabilitata.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole. Supporta anche macro utente come parte dell’elenco separato da virgole. Esempio: 200,200-{$M},{$M},200-400 Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
templateid	ID	ID della regola LLD del template parent. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
timeout	string	Timeout della richiesta di polling dei dati dell'item. Accetta un numero di secondi (ad esempio, 1, 60, 300) oppure un valore con un suffisso temporale (ad esempio, 30s, 1m). Accetta anche macro utente. L'intervallo dei valori possibili è: 1-600s. Predefinito: "" - utilizza le impostazioni del proxy/globali. Comportamento della proprietà: - *supportata* se `type` è impostato su "Zabbix agent" (0), "Simple check" (3) e `key_` non inizia con "`vmware.`" e "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) e `snmp_oid` inizia con "`walk[`" o "`get[`", "Script" (21), "Browser" (22) - *sola lettura* per gli oggetti ereditati, - se viene applicato il valore predefinito e l'item appartiene a un host monitorato da un proxy o da un gruppo di proxy non accessibile, viene utilizzato il timeout globale.
trapper_hosts	string	Host consentiti. Comportamento della proprietà: - *supportata* se `type` è impostato su "Zabbix trapper", oppure se `type` è impostato su "HTTP agent" e `allow_traps` è impostato su "Accettare dati in ingresso". Se la proprietà viene omessa, viene impostata automaticamente sulla macro utente `{$TRAPPER.ALLOWED_HOSTS}` (valore predefinito). Per impostazione predefinita, `{$TRAPPER.ALLOWED_HOSTS}` è `127.0.0.1,::1`. Se la proprietà è impostata su una stringa vuota, le connessioni in ingresso verranno negate da tutti gli host.
username	string	Nome utente per l'autenticazione. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "SSH agent", "TELNET agent", oppure se `type` è impostato su "JMX agent" e `password` è impostato - *supportato* se `type` è impostato su "Simple check", "Database monitor" o "HTTP agent" - *sola lettura* per gli oggetti ereditati (se `type` è impostato su "HTTP agent")
uuid	string	Identificatore univoco universale, utilizzato per collegare le regole LLD importate a quelle già esistenti. Generato automaticamente se non specificato. Comportamento della proprietà: - *supportato* se la regola LLD appartiene a un template
verify_host	integer	Indica se convalidare che il nome host della connessione corrisponda a quello presente nel certificato dell'host. Valori possibili: 0 - *(predefinito)* Non convalidare; 1 - Convalidare. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati

Property	Type	Description
verify_peer	integer	Indica se convalidare l'autenticità del certificato dell'host. Valori possibili: 0 - *(predefinito)* Non convalidare; 1 - Convalidare. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati

Intestazione HTTP

L'oggetto header ha le seguenti proprietà:

Proprietà	Type	Descrizione
name	string	Nome dell'intestazione HTTP. Comportamento della proprietà: - *obbligatorio*
value	string	Valore dell'intestazione. Comportamento della proprietà: - *obbligatorio*

Campo di query HTTP

L'oggetto campo di query definisce un nome e un valore utilizzati per specificare un parametro URL. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
name	string	Nome del parametro. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro. Comportamento della proprietà: - *obbligatorio*

Filtro della regola LLD

L'oggetto filtro della regola LLD definisce un insieme di condizioni che possono essere utilizzate per filtrare gli oggetti rilevati. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
conditions	object/array	Insieme di condizioni di filtro da utilizzare per filtrare i risultati. Le condizioni saranno ordinate nell'ordine del loro posizionamento nella formula. Property behavior: - *obbligatorio*
evaltype	integer	Metodo di valutazione della condizione di filtro. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Property behavior: - *obbligatorio*
eval_formula	string	Espressione generata che verrà utilizzata per valutare le condizioni di filtro. L'espressione contiene ID che fanno riferimento a specifiche condizioni di filtro tramite il loro `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Property behavior: - *sola lettura*
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a specifiche condizioni di filtro tramite il loro `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni di filtro: nessuna condizione può rimanere inutilizzata o omessa. Property behavior: - *obbligatorio* se `evaltype` è impostato su "custom expression"

Condizione di filtro della regola LLD

L'oggetto condizione di filtro della regola LLD definisce un controllo separato da eseguire sul valore di una macro LLD. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
macro	string	Macro LLD su cui eseguire il controllo. Comportamento della proprietà: - *obbligatoria*
value	string	Valore con cui confrontare. Comportamento della proprietà: - *obbligatoria* se `operator` è impostato su "corrisponde all'espressione regolare" o "non corrisponde all'espressione regolare"
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente durante la modifica delle condizioni di filtro, ma verrà generato nuovamente quando verranno richieste in seguito. Comportamento della proprietà: - *obbligatoria* se `evaltype` di oggetto filtro della regola LLD è impostato su "espressione personalizzata"

Proprietà	Type	Descrizione
operator	integer	Operatore della condizione. Valori possibili: 8 - *(predefinito)* corrisponde all'espressione regolare; 9 - non corrisponde all'espressione regolare; 12 - esiste; 13 - non esiste.

Note:
Per comprendere meglio come utilizzare i filtri con vari tipi di espressioni, vedere gli esempi nelle pagine dei metodi discoveryrule.get e discoveryrule.create.

Percorso della macro LLD

Il percorso della macro LLD ha le seguenti proprietà:

Proprietà	Type	Descrizione
lld_macro	string	Macro LLD. Comportamento della proprietà: - *obbligatorio*
path	string	Selettore per il valore che verrà assegnato alla macro corrispondente. Comportamento della proprietà: - *obbligatorio*

Preelaborazione della regola LLD

L'oggetto di preelaborazione della regola LLD presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
type	integer	Il tipo di opzione di preelaborazione. Valori possibili: 5 - Espressione regolare; 11 - XPath XML; 12 - JSONPath; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Verifica la presenza di errori in JSON; 17 - Verifica la presenza di errori in XML; 20 - Scarta se invariato con heartbeat; 21 - JavaScript; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Sostituisci; 27 - XML in JSON; 28 - Valore SNMP walk; 29 - SNMP walk in JSON; 30 - Valore SNMP get. Comportamento della proprietà: - *obbligatoria*

Proprietà	Tipo	Descrizione
params	string	Parametri aggiuntivi utilizzati dall'opzione di preelaborazione. Più parametri sono separati dal carattere di nuova riga (\n). Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "Espressione regolare" (5), "XPath XML" (11), "JSONPath" (12), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Verifica la presenza di errori in JSON" (16), "Verifica la presenza di errori in XML" (17), "Scarta se invariato con heartbeat" (20), "JavaScript" (21), "Prometheus in JSON" (23), "CSV in JSON" (24), "Sostituisci" (25), "Valore SNMP walk" (28), "SNMP walk in JSON" (29) o "Valore SNMP get" (30)
error_handler	integer	Tipo di azione utilizzato in caso di errore del passaggio di preelaborazione. Valori possibili: 0 - Il messaggio di errore viene impostato dal server Zabbix; 1 - Scarta il valore; 2 - Imposta un valore personalizzato; 3 - Imposta un messaggio di errore personalizzato. Comportamento della proprietà: - *obbligatorio* se `type` è impostato su "Espressione regolare" (5), "XPath XML" (11), "JSONPath" (12), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Verifica la presenza di errori in JSON" (16), "Verifica la presenza di errori in XML" (17), "Prometheus in JSON" (23), "CSV in JSON" (24), "XML in JSON" (27), "Valore SNMP walk" (28), "SNMP walk in JSON" (29) o "Valore SNMP get" (30)
error_handler_params	string	Parametri del gestore degli errori. Comportamento della proprietà: - *obbligatorio* se `error_handler` è impostato su "Imposta un valore personalizzato" o "Imposta un messaggio di errore personalizzato"

I seguenti parametri e gestori degli errori sono supportati per ogni tipo di preelaborazione.

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
5	Espressione regolare	pattern[1]	output[2]		0, 1, 2, 3
11	XPath XML	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Corrisponde all'espressione regolare	pattern[1]			0, 1, 2, 3
15	Non corrisponde all'espressione regolare	pattern[1]			0, 1, 2, 3

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori degli errori supportati
16	Verifica la presenza di errori in JSON	path[3]			0, 1, 2, 3
17	Verifica la presenza di errori in XML	path[3]			0, 1, 2, 3
20	Scarta se invariato con heartbeat	seconds^[4], 5^[5]			
21	JavaScript	script[2]			
23	Prometheus in JSON	pattern^[5], 6^[6]			0, 1, 2, 3
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Sostituisci	stringa di ricerca[2]	sostituzione[2]		
27	XML in JSON				0, 1, 2, 3
28	Valore SNMP walk	OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS		0, 1, 2, 3
29	SNMP walk in JSON[7]	Nome del campo[2]	Prefisso OID[2]	Formato: 0 - Invariato 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS	0, 1, 2, 3
30	Valore SNMP get	Formato: 1 - UTF-8 da Hex-STRING 2 - MAC da Hex-STRING 3 - Intero da BITS			0, 1, 2, 3

[1] espressione regolare
 [2] stringa
 [3] JSONPath o XPath XML
 [4] intero positivo (con supporto dei suffissi temporali, ad esempio 30s, 1m, 2h, 1d)
 [5] macro utente
 [6] pattern Prometheus conforme alla sintassi: `<metric name>{<label`

name>="<label value>", ...} == <value>. Ogni componente del pattern Prometheus (metrica, nome dell'etichetta, valore dell'etichetta e valore della metrica) può essere una macro utente.
 [7] Supporta più record "Nome del campo,Prefisso OID,Record del formato" delimitati da un carattere di nuova riga.

Override delle regole LLD

L'oggetto override delle regole LLD definisce un insieme di regole (filtri, condizioni e operazioni) utilizzate per sovrascrivere le proprietà di diversi oggetti prototipo. Ha le seguenti proprietà:

Property	Type	Description
name	string	Nome univoco dell'override. Comportamento della proprietà: - *obbligatoria*
step	integer	Numero d'ordine univoco dell'override. Comportamento della proprietà: - *obbligatoria*
stop	integer	Interrompe l'elaborazione degli override successivi in caso di corrispondenza. Valori possibili: 0 - *(predefinito)* non interrompere l'elaborazione degli override; 1 - interrompere l'elaborazione degli override se il filtro corrisponde.
filter	object	Filtro dell'override.
operations	object/array	Operazioni dell'override.

Filtro di override della regola LLD

L'oggetto filtro di override della regola LLD definisce un insieme di condizioni che, se corrispondono all'oggetto individuato, fanno sì che venga applicato l'override. Ha le seguenti proprietà:

Property	Type	Description
conditions	object/array	Insieme di condizioni del filtro di override da utilizzare per la corrispondenza con gli oggetti individuati. Le condizioni verranno ordinate nell'ordine in cui sono posizionate nella formula. Property behavior: - *required*
evaltype	integer	Metodo di valutazione delle condizioni del filtro di override. Valori possibili: 0 - And/Or; 1 - And; 2 - Or; 3 - Espressione personalizzata. Property behavior: - *required*
eval_formula	string	Espressione generata che verrà utilizzata per valutare le condizioni del filtro di override. L'espressione contiene ID che fanno riferimento a specifiche condizioni del filtro di override tramite il relativo `formulaid`. Il valore di `eval_formula` è uguale al valore di `formula` per i filtri con un'espressione personalizzata. Property behavior: - *read-only*

Property	Type	Description
formula	string	Espressione definita dall'utente da utilizzare per valutare le condizioni dei filtri di override con un'espressione personalizzata. L'espressione deve contenere ID che fanno riferimento a specifiche condizioni del filtro di override tramite il relativo `formulaid`. Gli ID utilizzati nell'espressione devono corrispondere esattamente a quelli definiti nelle condizioni del filtro di override: nessuna condizione può rimanere inutilizzata o omessa. Property behavior: - *required* se `evaltype` è impostato su "custom expression"

Condizione di filtro di override della regola LLD

L'oggetto condizione di filtro di override della regola LLD definisce un controllo separato da eseguire sul valore di una macro LLD. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
macro	string	Macro LLD su cui eseguire il controllo. Comportamento della proprietà: - *obbligatoria*
value	string	Valore con cui confrontare. Comportamento della proprietà: - *obbligatoria* se `operator` è impostato su "corrisponde all'espressione regolare" o "non corrisponde all'espressione regolare"
formulaid	string	ID univoco arbitrario utilizzato per fare riferimento alla condizione da un'espressione personalizzata. Può contenere solo lettere maiuscole. L'ID deve essere definito dall'utente durante la modifica delle condizioni di filtro, ma verrà generato nuovamente quando verranno richieste in seguito. Comportamento della proprietà: - *obbligatoria* se `evaltype` di oggetto filtro di override della regola LLD è impostato su "espressione personalizzata"
operator	integer	Operatore della condizione. Valori possibili: 8 - *(predefinito)* corrisponde all'espressione regolare; 9 - non corrisponde all'espressione regolare; 12 - esiste; 13 - non esiste.

Operazione di override della regola LLD

L'operazione di override della regola LLD è una combinazione di condizioni e azioni da eseguire sull'oggetto prototipo. Ha le seguenti proprietà:

Property	Type	Description
operationobject	integer	Tipo di oggetto individuato su cui eseguire l'azione. Valori possibili: 0 - Prototipo di item; 1 - Prototipo di trigger; 2 - Prototipo di grafico; 3 - Prototipo di host; 4 - Prototipo di discovery. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione di override. Valori possibili: 0 - *(predefinito)* uguale; 1 - diverso da; 2 - contiene; 3 - non contiene; 8 - corrisponde; 9 - non corrisponde.
value	string	Pattern da confrontare con il nome del prototipo di item, trigger, grafico o host, a seconda dell'oggetto selezionato.
opstatus	object	Oggetto stato dell'operazione di override per gli oggetti prototipo di item, trigger e host.
opdiscover	object	Oggetto stato di discovery dell'operazione di override (tutti i tipi di oggetto).
opperiod	object	Oggetto periodo dell'operazione di override (intervallo di aggiornamento) per l'oggetto prototipo di item.
ophistory	object	Oggetto cronologia dell'operazione di override per l'oggetto prototipo di item.
optrends	object	Oggetto trend dell'operazione di override per l'oggetto prototipo di item.
opseverity	object	Oggetto gravità dell'operazione di override per l'oggetto prototipo di trigger.
optag	object/array	Oggetto tag dell'operazione di override per gli oggetti prototipo di trigger e host.
optemplate	object/array	Oggetto template dell'operazione di override per l'oggetto prototipo di host.
opinventory	object	Oggetto inventario dell'operazione di override per l'oggetto prototipo di host.

Stato dell'operazione di override della regola LLD

Stato dell'operazione di override della regola LLD impostato per l'oggetto rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
status	integer	Esegue l'override dello stato per l'oggetto selezionato. Valori possibili: 0 - Crea abilitato; 1 - Crea disabilitato. Comportamento della proprietà: - *obbligatorio*

Operazione di override della regola LLD discover

Lo stato discover dell'operazione di override della regola LLD viene impostato sull'oggetto rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
discover	integer	Esegue l'override dello stato discover per l'oggetto selezionato. Valori possibili: 0 - Sì, continua a rilevare gli oggetti; 1 - No, i nuovi oggetti non verranno rilevati e quelli esistenti verranno contrassegnati come persi. Comportamento della proprietà: - *obbligatorio*

Periodo dell'operazione di override della regola LLD

Il periodo dell'operazione di override della regola LLD è un valore dell'intervallo di aggiornamento impostato per l'item rilevato. Presenta le seguenti proprietà:

Property	Type	Description
delay	string	Sostituisce l'intervallo di aggiornamento del prototipo dell'item. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h, 1d) oppure, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e intervalli di pianificazione. Accetta macro utente o macro LLD. Se utilizzato, il valore deve essere una singola macro. Non sono supportate più macro né macro combinate con testo. Gli intervalli flessibili possono essere scritti come due macro separate da una barra (ad esempio, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Esempio: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento della proprietà: - *required*

Cronologia delle operazioni di override della regola LLD

Valore della cronologia delle operazioni di override della regola LLD impostato sull'item rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
history	string	Esegue l'override della cronologia del prototipo di item, che è un'unità di tempo che definisce per quanto tempo i dati della cronologia devono essere conservati. Accetta anche macro utente e macro LLD. Comportamento della proprietà: - *obbligatorio*

Tendenze delle operazioni di override della regola LLD

Valore delle tendenze delle operazioni di override della regola LLD impostato sull'item individuato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
trends	string	Esegue l'override delle tendenze del prototipo di item, che rappresentano un'unità di tempo per quanto a lungo i dati di tendenza devono essere conservati. Accetta anche macro utente e macro LLD. Comportamento della proprietà: - *obbligatorio*

Gravità dell'operazione di override della regola LLD

Valore di gravità dell'operazione di override della regola LLD impostato sul trigger rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
severity	integer	Sovrascrive la gravità del prototipo di trigger. Valori possibili: 0 - *(predefinito)* non classificato; 1 - informazione; 2 - avviso; 3 - media; 4 - alta; 5 - disastro. Comportamento della proprietà: - *obbligatorio*

Tag dell'operazione di override della regola LLD

L'oggetto tag dell'operazione di override della regola LLD contiene il nome del tag e il valore impostati per l'oggetto rilevato. Ha le seguenti proprietà:

Property	Type	Description
tag	string	Nuovo nome del tag. Accetta anche macro utente e macro LLD. Property behavior: - *required*
value	string	Nuovo valore del tag. Accetta anche macro utente e macro LLD.

Template dell'operazione di override della regola LLD

Oggetto template dell'operazione di override della regola LLD collegato all'host rilevato. Ha le seguenti proprietà:

Property	Type	Description
templateid	ID	Esegue l'override del template dei template collegati al prototipo di host. Comportamento della proprietà: - *obbligatorio*

Modalità inventario dell'operazione di override della regola LLD

Valore della modalità di inventario dell'operazione di override della regola LLD impostato sul host rilevato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
inventory_mode	integer	Sovrascrive la modalità di inventario del prototipo di host. Valori possibili: -1 - disabilitato; 0 - *(predefinito)* manuale; 1 - automatico. Comportamento della proprietà: - *obbligatorio*

discoveryrule.create

Descrizione

`object discoveryrule.create(object/array lldRules)`

Questo metodo consente di creare nuove regole LLD.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Regole LLD da creare.

Oltre alle proprietà standard delle regole LLD, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
filter	object	Filtro della regola LLD per la regola LLD.
preprocessing	object/array	Opzioni di preprocessing della regola LLD.
lld_macro_paths	object/array	Opzioni lld_macro_path della regola LLD.
overrides	object/array	Opzioni di override della regola LLD.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle regole LLD create sotto la proprietà `itemids`. L'ordine degli ID restituiti corrisponde all'ordine delle regole LLD approvate.

Esempi

Creazione di una regola LLD

Creare una regola LLD di Zabbix agent per individuare i file system montati. Gli item individuati verranno aggiornati ogni 30 secondi.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Utilizzo di un filtro

Creare una regola LLD con un insieme di condizioni in base alle quali filtrare i risultati. Le condizioni verranno raggruppate utilizzando l'operatore logico "and".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex2",
                    "operator": "9"
                },
                {
                    "macro": "{#MACRO3}",
                    "value": "",
                    "operator": "12"
                },
                {
                    "macro": "{#MACRO4}",
                    "value": "",
                    "operator": "13"
                }
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
```

```
}
```

Creazione di una regola LLD con percorsi macro

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Regola LLD con percorsi macro LLD",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.path.1"
            },
            {
                "lld_macro": "{#MACRO2}",
                "path": "$.path.2"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Utilizzo di un filtro di espressione personalizzato

Creare una regola LLD con un filtro che utilizzi un'espressione personalizzata per valutare le condizioni. La regola LLD deve individuare solo gli oggetti per i quali il valore della macro "{#MACRO1}" corrisponde a entrambe le espressioni regolari "regex1" e "regex2", e il valore di "{#MACRO2}" corrisponde a "regex3" oppure "regex4". Gli ID formula "A", "B", "C" e "D" sono stati scelti arbitrariamente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 3,
            "formula": "(A and B) and (C or D)",
            "conditions": [
                {
```

```
                    "macro": "{#MACRO1}",
                    "value": "@regex1",
                    "formulaid": "A"
                },
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex2",
                    "formulaid": "B"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex3",
                    "formulaid": "C"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex4",
                    "formulaid": "D"
                }
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Utilizzo di campi di query e header personalizzati

Creare una regola LLD con campi di query e header personalizzati.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Creazione di una regola LLD con preprocessing

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Regola di discovery con preprocessing",
        "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Creazione di una regola LLD con override

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discover database host",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Discover MySQL host",
                "step": "1",
                "stop": "1",
                "filter": {
                    "evaltype": "2",
                    "conditions": [
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^mysqld\\.service$"
                        },
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^mariadb\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Database host",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10170"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "mysql"
                            }
                        ]
                    }
                ]
            },
            {
                "name": "Discover PostgreSQL host",
                "step": "2",
                "stop": "1",
                "filter": {
                    "evaltype": "0",
                    "conditions": [
                        {
```

```
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^postgresql\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Database host",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

Creare una regola LLD script

Creare una semplice raccolta dati utilizzando una regola LLD script.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Creare una regola LLD con un periodo di tempo specificato per la disabilitazione e senza eliminazione

Creare una regola LLD con un periodo di tempo personalizzato per disabilitare l'entità dopo che non viene più rilevata, con l'impostazione che non verrà mai eliminata.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Vedi anche

- Filtro della regola LLD
- Percorsi macro LLD
- Preelaborazione della regola LLD

Fonte

CDiscoveryRule::create() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Descrizione

`object discoveryrule.delete(array lldRuleIds)`

Questo metodo consente di eliminare le regole LLD.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 L'utente non potrà eliminare una regola LLD che rileva entità utilizzando un proxy non accessibile. Per ulteriori informazioni, vedere configurazione del gruppo utenti (*Elenco di accesso ai proxy*).

Parametri

`(array)` ID delle regole LLD da eliminare.

Restituisce valori

"(object)" Restituisce un oggetto contenente gli ID delle regole LLD eliminate sotto la proprietà `ruleids`.

Esempi

Eliminazione di più regole LLD

Elimina due regole LLD.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

Fonte

CDiscoveryRule::delete() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Descrizione

`integer/array discoveryrule.get(object parameters)`

Il metodo consente di recuperare le regole LLD in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
itemids	ID/array	Restituisce solo le regole LLD con gli ID specificati.
groupids	ID/array	Restituisce solo le regole LLD che appartengono agli host dei gruppi specificati.
hostids	ID/array	Restituisce solo le regole LLD che appartengono agli host specificati.
inherited	boolean	Se impostato su `true`, restituisce solo le regole LLD ereditate da un template.
interfaceids	ID/array	Restituisce solo le regole LLD che utilizzano le interfacce host specificate.
monitored	boolean	Se impostato su `true`, restituisce solo le regole LLD abilitate che appartengono a host monitorati.
templated	boolean	Se impostato su `true`, restituisce solo le regole LLD che appartengono ai template.
templateids	ID/array	Restituisce solo le regole LLD che appartengono ai template specificati.
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery della regola LLD. L'oggetto di discovery della regola LLD collega una regola LLD rilevata a un prototipo di regola LLD da cui è stata rilevata.
selectDiscoveryRulePrototypes	query	Restituisce una proprietà `discoveryRulePrototypes` con i prototipi di regola LLD che appartengono alla regola LLD. Supporta `count`.
selectFilter	query	Restituisce una proprietà `filter` con i dati del filtro utilizzato dalla regola LLD.
selectGraphs	query	Restituisce una proprietà `graphs` con i prototipi di grafico che appartengono alla regola LLD. Supporta `count`.
selectHostPrototypes	query	Restituisce una proprietà `hostPrototypes` con i prototipi di host che appartengono alla regola LLD. Supporta `count`.
selectHosts	query	Restituisce una proprietà `hosts` con un array di host a cui appartiene la regola LLD.
selectItems	query	Restituisce una proprietà `items` con i prototipi di item che appartengono alla regola LLD. Supporta `count`.
selectTriggers	query	Restituisce una proprietà `triggers` con i prototipi di trigger che appartengono alla regola LLD. Supporta `count`.
selectLLDMacroPaths	query	Restituisce una proprietà `lld_macro_paths` con un elenco di macro LLD e dei percorsi ai valori assegnati a ciascuna macro corrispondente.
selectPreprocessing	query	Restituisce una proprietà `preprocessing` con le opzioni di preprocessing della regola LLD.
selectOverrides	query	Restituisce una proprietà `lld_rule_overrides` con un elenco di filtri di override, condizioni e operazioni eseguite sugli oggetti prototipo.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text`. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene la regola LLD.
limitSelects	integer	Limita il numero di record restituiti dalle sottoquery. Si applica alle seguenti sottoquery: `selectItems`, `selectGraphs`, `selectTriggers`.

Parametro	Tipo	Descrizione
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero delle regole di individuazione da un host

Recupera tutte le regole di individuazione per uno specifico ID host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Individuazione delle interfacce di rete",
            "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": "Individuazione delle interfacce di rete come definito nell'espressione regolar
            "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": "Individuazione dei filesystem montati",
            "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": "Individuazione dei file system di tipi diversi come definito nell'espressione
            "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
}
```

Recupero delle condizioni del filtro

Recupera il nome della regola LLD "24681" e le relative condizioni del filtro. Il filtro utilizza il tipo di valutazione "and", quindi la proprietà `formula` è vuota e `eval_formula` viene generata automaticamente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Recuperare una regola LLD tramite URL

Recupera la regola LLD per un host in base al valore del campo URL della regola. È supportata solo la corrispondenza esatta della stringa URL definita per la regola LLD.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Recuperare una regola LLD con override

Recupera una regola LLD che ha varie impostazioni di override.

Richiesta:


```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
```

```
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Prototipo di grafico
- Host
- Prototipo di item
- Filtro della regola LLD
- Prototipo di trigger

Sorgente

CDiscoveryRule::get() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Descrizione

`object discoveryrule.update(object/array lldRules)`

Questo metodo consente di aggiornare le regole LLD esistenti.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà della regola LLD da aggiornare.

La proprietà `itemid` deve essere definita per ogni regola LLD, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard della regola LLD, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
filter	object	Filtro della regola LLD per sostituire il filtro esistente.
preprocessing	object/array	Opzioni di preprocessing della regola LLD per sostituire le opzioni di preprocessing esistenti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
lld_macro_paths	object/array	Opzioni lld_macro_path della regola LLD per sostituire le opzioni lld_macro_path esistenti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati
overrides	object/array	Opzioni di override della regola LLD per sostituire le opzioni di override esistenti. Comportamento del parametro: - *sola lettura* per gli oggetti ereditati

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle regole LLD aggiornate sotto la proprietà `itemids`.

Esempi

Aggiunta di un filtro a una regola LLD

Aggiungi un filtro in modo che il contenuto della macro *{#FSTYPE}* corrisponda all'espressione regolare *@File systems for discovery*.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Aggiunta di percorsi macro LLD

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Disabilitare il trapping

Disabilitare il trapping LLD per la regola di discovery.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Aggiornamento delle opzioni di preprocessing della regola LLD

Aggiorna una regola LLD con la regola di preprocessing "JSONPath".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Aggiornamento dello script della regola LLD

Aggiorna uno script di una regola LLD con uno script diverso e rimuove i parametri non necessari che erano utilizzati dallo script precedente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Aggiornamento del periodo di conservazione della regola LLD

Aggiorna la regola LLD per disabilitare l'entità non più rilevata dopo 12 ore e per eliminarla dopo 7 giorni.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Fonte

CDiscoveryRule::update() in *ui/include/classes/api/services/CDiscoveryRule.php*.

Report

Questa classe è progettata per funzionare con i report pianificati.

Riferimenti agli oggetti:

- Report
- Utenti
- Gruppi di utenti

Metodi disponibili:

- report.create - crea nuovi report pianificati
- report.delete - elimina report pianificati
- report.get - recupera report pianificati
- report.update - aggiorna report pianificati

Oggetto report

I seguenti oggetti sono direttamente correlati all'API `report`.

Report

L'oggetto report presenta le seguenti proprietà:

Property	Type	Description
reportid	ID	ID del report. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
userid	ID	ID dell'utente che ha creato il report. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Property	Type	Description
name	string	Nome univoco del report. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
dashboardid	ID	ID della dashboard su cui si basa il report. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
period	integer	Periodo per il quale verrà preparato il report. Valori possibili: 0 - *(predefinito)* giorno precedente; 1 - settimana precedente; 2 - mese precedente; 3 - anno precedente.
cycle	integer	Pianificazione con ripetizione del periodo. Valori possibili: 0 - *(predefinito)* giornaliera; 1 - settimanale; 2 - mensile; 3 - annuale.
start_time	integer	Ora del giorno, in secondi, in cui il report verrà preparato per l'invio. Predefinito: 0.
weekdays	integer	Giorni della settimana per l'invio del report. Valori bitmap possibili: 1 - lunedì; 2 - martedì; 4 - mercoledì; 8 - giovedì; 16 - venerdì; 32 - sabato; 64 - domenica. Questo è un campo con maschera di bit; è accettabile qualsiasi somma dei valori bitmap possibili (ad esempio, 21 per lunedì, mercoledì e venerdì). Predefinito: 0. Comportamento della proprietà: - *obbligatoria* se `cycle` è impostato su "weekly".
active_since	string	Data di inizio. Valori possibili: stringa vuota - *(predefinito)* non specificata (memorizzata come 0); data specifica nel formato YYYY-MM-DD (memorizzata come timestamp dell'inizio della giornata (00:00:00)).
active_till	string	Data di fine. Valori possibili: stringa vuota - *(predefinito)* non specificata (memorizzata come 0); data specifica nel formato YYYY-MM-DD (memorizzata come timestamp della fine della giornata (23:59:59)).
subject	string	Oggetto del messaggio del report.
message	string	Testo del messaggio del report.

Property	Type	Description
status	integer	Indica se il report è abilitato o disabilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.
description	text	Descrizione del report.
state	integer	Stato del report. Valori possibili: 0 - *(predefinito)* il report non è ancora stato elaborato; 1 - il report è stato generato e inviato correttamente a tutti i destinatari; 2 - la generazione del report non è riuscita; ”info” contiene informazioni sull’errore; 3 - il report è stato generato, ma l’invio ad alcuni o a tutti i destinatari non è riuscito; ”info” contiene informazioni sull’errore. Comportamento della proprietà: - *sola lettura*
lastsent	timestamp	Timestamp Unix dell’ultimo report inviato correttamente. Comportamento della proprietà: - *sola lettura*
info	string	Descrizione dell’errore o informazioni aggiuntive. Comportamento della proprietà: - *sola lettura*

Utenti

L’oggetto users ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
userid	ID	ID dell’utente a cui inviare il report. Comportamento della proprietà: - *obbligatoria*
access_userid	ID	ID dell’utente per conto del quale verrà generato il report. 0 - *(predefinito)* Genera il report per destinatario.
exclude	integer	Indica se escludere l’utente dalla mailing list. Valori possibili: 0 - *(predefinito)* Includi; 1 - Escludi.

Gruppi di utenti

L’oggetto dei gruppi di utenti ha le seguenti proprietà:

Proprietà	Type	Descrizione
usrgrpid	ID	ID del gruppo di utenti a cui inviare il report. Comportamento della proprietà: - *obbligatorio*
access_userid	ID	ID dell’utente per conto del quale verrà generato il report. 0 - *(predefinito)* Genera il report in base al destinatario.

report.create

Descrizione

`object report.create(object/array reports)`

Questo metodo consente di creare nuovi report pianificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. I permessi per richiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Report pianificati da creare.

Oltre alle proprietà standard dei report pianificati, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
users	object/array	Utenti a cui inviare il report. Comportamento del parametro: - *obbligatorio* se `user_groups` non è impostato
user_groups	object/array	Gruppi di utenti a cui inviare il report. Comportamento del parametro: - *obbligatorio* se `users` non è impostato

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei report pianificati creati nella proprietà `reportids`. L'ordine degli ID restituiti corrisponde all'ordine dei report pianificati passati.

Esempi

Creazione di un report pianificato

Creare un report settimanale che verrà preparato per la settimana precedente ogni lunedì-venerdì alle 12:00 dal 2021-04-01 al 2021-08-31.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Vedi anche

- Utenti
- Gruppi di utenti

Fonte

CReport::create() in *ui/include/classes/api/services/CReport.php*.

report.delete

Descrizione

`object report.delete(array reportids)`

Questo metodo consente di eliminare i report pianificati.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

ID "(array)" dei report pianificati da eliminare.

Valori di ritorno

`(oggetto)` Restituisce un oggetto contenente gli ID dei report pianificati eliminati sotto la proprietà `reportids`.

Esempi

Eliminazione di più report pianificati

Eliminare due report pianificati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
```

```
        "2"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CReport::delete() in *ui/include/classes/api/services/CReport.php*.

report.get

Descrizione

`integer/array report.get(object parameters)`

Il metodo consente di recuperare i report pianificati in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
reportids	ID/array	Restituisce solo i report pianificati con gli ID report specificati.
expired	boolean	Se impostato su `true`, restituisce solo i report pianificati scaduti.
selectUsers	query	Restituisce una proprietà `users` con gli utenti a cui è configurato l'invio del report.
selectUserGroups	query	Restituisce una proprietà `user_groups` con i gruppi di utenti a cui è configurato l'invio del report.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `reportid`, `name`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei dati del report

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": {
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "Report settimanale",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Report settimanale",
            "message": "Testo di accompagnamento del report",
            "status": "1",
            "description": "Descrizione del report",
            "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": "Report mensile",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Report mensile",
            "message": "Testo di accompagnamento del report",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```


Vedi anche

- Utenti
- Gruppi di utenti

Fonte

CReport::get() in *ui/include/classes/api/services/CReport.php*.

report.update

Descrizione

`object report.update(object/array reports)`

Questo metodo consente di aggiornare i report pianificati esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del report pianificato da aggiornare.

La proprietà `reportid` deve essere definita per ogni report pianificato, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del report pianificato, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
users	object/array	Utenti per sostituire gli utenti attualmente assegnati al report pianificato. Comportamento del parametro: - *obbligatorio* se `user_groups` non è impostato

Parametro	Tipo	Descrizione
user_groups	object/array	Gruppi di utenti per sostituire i gruppi di utenti attualmente assegnati al report pianificato. Comportamento del parametro: - *obbligatorio* se `users` non è impostato

Valori di ritorno

`(oggetto)` Restituisce un oggetto contenente gli ID dei report pianificati aggiornati sotto la proprietà `reportids`.

Esempi

Disabilitazione del report pianificato

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Vedi anche

- Utenti
- Gruppi di utenti

Fonte

CReport::update() in *ui/include/classes/api/services/CReport.php*.

Ruolo

Questa classe è progettata per funzionare con i ruoli utente.

Riferimenti agli oggetti:

- Role
- Role rules
  - Elemento dell'interfaccia utente
  - Servizio
  - Tag di servizio
  - Modulo
  - Azione

Metodi disponibili:

- role.create - crea nuovi ruoli utente
- role.delete - elimina i ruoli utente
- role.get - recupera i ruoli utente
- role.update - aggiorna i ruoli utente

Oggetto ruolo

I seguenti oggetti sono direttamente correlati all'API `role`.

Ruolo

L'oggetto ruolo ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
roleid	ID	ID del ruolo. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del ruolo. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
type	integer	Tipo di utente. Valori possibili: 1 - *(predefinito)* Utente; 2 - Admin; 3 - Super admin. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
readonly	integer	Indica se il ruolo è di sola lettura. Valori possibili: 0 - *(predefinito)* No; 1 - Sì. Comportamento della proprietà: - *sola lettura*

Regole dei ruoli

L'oggetto delle regole del ruolo presenta le seguenti proprietà:

Proprietà	Tipo	Descrizione
ui	array	Array di oggetti elemento dell'interfaccia utente.
ui.default_access	integer	Indica se l'accesso ai nuovi elementi dell'interfaccia utente è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.
profile.redirect.enforce	integer	Indica se l'URL di reindirizzamento a livello di ruolo dopo l'autenticazione è obbligatorio e non può essere sovrascritto dall'utente. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato.
profile.redirect.url	string	URL utilizzato per il reindirizzamento dell'utente dopo l'autenticazione.
services.read.mode	integer	Accesso in sola lettura ai servizi. Valori possibili: 0 - Accesso in sola lettura ai servizi specificati dalle proprietà `services.read.list` o corrispondenti alla proprietà `services.read.tag`; 1 - *(predefinito)* Accesso in sola lettura a tutti i servizi.

Proprietà	Tipo	Descrizione
services.read.list	array	Array di oggetti Service. Ai servizi specificati, inclusi i servizi figlio, verrà concesso l'accesso in sola lettura al ruolo utente. L'accesso in sola lettura non sovrascriverà l'accesso in lettura-scrittura ai servizi. Comportamento della proprietà: - *supportata* se `services.read.mode` è impostato su "0".
services.read.tag	object	Array di oggetti Service tag. Ai servizi corrispondenti ai tag, inclusi i servizi figlio, verrà concesso l'accesso in sola lettura al ruolo utente. L'accesso in sola lettura non sovrascriverà l'accesso in lettura-scrittura ai servizi. Comportamento della proprietà: - *supportata* se `services.read.mode` è impostato su "0".
services.write.mode	integer	Accesso in lettura-scrittura ai servizi. Valori possibili: 0 - *(predefinito)* Accesso in lettura-scrittura ai servizi specificati dalle proprietà `services.write.list` o corrispondenti alla proprietà `services.write.tag`; 1 - Accesso in lettura-scrittura a tutti i servizi.
services.write.list	array	Array di oggetti Service. Ai servizi specificati, inclusi i servizi figlio, verrà concesso l'accesso in lettura-scrittura al ruolo utente. L'accesso in lettura-scrittura sovrascriverà l'accesso in sola lettura ai servizi. Comportamento della proprietà: - *supportata* se `services.write.mode` è impostato su "0".
services.write.tag	object	Array di oggetti Service tag. Ai servizi corrispondenti ai tag, inclusi i servizi figlio, verrà concesso l'accesso in lettura-scrittura al ruolo utente. L'accesso in lettura-scrittura sovrascriverà l'accesso in sola lettura ai servizi. Comportamento della proprietà: - *supportata* se `services.write.mode` è impostato su "0".
modules	array	Array di oggetti module.
modules.default_access	integer	Indica se l'accesso ai nuovi moduli è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.
api.access	integer	Indica se l'accesso all'API è abilitato. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato.
api.mode	integer	Modalità per la gestione dei metodi API elencati nella proprietà `api`. Valori possibili: 0 - *(predefinito)* Elenco di negazione; 1 - Elenco di autorizzazione. Comportamento della proprietà: - *supportata* se `api.access` è impostato su "1".

Proprietà	Tipo	Descrizione
api	array	Array di metodi API. I caratteri jolly sono supportati, ad esempio: `*` (qualsiasi metodo), `dashboard.*`, `*.export`. Se non viene specificato alcun metodo, l'accesso a tutti i metodi API viene negato se `api.mode` è impostato su "1" (Elenco di autorizzazione) e consentito se `api.mode` è impostato su "0" (Elenco di negazione). Comportamento della proprietà: - *supportata* se `api.access` è impostato su "1".
actions	array	Array di oggetti action.
actions.default_access	integer	Indica se l'accesso alle nuove azioni è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.
devices.access	integer	Indica se l'accesso ai dispositivi mobili dell'utente per l'utilizzo con Zabbix mobile app è abilitato. Valori possibili: 0 - (*predefinito per User e Admin*) Disabilitato; 1 - (*predefinito per Super admin*) Abilitato. Comportamento della proprietà: - *supportata* se `EnableMobileDevices` è abilitato
devices.actions	array	Array di oggetti Device action. Comportamento della proprietà: - *supportata* se `devices.access` è impostato su "1"
devices.actions.default_access	integer	Indica se l'accesso alle nuove azioni dei dispositivi è abilitato. Valori possibili: 0 - (*predefinito per User e Admin*) Disabilitato; 1 - (*predefinito per Super admin*) Abilitato. Comportamento della proprietà: - *supportata* se `EnableMobileDevices` è abilitato

Elemento dell'interfaccia utente

L'oggetto elemento dell'interfaccia utente ha le seguenti proprietà:

Property	Type	Description
name	string	Nome dell'elemento dell'interfaccia utente. Valori possibili se `type` dell'oggetto Role è impostato su "User", "Admin" o "Super admin": `monitoring.dashboard` - *Dashboard*; `monitoring.problems` - *Monitoraggio* > *Problemi*; `monitoring.hosts` - *Monitoraggio* > *Host*; `monitoring.latest_data` - *Monitoraggio* > *Ultimi dati*; `monitoring.maps` - *Monitoraggio* > *Mappe*; `services.services` - *Servizi* > *Servizi*; `services.sla_report` - *Servizi* > *Report SLA*; `inventory.overview` - *Inventario* > *Panoramica*; `inventory.hosts` - *Inventario* > *Host*; `reports.availability_report` - *Report* > *Report sulla disponibilità*; `reports.top_triggers` - *Report* > *Top 100 trigger*. Valori possibili se `type` dell'oggetto Role è impostato su "Admin" o "Super admin": `monitoring.discovery` - *Monitoraggio* > *Rilevamento*; `services.sla` - *Servizi* > *SLA*; `reports.scheduled_reports` - *Report* > *Report pianificati*; `reports.notifications` - *Report* > *Notifiche*; `configuration.template_groups` - *Raccolta dati* > *Gruppi di template*; `configuration.host_groups` - *Raccolta dati* > *Gruppi di host*; `configuration.templates` - *Raccolta dati* > *Template*; `configuration.hosts` - *Raccolta dati* > *Host*; `configuration.maintenance` - *Raccolta dati* > *Manutenzione*; `configuration.discovery` - *Raccolta dati* > *Rilevamento*; `configuration.trigger_actions` - *Avvisi* > *Azioni* > *Azioni trigger*; `configuration.service_actions` - *Avvisi* > *Azioni* > *Azioni dei servizi*; `configuration.discovery_actions` - *Avvisi* > *Azioni* > *Azioni di rilevamento*; `configuration.autoregistration_actions` - *Avvisi* > *Azioni* > *Azioni di autoregistrazione*; `configuration.internal_actions` - *Avvisi* > *Azioni* > *Azioni interne*. Valori possibili se `type` dell'oggetto Role è impostato su "Super admin": `reports.system_info` - *Report* > *Informazioni di sistema*; `reports.audit` - *Report* > *Registro di audit*; `reports.action_log` - *Report* > *Registro delle azioni*; `configuration.event_correlation` - *Raccolta dati* > *Correlazione degli eventi*; `administration.media_types` - *Avvisi* > *Tipi di supporto*; `administration.scripts` - *Avvisi* > *Script*; `administration.user_groups` - *Utenti* > *Gruppi di utenti*; `administration.user_roles` - *Utenti* > *Ruoli utente*; `administration.users` - *Utenti* > *Utenti*; `administration.api_tokens` - *Utenti* > *Token API*; `administration.authentication` - *Utenti* > *Autenticazione*; `administration.linked_devices` - *Utenti* > *Dispositivi*; `administration.general` - *Amministrazione* > *Generale*; `administration.audit_log` - *Amministrazione* > *Registro di audit*; `administration.housekeeping` - *Amministrazione* > *Housekeeping*; `administration.proxy_groups` - *Amministrazione* > *Gruppi di proxy*; `administration.proxies` - *Amministrazione* > *Proxy*; `administration.macros` - *Amministrazione* > *Macro*; `administration.queue` - *Amministrazione* > *Coda*.

Property	Type	Description
status	integer	Indica se l'accesso all'elemento dell'interfaccia utente è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(default)* Abilitato.

Servizio

Proprietà	Type	Descrizione
serviceid	ID	ID del servizio. Comportamento della proprietà: - *obbligatorio*

Tag del servizio

Property	Type	Description
tag	string	Nome del tag. Se viene specificata una stringa vuota, il tag del servizio non verrà utilizzato per la corrispondenza del servizio. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del tag. Se non viene specificato alcun valore o viene specificata una stringa vuota, per la corrispondenza del servizio verrà utilizzato solo il nome del tag.

Modulo

L'oggetto modulo ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
moduleid	ID	ID del modulo. Comportamento della proprietà: - *obbligatorio*
status	integer	Indica se l'accesso al modulo è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.

Azione

L'oggetto action presenta le seguenti proprietà:

Property	Type	Description
name	string	Nome dell'action. Valori possibili se `type` dell'oggetto Role è impostato su "User", "Admin" o "Super admin": `edit_dashboards` - Creare e modificare dashboard; `edit_maps` - Creare e modificare mappe; `add_problem_comments` - Aggiungere commenti ai problemi; `change_severity` - Modificare la gravità dei problemi; `acknowledge_problems` - Confermare i problemi; `suppress_problems` - Sopprimere i problemi; `close_problems` - Chiudere i problemi; `execute_scripts` - Eseguire script; `manage_api_tokens` - Gestire i token API; `change_problem_ranking` - Modificare la classificazione del problema da causa a sintomo e viceversa; `edit_own_media` - Consentire agli utenti di creare e modificare i propri media. Valori possibili se `type` dell'oggetto Role è impostato su "Admin" o "Super admin": `edit_maintenance` - Creare e modificare le attività di manutenzione; `manage_scheduled_reports` - Gestire i report pianificati, `manage_sla` - Gestire gli SLA, `select_server_for_monitoring` - Consentire agli utenti di creare host monitorati dal server e dalle regole di network discovery eseguite dal server. Valori possibili se `type` dell'oggetto Role è impostato su "User" o "Admin": `invoke_execute_now` - Consentire agli utenti con sole autorizzazioni di lettura su un host di eseguire i controlli degli item. Valori possibili se `type` dell'oggetto Role è impostato su "Super admin": `edit_user_media` - Consentire agli utenti di creare e modificare i media per altri utenti. Comportamento della proprietà: - *required*
status	integer	Indica se l'accesso per eseguire l'azione è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(default)* Abilitato.

Azione del dispositivo

L'oggetto azione del dispositivo presenta le seguenti proprietà:

Property	Type	Description
name	string	Nome dell'azione del dispositivo. Valori possibili se `type` dell'oggetto Role è impostato su "User", "Admin" o "Super admin": `manage_own` - Consente agli utenti di aggiungere o rimuovere i propri dispositivi mobili. Valori possibili se `type` dell'oggetto Role è impostato su "Super admin": `manage_user` - Consente agli utenti di aggiungere o rimuovere dispositivi mobili per tutti gli utenti. Comportamento della proprietà: - *required*
status	integer	Indica se l'accesso per eseguire l'azione è abilitato. Valori possibili: 0 - Disabilitato; 1 - (*default*) Abilitato.

role.create

Descrizione

`object role.create(object/array roles)`

Questo metodo consente di creare nuovi ruoli.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Ruoli da creare.

Oltre alle proprietà standard del ruolo, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
rules	array	Regole del ruolo da creare per il ruolo.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei ruoli creati sotto la proprietà `roleids`. L'ordine degli ID restituiti corrisponde all' ordine dei ruoli passati.

Esempi

Creazione di un ruolo

Creare un ruolo di tipo "User" con accesso negato a due elementi dell'interfaccia utente.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Vedi anche

- Regole del ruolo
- Elemento dell'interfaccia utente
- Modulo
- Azione

Fonte

CRole::create() in *ui/include/classes/api/services/CRole.php*.

role.delete

Descrizione

`object role.delete(array roleids)`

Questo metodo consente di eliminare i ruoli.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID dei ruoli da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei ruoli eliminati sotto la proprietà `roleids`.

Esempi

Eliminazione di più ruoli utente

Elimina due ruoli utente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
```

```
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CRole::delete() in *ui/include/classes/api/services/CRole.php*.

role.get

Descrizione

`integer/array role.get(object parameters)`

Il metodo consente di recuperare i ruoli in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
roleids	ID/array	Restituisce solo i ruoli con gli ID specificati.
current	boolean	Se impostato su `true`, restituisce solo il ruolo dell'utente corrente.
selectRules	query	Restituisce una proprietà `rules` con le regole del ruolo.
selectUsers	query	Restituisce una proprietà `users` con gli utenti a cui è assegnato il ruolo. Vedere `user.get` per le limitazioni basate sul tipo di utente.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `roleid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei dati del ruolo

Recupera i dati del ruolo "Super admin role" e le relative regole di accesso.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.problems",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.hosts",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.latest_data",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.maps",
                        "status": "1"
                    },
                    {
                        "name": "services.services",
                        "status": "1"
                    },
                    {
                        "name": "services.sla_report",
                        "status": "1"
                    },
                    {
```

```
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
        },
        {
            "name": "configuration.service_actions",
            "status": "1"
```

```
        },
        {
            "name": "configuration.discovery_actions",
            "status": "1"
        },
        {
            "name": "configuration.autoregistration_actions",
            "status": "1"
        },
        {
            "name": "configuration.internal_actions",
            "status": "1"
        },
        {
            "name": "reports.system_info",
            "status": "1"
        },
        {
            "name": "reports.audit",
            "status": "1"
        },
        {
            "name": "reports.action_log",
            "status": "1"
        },
        {
            "name": "configuration.event_correlation",
            "status": "1"
        },
        {
            "name": "administration.media_types",
            "status": "1"
        },
        {
            "name": "administration.scripts",
            "status": "1"
        },
        {
            "name": "administration.user_groups",
            "status": "1"
        },
        {
            "name": "administration.user_roles",
            "status": "1"
        },
        {
            "name": "administration.users",
            "status": "1"
        },
        {
            "name": "administration.api_tokens",
            "status": "1"
        },
        {
            "name": "administration.authentication",
            "status": "1"
        },
        {
            "name": "administration.general",
            "status": "1"
        },
        {
```

```
                "name": "administration.audit_log",
                "status": "1"
            },
            {
                "name": "administration.housekeeping",
                "status": "1"
            },
            {
                "name": "administration.proxy_groups",
                "status": "1"
            },
            {
                "name": "administration.proxies",
                "status": "1"
            },
            {
                "name": "administration.macros",
                "status": "1"
            },
            {
                "name": "administration.queue",
                "status": "1"
            }
        ],
        "ui.default_access": "1",
        "profile.redirect.enforce": "0",
        "profile.redirect.url": "",
        "services.read.mode": "1",
        "services.read.list": [],
        "services.read.tag": {
            "tag": "",
            "value": ""
        },
        "services.write.mode": "1",
        "services.write.list": [],
        "services.write.tag": {
            "tag": "",
            "value": ""
        },
        "modules": [
            {
                "moduleid": 1,
                "status": "1"
            },
            {
                "moduleid": 2,
                "status": "1"
            },
            {
                "moduleid": 3,
                "status": "1"
            },
            {
                "moduleid": 4,
                "status": "1"
            },
            {
                "moduleid": 5,
                "status": "1"
            },
            {
                "moduleid": 6,
```

```
            "status": "1"
        },
        {
            "moduleid": 7,
            "status": "1"
        },
        {
            "moduleid": 8,
            "status": "1"
        },
        {
            "moduleid": 9,
            "status": "1"
        },
        {
            "moduleid": 10,
            "status": "1"
        },
        {
            "moduleid": 11,
            "status": "1"
        },
        {
            "moduleid": 12,
            "status": "1"
        },
        {
            "moduleid": 13,
            "status": "1"
        },
        {
            "moduleid": 14,
            "status": "1"
        },
        {
            "moduleid": 15,
            "status": "1"
        },
        {
            "moduleid": 16,
            "status": "1"
        },
        {
            "moduleid": 17,
            "status": "1"
        },
        {
            "moduleid": 18,
            "status": "1"
        },
        {
            "moduleid": 19,
            "status": "1"
        },
        {
            "moduleid": 20,
            "status": "1"
        },
        {
            "moduleid": 21,
            "status": "1"
        },
```

```
            {
                "moduleid": 22,
                "status": "1"
            },
            {
                "moduleid": 23,
                "status": "1"
            },
            {
                "moduleid": 24,
                "status": "1"
            },
            {
                "moduleid": 25,
                "status": "1"
            },
            {
                "moduleid": 26,
                "status": "1"
            },
            {
                "moduleid": 27,
                "status": "1"
            },
            {
                "moduleid": 28,
                "status": "1"
            },
            {
                "moduleid": 29,
                "status": "1"
            },
            {
                "moduleid": 30,
                "status": "1"
            },
            {
                "moduleid": 31,
                "status": "1"
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
            {
                "name": "edit_maps",
                "status": "1"
            },
```

```
                {
                    "name": "acknowledge_problems",
                    "status": "1"
                },
                {
                    "name": "suppress_problems",
                    "status": "1"
                },
                {
                    "name": "close_problems",
                    "status": "1"
                },
                {
                    "name": "change_severity",
                    "status": "1"
                },
                {
                    "name": "add_problem_comments",
                    "status": "1"
                },
                {
                    "name": "execute_scripts",
                    "status": "1"
                },
                {
                    "name": "manage_api_tokens",
                    "status": "1"
                },
                {
                    "name": "edit_maintenance",
                    "status": "1"
                },
                {
                    "name": "manage_scheduled_reports",
                    "status": "1"
                },
                {
                    "name": "manage_sla",
                    "status": "1"
                },
                {
                    "name": "invoke_execute_now",
                    "status": "1"
                },
                {
                    "name": "change_problem_ranking",
                    "status": "1"
                },
                {
                    "name": "edit_own_media",
                    "status": "1"
                },
                {
                    "name": "edit_user_media",
                    "status": "1"
                }
            ],
            "actions.default_access": "1"
        }
    }
],
"id": 1
```

```
}
```

Vedi anche

- Regole del ruolo
- Utente

Fonte

CRole::get() in *ui/include/classes/api/services/CRole.php*.

role.update

Descrizione

`object role.update(object/array roles)`

Questo metodo consente di aggiornare i ruoli esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà del ruolo da aggiornare.

La proprietà `roleid` deve essere definita per ciascun ruolo, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del ruolo, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
rules	array	Regole di accesso da aggiornare per il ruolo.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei ruoli aggiornati sotto la proprietà `roleids`.

Esempi

Disabilitazione della possibilità di eseguire script

Aggiorna il ruolo con ID "5", disabilitando la possibilità di eseguire script.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Limitazione dell'accesso all'API

Aggiorna il ruolo con ID "5", negando la chiamata di qualsiasi metodo "create", "update" o "delete".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CRole::update() in *ui/include/classes/api/services/CRole.php*.

Scenario web

Questa classe è progettata per funzionare con gli scenari web.

Riferimenti agli oggetti:

- Scenario web
- Passo dello scenario
  - Campo HTTP
- Tag dello scenario web

Metodi disponibili:

- httptest.create - crea nuovi scenari web
- httptest.delete - elimina scenari web
- httptest.get - recupera scenari web
- httptest.update - aggiorna scenari web

Oggetto scenario web

I seguenti oggetti sono direttamente correlati all'API `webcheck`.

Scenario web

L'oggetto scenario web presenta le seguenti proprietà.

Property	Type	Description
httptestid	ID	ID dello scenario web. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
hostid	ID	ID dell'host a cui appartiene lo scenario web. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
name	string	Nome dello scenario web. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
agent	string	Stringa user agent utilizzata dallo scenario web. Predefinito: Zabbix
authentication	integer	Metodo di autenticazione utilizzato dallo scenario web. Valori possibili: 0 - *(predefinito)* nessuno; 1 - autenticazione HTTP di base; 2 - autenticazione NTLM; 3 - autenticazione Kerberos; 4 - autenticazione Digest.
delay	string	Intervallo di esecuzione dello scenario web. Accetta un numero di secondi (ad esempio, `1`, `60`, `300`), un valore con un suffisso temporale (ad esempio, `30s`, `1m`, `2h`, `1d`) o una macro utente. Predefinito: 1m.
headers	array	Intestazioni HTTP inviate durante l'esecuzione di una richiesta.
http_password	string	Password utilizzata per l'autenticazione HTTP di base, NTLM, Kerberos o Digest.
http_proxy	string	proxy utilizzato dallo scenario web, nel formato *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Nome utente utilizzato per l'autenticazione HTTP di base, NTLM, Kerberos o Digest.
retries	integer	Numero di volte in cui uno scenario web tenterà di eseguire ogni passaggio prima di non riuscire. Predefinito: 1.
ssl_cert_file	string	Nome del file del certificato SSL utilizzato per l'autenticazione del client (deve essere in formato PEM).
ssl_key_file	string	Nome del file della chiave privata SSL utilizzata per l'autenticazione del client (deve essere in formato PEM).
ssl_key_password	string	Password della chiave privata SSL.
status	integer	Indica se lo scenario web è abilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.

Property	Type	Description
templateid	ID	ID dello scenario web del template principale. Comportamento della proprietà: - *sola lettura*
variables	array	Variabili dello scenario web.
verify_host	integer	Indica se verificare che il nome host della connessione corrisponda a quello presente nel certificato dell'host. Valori possibili: 0 - *(predefinito)* non verificare l'host; 1 - verificare l'host.
verify_peer	integer	Indica se verificare che il certificato dell'host sia autentico. Valori possibili: 0 - *(predefinito)* non verificare il peer; 1 - verificare il peer.
uuid	string	Identificatore univoco globale, utilizzato per collegare gli scenari web importati a quelli già esistenti. Generato automaticamente se non specificato. Comportamento della proprietà: - *supportato* se lo scenario web appartiene a un template

Passaggio dello scenario

L'oggetto del passaggio dello scenario definisce un controllo specifico dello scenario web. Presenta le seguenti proprietà.

Property	Type	Description
name	string	Nome del passaggio dello scenario. Comportamento della proprietà: - *obbligatoria*
no	integer	Numero progressivo del passaggio in uno scenario web. Comportamento della proprietà: - *obbligatoria*
url	string	URL da controllare. Comportamento della proprietà: - *obbligatoria*
follow_redirects	integer	Indica se seguire i reindirizzamenti HTTP. Valori possibili: 0 - non seguire i reindirizzamenti; 1 - *(predefinito)* seguire i reindirizzamenti.
headers	array	Intestazioni HTTP da inviare durante l'esecuzione di una richiesta. Le intestazioni del passaggio dello scenario sovrascriveranno quelle specificate per lo scenario web.
posts	string/array	Variabili HTTP POST come stringa (dati post non elaborati) o come array di campi HTTP (dati dei campi del modulo).
required	string	Testo che deve essere presente nella risposta.
retrieve_mode	integer	Parte della risposta HTTP che il passaggio dello scenario deve recuperare. Valori possibili: 0 - *(predefinito)* solo corpo; 1 - solo intestazioni; 2 - intestazioni e corpo.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole.

Property	Type	Description
timeout	string	Timeout della richiesta in secondi. Accetta un numero di secondi (ad esempio, 1, 60, 300), un valore con un suffisso temporale (ad esempio, 30s, 1m, 1h) o una macro utente. Predefinito: 15s. Massimo: 1h. Minimo: 1s.
variables	array	Variabili del passaggio dello scenario.
query_fields	array	Campi della query - array di campi HTTP che verranno aggiunti all'URL durante l'esecuzione di una richiesta.

Campo HTTP

L'oggetto campo HTTP definisce il nome e il valore utilizzati per specificare le variabili dello scenario web, le intestazioni HTTP e i campi POST o i campi di query. Ha le seguenti proprietà.

Property	Type	Description
name	string	Nome dell'intestazione/variabile/campo POST o GET. Property behavior: - *obbligatorio*
value	string	Valore dell'intestazione/variabile/campo POST o GET. Property behavior: - *obbligatorio*

Tag dello scenario web

L'oggetto tag dello scenario web ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tag	string	Nome del tag dello scenario web. Comportamento della proprietà: - *obbligatoria*
value	string	Valore del tag dello scenario web. Comportamento della proprietà: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template; 1 - host. Comportamento della proprietà: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.

httptest.create

Descrizione

`object httptest.create(object/array webScenarios)`

Questo metodo consente di creare nuovi web scenario.

Note:
La creazione di un web scenario creerà automaticamente un insieme di item di monitoraggio web.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Scenari web da creare.

Oltre alle proprietà standard dello scenario web, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
steps	array	Passi dello scenario. Comportamento del parametro: - *obbligatorio*
tags	array	Tag dello scenario web.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli scenari web creati sotto la proprietà `httptestids`. L'ordine degli ID restituiti corrispondono all'ordine degli scenari Web passati.

Esempi

Creazione di uno scenario web

Creare uno scenario web per monitorare la home page aziendale. Lo scenario avrà due passaggi, per controllare la home page e la pagina "About" e assicurarsi che restituiscano il codice di stato HTTP 200.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
```

```
        ]
    },
    "id": 1
}
```

Vedi anche

- Passo dello scenario

Fonte

CHttpTest::create() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

Descrizione

`object httptest.delete(array webScenarioIds)`

Questo metodo consente di eliminare gli scenari web.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID degli scenari web da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli scenari web cancellati sotto la proprietà `httptestids`.

Esempi

Eliminazione di più scenari web

Eliminare due scenari web.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Fonte

CHttpTest::delete() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

Descrizione

`integer/array httptest.get(object parameters)`

Il metodo consente di recuperare gli scenari web in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
groupids	ID/array	Restituisce solo gli scenari web che appartengono ai gruppi di host specificati.
hostids	ID/array	Restituisce solo gli scenari web che appartengono agli host specificati.
httptestids	ID/array	Restituisce solo gli scenari web con gli ID specificati.
inherited	boolean	Se impostato su `true`, restituisce solo gli scenari web ereditati da un template.
inheritedTags	boolean	Restituisce gli scenari web che hanno i tags specificati anche in template/host/template collegati. Valori possibili: `true` - template/host/template collegati devono avere anch'essi i tag specificati; `false` - *(predefinito)* i tag da template/host/template collegati vengono ignorati.
monitored	boolean	Se impostato su `true`, restituisce solo gli scenari web abilitati che appartengono a host monitorati.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag presenti in template/host/template collegati.
templated	boolean	Se impostato su `true`, restituisce solo gli scenari web che appartengono ai template.
templateids	ID/array	Restituisce solo gli scenari web che appartengono ai template specificati.
expandName	flag	Espande le macro nel nome dello scenario web.
expandStepName	flag	Espande le macro nei nomi dei passaggi dello scenario.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
tags	array	Restituisce solo gli scenari web con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti gli scenari web. Valori possibili di operator: 0 - *(predefinito)* Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
selectHosts	query	Restituisce gli host a cui appartiene lo scenario web come array nella proprietà `hosts`.

Parametro	Type	Descrizione
selectSteps	query	Restituisce i passaggi dello scenario web nella proprietà `steps`. Supporta `count`.
selectTags	query	Restituisce i tag dello scenario web nella proprietà `tags`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `httptestid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero di uno scenario web

Recupera tutti i dati sullo scenario web "9".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Host
- Passo dello scenario

Fonte

CHttpTest::get() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Descrizione

`object httptest.update(object/array webScenarios)`

Questo metodo consente di aggiornare gli scenari web esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà dello scenario web da aggiornare.

La proprietà `httptestid` deve essere definita per ciascuno scenario web, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard dello scenario web, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
steps	array	Passi dello scenario per sostituire i passi esistenti.
tags	array	Tag dello scenario web.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli scenari web aggiornati sotto la proprietà `httptestids`.

Esempi

Abilitazione di uno scenario web

Abilitare uno scenario web, cioè impostarne lo stato su ”0”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Vedi anche

- Passo dello scenario

Fonte

CHttpTest::update() in *ui/include/classes/api/services/CHttpTest.php*.

Script

Questa classe è progettata per funzionare con gli script.

Riferimenti agli oggetti:

- Script
  - Parametri webhook
- Debug
  - Voce di log

Metodi disponibili:

- script.create - crea nuovi script
- script.delete - elimina gli script
- script.execute - esegue gli script
- script.get - recupera gli script
- script.getscriptsbyevents - recupera gli script per gli eventi
- script.getscriptsbyhosts - recupera gli script per gli host
- script.update - aggiorna gli script

Oggetto Script

I seguenti oggetti sono direttamente correlati all'API `script`.

Script

L'oggetto script presenta le seguenti proprietà.

Property	Type	Description
scriptid	ID	ID dello script. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome dello script. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
type	integer	Tipo di script. Valori possibili se `scope` è impostato su "action operation": 0 - Script; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook. Valori possibili se `scope` è impostato su "manual host action" o "manual event action": 6 - URL. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
command	string	Comando da eseguire. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Script", "IPMI", "SSH", "TELNET" o "Webhook"
scope	integer	Ambito dello script. Valori possibili: 1 - action operation; 2 - manual host action; 4 - manual event action. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Property	Type	Description
execute_on	integer	Dove eseguire lo script. Valori possibili: 0 - esecuzione su Zabbix agent; 1 - esecuzione su Zabbix server. È *supportato* solo se l'esecuzione degli script globali è abilitata su Zabbix server; 2 - *(predefinito)* esecuzione su Zabbix server o proxy. Comportamento della proprietà: - *supportata* se `type` è impostato su "Script"
menu_path	string	Cartelle separate da barre che formano una navigazione simile a un menu nel frontend quando si fa clic su un host o un evento. Comportamento della proprietà: - *supportata* se `scope` è impostato su "manual host action" o "manual event action"
authtype	integer	Metodo di autenticazione utilizzato per gli script SSH. Valori possibili: 0 - password; 1 - chiave pubblica. Comportamento della proprietà: - *supportata* se `type` è impostato su "SSH"
username	string	Nome utente utilizzato per l'autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SSH" o "TELNET"
password	string	Password utilizzata per gli script SSH con autenticazione tramite password e per gli script TELNET. Comportamento della proprietà: - *supportata* se `type` è impostato su "SSH" e `authtype` è impostato su "password", oppure se `type` è impostato su "TELNET"
publickey	string	Nome del file della chiave pubblica utilizzato per gli script SSH con autenticazione tramite chiave pubblica. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SSH" e `authtype` è impostato su "public key"
privatekey	string	Nome del file della chiave privata utilizzato per gli script SSH con autenticazione tramite chiave pubblica. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SSH" e `authtype` è impostato su "public key"
port	string	Numero di porta utilizzato per gli script SSH e TELNET. Comportamento della proprietà: - *supportata* se `type` è impostato su "SSH" o "TELNET"
groupid	ID	ID del gruppo di host su cui è possibile eseguire lo script. Se impostato su "0", lo script sarà disponibile per tutti i gruppi di host. Predefinito: 0.

Property	Type	Description
usrgrpid	ID	ID del gruppo di utenti autorizzato a eseguire lo script. Se impostato su ”0”, lo script sarà disponibile per tutti i gruppi di utenti. Predefinito: 0. Comportamento della proprietà: - *supportata* se `scope` è impostato su ”manual host action” o ”manual event action”
host_access	integer	Permessi sull’host necessari per eseguire lo script. Valori possibili: 2 - *(predefinito)* lettura; 3 - scrittura. Comportamento della proprietà: - *supportata* se `scope` è impostato su ”manual host action” o ”manual event action”
confirmation	string	Testo della finestra popup di conferma. La finestra popup viene visualizzata quando si tenta di eseguire lo script dal frontend di Zabbix. Comportamento della proprietà: - *supportata* se `scope` è impostato su ”manual host action” o ”manual event action”
timeout	string	Timeout di esecuzione dello script webhook in secondi. Sono supportati i suffissi temporali (ad esempio, `30s`, `1m`). Valori possibili: 1-60s. Predefinito: 30s. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”Webhook”
parameters	array	Array dei parametri di input del webhook. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Webhook”
description	string	Descrizione dello script.
url	string	URL definito dall’utente. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”URL”
new_window	integer	Apri l’URL in una nuova finestra. Valori possibili: 0 - No; 1 - *(predefinito)* Sì. Comportamento della proprietà: - *supportata* se `type` è impostato su ”URL”
manualinput	integer	Indica se lo script accetta input forniti dall’utente. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato; Comportamento della proprietà: - *supportata* se `scope` è impostato su ”manual host action” o ”manual event action”

Property	Type	Description
manualinput_prompt	string	Testo del prompt per l'input manuale. Comportamento della proprietà: - *obbligatoria* se `manualinput` è impostato su "Enabled"
manualinput_validator	string	Campo stringa utilizzato per convalidare l'input fornito dall'utente. La stringa è costituita da un'espressione regolare oppure da un insieme di valori separati da virgole. Comportamento della proprietà: - *obbligatoria* se `manualinput` è impostato su "Enabled"
manualinput_validator_type	integer	Determina il tipo di input utente previsto. Valori possibili: 0 - *(predefinito)* Stringa. Indica che manualinput_validator deve essere trattato come un'espressione regolare; 1 - Elenco. Indica che manualinput_validator deve essere trattato come un elenco separato da virgole di possibili valori di input.
manualinput_default_value	string	Valore predefinito per il completamento automatico dell'input utente. Comportamento della proprietà: - *supportata* se `manualinput` è impostato su "Enabled"
		Comportamento della proprietà: - *supportata* se `manualinput_validator_type` è impostato su "String"

Parametri del webhook

I parametri passati allo script webhook quando viene chiamato hanno le seguenti proprietà.

Proprietà	Tipo	Descrizione
name	string	Nome del parametro. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro. Supporta le macro.

Debug

Informazioni di debug dello script webhook eseguito. L'oggetto debug ha le seguenti proprietà.

Proprietà	Type	Descrizione
logs	array	Array di voci di log.
ms	string	Durata dell'esecuzione dello script in millisecondi.

Voce di log

L'oggetto voce di log ha le seguenti proprietà.

Proprietà	Type	Descrizione
level	integer	Livello di log.
ms	string	Il tempo trascorso in millisecondi dall'esecuzione dello script prima che la voce di log fosse aggiunta.
message	string	Messaggio di log.

script.create

Descrizione

`object script.create(object/array scripts)`

Questo metodo consente di creare nuovi script.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Script da creare.

Il metodo accetta script con le proprietà standard degli script.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID degli scripts creati sotto la proprietà `scriptids`. L'ordine degli ID restituiti corrisponde a ordine degli scripts passati.

Esempi

Creare uno script webhook

Creare uno script webhook che invii una richiesta HTTP a un servizio esterno.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
```

```
    "id": 1
}
```

Creare uno script SSH

Creare uno script SSH con autenticazione a chiave pubblica che possa essere eseguito su un host e che abbia un menu contestuale.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Creare uno script personalizzato

Creare uno script personalizzato che riavvierà un server. Lo script richiederà l'accesso in scrittura al host e chiederà all'utente un input manuale. Dopo l'invio corretto dell'input, lo script mostrerà un messaggio di conferma nel frontend.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

Creare uno script di tipo URL

Creare uno script di tipo URL per l'ambito host che rimanga nella stessa finestra e abbia un testo di conferma.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Creare uno script di tipo URL con input manuale

Creare uno script di tipo URL per l'ambito evento che si apra in una nuova finestra e abbia input manuale.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script with manual input",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Select a page to open:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

Fonte

CScript::create() in *ui/include/classes/api/services/CScript.php*.

script.delete

Descrizione

`object script.delete(array scriptIds)`

Questo metodo consente di eliminare gli script.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID degli script da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli script eliminati sotto la proprietà `scriptids`.

Esempi

Eliminare più script

Eliminare due script.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Fonte

CScript::delete() in *ui/include/classes/api/services/CScript.php*.

script.execute

Descrizione

`object script.execute(object parameters)`

Questo metodo consente di eseguire uno script su un host o su un evento. Fanno eccezione gli script di tipo URL; questi non sono eseguibili.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri contenenti l'ID dello script da eseguire, oppure l'ID dell'host o l'ID dell'evento e il valore manualinput.

Parametro	Type	Descrizione
scriptid	ID	ID dello script da eseguire. Comportamento del parametro: - *obbligatorio*
hostid	ID	ID dell'host su cui eseguire lo script. Comportamento del parametro: - *obbligatorio* se `eventid` non è impostato
eventid	ID	ID dell'evento su cui eseguire lo script. Comportamento del parametro: - *obbligatorio* se `hostid` non è impostato
manualinput	string	Valore fornito dall'utente con cui eseguire lo script, in sostituzione della macro {MANUALINPUT}.

Valori restituiti

`(object)` Restituisce il risultato dell'esecuzione dello script.

Proprietà	Type	Descrizione
response	string	Indica se lo script è stato eseguito correttamente. Valore possibile - `success`.
value	string	Output dello script.
debug	object	Contiene un oggetto debug se viene eseguito uno script webhook. Per altri tipi di script, contiene un oggetto vuoto.

Esempi

Eseguire uno script webhook

Eseguire uno script webhook che invia una richiesta HTTP a un servizio esterno.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

Risposta:


```
{
    "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
}
```


Eseguire uno script personalizzato

Eseguire uno script ”ping” su un host.

Richiesta:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```


Risposta:


```
{
    "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
}
```


Eseguire uno script personalizzato con input manuale

Eseguire uno script ”ping” con il comando ”ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac” su un host.

Richiesta:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Risposta:

```
{
    "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
}
```


Fonte

CScript::execute() in *ui/include/classes/api/services/CScript.php*.

script.get

Descrizione

`integer/array script.get(object parameters)`

Il metodo consente di recuperare gli script in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
groupids	ID/array	Restituisce solo gli script che possono essere eseguiti sui gruppi di host specificati.
hostids	ID/array	Restituisce solo gli script che possono essere eseguiti sugli host specificati.
scriptids	ID/array	Restituisce solo gli script con gli ID specificati.
usrgrpids	ID/array	Restituisce solo gli script che possono essere eseguiti dagli utenti nei gruppi di utenti specificati.
selectHostGroups	query	Restituisce una proprietà `hostgroups` con i gruppi di host sui quali lo script può essere eseguito.
selectHosts	query	Restituisce una proprietà `hosts` con gli host sui quali lo script può essere eseguito.
selectActions	query	Restituisce una proprietà `actions` con le azioni a cui lo script è associato.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `scriptid`, `name`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

(integer/array) Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare tutti gli script

Recupera tutti gli script configurati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
```

```
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Webhook",
            "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "5",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
```

```
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": [
                {
                    "name": "token",
                    "value": "{$WEBHOOK.TOKEN}"
                },
                {
                    "name": "host",
                    "value": "{HOST.HOST}"
                },
                {
                    "name": "v",
                    "value": "2.2"
                }
            ]
        },
        {
            "scriptid": "5",
            "name": "URL",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Go to {HOST.NAME}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action=latest.view&hostids[]={HOST.ID}",
            "new_window": "0",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "6",
            "name": "URL with user input",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
```

```
            "groupid": "0",
            "description": "",
            "confirmation": "Open zabbix page {MANUALINPUT}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
            "new_window": "0",
            "manualinput": "1",
            "manualinput_prompt": "Select a page to open:",
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```


Vedi anche

- Host
- Gruppo host

Fonte

CScript::get() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Descrizione

`object script.getscriptsbyevents(object parameters)`

Questo metodo consente di recuperare tutti gli script disponibili per l'evento specificato oppure uno script specifico se viene fornito l'ID dello script. Quando viene fornito manualinput, sostituisce la macro `{MANUALINPUT}` con il valore specificato.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Il metodo accetta un oggetto o un array di oggetti con i seguenti parametri.

Parametro	Tipo	Descrizione
eventid	ID	ID dell'evento per cui restituire gli script. Deve essere univoco. Comportamento dei parametri: - *obbligatorio*
scriptid	ID	ID dello script da restituire.
manualinput	string	Valore della macro {MANUALINPUT} fornita dall'utente.

Valori restituiti

(`object`) Restituisce un oggetto con gli ID evento come proprietà e array di script disponibili come valori. Se viene fornito l'ID dello script, il valore associato è un array contenente lo script specifico.

> **Note:**
> Il metodo espanderà automaticamente le macro nel testo di `confirmation`, nel testo del prompt di `manualinput` e in `url`.
> Se viene fornito il parametro manualinput, la macro {MANUALINPUT} verrà risolta nel valore specificato.

Esempi

Recuperare gli script in base agli ID evento

Recupera tutti gli script disponibili per gli eventi "632" e "614".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Rileva il sistema operativo",
                "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": "Apri la pagina di Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Aprire davvero la pagina *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Pagina di Zabbix da aprire:",
            "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": "Rileva il sistema operativo",
            "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": "Apri la pagina di Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Aprire davvero la pagina *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Pagina di Zabbix da aprire:",
            "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
}
```

Recuperare uno script specifico con il valore manualinput.

Recupera lo script con ID ”4” sull’evento ”632” con il valore manualinput ”dashboard.view”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Fonte

CScript::getScriptsByEvents() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Descrizione

`object script.getscriptsbyhosts(object parameters)`

Questo metodo consente di recuperare tutti gli script disponibili sull'host specificato oppure uno script specifico se viene fornito l'ID dello script. Quando viene fornito manualinput, sostituisce la macro `{MANUALINPUT}` con il valore specificato.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Il metodo accetta un oggetto o un array di oggetti con i seguenti parametri.

Parametro	Type	Descrizione
hostid	ID	ID dell'host per cui restituire gli script. Deve essere univoco. Comportamento del parametro: - *obbligatorio*
scriptid	ID	ID dello script da restituire.
manualinput	string	Valore della macro {MANUALINPUT} fornita dall'utente.

Valori restituiti

`(object)` Restituisce un oggetto con gli ID degli host come proprietà e array di script disponibili come valori. Se viene fornito l'ID dello script, il valore associato è un array contenente lo script specifico.

> **Note:**
> Il metodo espanderà automaticamente le macro nel testo di `confirmation`, nel testo del prompt di `manualinput` e nell'`url`.
> Se viene fornito il parametro manualinput, la macro {MANUALINPUT} verrà risolta nel valore specificato.

Esempi

Recuperare gli script per ID host

Recupera tutti gli script disponibili sugli host "30079" e "30073".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
       }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Rileva il sistema operativo",
                "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": "Apri la pagina Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Sei sicuro di voler aprire la pagina *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": "Pagina Zabbix da aprire:",
            "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": "Rileva il sistema operativo",
                "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": "Apri la pagina Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Sei sicuro di voler aprire la pagina *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": "Pagina Zabbix da aprire:",
            "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
}
```

Recuperare uno script specifico con valore manualinput.

Recupera lo script con ID "4" sull'host "30079" con valore manualinput "dashboard.view".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Fonte

CScript::getScriptsByHosts() in *ui/include/classes/api/services/CScript.php*.

script.update

Descrizione

`object script.update(object/array scripts)`

Questo metodo consente di aggiornare gli script esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Proprietà dello script da aggiornare.

La proprietà `scriptid` deve essere definita per ciascuno script, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate. Fa eccezione la modifica della proprietà `type` da 5 (Webhook) a un altro valore: la proprietà `parameters` verrà ripulita.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli script aggiornati sotto la proprietà `scriptids`.

Esempi

Modificare il comando dello script

Modificare il comando dello script in "/bin/ping -c 10 {HOST.CONN} 2>&1".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Modificare il comando dello script e aggiungere input manuale

Modificare il comando dello script in "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1".

Richiesta:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "Specificare il numero di pacchetti ICMP da inviare con il comando ping",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CScript::update() in *ui/include/classes/api/services/CScript.php*.

Servizio

Questa classe è progettata per funzionare con i servizi aziendali/dell'infrastruttura IT.

Riferimenti agli oggetti:

- Service
- Status rule
- Service tag
- Service alarm
- Problem tag

Metodi disponibili:

- service.create - crea nuovi servizi
- service.delete - elimina servizi
- service.get - recupera servizi
- service.update - aggiorna servizi

Oggetto servizio

I seguenti oggetti sono direttamente correlati all'API `service`.

Service

L'oggetto service ha le seguenti proprietà.

Property	Type	Description
serviceid	ID	ID del service. Property behavior: - *read-only* - *required* per le operazioni di aggiornamento

Property	Type	Description
algorithm	integer	Regola di calcolo dello stato. Applicabile solo se esistono service figlio. Valori possibili: 0 - imposta lo stato su OK; 1 - il più critico se tutti i figli hanno problemi; 2 - il più critico tra i service figlio. Property behavior: - *required* per le operazioni di creazione
name	string	Nome del service. Property behavior: - *required* per le operazioni di creazione
sortorder	integer	Posizione del service usata per l'ordinamento. Valori possibili: 0-999. Property behavior: - *required* per le operazioni di creazione
weight	integer	Peso del service. Valori possibili: 0-1000000. Predefinito: 0.
propagation_rule	integer	Regola di propagazione dello stato. Valori possibili: 0 - *(predefinito)* propaga lo stato del service così com'è, senza alcuna modifica; 1 - aumenta lo stato propagato di un dato `propagation_value` (di 1-5 severità); 2 - diminuisce lo stato propagato di un dato `propagation_value` (di 1-5 severità); 3 - ignora questo service - lo stato non viene propagato al service padre; 4 - imposta uno stato fisso del service usando un dato `propagation_value`. Property behavior: - *required* se `propagation_value` è impostato

Property	Type	Description
propagation_value	integer	Valore di propagazione dello stato. Valori possibili se `propagation_rule` è impostato su ”0” o ”3”: 0 - Non classificato. Valori possibili se `propagation_rule` è impostato su ”1” o ”2”: 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Valori possibili se `propagation_rule` è impostato su ”4”: -1 - OK; 0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Property behavior: - *required* se `propagation_rule` è impostato
status	integer	Indica se il service si trova nello stato OK o di problema. Se il service si trova in uno stato di problema, `status` è uguale alla gravità del problema più critico oppure allo stato più alto di un service figlio in stato di problema. Se il service si trova nello stato OK, `status` è uguale a: -1. Property behavior: - *read-only*
description	string	Descrizione del service.
uuid	string	Identificatore univoco universale, usato per collegare i service importati a quelli già esistenti. Generato automaticamente, se non specificato.
created_at	integer	Timestamp Unix in cui il service è stato creato.
readonly	integer	Accesso al service. Valori possibili: 0 - Lettura e scrittura; 1 - Sola lettura. Property behavior: - *read-only*

Regola di stato

L’oggetto regola di stato ha le seguenti proprietà.

Proprietà	Type	Descrizione
type	integer	Condizione per impostare lo stato (Nuovo stato). Valori possibili: 0 - se almeno (N) servizi figli hanno lo stato (Stato) o superiore; 1 - se almeno il (N%) dei servizi figli ha lo stato (Stato) o superiore; 2 - se meno di (N) servizi figli hanno lo stato (Stato) o inferiore; 3 - se meno del (N%) dei servizi figli ha lo stato (Stato) o inferiore; 4 - se il peso dei servizi figli con stato (Stato) o superiore è almeno (W); 5 - se il peso dei servizi figli con stato (Stato) o superiore è almeno (N%); 6 - se il peso dei servizi figli con stato (Stato) o inferiore è inferiore a (W); 7 - se il peso dei servizi figli con stato (Stato) o inferiore è inferiore a (N%). Dove: - N (W) è `limit_value`; - (Stato) è `limit_status`; - (Nuovo stato) è `new_status`. Comportamento della proprietà: - *required*
limit_value	integer	Valore limite. Valori possibili: - per N e W: 1-100000; - per N%: 1-100. Comportamento della proprietà: - *required*
limit_status	integer	Stato limite. Valori possibili: -1 - OK; 0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Comportamento della proprietà: - *required*
new_status	integer	Valore del nuovo stato. Valori possibili: 0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Medio; 4 - Alto; 5 - Disastro. Comportamento della proprietà: - *required*

Tag del servizio

L'oggetto tag del servizio ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag del servizio. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del tag del servizio.

Allarme del servizio

Note:
Gli allarmi del servizio non possono essere creati, aggiornati o eliminati direttamente tramite la Zabbix API.

Gli oggetti allarme del servizio rappresentano una modifica dello stato di un servizio. Ha le seguenti proprietà.

Property	Type	Description
clock	timestamp	Ora in cui si è verificata la modifica dello stato del servizio.
value	integer	Stato del servizio. Fare riferimento alla proprietà `status` del servizio per un elenco dei valori possibili.

Tag del problema

I tag del problema consentono di collegare i servizi con gli eventi di problema. L'oggetto tag del problema ha le seguenti proprietà.

Property	Type	Description
tag	string	Nome del tag del problema. Property behavior: - *required*
operator	integer	Operatore della condizione di mapping. Valori possibili: 0 - *(default)* Uguale; 2 - Contiene.
value	string	Valore del tag del problema.

service.create

Descrizione

`object service.create(object/array services)`

Questo metodo consente di creare nuovi servizi.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` servizi da creare.

Oltre alle proprietà standard del servizio, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
children	array	Servizi figli da collegare al servizio. I servizi figli devono avere definita solo la proprietà `serviceid`.

Parametro	Tipo	Descrizione
parents	array	Servizi padre da collegare al servizio. I servizi padre devono avere definita solo la proprietà `serviceid`.
tags	array	Tag di servizio da creare per il servizio.
problem_tags	array	Tag di problema da creare per il servizio.
status_rules	array	Regole di stato da creare per il servizio.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID dei servizi creati sotto la proprietà `serviceids`. L'ordine degli ID restituiti corrisponde a ordine dei servizi passati.

Esempi

Creazione di un servizio

Creare un servizio che verrà impostato nello stato di problema se almeno un figlio presenta un problema.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CService::create() in *ui/include/classes/api/services/CService.php*.

service.delete

Descrizione

`object service.delete(array serviceIds)`

Questo metodo consente di eliminare i servizi.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID dei servizi da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei servizi eliminati sotto la proprietà `serviceids`.

Esempi

Eliminazione di più servizi

Eliminare due servizi.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CService::delete() in *ui/include/classes/api/services/CService.php*.

service.get

Descrizione

`integer/array service.get(object parameters)`

Il metodo consente di recuperare i servizi in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
serviceids	ID/array	Restituisce solo i servizi con gli ID specificati.
parentids	ID/array	Restituisce solo i servizi collegati ai servizi padre specificati.
deep_parentids	flag	Restituisce tutti i servizi figli diretti e indiretti. Usato insieme a `parentids`.
childids	ID/array	Restituisce solo i servizi collegati ai servizi figli specificati.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.

Parametro	Type	Descrizione
tags	object/array	Restituisce solo i servizi con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i servizi. Valori possibili di operator: 0 - *(predefinito)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
problem_tags	object/array	Restituisce solo i servizi con i tag problema specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i servizi. Valori possibili di operator: 0 - *(predefinito)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
without_problem_tags	flag	Restituisce solo i servizi senza tag problema.
slaids	ID/array	Restituisce solo i servizi collegati agli SLA specificati.
selectChildren	query	Restituisce una proprietà `children` con i servizi figli. Supporta `count`.
selectParents	query	Restituisce una proprietà `parents` con i servizi padre. Supporta `count`.
selectTags	query	Restituisce una proprietà `tags` con i tag del servizio. Supporta `count`.
selectProblemEvents	query	Restituisce una proprietà `problem_events` con un array di oggetti evento problema. L'oggetto evento problema ha le seguenti proprietà: `eventid` - `(ID)` ID evento; `severity` - `(string)` severità corrente dell'evento; `name` - `(string)` nome dell'evento risolto. Supporta `count`.
selectProblemTags	query	Restituisce una proprietà `problem_tags` con i tag problema. Supporta `count`.
selectStatusRules	query	Restituisce una proprietà `status_rules` con le regole di stato. Supporta `count`.
selectStatusTimeline	object/array	Restituisce una proprietà `status_timeline` contenente i cambiamenti di stato del servizio per i periodi specificati. Formato `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - dove `period_from` è una data di inizio (inclusiva; timestamp intero) e `period_to` è una data di fine (esclusiva; timestamp intero) per il periodo di interesse. Restituisce un array di voci contenenti una proprietà `start_value` e un array `alarms` per i cambiamenti di stato all'interno dei periodi specificati.

Parametro	Type	Descrizione
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero di tutti i servizi

Recupera tutti i dati relativi a tutti i servizi e alle loro dipendenze.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Fonte

CService::get() in *ui/include/classes/api/services/CService.php*.

service.update

Descrizione

`object service.update(object/array services)`

Questo metodo consente di aggiornare i servizi esistenti.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` proprietà del servizio da aggiornare.

La proprietà `serviceid` deve essere definita per ciascun servizio, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del servizio, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
children	array	Servizi figli per sostituire gli attuali servizi figli. I servizi figli devono avere definita solo la proprietà `serviceid`.
parents	array	Servizi padre per sostituire gli attuali servizi padre. I servizi padre devono avere definita solo la proprietà `serviceid`.
tags	array	Tag del servizio per sostituire gli attuali tag del servizio.
problem_tags	array	Tag del problema per sostituire gli attuali tag del problema.
status_rules	array	Regole di stato per sostituire le attuali regole di stato.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei servizi aggiornati sotto la proprietà `serviceids`.

Esempi

Impostazione del parent per un servizio

Impostare il servizio con ID ”3” come parent del servizio con ID ”5”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
```

```
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Aggiunta di un periodo di inattività pianificato

Aggiungere un periodo di inattività per il servizio con ID ”4”, pianificato settimanalmente dal lunedì alle 22:00 fino al martedì alle 10:00.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

Fonte

CService::update() in *ui/include/classes/api/services/CService.php*.

Servizio di pulizia

Questa classe è progettata per lavorare con il servizio di pulizia.

Riferimenti degli oggetti:

- Servizio di pulizia

Metodi disponibili:

- housekeeping.get - recupera il servizio di pulizia
- housekeeping.update - aggiorna il servizio di pulizia

Oggetto del servizio di pulizia

I seguenti oggetti sono direttamente correlati all'API `housekeeping`.

Pulizia

L'oggetto delle impostazioni presenta le seguenti proprietà.

Property	Type	Description
hk_events_mode	integer	Abilita la pulizia interna per eventi e avvisi. Valori possibili: 0 - Disabilitata; 1 - *(predefinita)* Abilitata.
hk_events_trigger	string	Periodo di conservazione dei dati dei trigger. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 365d.
hk_events_service	string	Periodo di conservazione dei dati dei servizi. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 1d.
hk_events_internal	string	Periodo di conservazione dei dati interni. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 1d.
hk_events_discovery	string	Periodo di conservazione dei dati di rilevamento della rete. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 1d.
hk_events_autoreg	string	Periodo di conservazione dei dati di autoregistrazione. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 1d.
hk_services_mode	integer	Abilita la pulizia interna per i servizi. Valori possibili: 0 - Disabilitata; 1 - *(predefinita)* Abilitata.
hk_services	string	Periodo di conservazione dei dati dei servizi. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 365d.
hk_audit_mode	integer	Abilita la pulizia interna per l'audit. Valori possibili: 0 - Disabilitata; 1 - *(predefinita)* Abilitata.
hk_audit	string	Periodo di conservazione dei dati di audit. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 31d.

Property	Type	Description
hk_sessions_mode	integer	Abilita la pulizia interna per le sessioni. Valori possibili: 0 - Disabilitata; 1 - *(predefinita)* Abilitata.
hk_sessions	string	Periodo di conservazione dei dati delle sessioni. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 31d.
hk_history_mode	integer	Abilita la pulizia interna per la cronologia. Valori possibili: 0 - Disabilitata; 1 - *(predefinita)* Abilitata.
hk_history_global	integer	Sostituisce il periodo della cronologia dell'item. Valori possibili: 0 - Non sostituire; 1 - *(predefinito)* Sostituire.
hk_history	string	Periodo di conservazione dei dati della cronologia. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 31d.
hk_trends_mode	integer	Abilita la pulizia interna per i trend. Valori possibili: 0 - Disabilitata; 1 - *(predefinita)* Abilitata.
hk_trends_global	integer	Sostituisce il periodo dei trend dell'item. Valori possibili: 0 - Non sostituire; 1 - *(predefinito)* Sostituire.
hk_trends	string	Periodo di conservazione dei dati dei trend. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 365d.
db_extension	string	Flag di configurazione dell'estensione DB. Se questo flag è impostato su "timescaledb", il server modifica il proprio comportamento per la pulizia e l'eliminazione degli item. Comportamento della proprietà: - *sola lettura*
compression_availability	integer	Indica se la compressione dei dati è supportata dal database o dalla relativa estensione. Valori possibili: 0 - Non disponibile; 1 - Disponibile. Comportamento della proprietà: - *sola lettura*
compression_status	integer	Abilita la compressione TimescaleDB per la cronologia e i trend. Valori possibili: 0 - *(predefinito)* Disattivata; 1 - Attivata.

Property	Type	Description
compress_older	string	Comprimi i record della cronologia e dei trend più vecchi del periodo specificato. Accetta un numero di secondi (ad es. 1, 60, 300) oppure un valore con un suffisso temporale (ad es. 30s, 1m, 2h, 1d). Predefinito: 7d.

housekeeping.get

Descrizione

`object housekeeping.get(object parameters)`

Il metodo consente di recuperare un oggetto di housekeeping in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta un solo parametro.

Parametro	Tipo	Descrizione
output	query	Questo parametro è descritto nel commento di riferimento.

Valori restituiti

"(object)" Restituisce l'oggetto di pulizia.

Esempi

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Fonte

CHousekeeping ::get() in *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

Descrizione

`object housekeeping.update(object housekeeping)`

Questo metodo consente di aggiornare le impostazioni di housekeeping esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente .

Parametri

`(object)` Proprietà di housekeeping da aggiornare.

Valori restituiti

”(array)” Restituisce un array con i nomi dei parametri aggiornati.

Esempi

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

Fonte

CHousekeeping::update() in *ui/include/classes/api/services/CHousekeeping.php*.

Servizio individuato

Questa classe è progettata per funzionare con i servizi individuati.

Riferimenti agli oggetti:

- Servizio individuato

Metodi disponibili:

- dservice.get - recupera i servizi individuati

Oggetto del servizio discoverato

I seguenti oggetti sono direttamente correlati all'API `dservice`.

Servizio rilevato

Note:
I servizi rilevati vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

L'oggetto servizio rilevato contiene informazioni su un servizio rilevato da una regola di rilevamento di rete su un host. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
dserviceid	ID	ID del servizio rilevato.
dcheckid	ID	ID del controllo di rilevamento utilizzato per individuare il servizio.
dhostid	ID	ID dell'host rilevato che esegue il servizio.
dns	string	DNS dell'host che esegue il servizio.
ip	string	Indirizzo IP dell'host che esegue il servizio.
lastdown	timestamp	Ora in cui il servizio rilevato è andato giù l'ultima volta.
lastup	timestamp	Ora in cui il servizio rilevato è tornato su l'ultima volta.
port	integer	Numero di porta del servizio.
status	integer	Stato del servizio. Valori possibili: 0 - servizio attivo; 1 - servizio non attivo.
value	string	Valore restituito dal servizio durante l'esecuzione di un controllo di rilevamento Zabbix agent, SNMPv1, SNMPv2 o SNMPv3.

dservice.get

Descrizione

`integer/array dservice.get(object parameters)`

Il metodo consente di recuperare i servizi rilevati in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
dserviceids	ID/array	Restituisce solo i servizi rilevati con gli ID specificati.
dhostids	ID/array	Restituisce solo i servizi rilevati che appartengono agli host rilevati specificati.

Parametro	Type	Descrizione
dcheckids	ID/array	Restituisce solo i servizi rilevati che sono stati individuati dai controlli di discovery specificati.
druleids	ID/array	Restituisce solo i servizi rilevati che sono stati individuati dalle regole di discovery specificate.
selectDRules	query	Restituisce una proprietà `drules` con un array delle regole di discovery che hanno individuato il servizio.
selectDHosts	query	Restituisce una proprietà `dhosts` con un array degli host rilevati a cui appartiene il servizio.
selectHosts	query	Restituisce una proprietà `hosts` con gli host con lo stesso indirizzo IP e proxy del servizio. Supporta `count`.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectHosts` - il risultato sarà ordinato per `hostid`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare i servizi rilevati su un host

Recupera tutti i servizi rilevati individuati sull'host rilevato "11".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```


Vedi anche

- Host rilevato
- Controllo di discovery
- Host

Fonte

CDService::get() in *ui/include/classes/api/services/CDService.php*.

SLA

Questa classe è progettata per funzionare con gli oggetti SLA (Service Level Agreement) utilizzati per stimare le prestazioni dell'infrastruttura IT e dei servizi aziendali.

Riferimenti agli oggetti:

- SLA
- Pianificazione SLA
- Tempo di inattività escluso SLA
- Tag di servizio SLA

Metodi disponibili:

- sla.create - crea nuovi SLA
- sla.delete - elimina gli SLA
- sla.get - recupera gli SLA
- sla.getsli - recupera i dati del Service Level Indicator (SLI) per gli SLA
- sla.update - aggiorna gli SLA

Oggetto SLA

I seguenti oggetti sono direttamente correlati all'API `sla` (Service Level Agreement).

SLA

L'oggetto SLA ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
slaid	ID	ID dello SLA. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome dello SLA. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
period	integer	Periodo di reporting dello SLA. Valori possibili: 0 - giornaliero; 1 - settimanale; 2 - mensile; 3 - trimestrale; 4 - annuale. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
slo	float	Obiettivo minimo accettabile del livello di servizio espresso in percentuale. Se l'Indicatore del livello di servizio (SLI) scende al di sotto di questo valore, lo SLA viene considerato in stato di problema/non soddisfatto. Valori possibili: 0-100 (fino a 4 cifre decimali). Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
effective_date	integer	Data di entrata in vigore dello SLA. Valori possibili: timestamp della data in UTC.
timezone	string	Fuso orario di reporting, ad esempio: `Europe/London`, `UTC`. Per l'elenco completo dei fusi orari supportati, fare riferimento alla documentazione PHP. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
status	integer	Stato dello SLA. Valori possibili: 0 - *(predefinito)* SLA disabilitato; 1 - SLA abilitato.
description	string	Descrizione dello SLA.

Pianificazione SLA

L'oggetto di pianificazione SLA definisce i periodi in cui i servizi collegati sono pianificati per essere operativi. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
period_from	integer	Ora di inizio del periodo di tempo settimanale ricorrente (inclusa). Valori possibili: numero di secondi (conteggiati a partire da domenica). Comportamento della proprietà: - *obbligatorio*

Proprietà	Type	Descrizione
period_to	integer	Ora di fine del periodo di tempo settimanale ricorrente (esclusa). Valori possibili: numero di secondi (conteggiati a partire da domenica). Comportamento della proprietà: - *obbligatorio*

Tempo di inattività escluso dall'SLA

L'oggetto del tempo di inattività escluso definisce i periodi in cui i servizi collegati sono pianificati per essere fuori servizio, senza influire sullo SLI, ad esempio durante una manutenzione pianificata. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
name	string	Nome del periodo di inattività escluso. Comportamento della proprietà: - *obbligatoria*
period_from	integer	Ora di inizio del periodo di inattività escluso (inclusa). Valori possibili: timestamp. Comportamento della proprietà: - *obbligatoria*
period_to	integer	Ora di fine del periodo di inattività escluso (esclusa). Valori possibili: timestamp. Comportamento della proprietà: - *obbligatoria*

Etichetta di servizio SLA

L'oggetto tag di servizio SLA collega i servizi da includere nei calcoli per lo SLA. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag del servizio SLA. Comportamento della proprietà: - *obbligatoria*
operator	integer	Operatore del tag del servizio SLA. Valori possibili: 0 - *(predefinito)* Uguale; 2 - Contiene.
value	string	Valore del tag del servizio SLA.

sla.create

Descrizione

`object sla.create(object/array SLAs)`

Questo metodo consente di creare nuovi oggetti SLA.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

(`object/array`) Oggetti SLA da creare.

Oltre alle proprietà SLA standard, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
service_tags	array	Tag di servizio SLA da creare per lo SLA. Comportamento del parametro: - *obbligatorio*
schedule	array	Pianificazione SLA da creare per lo SLA. Specificare un parametro vuoto sarà interpretato come una pianificazione 24x7. Predefinito: pianificazione 24x7.
excluded_downtimes	array	Tempi di inattività esclusi SLA da creare per lo SLA.

Valori restituiti

(`object`) Restituisce un oggetto contenente gli ID degli SLA creati nella proprietà `slaids`. L'ordine degli ID restituiti corrisponde all'ordine degli SLA passati.

Esempi

Creazione di un SLA

Indicare di creare una voce SLA per: * il monitoraggio dell'uptime dei servizi correlati al motore SQL; * una pianificazione personalizzata di tutti i giorni feriali, escludendo l'ultima ora del sabato; * una data di entrata in vigore corrispondente all'ultimo giorno dell'anno 2022; * con un periodo di inattività pianificata della durata di 1 ora e 15 minuti, a partire dalla mezzanotte del 4 luglio; * il calcolo del report settimanale dell'SLA sarà attivo; * l'SLO minimo accettabile sarà del 99,9995%.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Uptime del database",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Fornire un uptime eccellente per i principali motori di database.",
            "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": "Distribuzione dell'aggiornamento della versione del software",
                    "period_from": "1648760400",
```

```
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CSla::create() in *ui/include/classes/api/services/CSla.php*.

sla.delete

Descrizione

`object sla.delete(array slaids)`

Questo metodo consente di eliminare le voci SLA.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

IDs (`array`) degli SLA da eliminare.

Restituisce i valori

(`object`) Restituisce un oggetto contenente gli ID degli SLA eliminati sotto la proprietà `slaids`.

Esempi

Eliminazione di più SLA

Eliminare due voci SLA.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
```

```
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CSla::delete() in *ui/include/classes/api/services/CSla.php*.

sla.get

Descrizione

`integer/array sla.get(object parameters)`

Il metodo consente di recuperare gli oggetti SLA in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
slaids	ID/array	Restituisce solo gli SLA con gli ID specificati.
serviceids	ID/array	Restituisce solo gli SLA corrispondenti ai servizi specifici.
selectSchedule	query	Restituisce una proprietà `schedule` con le pianificazioni SLA. Supporta `count`.
selectExcludedDowntimes	query	Restituisce una proprietà `excluded_downtimes` con i tempi di inattività esclusi dell'SLA. Supporta `count`.
selectServiceTags	query	Restituisce una proprietà `service_tags` con i tag di servizio dell'SLA. Supporta `count`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero di tutti gli SLA

Recupera tutti i dati relativi a tutti gli SLA e alle loro proprietà.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "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
}
```

Fonte

CSla:get() in *ui/include/classes/api/services/CSla.php*.

sla.getsli

Descrizione

`object sla.getsli(object parameters)`

Questo metodo consente di calcolare i dati dell'Indicatore di livello di servizio (SLI) per un Accordo sul livello di servizio (SLA).

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri contenenti l'ID dello SLA, i periodi di reporting e, facoltativamente, gli ID dei servizi per i quali calcolare lo SLI.

Parametro	Type	Descrizione
slaid	ID	ID dello SLA per il quale restituire le informazioni sulla disponibilità. Comportamento del parametro: - *obbligatorio*
period_from	timestamp	Timestamp iniziale (incluso) per il quale riportare lo SLI. Valori possibili: timestamp Unix.
period_to	timestamp	Timestamp finale (incluso) per il quale riportare lo SLI. Valori possibili: timestamp Unix.
periods	integer	Numero di periodi da riportare. Valori possibili: 1-100
serviceids	ID/array	ID dei servizi per i quali restituire lo SLI.

Suddivisione dei periodi

La tabella seguente mostra la disposizione delle sezioni di periodo restituite in base alle combinazioni dei parametri.

Note:
I periodi restituiti non precedono il primo periodo disponibile (in base alla data di entrata in vigore dello SLA) e non superano il periodo corrente.

Parametri			Periodi restituiti
period_from	**period_to**	**periods**	
-	-	-	Ultimi 20 periodi, incluso quello corrente.
-	-	N	Ultimi N periodi.
-	specified	-	Ultimi 20 periodi prima di `period_to`.
-	specified	N	Ultimi N periodi prima di `period_to`.
specified	-	-	Primi 20 periodi a partire da `period_from`.
specified	-	N	Primi N periodi a partire da `period_from`.
specified	specified	-	Fino a 100 periodi nell'intervallo specificato.
specified	specified	N	N periodi nell'intervallo specificato.

Valori restituiti

`(object)` Restituisce i risultati del calcolo.

Proprietà	Type	Descrizione
periods	array	Elenco dei periodi riportati. Ciascun periodo riportato è rappresentato come un oggetto composto da: - `period_from` - (`timestamp`) Timestamp iniziale (incluso) del periodo riportato. - `period_to` - (`timestamp`) Timestamp finale (escluso) del periodo riportato. I periodi sono ordinati per `period_from`, con i periodi più vecchi visualizzati per primi.
serviceids	array	Elenco degli ID dei servizi nei periodi riportati. L'ordine di ordinamento dell'elenco non è definito. Anche se il parametro `serviceids` è stato passato al metodo `sla.getsli`.
sli	array	Dati SLI (come **array bidimensionale**) per ciascun periodo riportato e servizio. L'indice della proprietà `periods` viene utilizzato come **prima** dimensione della proprietà `sli`. L'indice della proprietà `serviceids` viene utilizzato come **seconda** dimensione della proprietà `sli`.

Dati SLI

I dati SLI restituiti per ciascun periodo riportato e servizio consistono in:

Proprietà	Type	Descrizione
uptime	integer	Quantità di tempo in cui il servizio è rimasto in stato *OK* durante il tempo di attività pianificato, al netto dei tempi di inattività esclusi.
downtime	integer	Quantità di tempo in cui il servizio è rimasto in stato *not OK* durante il tempo di attività pianificato, al netto dei tempi di inattività esclusi.
sli	float	SLI (percentuale del tempo di attività totale), basato su uptime e downtime.
error_budget	integer	Budget di errore (in secondi), basato su SLI e SLO.
excluded_downtimes	array	Array dei tempi di inattività esclusi in questo periodo di reporting. Ciascun oggetto conterrà i seguenti parametri: - `name` - (`string`) Nome del tempo di inattività escluso. - `period_from` - (`timestamp`) Timestamp iniziale (incluso) del tempo di inattività escluso. - `period_to` - (`timestamp`) Timestamp finale (escluso) del tempo di inattività escluso. I tempi di inattività esclusi sono ordinati per `period_from`, con i periodi più iniziali visualizzati per primi.

Esempi

Calcolo dell'SLI per un SLA giornaliero

Recupera i dati SLI per i servizi con ID "1" e "4" collegati all'SLA con ID "1". Recupera i dati per un singolo periodo fino a "1761861599" (30 ott 2025 23:59:59 GMT+0200). Poiché il periodo di reporting dell'SLA è giornaliero, i dati SLI vengono recuperati da "1761775200" (30 ott 2025 00:00:00 GMT+0200) a "1761861600" (31 ott 2025 00:00:00 GMT+0200).

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
```

```
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Calcolo dell'SLI per un SLA mensile

Recupera i dati SLI sui servizi con ID "50", "60" e "70" collegati all'SLA con ID "5". Recupera i dati per tre periodi a partire da "1635724800" (01 nov 2021 00:00:00 UTC). Poiché il periodo di reporting dell'SLA è mensile, i dati SLI vengono recuperati per i tre mesi seguenti:

- Da "1635724800" (01 nov 2021 00:00:00 UTC) a "1638316800" (01 dic 2021 00:00:00 UTC)
- Da "1638316800" (01 dic 2021 00:00:00 UTC) a "1640995200" (01 gen 2022 00:00:00 UTC)
- Da "1640995200" (01 gen 2022 00:00:00 UTC) a "1643673600" (01 feb 2022 00:00:00 UTC)

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```

Risposta:

```
{
    "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": 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
}
```


Fonte

CSla::getSli() in *ui/include/classes/api/services/CSla.php*

sla.update

Descrizione

`object sla.update(object/array slaids)`

Questo metodo consente di aggiornare le voci SLA esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dell'SLA da aggiornare.

La proprietà `slaid` deve essere definita per ogni SLA, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà SLA standard, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
service_tags	array	Tag di servizio SLA per sostituire i tag di servizio SLA correnti.
schedule	array	Pianificazione SLA per sostituire quella corrente. Specificare il parametro come vuoto sarà interpretato come una pianificazione 24x7.
excluded_downtimes	array	Tempi di inattività esclusi dell'SLA per sostituire quelli correnti.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli SLA aggiornati nella proprietà `slaids`.

Esempi

Aggiornamento dei tag del servizio

Fare in modo che l'SLA con ID "5" venga calcolato a intervalli mensili per i servizi correlati a NoSQL, senza modificarne la pianificazione o i periodi di inattività esclusi; impostare l'SLO al 95%.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Modifica della pianificazione di un SLA

Imposta l'SLA con ID "5" su una pianificazione 24x7.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
```

```
    },
    "id": 1
}
```

Modifica dei periodi di inattività esclusi per un SLA

Aggiungere un periodo di inattività pianificato di 4 ore per un aggiornamento della RAM il 6 aprile 2022, mantenendo (deve essere definito nuovamente) un aggiornamento software pianificato precedentemente esistente il 4 luglio per l'SLA con ID "5".

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CSla::update() in *ui/include/classes/api/services/CSla.php*.

Span APM

Questa classe è progettata per funzionare con gli span raccolti per il monitoraggio delle prestazioni delle applicazioni (APM).

Riferimenti agli oggetti:

- Span APM

Metodi disponibili:

- apmspan.get - recupera gli span

Oggetto span APM

I seguenti oggetti sono direttamente correlati all'API `apmspan`.

Note:
Ogni proprietà corrisponde a una colonna nella tabella `otel_traces` dell'origine dati di telemetria. Per l'elenco delle colonne, vedere Colonne dei dati di telemetria supportate.

Span APM

L'oggetto span APM rappresenta uno span (una singola operazione all'interno di una trace, ad esempio una richiesta HTTP, una query al database o una chiamata di funzione).

Gli span formano una gerarchia che inizia dallo span radice, che rappresenta la prima operazione della richiesta. Ogni altro span è uno span figlio dello span durante il quale è stata avviata la relativa operazione.

L'oggetto span APM ha le seguenti proprietà:

Property	Type	Description
timestamp	integer	Ora Unix (in nanosecondi) di avvio dello span. Comportamento della proprietà: - *sola lettura*
traceid	string	ID della trace a cui appartiene lo span. Comportamento della proprietà: - *sola lettura*
spanid	string	ID dello span. Comportamento della proprietà: - *sola lettura*
parent_spanid	string	ID dello span padre. Vuoto per lo span radice. Comportamento della proprietà: - *sola lettura*
trace_state	string	Dati aggiuntivi, se presenti, associati alla trace dagli strumenti di tracing (vedere W3C Trace Context). Comportamento della proprietà: - *sola lettura*
span_name	string	Nome dell'operazione rappresentata dallo span (ad esempio, `mysqli_query`). Comportamento della proprietà: - *sola lettura*
span_kind	string	Ruolo dello span. Valori possibili: `Unspecified` - il tipo di span non è specificato; `Internal` - operazione eseguita all'interno di un'applicazione; `Server` - gestione di una richiesta in ingresso; `Client` - richiesta in uscita; `Producer` - creazione di un'operazione asincrona (ad esempio, l'invio di un messaggio); `Consumer` - elaborazione di un'operazione asincrona. Comportamento della proprietà: - *sola lettura*
service_name	string	Nome del servizio che ha prodotto lo span (ad esempio, `zabbix-frontend`), ricavato dall'attributo della risorsa OpenTelemetry `service.name`. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
resource_attributes	object	Attributi della risorsa chiave-valore dello span (ad esempio, `host.name: eabf8c2a5ddc`). Comportamento della proprietà: - *sola lettura*
scope_name	string	Nome dell'ambito di strumentazione che ha prodotto lo span. Comportamento della proprietà: - *sola lettura*
scope_version	string	Versione dell'ambito di strumentazione. Comportamento della proprietà: - *sola lettura*
span_attributes	object	Attributi dello span chiave-valore dello span (ad esempio, `db.system: mysql`). Comportamento della proprietà: - *sola lettura*
duration	integer	Durata dello span in nanosecondi. Comportamento della proprietà: - *sola lettura*
status_code	string	Stato dell'operazione rappresentata dallo span. Valori possibili: `Unset` - allo span non è stato assegnato alcuno stato esplicito; `Ok` - l'operazione è stata completata correttamente; `Error` - l'operazione è stata completata con un errore. Comportamento della proprietà: - *sola lettura*
status_message	string	Informazioni aggiuntive, se presenti, sullo stato dello span. Comportamento della proprietà: - *sola lettura*
events	array	Array di oggetti event dello span, ovvero momenti significativi durante l'operazione (ad esempio, eccezioni). Comportamento della proprietà: - *sola lettura*
links	array	Array di oggetti link dello span, che fanno riferimento a span correlati nella stessa trace o in un'altra trace. Comportamento della proprietà: - *sola lettura*

apmspan.get

Descrizione

`integer/array apmspan.get(object parameters)`

Il metodo consente di recuperare gli span in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
time_from	timestamp	Restituisce solo gli span iniziati all'ora specificata o successivamente. Comportamento del parametro: - *obbligatorio*
time_till	timestamp	Restituisce solo gli span iniziati prima dell'ora specificata. Comportamento del parametro: - *obbligatorio*
traceids	string/array	Restituisce solo gli span appartenenti alle trace specificate.
spanids	string/array	Restituisce solo gli span con gli ID specificati.
parent_spanids	string/array	Restituisce solo gli span con gli span padre specificati.
resource_attributes	object/array	Restituisce solo gli span con gli attributi della risorsa specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti gli span. Valori possibili di operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
resource_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `resource_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
span_attributes	object/array	Restituisce solo gli span con gli attributi dello span specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti gli span. Valori possibili di operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
span_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `span_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
min_duration	integer	Restituisce solo gli span la cui durata è maggiore o uguale al valore specificato, in nanosecondi.
max_duration	integer	Restituisce solo gli span la cui durata è inferiore al valore specificato, in nanosecondi.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore o un array di valori da confrontare. Proprietà supportate: `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.

Parametro	Tipo	Descrizione
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono specificate opzioni aggiuntive, viene eseguita una ricerca `LIKE "%…%"`. Proprietà supportate: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Numero di record da ignorare prima di restituire il risultato.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero della gerarchia degli span di una traccia

Recupera i primi quattro span della traccia "dee625f33cb0994e86c5cd4640e3b09c", ordinati in base all'ora di inizio. La proprietà `parent_spanid` mostra la relazione tra gli span: lo span radice non ha un elemento padre, mentre ogni altro span fa riferimento allo span che lo ha avviato.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
```

```
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```


Vedi anche

- Traccia APM
- Log APM
- Metrica APM

Origine

CApmSpan::get() in *ui/include/classes/api/services/CApmSpan.php*.

Storico

Questa classe è progettata per funzionare con i dati dello storico.

Riferimenti agli oggetti:

- Storico float
- Storico intero
- Storico stringa
- Storico testo
- Storico log

Metodi disponibili:

- history.clear - cancella i dati dello storico
- history.get - recupera i dati dello storico
- history.push - invia i dati dello storico al server Zabbix

Oggetto Storico

I seguenti oggetti sono direttamente correlati all'API `history`.

Note:
Gli oggetti Storico variano a seconda del tipo di informazione dell'elemento. Vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

Cronologia float

L'oggetto cronologia float ha le seguenti proprietà.

Proprietà	Type	Descrizione
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
ns	integer	Nanosecondi al momento della ricezione del valore.
value	float	Valore ricevuto.

Storico intero

L'oggetto storico intero ha le seguenti proprietà.

Proprietà	Type	Descrizione
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
ns	integer	Nanosecondi al momento della ricezione del valore.
value	integer	Valore ricevuto.

Cronologia delle stringhe

L'oggetto cronologia delle stringhe ha le seguenti proprietà.

Proprietà	Type	Descrizione
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
ns	integer	Nanosecondi in cui il valore è stato ricevuto.
value	string	Valore ricevuto.

Cronologia del testo

L'oggetto cronologia del testo ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
id	ID	ID della voce della cronologia.
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
ns	integer	Nanosecondi in cui il valore è stato ricevuto.
value	text	Valore ricevuto.

Cronologia dei log

L'oggetto cronologia dei log ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
id	ID	ID della voce della cronologia.
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
logeventid	integer	ID della voce del registro eventi di Windows.
ns	integer	Nanosecondi in cui il valore è stato ricevuto.
severity	integer	Livello della voce del registro eventi di Windows.
source	string	Origine della voce del registro eventi di Windows.
timestamp	timestamp	Ora della voce del registro eventi di Windows.

Proprietà	Tipo	Descrizione
value	text	Valore ricevuto.

Storico binario

L'oggetto dello storico binario ha le seguenti proprietà.

Property	Type	Description
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
ns	integer	Nanosecondi in cui il valore è stato ricevuto.
value	text	Valore ricevuto.

Cronologia JSON

L'oggetto cronologia JSON ha le seguenti proprietà.

Proprietà	Type	Descrizione
clock	timestamp	Ora in cui quel valore è stato ricevuto.
itemid	ID	ID dell'item correlato.
ns	integer	Nanosecondi al momento della ricezione del valore.
value	text	Valore ricevuto.

history.clear

Descrizione

`object history.clear(array itemids)`

Questo metodo consente di cancellare la cronologia degli item.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID degli elementi da ripulire.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli elementi ripuliti nella proprietà `itemids`.

Esempi

Ripulisci lo storico

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "history.clear",
"params": [
"10325",
"13205"
],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": {
"itemids": [
"10325",
"13205"
]
},
"id": 1
}
```


Sorgente

CHistory::clear() in *ui/include/classes/api/services/CHistory.php*.

history.get

Descrizione

`integer/array history.get(object parameters)`

Il metodo consente di recuperare i dati storici in base ai parametri specificati.

Attention:
Questo metodo può restituire dati storici di una entità eliminata se tali dati non sono ancora stati rimossi dall'housekeeper.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
history	integer	Tipi di oggetti history da restituire. Valori possibili: 0 - float numerico; 1 - carattere; 2 - log; 3 - *(predefinito)* unsigned numerico; 4 - testo; 5 - binario; 6 - JSON.
hostids	ID/array	Restituisce solo la history degli host specificati.
itemids	ID/array	Restituisce solo la history degli item specificati.
maxValueSize	integer	Numero massimo di byte da restituire nella proprietà `value`. Intervallo dei valori possibili: 1-134217728 (da 1 byte a 128MiB). Se impostato su `null`, non viene applicato alcun limite. Predefinito: 65536 (64KiB). Comportamento del parametro: - *supportato* se `history` è impostato su "binary" o "JSON"
time_from	timestamp	Restituisce solo i valori ricevuti dopo o alla data e ora specificata.
time_till	timestamp	Restituisce solo i valori ricevuti prima o alla data e ora specificata.

Parametro	Tipo	Descrizione
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `itemid`, `clock`, `ns`. Non supporta l'ordinamento per `ns` se i dati history sono archiviati in ClickHouse.
search	object	Restituisce i risultati che corrispondono al pattern specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Non supporta la ricerca per `value` se `history` è impostato su 6 (JSON).
countOutput	boolean	Questi parametri sono descritti nella reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il conteggio degli oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recuperare dei dati dallo storico degli elementi

Restituisce gli ultimi 10 valori ricevuti da un elemento numerico (float).

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "history.get",
"params": {
"output": "extend",
"history": 0,
"itemids": "23296",
"sortfield": "clock",
"sortorder": "DESC",
"limit": 10
},
"id": 1
}
```

Risposta:

```
{
"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
}
```


Sorgente

CHistory::get() in *ui/include/classes/api/services/CHistory.php*.

history.push

Descrizione

`object history.push(object/array itemHistoryData)`

Questo metodo consente di inviare i dati dello storico degli elementi al server Zabbix.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Consultare Ruoli utente per ulteriori informazioni.

Parametri

`(object/array)` Dati della cronologia degli item da inviare.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
itemid	ID	ID dell'item correlato. Comportamento del parametro: - *obbligatorio* se `host` e `key` non sono impostati
host	string	Nome tecnico del host. Comportamento del parametro: - *obbligatorio* se `itemid` non è impostato
key	string	Chiave dell'item. Comportamento del parametro: - *obbligatorio* se `itemid` non è impostato
value	mixed	Valore dell'item. Comportamento del parametro: - *obbligatorio*
clock	timestamp	Ora in cui il valore è stato ricevuto.
ns	integer	Nanosecondi in cui il valore è stato ricevuto.

Valori restituiti

`(oggetto)` Restituisce il risultato dell'operazione di invio dei dati.

Esempi

Invia i dati dello storico degli elementi

Invia i dati dello storico degli elementi al server Zabbix per gli elementi "10600", "10601" e "999999".

Richiesta:

```
{
"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/html/robots.tx
},
{
"itemid": 999999,
"value": 123
}
],
"id": 1
}
```


Risposta:


```
{
"jsonrpc": "2.0",
"result": {
"response": "success",
"data": [
{
"itemid": "10600"
},
{
"itemid": "10600"
},
{
"itemid": "10601",
"error": "Item is disabled."
},
{
"error": "No permissions to referred object or it does not exist."
}
]
},
"id": 1
}
```


Vedi anche

- Elementi Trapper
- HTTP agent elementi
- Host
- Elemento

Sorgente

CHistory::push() in *ui/include/classes/api/services/CHistory.php*.

Task

Questa classe è progettata per funzionare con i task (come il controllo degli item o delle regole di low-level discovery senza ricaricare la configurazione).

Riferimenti agli oggetti:

- Task
  - oggetto richiesta 'Execute now'
  - oggetto richiesta 'Refresh proxy configuration'
  - oggetto richiesta 'Diagnostic information'
    - oggetto richiesta Statistic
    - oggetto risultato Statistic

Metodi disponibili:

- task.create - crea nuovi task
- task.get - recupera i task

Oggetto task

I seguenti oggetti sono direttamente correlati all'API `task`.

Task

L'oggetto task ha le seguenti proprietà:

Property	Type	Description
taskid	ID	ID del task. Property behavior: - *sola lettura*
type	integer	Tipo del task. Valori possibili: 1 - Informazioni diagnostiche; 2 - Aggiorna configurazione proxy; 6 - Esegui ora. Gli utenti di tipo *Admin* e *User* possono creare solo task di tipo "Esegui ora". Si noti che `task.get` restituisce sempre "7" (Riepilogo esecuzione task). Property behavior: - *obbligatorio*
status	integer	Stato del task. Valori possibili: 1 - nuovo task; 2 - task in corso; 3 - task completato; 4 - task scaduto. Property behavior: - *sola lettura*
clock	timestamp	Ora in cui il task è stato creato. Property behavior: - *sola lettura*
ttl	integer	Tempo in secondi dopo il quale il task scade. Property behavior: - *sola lettura*
proxyid	ID	ID del proxy per il quale vengono raccolte le statistiche delle informazioni diagnostiche. Property behavior: - *supportato* se `type` è impostato su "Informazioni diagnostiche" o "Aggiorna configurazione proxy"
request	object	Oggetto richiesta del task in base al tipo di task: L'oggetto del task "Esegui ora" è descritto in dettaglio di seguito; L'oggetto del task "Aggiorna configurazione proxy" è descritto in dettaglio di seguito; L'oggetto del task "Informazioni diagnostiche" è descritto in dettaglio di seguito. Property behavior: - *obbligatorio*

Property	Type	Description
result	object	Oggetto risultato del task di informazioni diagnostiche. Può contenere NULL se il risultato non è ancora pronto. L'oggetto risultato è descritto in dettaglio di seguito. Property behavior: - *sola lettura*

Oggetto della richiesta 'Execute now'

L'oggetto della richiesta del task 'Execute now' ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
itemid	ID	ID di item e regole di low-level discovery. Gli utenti di tipo *Admin* e *User* possono eseguire 'Execute now' sugli item dei host per i quali dispongono del permesso in *lettura-scrittura*, oppure del permesso in *sola lettura* con l'azione `invoke_execute_now` abilitata per il loro ruolo. Lo stesso vale per gli utenti di tipo *Admin* per le regole di low-level discovery (LLD).

Oggetto richiesta 'Refresh proxy configuration'

L'oggetto richiesta del task 'Refresh proxy configuration' ha le seguenti proprietà.

Proprietà	Type	Descrizione
proxyids	array	ID dei proxy.

Oggetto richiesta 'Informazioni diagnostiche'

L'oggetto richiesta dell'attività di informazioni diagnostiche ha le seguenti proprietà. L'oggetto richiesta delle statistiche per tutti i tipi di proprietà è descritto in dettaglio di seguito.

Proprietà	Tipo	Descrizione
historycache	object	Richiesta di statistiche della cache della cronologia. Disponibile su server e proxy.
valuecache	object	Richiesta di statistiche della cache degli item. Disponibile sul server.
preprocessing	object	Richiesta di statistiche del gestore di preprocessing. Disponibile su server e proxy.
alerting	object	Richiesta di statistiche del gestore degli avvisi. Disponibile sul server.
lld	object	Richiesta di statistiche del gestore LLD. Disponibile sul server.

Oggetto di richiesta delle statistiche

L'oggetto di richiesta delle statistiche viene utilizzato per definire quale tipo di informazioni deve essere raccolto sui processi interni di server/proxy. Ha le seguenti proprietà.

Property	Type	Description
stats	query	Proprietà dell'oggetto statistico da restituire. L'elenco dei campi disponibili per ciascun tipo di statistica delle informazioni diagnostiche è descritto in dettaglio di seguito. Predefinito: `extend` restituirà tutti i campi statistici disponibili.
top	object	Oggetto per ordinare e limitare i valori statistici restituiti. L'elenco dei campi disponibili per ciascun tipo di statistica delle informazioni diagnostiche è descritto in dettaglio di seguito. Esempio: { "source.alerts": 10 }

Elenco dei campi statistici disponibili per ciascun tipo di richiesta di informazioni diagnostiche

I seguenti campi statistici possono essere richiesti per ciascun tipo di proprietà della richiesta di informazioni diagnostiche.

Tipo diagnostico	Campi disponibili	Descrizione
historycache	items	Numero di item nella cache.
	values	Numero di valori nella cache.
	memory	Statistiche della memoria condivisa (spazio libero, numero di blocchi utilizzati, numero di blocchi liberi, dimensione massima del blocco libero).
	memory.data	Statistiche della memoria condivisa della cache dei dati storici.
	memory.index	Statistiche della memoria condivisa della cache dell'indice storico.
valuecache	items	Numero di item nella cache.
	values	Numero di valori nella cache.
	memory	Statistiche della memoria condivisa (spazio libero, numero di blocchi utilizzati, numero di blocchi liberi, dimensione massima del blocco libero).
	mode	Modalità della cache dei valori.
preprocessing	values	Numero di valori in coda.
	preproc.values	Numero di valori in coda con passaggi di preprocessing.
alerting	alerts	Numero di avvisi in coda.
lld	rules	Numero di regole in coda.
	values	Numero di valori in coda.

Elenco dei campi di ordinamento disponibili per ciascun tipo di richiesta di informazioni diagnostiche

I seguenti campi statistici possono essere utilizzati per ordinare e limitare le informazioni richieste.

Tipo diagnostico	Campi disponibili	Tipo
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Oggetto risultato statistico

L'oggetto risultato statistico viene recuperato nel campo `result` dell'oggetto task.

Proprietà	Tipo	Descrizione
status	integer	Stato del risultato del task. Valori possibili: -1 - si è verificato un errore durante l'esecuzione del task; 0 - il risultato del task è stato creato. Comportamento della proprietà: - *sola lettura*
data	string/object	Risultati in base all'oggetto richiesta statistica del particolare task di informazioni diagnostiche. Contiene una stringa con il messaggio di errore se si è verificato un errore durante l'esecuzione del task.

task.create

Descrizione

```
object task.create(object/array tasks)
```

Questo metodo consente di creare task.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Un task da creare.

Il metodo accetta task con le proprietà standard del task.

Si noti che i task "Execute now" possono essere creati solo per i seguenti tipi di item/regole di discovery:

- Zabbix agent (passivo)
- Controllo semplice
- SNMP agent (v1/v2/v3)
- Zabbix interno
- Controllo esterno
- Monitor database
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent
- Calcolato
- Dependent item
- Script
- Browser

Se l'item/regola di discovery è di tipo "Dependent item", anche il suo item master deve essere uno dei tipi sopra elencati.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID delle attività create nella proprietà `taskids`. Viene creata un'attività per ciascun item e regola di low-level discovery. L'ordine degli ID restituiti corrisponde all'ordine degli `itemids` passati.

Esempi

Creazione di un task

Creare un task 'Execute now' per un item e una regola di low-level discovery.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Creare un task 'Refresh proxy configuration' per due proxy.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

Creare un task 'Diagnostic information'.

Request:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- Task
- Oggetto richiesta 'Execute now'
- Oggetto richiesta 'Refresh proxy configuration'
- Oggetto richiesta 'Diagnostic information'

Fonte

CTask::create() in *ui/include/classes/api/services/CTask.php*.

task.get

Descrizione

`integer/array task.get(object parameters)`

Il metodo consente di recuperare i task in base ai parametri specificati.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
taskids	ID/array	Restituisce solo le attività con gli ID specificati.
output	query	Questi parametri sono descritti nel commento di riferimento.
preservekeys	boolean	

Valori di ritorno

`(integer/array)` Restituisce un array di oggetti.

Esempi

Recuperare attività per ID

Recupera tutti i dati relativi all'attività con ID "1".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
```

```
        "taskids": "1"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Task
- Oggetto richiesta 'Execute now'
- Oggetto richiesta 'Refresh proxy configuration'
- Oggetto richiesta 'Diagnostic information'

Fonte

CTask::get() in *ui/include/classes/api/services/CTask.php*.

Template

Questa classe è progettata per funzionare con i template.

Riferimenti agli oggetti:

- Template
- Tag del template

Metodi disponibili:

- template.create - crea nuovi template
- template.delete - elimina i template
- template.get - recupera i template
- template.massadd - aggiunge oggetti correlati ai template
- template.massremove - rimuove oggetti correlati dai template
- template.update - aggiorna i template

Oggetto template

I seguenti oggetti sono direttamente correlati all'API `template`.

Template

L'oggetto template ha le seguenti proprietà.

Property	Type	Description
templateid	ID	ID del template. Property behavior: - *sola lettura* - *obbligatorio* per le operazioni di aggiornamento
description	text	Descrizione del template.
host	string	Nome tecnico del template. Property behavior: - *obbligatorio* per le operazioni di creazione
name	string	Nome visibile del template. Predefinito: valore della proprietà `host`.
readme	text	Istruzioni di configurazione specifiche del template da visualizzare nella Host Wizard. Supporta la formattazione Markdown.
uuid	string	Identificatore univoco universale, utilizzato per collegare i template importati a quelli già esistenti. Generato automaticamente, se non specificato.
vendor_name	string	Nome del fornitore del template. Per le operazioni di creazione, `vendor_name` e `vendor_version` devono essere entrambi impostati oppure entrambi lasciati vuoti. Per le operazioni di aggiornamento, `vendor_version` può essere lasciato vuoto se ha un valore nel database.
vendor_version	string	Versione del fornitore del template. Per le operazioni di creazione, `vendor_name` e `vendor_version` devono essere entrambi impostati oppure entrambi lasciati vuoti. Per le operazioni di aggiornamento, `vendor_name` può essere lasciato vuoto se ha un valore nel database.

Property	Type	Description
wizard_ready	integer	Indica se il template è disponibile per la selezione nella Host Wizard. Valori possibili: 0 - *(predefinito)* Non disponibile. 1 - Disponibile.

Tag del template

L'oggetto tag del template ha le seguenti proprietà.

Property	Type	Description
tag	string	Nome del tag del template. Property behavior: - *obbligatorio*
value	string	Valore del tag del template. Property behavior: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template. Property behavior: - *sola lettura*.
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Property behavior: - *sola lettura*.

template.create

Descrizione

`object template.create(object/array templates)`

Questo metodo consente di creare nuovi template.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Template da creare.

Oltre alle proprietà standard del template, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
groups	object/array	Gruppi di template a cui aggiungere il template. I gruppi di template devono avere definita solo la proprietà `groupid`. Comportamento del parametro: - *obbligatorio*
tags	object/array	Tag del template.
templates	object/array	Template da collegare al template. I template devono avere definita solo la proprietà `templateid`.

Parametro	Tipo	Descrizione
macros	object/array	Macro utente da creare per il template.

Valori di ritorno

(`object`) Restituisce un oggetto contenente gli ID dei modelli creati sotto la proprietà `templateids`. L'ordine degli ID restituiti corrisponde all'ordine dei modelli passati.

Esempi

Creazione di un template

Creare un template con tag e collegare due template a questo template.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

Fonte

CTemplate::create() in *ui/include/classes/api/services/CTemplate.php*.

template.delete

Descrizione

`object template.delete(array templateIds)`

Questo metodo consente di eliminare i template.

L'eliminazione di un template rimuove anche tutte le sue entità (item, trigger, grafici, ecc.) da tutti gli host o template a cui è collegato. Per conservare queste entità, scollega prima il template dagli host o template interessati usando i metodi `host.update`, `host.massremove`, `template.update` o `template.massremove`.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, consulta Ruoli utente.

Parametri

`(array)` ID dei modelli da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei modelli eliminati sotto la proprietà `templateids`.

Esempi

Eliminazione di più template

Elimina due template.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Fonte

CTemplate::delete() in *ui/include/classes/api/services/CTemplate.php*.

template.get

Descrizione

`integer/array template.get(object parameters)`

Il metodo consente di recuperare i template in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
templateids	ID/array	Restituisce solo i template con gli ID template specificati.
groupids	ID/array	Restituisce solo i template che appartengono ai gruppi di template specificati.
parentTemplateids	ID/array	Restituisce solo i template ai quali il template specificato è collegato.
hostids	ID/array	Restituisce solo i template collegati agli host/template specificati.
graphids	ID/array	Restituisce solo i template che contengono i grafici specificati.
itemids	ID/array	Restituisce solo i template che contengono gli item specificati.
triggerids	ID/array	Restituisce solo i template che contengono i trigger specificati.
with_items	flag	Restituisce solo i template che hanno item.
with_triggers	flag	Restituisce solo i template che hanno trigger.
with_graphs	flag	Restituisce solo i template che hanno grafici.
with_httptests	flag	Restituisce solo i template che hanno scenari web.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
tags	object/array	Restituisce solo i template con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i template. Valori possibili di operator: 0 - *(predefinito)* Contiene; 1 - Uguale a; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
inheritedTags	boolean	Restituisce i template che hanno i `tags` specificati anche nei template collegati. Valori possibili: `true` - anche i template collegati devono avere i tag specificati; `false` - *(predefinito)* i tag dei template collegati vengono ignorati.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag presenti nei template collegati.
selectTags	query	Restituisce i tag del template nella proprietà `tags`.
selectDiscoveryRules	query	Restituisce una proprietà `discoveryRules` con le regole LLD del template. Supporta `count`.
selectHosts	query	Restituisce gli host collegati al template nella proprietà `hosts`. Supporta `count`.
selectTemplateGroups	query	Restituisce i gruppi di template a cui il template appartiene nella proprietà `templategroups`.
selectTemplates	query	Restituisce i template ai quali il template specificato è collegato nella proprietà `templates`. Supporta `count`.
selectParentTemplates	query	Restituisce i template che sono collegati al template specificato nella proprietà `parentTemplates`. Supporta `count`.
selectHttpTests	query	Restituisce gli scenari web del template nella proprietà `httpTests`. Supporta `count`.
selectItems	query	Restituisce gli item del template nella proprietà `items`. Supporta `count`.

Parametro	Type	Descrizione
selectTriggers	query	Restituisce i trigger del template nella proprietà `triggers`. Supporta `count`.
selectGraphs	query	Restituisce i grafici del template nella proprietà `graphs`. Supporta `count`.
selectMacros	query	Restituisce le macro del template nella proprietà `macros`.
selectDashboards	query	Restituisce le dashboard del template nella proprietà `dashboards`. Supporta `count`.
selectValueMaps	query	Restituisce una proprietà `valuemaps` con le mappe di valori del template.
limitSelects	integer	Limita il numero di record restituiti dalle sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectTemplates` - i risultati saranno ordinati per `name`; `selectHosts` - ordinati per `host`; `selectParentTemplates` - ordinati per `host`; `selectItems` - ordinati per `name`; `selectDiscoveryRules` - ordinati per `name`; `selectTriggers` - ordinati per `description`; `selectGraphs` - ordinati per `name`; `selectDashboards` - ordinati per `name`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
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	Restituisce le low-level discovery del template nella proprietà `discoveries`. Supporta `count`. Questa query è **deprecata**, usare invece `selectDiscoveryRules`.

Valori restituiti

`(integer/array)` Restituisce uno dei seguenti:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei template per nome

Recupera tutti i dati relativi a due template denominati ”Linux by Zabbix agent” e ”Windows by Zabbix agent”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
```

```
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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": "Questo è un template Linux ufficiale. Richiede Zabbix agent 8.0 o versione suc
            "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": "## Panoramica\r\n\r\nQuesto è un template Linux ufficiale. Richiede Zabbix agent 8.
        },
        {
            "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": "Questo è un template Windows ufficiale. Richiede Zabbix agent 8.0 o versione s
```

```
            "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": "## Panoramica\r\n\r\nQuesto è un template Windows ufficiale. Richiede Zabbix agent
        }
    ],
    "id": 1
}
```


Recupero dei gruppi di template

Recupera i gruppi di template di cui il template "Linux by Zabbix agent" è membro.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```

Recupero degli host per template

Recupera gli host a cui è collegato il template "10001" (*Linux by Zabbix agent*).

Richiesta:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Ricerca per tag del template

Recupera i template che hanno il tag ”host-name” uguale a ”{HOST.NAME}”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10402",
```

```
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
    ],
    "id": 1
}
```

Vedi anche

- Gruppo di template
- Template
- Macro utente
- Interfaccia host

Fonte

CTemplate::get() in *ui/include/classes/api/services/CTemplate.php*.

template.massadd

Descrizione

`object template.massadd(object parameters)`

Questo metodo consente di aggiungere simultaneamente più oggetti correlati ai template specificati.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID dei template da aggiornare e gli oggetti da aggiungere ai template.

Il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
templates	object/array	Template da aggiornare. I template devono avere definita solo la proprietà `templateid`. Comportamento dei parametri: - *obbligatorio*
groups	object/array	Gruppi di template a cui aggiungere i template specificati. I gruppi di template devono avere definita solo la proprietà `groupid`.
macros	object/array	Macro utente da creare per i template specificati.
templates_link	object/array	Template da collegare ai template specificati. I template devono avere definita solo la proprietà `templateid`.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei modelli aggiornati sotto la proprietà `templateids`.

Esempi

Collegare un gruppo ai template

Aggiungere il gruppo di template "2" a due template.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Collega due template a un template

Collega i template ”10106” e ”10104” al template ”10073”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

Vedi anche

- template.update
- Host
- Gruppo di template
- Macro utente

Fonte

CTemplate::massAdd() in *ui/include/classes/api/services/CTemplate.php*.

template.massremove

Descrizione

`object template.massremove(object parameters)`

Questo metodo consente di rimuovere gli oggetti correlati da più template.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti gli ID dei template da aggiornare e gli oggetti che devono essere rimossi.

Parametro	Tipo	Descrizione
templateids	ID/array	ID dei template da aggiornare. Comportamento del parametro: - *obbligatorio*
groupids	ID/array	ID dei gruppi di template dai quali rimuovere i template specificati.
macros	string/array	ID delle macro utente da eliminare dai template specificati.
templateids_clear	ID/array	ID dei template da scollegare e rimuovere dai template specificati (a monte).
templateids_link	ID/array	ID dei template da scollegare dai template specificati (a monte).

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei modelli aggiornati sotto la proprietà `templateids`.

Esempi

Rimozione di template da un gruppo

Rimuovere due template dal gruppo "2".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
```

```
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Scollegamento dei template da un host

Scollega i template ”10106” e ”10104” dal template ”10085”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

Vedi anche

- template.update
- Macro utente

Fonte

CTemplate::massRemove() in *ui/include/classes/api/services/CTemplate.php*.

template.update

Descrizione

`object template.update(object/array templates)`

Questo metodo consente di aggiornare i template esistenti.

Note:
Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del template da aggiornare.

La proprietà `templateid` deve essere definita per ogni template, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del template, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
groups	object/array	Gruppi di template per sostituire i gruppi di template correnti a cui appartengono i template. I gruppi di template devono avere definita solo la proprietà `groupid`.
tags	object/array	Tag del template per sostituire i tag del template correnti.
macros	object/array	Macro utente per sostituire le macro utente correnti nei template specificati.
templates	object/array	Template per sostituire i template attualmente collegati. I template non passati vengono solo scollegati. I template devono avere definita solo la proprietà `templateid`.
templates_clear	object/array	Template da scollegare e rimuovere dai template specificati. I template devono avere definita solo la proprietà `templateid`.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei modelli aggiornati sotto la proprietà `templateids`.

Esempi

Modifica delle proprietà standard del template

Modificare il nome tecnico del template in "Linux by Zabbix agent Custom", il nome visibile in "My template" e aggiornare la descrizione del template.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Aggiornamento dei gruppi di template

Sostituisce tutti i gruppi di template del template specificato con uno diverso.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Aggiornamento di più gruppi di template

Sostituisce tutti i gruppi di template per più template con altri diversi.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Aggiornamento dei tag del template

Sostituisci tutti i tag del template con uno diverso.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Aggiornamento delle macro del template

Sostituisce tutte le macro del template con un'altra.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
```

```
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Aggiornamento di più macro di template

Sostituisce tutte le macro utente con la macro utente specificata su più template.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Timeout dopo il quale l'agent viene considerato non disponibile. F
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Timeout dopo il quale l'agent viene considerato non disponibile. F
                    }
                ]
            }
        ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
```

```
}
```

Aggiornamento dei template collegati di un template

Scollega (senza cancellare) tutti i template dal template specificato e collegane uno diverso.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10001"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Rimozione di più template collegati da template

Scollega e rimuovi un template collegato specifico dai template indicati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Fonte

CTemplate::update() in *ui/include/classes/api/services/CTemplate.php*.

Tipo di supporto

Questa classe è progettata per funzionare con i tipi di supporto.

Riferimenti agli oggetti:

- Tipo di supporto
  - Parametri webhook
  - Parametri script
- Template di messaggio

Metodi disponibili:

- mediatype.create - crea nuovi tipi di supporto
- mediatype.delete - elimina i tipi di supporto
- mediatype.get - recupera i tipi di supporto
- mediatype.update - aggiorna i tipi di supporto

Oggetto tipo di supporto

I seguenti oggetti sono direttamente correlati all'API `mediatype`.

Tipo di media

L'oggetto tipo di media presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
mediatypeid	ID	ID del tipo di media. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del tipo di media. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
type	integer	Trasporto utilizzato dal tipo di media. Valori possibili: 0 - Email; 1 - Script; 2 - SMS; 4 - Webhook; 5 - Push. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Proprietà	Tipo	Descrizione
exec_path	string	Nome del file script (ad esempio, notification.sh) che si trova nella directory specificata nel parametro di configurazione del server AlertScriptsPath. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”Script”
gsm_modem	string	Nome del dispositivo seriale del modem GSM. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SMS”
passwd	string	Password di autenticazione. Comportamento della proprietà: - *supportata* se `smtp_authentication` è impostato su ”Normal password”
provider	integer	Provider email. Valori possibili: 0 - *(predefinito)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.
smtp_email	string	Indirizzo email dal quale verranno inviate le notifiche. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”Email”
smtp_helo	string	HELO SMTP. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Email”
smtp_server	string	server SMTP. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”Email”
smtp_port	integer	Porta del server SMTP a cui connettersi. Predefinito: 25. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Email”
smtp_security	integer	Livello di sicurezza della connessione SMTP da utilizzare. Valori possibili: 0 - *(predefinito)* None; 1 - STARTTLS; 2 - SSL/TLS. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Email”
smtp_verify_host	integer	Verifica SSL dell’host per SMTP. Valori possibili: 0 - *(predefinito)* No; 1 - Yes. Comportamento della proprietà: - *supportata* se `smtp_security` è impostato su ”STARTTLS” o ”SSL/TLS”

Proprietà	Tipo	Descrizione
smtp_verify_peer	integer	Verifica SSL del peer per SMTP. Valori possibili: 0 - *(predefinito)* No; 1 - Yes. Comportamento della proprietà: - *supportata* se `smtp_security` è impostato su ”STARTTLS” o ”SSL/TLS”
smtp_authentication	integer	Metodo di autenticazione SMTP da utilizzare. Valori possibili: 0 - *(predefinito)* None; 1 - Normal password; 2 - OAuth token. L’autenticazione OAuth non è consentita per il provider email *Office365 relay*. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Email”
redirection_url	string	URL del frontend Zabbix a cui reindirizzare dopo l’autorizzazione OAuth. Predefinito: Valore della proprietà `url` delle impostazioni API con la parte `zabbix.php?action=oauth.authorize` Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”
client_id	string	Identificatore client registrato nel server di autorizzazione OAuth. Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”
client_secret	string	Segreto client registrato nel server di autorizzazione OAuth. Accessibile solo agli utenti di tipo Super Admin. Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”
authorization_url	string	URL OAuth, con parametri, per ottenere i token di accesso e aggiornamento. Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”
token_url	string	URL OAuth per scambiare il token di autorizzazione con i token di accesso e aggiornamento.<brQuesto URL viene utilizzato anche dal server per aggiornare un token di accesso non valido. Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”

Proprietà	Tipo	Descrizione
tokens_status	integer	Maschera di bit dello stato dei token. Valori possibili: 0 - (predefinito) Entrambi i token contengono un valore non valido 1 - Il token di accesso contiene un valore valido 2 - Il token di aggiornamento contiene un valore valido 3 - Entrambi i token contengono un valore valido. Comportamento della proprietà: - *supportata* se `smtp_authentication` è impostato su ”OAuth token”
access_token	string	Valore del token di accesso OAuth. Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”
access_token_updated	timestamp	Timestamp dell’ultima modifica di `access_token` eseguita dal server durante l’aggiornamento con `refresh_token` o dall’API in seguito a modifiche del token. Comportamento della proprietà: - *supportata* se `smtp_authentication` è impostato su ”OAuth token”
access_expires_in	integer	Tempo in secondi dopo il quale `access_token` diventerà obsoleto e sarà necessario effettuare una richiesta a `refresh_url`. Viene impostato dal server Zabbix durante l’aggiornamento di `access_token` o dall’API in seguito a modifiche del token. Il timestamp viene calcolato aggiungendo il valore di `access_token_updated`. Comportamento della proprietà: - *supportata* se `smtp_authentication` è impostato su ”OAuth token”
refresh_token	string	Valore del token di aggiornamento OAuth. Comportamento della proprietà: - *obbligatoria* se `smtp_authentication` è impostato su ”OAuth token”
status	integer	Indica se il tipo di media è abilitato. Valori possibili: 0 - *(predefinito)* Abilitato; 1 - Disabilitato.
username	string	Nome utente. Comportamento della proprietà: - *supportata* se `smtp_authentication` è impostato su ”Normal password”
maxsessions	integer	Numero massimo di avvisi che possono essere elaborati in parallelo. Valori possibili se `type` è impostato su ”SMS”: 1. Valori possibili se `type` è impostato su ”Email”, ”Script” o ”Webhook”: 0-100. Predefinito: 1.
maxattempts	integer	Numero massimo di tentativi di invio di un avviso. Valori possibili: 1-100. Predefinito: 3.

Proprietà	Tipo	Descrizione
attempt_interval	string	Intervallo tra i tentativi successivi. Accetta un numero di secondi (ad esempio, 1, 30, 60) o un valore con un suffisso temporale (ad esempio, 30s, 1m, 1h). Valori possibili: 0-1h. Predefinito: 10s.
message_format	integer	Formato del messaggio. Valori possibili: 0 - Testo normale; 1 - *(predefinito)* HTML. Comportamento della proprietà: - *supportata* se `type` è impostato su "Email"
script	text	Corpo dello script Webhook (JavaScript). Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Webhook"
timeout	string	Timeout dello script Webhook. Accetta un numero di secondi (ad esempio, 1, 60) o un valore con un suffisso temporale (ad esempio, 30s, 1m) Valori possibili: 1-60s. Predefinito: 30s.
process_tags	integer	Comportamento della proprietà: - *supportata* se `type` è impostato su "Webhook" Elabora i valori delle proprietà JSON nella risposta dello script Webhook come tag. Questi tag vengono aggiunti a qualsiasi tag del problema esistente. Valori possibili: 0 - *(predefinito)* Ignora la risposta dello script webhook; 1 - Elabora la risposta dello script webhook come tag.
show_event_menu	integer	Comportamento della proprietà: - *supportata* se `type` è impostato su "Webhook" Include una voce nel menu degli eventi che rimanda a un URL personalizzato. Aggiunge inoltre la proprietà `urls` all'output di problem.get e event.get. Valori possibili: 0 - *(predefinito)* Non includere la voce nel menu degli eventi né la proprietà `urls`; 1 - Includere la voce nel menu degli eventi e la proprietà `urls`.
event_menu_url	string	Comportamento della proprietà: - *supportata* se `type` è impostato su "Webhook" URL utilizzato nella voce del menu degli eventi e nella proprietà `urls` restituita da problem.get e event.get.
event_menu_name	string	Comportamento della proprietà: - *supportata* se `show_event_menu` è impostato su "Include event menu entry and `urls` property" Nome utilizzato per la voce del menu degli eventi e nella proprietà `urls` restituita da problem.get e event.get. Comportamento della proprietà: - *supportata* se `show_event_menu` è impostato su "Include event menu entry and `urls` property"

Proprietà	Tipo	Descrizione
parameters	array	Parametri Webhook o script. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Webhook” o ”Script”
description	text	Descrizione del tipo di media.

Parametri webhook

I parametri del webhook hanno le seguenti proprietà.

Proprietà	Tipo	Descrizione
name	string	Nome del parametro. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro, supporta le macro. Le macro supportate sono descritte nella pagina Macro supportate.

Parametri dello script

I parametri dello script hanno le seguenti proprietà.

Proprietà	Type	Descrizione
sortorder	integer	L’ordine in cui i valori dei parametri verranno passati allo script come argomenti della riga di comando, a partire da 0 come primo. Comportamento della proprietà: - *obbligatorio*
value	string	Valore del parametro, supporta le macro. Le macro supportate sono descritte nella pagina Macro supportate.

Modello di messaggio

L’oggetto modello di messaggio definisce un modello che verrà utilizzato come messaggio predefinito per le operazioni di azione per inviare una notifica. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
eventsource	integer	Origine dell’evento. Valori possibili: 0 - Trigger; 1 - Individuazione; 2 - Registrazione automatica; 3 - Interno; 4 - Servizi. Comportamento della proprietà: - *obbligatoria*
recovery	integer	Modalità operativa. Valori possibili: 0 - Operazioni; 1 - Operazioni di ripristino; 2 - Operazioni di aggiornamento. Comportamento della proprietà: - *obbligatoria*
subject	string	Oggetto del messaggio.

Proprietà	Type	Descrizione
message	string	Testo del messaggio.

mediatype.create

Descrizione

`object mediatype.create(object/array mediaTypes)`

Questo metodo consente di creare nuovi tipi di supporto.

Note:
Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se i metodi di consegna sono specificati nel flag di funzionalità `$ZBX_FEATURE_FLAGS['media_type_denylist']` nel file di configurazione del frontend, all'utente non è consentito chiamare il metodo per i tipi di supporto corrispondenti.

Parametri

`(object/array)` Tipi di supporto da creare.

Oltre alle proprietà standard del tipo di supporto, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
message_templates	array	Template di messaggio da creare per il tipo di supporto.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID dei tipi di supporto creati nella proprietà `mediatypeids`. L'ordine degli ID restituiti corrisponde all'ordine dei tipi di supporto passati.

Esempi

Creazione di un tipo di supporto email

Creare un nuovo tipo di supporto email con una porta SMTP personalizzata e modelli di messaggio.

Richiesta:

```
{
    "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": "Problema: {EVENT.NAME}",
                "message": "Il problema \"{EVENT.NAME}\" sul host \"{HOST.NAME}\" è iniziato alle {EVENT.T
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Risolto in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Il problema \"{EVENT.NAME}\" sul host \"{HOST.NAME}\" è stato risolto alle {EV
            },
            {
```

```
                "eventsource": "0",
                "recovery": "2",
                "subject": "Problema aggiornato in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} ha {EVENT.UPDATE.ACTION} il problema \"{EVENT.NAME}\" sul host
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

Creazione di un tipo di supporto script

Creare un nuovo tipo di supporto script con un valore personalizzato per il numero di tentativi e l'intervallo tra di essi.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push notifications",
        "exec_path": "push-notification.sh",
        "maxattempts": "5",
        "attempt_interval": "11s",
        "parameters": [
            {
                "sortorder": "0",
                "value": "{ALERT.SENDTO}"
            },
            {
                "sortorder": "1",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "sortorder": "2",
                "value": "{ALERT.MESSAGE}"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Creazione di un tipo di supporto webhook

Creare un nuovo tipo di supporto webhook.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

Fonte

CMediaType::create() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

Descrizione

`object mediatype.delete(array mediaTypeIds)`

Questo metodo consente di eliminare i tipi di supporto.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se i metodi di consegna sono specificati nel flag di funzionalità `$ZBX_FEATURE_FLAGS['media_type_denylist']` nel file di configurazione del frontend, all'utente non è consentito chiamare il metodo per i tipi di supporto corrispondenti.

Parametri

(array) ID dei tipi di media da eliminare.

Restituisce i valori

"(object)" Restituisce un oggetto contenente gli ID del supporto eliminato tipi nella proprietà "mediatypeids".

Esempi

Eliminazione di più tipi di supporto

Elimina due tipi di supporto.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CMediaType::delete() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

Descrizione

`integer/array mediatype.get(object parameters)`

Il metodo consente di recuperare i tipi di supporto in base ai parametri specificati.

Note:
Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se i metodi di consegna sono specificati nel flag di funzionalità `$ZBX_FEATURE_FLAGS['media_type_denylist']` nel file di configurazione del frontend, all'utente non è consentito chiamare il metodo per i tipi di supporto corrispondenti.

Parametri

(object) Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Note:
Quando si richiedono informazioni relative all'utente dei tipi di supporto, gli utenti di tipo *Admin* e *User* possono recuperare dati solo sul proprio utente. Per un esempio, vedere Recupero dei tipi di supporto come *Admin*.

Parametro	Type	Descrizione
mediatypeids	ID/array	Restituisce solo i tipi di supporto con gli ID specificati.
mediaids	ID/array	Restituisce solo i tipi di supporto utilizzati dai media specificati.

Parametro	Type	Descrizione
userids	ID/array	Restituisce solo i tipi di supporto utilizzati dagli utenti specificati.
selectActions	query	Restituisce una proprietà `actions` con le azioni che utilizzano il tipo di supporto.
selectMessageTemplates	query	Restituisce una proprietà `message_templates` con un array di messaggi del tipo di supporto. Comportamento del parametro: - *supportato* per utenti di tipo *Super admin*
selectUsers	query	Restituisce una proprietà `users` con gli utenti che utilizzano il tipo di supporto. Vedere `user.get` per le restrizioni basate sul tipo di utente.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `mediatypeid`.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text`. Possibili proprietà del Media type object per utenti di tipo *Admin* e *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`.
output	query	Proprietà del Media type object da restituire. Gli utenti di tipo *Admin* e *User* possono recuperare solo le seguenti proprietà: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Per un esempio, vedere Recupero dei tipi di supporto come *Admin*. Predefinito: `extend`.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Supporta solo proprietà del tipo di dati `string` e `text`. Possibili proprietà del Media type object per utenti di tipo *Admin* e *User*: `name`, `description`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei tipi di supporto

Recupera tutti i tipi di supporto configurati. L'esempio seguente restituisce due tipi di supporto:

- tipo di supporto email;
- tipo di supporto SMS.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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 started at {EVENT.TIME} on {EVENT.DATE}\r\nProblem name: {EVENT.NA
                },
                {
                    "eventsource": "0",
                    "recovery": "1",
                    "subject": "Resolved: {EVENT.NAME}",
                    "message": "Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE
                },
```

```
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                "message": "Discovery rule: {DISCOVERY.RULE.NAME}\r\n\r\nDevice IP: {DISCOVERY.DEVICE.
            },
            {
                "eventsource": "2",
                "recovery": "0",
                "subject": "Autoregistration: {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": "RESOLVED: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
```

```
                },
                {
                    "eventsource": "0",
                    "recovery": "2",
                    "subject": "",
                    "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
                },
                {
                    "eventsource": "1",
                    "recovery": "0",
                    "subject": "",
                    "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
                },
                {
                    "eventsource": "2",
                    "recovery": "0",
                    "subject": "",
                    "message": "Autoregistration: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.P
                }
            ],
            "parameters": []
        }
    ],
    "id": 1
}
```

Recupero dei tipi di supporto come *Admin*

Come utente di tipo *Admin*, recupera tutti i tipi di supporto abilitati, con gli utenti che utilizzano questi tipi di supporto. L'esempio seguente restituisce due tipi di supporto:

- tipo di supporto email con un utente (solo l'utente stesso di tipo *Admin*);
- tipo di supporto SMS senza utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": [
                {
                    "userid": "3",
                    "username": "database-admin",
                    "name": "John",
```

```
                    "surname": "Doe",
                    "url": "",
                    "autologin": "0",
                    "autologout": "0",
                    "lang": "default",
                    "refresh": "30s",
                    "theme": "default",
                    "attempt_failed": "0",
                    "attempt_ip": "",
                    "attempt_clock": "0",
                    "rows_per_page": "50",
                    "timezone": "default",
                    "roleid": "2",
                    "provisioned": "0"
                }
            ]
        },
        {
            "mediatypeid": "3",
            "type": "2",
            "name": "SMS",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": []
        }
    ],
    "id": 1
}
```

Retrieve script and webhook media types

The following example returns three media types:

- script media type with parameters;
- script media type without parameters;
- webhook media type with parameters.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
```

```
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```


Guarda anche

- Utente

Fonte

CMediaType::get() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

Descrizione

`object mediatype.update(object/array mediaTypes)`

Questo metodo consente di aggiornare i media type esistenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.
 Se i metodi di consegna sono specificati nel feature flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` nel file di configurazione del frontend, l'utente non può chiamare il metodo per i media type corrispondenti.

Parametri

`(object/array)` Proprietà del tipo di supporto da aggiornare.

La proprietà `mediatypeid` deve essere definita per ciascun tipo di supporto, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del tipo di supporto, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
message_templates	array	Modelli di messaggio che sostituiranno i modelli di messaggio correnti.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei tipi di supporto aggiornati sotto la proprietà `mediatypeids`.

Esempi

Abilitazione di un tipo di supporto

Abilitare un tipo di supporto, ovvero impostarne lo stato a "0".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CMediaType::update() in *ui/include/classes/api/services/CMediaType.php*.

Token

Questa classe è progettata per lavorare con i token.

Riferimenti degli oggetti:

- Token

Metodi disponibili:

- token.create - crea nuovi token
- token.delete - elimina i token
- token.get - recupera i token
- token.update - aggiorna i token
- token.generate - genera i token

Oggetto token

I seguenti oggetti sono direttamente correlati all'API `token`.

Token

L'oggetto token ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tokenid	ID	ID del token. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del token. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
description	text	Descrizione del token.
userid	ID	ID dell'utente a cui è stato assegnato il token. Predefinito: *utente corrente*. Comportamento della proprietà: - *costante*
lastaccess	timestamp	Data e ora più recenti in cui il token è stato autenticato. "0" se il token non è mai stato autenticato. Comportamento della proprietà: - *sola lettura*
status	integer	Stato del token. Valori possibili: 0 - token abilitato *(predefinito)*; 1 - token disabilitato.
expires_at	timestamp	Data e ora di scadenza del token. "0" per i token che non scadono mai.
created_at	timestamp	Data e ora di creazione del token. Comportamento della proprietà: - *sola lettura*
creator_userid	ID	ID dell'utente che ha creato il token. Comportamento della proprietà: - *sola lettura*

token.create

Descrizione

`object token.create(object/array tokens)`

Questo metodo consente di creare nuovi token.

Note:
Per il ruolo utente è necessaria l'autorizzazione *Manage API tokens* per gestire i token di altri utenti.

Attention:
Un token creato con questo metodo deve anche essere generato prima di poter essere utilizzato.

Parametri

`(object/array)` Token da creare.

Il metodo accetta token con proprietà token standard.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei token creati sotto la proprietà `tokenids`. L'ordine degli ID restituiti corrisponde all'ordine dei token passati.

Esempi

Creare un token

Creare un token abilitato che non scade mai e autentica l'utente con ID 2.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

Creare un token disabilitato che scade il 21 gennaio 2021. Questo token autenticherà l'utente corrente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
```

```
        ]
    },
    "id": 1
}
```

Fonte

CToken::create() in *ui/include/classes/api/services/CToken.php*.

token.delete

Descrizione

`object token.delete(array tokenids)`

Questo metodo consente di eliminare i token.

> **Note:**
> Per il ruolo utente è necessaria l'autorizzazione *Gestisci token API* per gestire i token di altri utenti.

Parametri

`(array)` ID dei token da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei token eliminati sotto la proprietà `tokenids`.

Esempi

Eliminare più token

Eliminare due token.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Fonte

CToken::delete() in *ui/include/classes/api/services/CToken.php*.

token.generate

Descrizione

`object token.generate(array tokenids)`

Questo metodo consente di generare token.

Note:
L'autorizzazione *Manage API tokens* permission è richiesta affinché il ruolo utente possa gestire i token di altri utenti.

Attention:
Un token può essere generato con questo metodo solo se è stato creato.

Parametri

`(array)` ID dei token da generare.

Valori restituiti

`(array)` Restituisce un array di oggetti contenente l'ID del token generato nella proprietà `tokenid` e la stringa di autorizzazione generata nella proprietà `token`.

Proprietà	Type	Descrizione
tokenid	ID	ID del token.
token	string	La stringa di autorizzazione generata per questo token.

Esempi

Generare più token

Genera due token.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

Fonte

CToken::generate() in *ui/include/classes/api/services/CToken.php*.

token.get

Descrizione

`integer/array token.get(object parameters)`

Il metodo consente di recuperare i token in base ai parametri specificati.

> **Note:**
> Solo il tipo di utente *Super admin* può visualizzare i token di altri utenti.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
tokenids	ID/array	Restituisce solo i token con gli ID specificati.
userids	ID/array	Restituisce solo i token creati per gli utenti specificati.
token	string	Restituisce solo i token creati per il *token di autenticazione* specificato.
valid_at	timestamp	Restituisce solo i token che sono validi (non scaduti) alla data e ora specificate.
expired_at	timestamp	Restituisce solo i token che sono scaduti (non validi) alla data e ora specificate.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recuperare un token

Recupera tutti i dati del token con ID "2".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Fonte

CToken::get() in *ui/include/classes/api/services/CToken.php*.

token.update

Descrizione

`object token.update(object/array tokens)`

Questo metodo consente di aggiornare i token esistenti.

> **Note:**
> È necessaria l'autorizzazione *Gestisci token API* per il ruolo utente per gestire i token di altri utenti.

Parametri

`(object/array)` Proprietà del token da aggiornare.

La proprietà `tokenid` deve essere definita per ciascun token; tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, mentre tutte le altre rimarranno invariate.

Il metodo accetta token con le proprietà standard del token.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei token aggiornati sotto la proprietà `tokenids`.

Esempi

Rimuovere la scadenza del token

Rimuovere la data di scadenza dal token.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
```

```
    "id": 1
}
```

Fonte

CToken::update() in *ui/include/classes/api/services/CToken.php*.

Traccia APM

Questa classe è progettata per funzionare con le tracce raccolte per il monitoraggio delle prestazioni delle applicazioni (APM).

Riferimenti agli oggetti:

- Traccia APM
- Evento
- Collegamento

Metodi disponibili:

- apmtrace.get - recupera le tracce

Oggetto di traccia APM

I seguenti oggetti sono direttamente correlati all'API `apmtrace`.

Note:
Ogni proprietà, ad eccezione di `span_count` e `error_count`, corrisponde a una colonna nella tabella `otel_traces` dell'origine dati di telemetria. La maggior parte delle proprietà corrisponde alla colonna con lo stesso nome (ad esempio, la proprietà `span_name` corrisponde alla colonna `SpanName`). Quando i nomi differiscono, le colonne dell'origine dati sono specificate nelle tabelle seguenti. Per l'elenco delle colonne, vedere Colonne di dati di telemetria supportate.

Traccia APM

L'oggetto traccia APM rappresenta una traccia (il percorso seguito da una richiesta attraverso le applicazioni e i servizi). Una traccia è costituita da uno o più span; ogni span rappresenta una singola operazione.

L'oggetto traccia APM si basa sullo span radice della traccia (lo span che avvia la traccia e non ha uno span padre). Tutte le proprietà contengono i dati dello span radice, ad eccezione di `span_count` e `error_count`, che vengono calcolati da tutti gli span della traccia.

L'oggetto traccia APM presenta le seguenti proprietà:

Property	Type	Description
timestamp	integer	Ora Unix (in nanosecondi) di avvio dello span radice. Comportamento della proprietà: - *sola lettura*
traceid	string	ID della traccia, condiviso da tutti gli span appartenenti alla traccia. Comportamento della proprietà: - *sola lettura*
spanid	string	ID dello span radice. Comportamento della proprietà: - *sola lettura*
trace_state	string	Dati aggiuntivi, se presenti, associati alla traccia dagli strumenti di tracing (vedere W3C Trace Context). Comportamento della proprietà: - *sola lettura*
span_name	string	Nome dell'operazione rappresentata dallo span radice (ad esempio, `POST /zabbix.php`). Comportamento della proprietà: - *sola lettura*

Property	Type	Description
span_kind	string	Ruolo dello span radice. Valori possibili: `Unspecified` - il tipo di span non è specificato; `Internal` - operazione eseguita all'interno di un'applicazione; `Server` - gestione di una richiesta in ingresso; `Client` - richiesta in uscita; `Producer` - creazione di un'operazione asincrona (ad esempio, l'invio di un messaggio); `Consumer` - elaborazione di un'operazione asincrona. Comportamento della proprietà: - *sola lettura*
service_name	string	Nome del servizio che ha prodotto lo span radice (ad esempio, `zabbix-nginx`), ricavato dall'attributo della risorsa OpenTelemetry `service.name`. Comportamento della proprietà: - *sola lettura*
resource_attributes	object	Attributi della risorsa chiave-valore dello span radice (ad esempio, `host.name: eabf8c2a5ddc`). Comportamento della proprietà: - *sola lettura*
scope_name	string	Nome dell'ambito di strumentazione che ha prodotto lo span radice. Comportamento della proprietà: - *sola lettura*
scope_version	string	Versione dell'ambito di strumentazione. Comportamento della proprietà: - *sola lettura*
span_attributes	object	Attributi dello span chiave-valore dello span radice (ad esempio, `http.method: POST`). Comportamento della proprietà: - *sola lettura*
duration	integer	Durata della traccia in nanosecondi. Comportamento della proprietà: - *sola lettura*
status_code	string	Stato dell'operazione rappresentata dallo span radice. Valori possibili: `Unset` - non è stato assegnato alcuno stato esplicito allo span radice; `Ok` - l'operazione è stata completata correttamente; `Error` - l'operazione è stata completata con un errore. Comportamento della proprietà: - *sola lettura*
status_message	string	Informazioni aggiuntive, se presenti, sullo stato dello span radice. Comportamento della proprietà: - *sola lettura*
events	array	Array di oggetti evento dello span radice, ovvero momenti significativi durante l'operazione (ad esempio, eccezioni). Comportamento della proprietà: - *sola lettura*

Property	Type	Description
links	array	Array di oggetti link dello span radice, che fanno riferimento a span correlati nella stessa traccia o in un'altra traccia. Comportamento della proprietà: - *sola lettura*
span_count	integer	Numero totale di span nella traccia. Comportamento della proprietà: - *sola lettura*
error_count	integer	Numero di span nella traccia con `status_code` impostato su `Error`. Comportamento della proprietà: - *sola lettura*

Evento

L'oggetto evento ha le seguenti proprietà.

Property	Type	Description
timestamp	integer	Ora Unix (in nanosecondi) in cui si è verificato l'evento. Colonna dell'origine dati: `Events.Timestamp`
name	string	Nome dell'evento. Colonna dell'origine dati: `Events.Name`
attributes	object	Attributi chiave-valore con informazioni aggiuntive sull'evento, se presenti. Colonna dell'origine dati: `Events.Attributes`

Collegamento

L'oggetto collegamento presenta le seguenti proprietà.

Proprietà	Tipo	Descrizione
traceid	string	ID della traccia che contiene lo span collegato. Colonna dell'origine dati: `Links.TraceId`
spanid	string	ID dello span collegato. Insieme a `traceid`, identifica in modo univoco lo span collegato. Colonna dell'origine dati: `Links.SpanId`
trace_state	string	Stato della traccia, se presente, dello span collegato (vedere W3C Trace Context). Colonna dell'origine dati: `Links.TraceState`
attributes	object	Attributi chiave-valore con informazioni aggiuntive sul collegamento. Colonna dell'origine dati: `Links.Attributes`

apmtrace.get

Descrizione

`integer/array apmtrace.get(object parameters)`

Il metodo consente di recuperare le tracce in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per gli utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
time_from	timestamp	Restituisce solo le tracce il cui span radice è iniziato al momento specificato o successivamente. Comportamento del parametro: - *obbligatorio*
time_till	timestamp	Restituisce solo le tracce il cui span radice è iniziato prima del momento specificato. Comportamento del parametro: - *obbligatorio*
traceids	string/array	Restituisce solo le tracce con gli ID specificati.
resource_attributes	object/array	Restituisce solo le tracce il cui span radice presenta gli attributi della risorsa specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutte le tracce. Valori possibili per operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non è uguale; 4 - Esiste; 5 - Non esiste.
resource_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `resource_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
span_attributes	object/array	Restituisce solo le tracce il cui span radice presenta gli attributi dello span specificati. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutte le tracce. Valori possibili per operator: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Non è uguale; 4 - Esiste; 5 - Non esiste.
span_attributes_evaltype	integer	Metodo di valutazione per le condizioni in `span_attributes`. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
min_duration	integer	Restituisce solo le tracce la cui `duration` dello span radice è maggiore o uguale al valore specificato, in nanosecondi.
max_duration	integer	Restituisce solo le tracce la cui `duration` dello span radice è inferiore al valore specificato, in nanosecondi.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore o un array di valori da confrontare. Proprietà supportate: `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.

Parametro	Tipo	Descrizione
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, viene eseguita una ricerca `LIKE "%…%"`. Proprietà supportate: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `timestamp`, `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Numero di record da ignorare prima di restituire il risultato.
countOutput	boolean	Questi parametri sono descritti nella documentazione di riferimento.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero delle tracce lente

Recupera le due tracce più lente del servizio `zabbix-nginx`. Cerca le tracce dalle 10:00 alle 11:00 del 1° ottobre 2026 (UTC) e includi solo quelle con una durata di almeno 1 secondo.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
```

```
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
        },
        {
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```


Vedi anche

- APM span
- APM metric
- APM log

Origine

CApmTrace::get() in *ui/include/classes/api/services/CApmTrace.php*.

Trend

Questa classe è progettata per funzionare con i dati di trend.

Riferimenti agli oggetti:

- Trend float
- Trend integer

Metodi disponibili:

- trend.get - recupera i trend

Oggetto trend

I seguenti oggetti sono direttamente correlati all'API `trend`.

> **Note:**
> Gli oggetti trend differiscono a seconda del tipo di informazioni dell'elemento. Vengono creati dal server Zabbix e non possono essere| modificati tramite l'API.

Trend float

L'oggetto trend float ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
clock	timestamp	Timestamp dell'ora per la quale è stato calcolato il valore. Ad esempio, il timestamp "04:00:00" indica i valori calcolati per il periodo "04:00:00-04:59:59".
itemid	ID	ID dell'item correlato.
num	integer	Numero di valori disponibili per l'ora.
value_min	float	Valore minimo orario.
value_avg	float	Valore medio orario.

Proprietà	Tipo	Descrizione
value_max	float	Valore massimo orario.

Trend intero

L'oggetto trend intero ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
clock	timestamp	Timestamp dell'ora per la quale è stato calcolato il valore. Ad esempio, il timestamp di "04:00:00" indica i valori calcolati per il periodo "04:00:00-04:59:59".
itemid	ID	ID dell'item correlato.
num	integer	Numero di valori disponibili per l'ora.
value_min	integer	Valore minimo orario.
value_avg	integer	Valore medio orario.
value_max	integer	Valore massimo orario.

trend.get

Descrizione

`integer/array trend.get(object parameters)`

Il metodo consente di recuperare i dati di trend in base ai parametri specificati.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
itemids	ID/array	Restituisce solo i trend con gli ID item specificati.
time_from	timestamp	Restituisce solo i valori raccolti dopo o al momento specificato.
time_till	timestamp	Restituisce solo i valori raccolti prima o al momento specificato.
countOutput	boolean	Conta il numero di oggetti recuperati.
limit	integer	Limita il numero di oggetti recuperati.
output	query	Imposta le proprietà dell'oggetto Trend da restituire.

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero dei dati di trend degli item

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
```

```
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```

Fonte

CTrend::get() in *ui/include/classes/api/services/CTrend.php*.

Trigger

Questa classe è progettata per funzionare con i trigger.

Riferimenti agli oggetti:

- Trigger
- Tag del trigger

Metodi disponibili:

- trigger.create - crea nuovi trigger
- trigger.delete - elimina i trigger
- trigger.get - recupera i trigger
- trigger.update - aggiorna i trigger

Oggetto trigger

I seguenti oggetti sono direttamente correlati all'API `trigger`.

Trigger

L'oggetto trigger ha le seguenti proprietà.

Property	Type	Description
triggerid	ID	ID del trigger. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento

Property	Type	Description
description	string	Nome del trigger. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
expression	string	Espressione del trigger. Gli item con `value_type` impostato su 5 (binario) o 6 (JSON) non possono essere utilizzati nelle espressioni del trigger. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
event_name	string	Nome dell'evento generato dal trigger.
opdata	string	Dati operativi.
comments	string	Descrizione aggiuntiva del trigger.
error	string	Testo dell'errore se si sono verificati problemi durante l'aggiornamento dello stato del trigger. Comportamento della proprietà: - *sola lettura*
flags	integer	Origine del trigger. Valori possibili: 0 - *(predefinito)* un trigger normale; 4 - un trigger convertito da prototipo. Comportamento della proprietà: - *sola lettura*
lastchange	timestamp	Ora dell'ultima modifica di stato del trigger. Comportamento della proprietà: - *sola lettura*
priority	integer	Gravità del trigger. Valori possibili: 0 - *(predefinito)* non classificato; 1 - informazione; 2 - avviso; 3 - media; 4 - alta; 5 - disastro.
state	integer	Stato dell'espressione del trigger. Valori possibili: 0 - *(predefinito)* lo stato del trigger è aggiornato; 1 - lo stato corrente del trigger è sconosciuto. Comportamento della proprietà: - *sola lettura*
status	integer	Indica se il trigger è abilitato o disabilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.
templateid	ID	ID del trigger template padre. Comportamento della proprietà: - *sola lettura*
type	integer	Indica se il trigger può generare più eventi problema. Valori possibili: 0 - *(predefinito)* non generare eventi multipli; 1 - generare eventi multipli.

Property	Type	Description
url	string	URL associato al trigger.
url_name	string	Etichetta dell'URL associato al trigger.
value	integer	Indica se il trigger è nello stato OK o problema. Valori possibili: 0 - *(predefinito)* OK; 1 - problema. Comportamento della proprietà: - *sola lettura*
recovery_mode	integer	Modalità di generazione dell'evento OK. Valori possibili: 0 - *(predefinito)* Espressione; 1 - Espressione di ripristino; 2 - Nessuno.
recovery_expression	string	Espressione di ripristino del trigger. Gli item con `value_type` impostato su 5 (binario) o 6 (JSON) non possono essere utilizzati nelle espressioni di ripristino del trigger.
correlation_mode	integer	Chiusura dell'evento OK. Valori possibili: 0 - *(predefinito)* Tutti i problemi; 1 - Tutti i problemi se i valori dei tag corrispondono.
correlation_tag	string	Tag per la corrispondenza.
manual_close	integer	Consenti chiusura manuale. Valori possibili: 0 - *(predefinito)* No; 1 - Sì.
uuid	string	Identificatore univoco universale, utilizzato per collegare i trigger importati a quelli già esistenti. Generato automaticamente, se non specificato. Comportamento della proprietà: - *supportata* se il trigger appartiene a un template

Tag trigger

L'oggetto tag del trigger ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tag	string	Nome del tag del trigger. Comportamento della proprietà: - *obbligatoria*
value	string	Valore del tag del trigger. Comportamento della proprietà: - *sola lettura*.
object	integer	Tipo di oggetto da cui il tag è stato ereditato. Valori possibili: 0 - template; 1 - host; 3 - item. Comportamento della proprietà: - *sola lettura*.

Proprietà	Tipo	Descrizione
objectid	ID	ID dell'oggetto da cui il tag è stato ereditato. Comportamento della proprietà: - *sola lettura*.
automatic	integer	Tipo di tag del trigger: Valori possibili: 0 - tag *(predefinito)* creato dall'utente; 1 - tag creato da LLD;Comportamento della proprietà: - *sola lettura*.

trigger.create

Descrizione

`object trigger.create(object/array triggers)`

Questo metodo consente di creare nuovi trigger.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Trigger da creare.

Oltre alle proprietà standard del trigger, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
dependencies	array	Trigger da cui il trigger dipende. I trigger devono avere definita solo la proprietà `triggerid`.
tags	array	Tag del trigger.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei trigger creati sotto la proprietà `triggerids`. L'ordine degli ID restituiti corrisponde all'ordine dei trigger passati.

Esempi

Creazione di un trigger

Creare due trigger, ciascuno dipendente da un altro trigger.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Il carico del processore è troppo elevato su {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Il servizio si è arrestato",
```

```
            "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::create() in *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

Descrizione

`object trigger.delete(array triggerIds)`

Questo metodo consente di eliminare i trigger.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID dei trigger da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei trigger eliminati sotto la proprietà `triggerids`.

Esempi

Eliminare più trigger

Eliminare due trigger.

Richiesta:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::delete() in *ui/include/classes/api/services/CTrigger.php*.

trigger.get

Descrizione

`integer/array trigger.get(object parameters)`

Il metodo consente di recuperare i trigger in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
triggerids	ID/array	Restituisce solo i trigger con gli ID specificati.
groupids	ID/array	Restituisce solo i trigger appartenenti a host o template dei gruppi di host o dei gruppi di template specificati.
templateids	ID/array	Restituisce solo i trigger appartenenti ai template specificati.
hostids	ID/array	Restituisce solo i trigger appartenenti agli host specificati.
itemids	ID/array	Restituisce solo i trigger che contengono gli item specificati.
maintenanceids	ID/array	Restituisce solo i trigger interessati dalle manutenzioni specificate.
functions	string/array	Restituisce solo i trigger che utilizzano le funzioni specificate. Per un elenco delle funzioni supportate, consultare la pagina delle funzioni supportate.
group	string	Restituisce solo i trigger appartenenti a host o template del gruppo di host o del gruppo di template con il nome specificato.
host	string	Restituisce solo i trigger appartenenti all'host con il nome tecnico specificato.
inherited	boolean	Se impostato su `true`, restituisce solo i trigger ereditati da un template.
templated	boolean	Se impostato su `true`, restituisce solo i trigger appartenenti ai template.

Parametro	Tipo	Descrizione
dependent	boolean	Se impostato su `true`, restituisce solo i trigger che hanno dipendenze. Se impostato su `false`, restituisce solo i trigger che non hanno dipendenze.
monitored	flag	Restituisce solo i trigger abilitati appartenenti a host monitorati e contenenti esclusivamente item abilitati.
active	flag	Restituisce solo i trigger abilitati appartenenti a host monitorati.
maintenance	boolean	Se impostato su `true`, restituisce solo i trigger abilitati appartenenti a host in manutenzione.
withUnacknowledgedEvents	flag	Restituisce solo i trigger con eventi non riconosciuti.
withAcknowledgedEvents	flag	Restituisce solo i trigger con tutti gli eventi riconosciuti.
withLastEventUnacknowledged	flag	Restituisce solo i trigger con l'ultimo evento non riconosciuto.
skipDependent	flag	Ignora i trigger in stato di problema che dipendono da altri trigger. Gli altri trigger vengono ignorati se sono disabilitati, se hanno item disabilitati o se appartengono a host con item disabilitati.
lastChangeSince	timestamp	Restituisce solo i trigger che hanno cambiato stato dopo l'ora specificata.
lastChangeTill	timestamp	Restituisce solo i trigger che hanno cambiato stato prima dell'ora specificata.
only_true	flag	Restituisce solo i trigger che sono stati recentemente in stato di problema (entro il periodo `ok_period`).
min_severity	integer	Restituisce solo i trigger con una gravità maggiore o uguale a quella specificata.
evaltype	integer	Metodo di valutazione dei tag. Valori possibili: 0 - *(predefinito)* And/Or; 2 - Or.
tags	array	Restituisce solo i trigger con i tag specificati. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un array vuoto restituisce tutti i trigger. Valori possibili dell'operatore: 0 - *(predefinito)* Contiene; 1 - Uguale; 2 - Non contiene; 3 - Diverso da; 4 - Esiste; 5 - Non esiste.
expandComment	flag	Espande le macro nella descrizione del trigger.
expandDescription	flag	Espande le macro nel nome del trigger.
expandExpression	flag	Espande le funzioni e le macro nell'espressione del trigger.
selectHostGroups	query	Restituisce i gruppi di host a cui appartiene il trigger nella proprietà `hostgroups`.
selectHosts	query	Restituisce gli host a cui appartiene il trigger nella proprietà `hosts`.
selectItems	query	Restituisce gli item contenuti dal trigger nella proprietà `items`.
selectFunctions	query	Restituisce le funzioni utilizzate nel trigger nella proprietà `functions`. Gli oggetti funzione rappresentano le funzioni utilizzate nell'espressione del trigger e hanno le seguenti proprietà: `functionid` - `(ID)` ID della funzione; `itemid` - `(ID)` ID dell'item utilizzato nella funzione; `function` - `(string)` nome della funzione; `parameter` - `(string)` parametro passato alla funzione. Il parametro della query viene sostituito dal simbolo $ nella stringa restituita.
selectDependencies	query	Restituisce i trigger da cui dipende il trigger nella proprietà `dependencies`.

Parametro	Tipo	Descrizione
selectDiscoveryData	query	Restituisce una proprietà `discoveryData` con i dati dell'oggetto di discovery del trigger. L'oggetto di discovery del trigger collega un trigger rilevato a un prototipo di trigger dal quale è stato rilevato. Presenta le seguenti proprietà: `parent_triggerid` - `(ID)` ID del prototipo di trigger dal quale è stato creato il trigger; `status` - `(int)` stato di discovery del trigger: 0 - *(predefinito)* il trigger è rilevato, 1 - il trigger non è più rilevato; `ts_delete` - `(timestamp)` momento in cui un trigger non più rilevato verrà eliminato; `ts_disable` - `(timestamp)` momento in cui un trigger non più rilevato verrà disabilitato; `disable_source` - `(int)` indicatore che specifica se il trigger è stato disabilitato da una regola LLD o manualmente: 0 - *(predefinito)* disabilitato automaticamente, 1 - disabilitato da una regola LLD.
selectDiscoveryRule	query	Restituisce la regola di low-level discovery che ha creato il trigger nella proprietà `discoveryRule`.
selectLastEvent	query	Restituisce l'ultimo evento significativo del trigger nella proprietà `lastEvent`.
selectTags	query	Restituisce i tag del trigger nella proprietà `tags`.
inheritedTags	boolean	Restituisce i trigger che hanno i `tags` specificati anche nel template, nell'host o nei template collegati. Valori possibili: `true` - anche il template, l'host o i template collegati devono avere i tag specificati; `false` - *(predefinito)* i tag del template, dell'host o dei template collegati vengono ignorati.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag ereditati dai template, dai template collegati e dagli host, nonché dagli item a cui si fa riferimento nelle espressioni del trigger o di ripristino.
selectTemplateGroups	query	Restituisce i gruppi di template a cui appartiene il trigger nella proprietà `templategroups`.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore o un array di valori da confrontare. Non supporta le proprietà del tipo di dati `text`. Supporta proprietà aggiuntive: `host` - nome tecnico dell'host a cui appartiene il trigger; `hostid` - ID dell'host a cui appartiene il trigger.
limitSelects	integer	Limita il numero di record restituiti dalle sottoquery. Si applica alle seguenti sottoquery: `selectHosts` - i risultati verranno ordinati per `host`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	Questi parametri sono descritti nella documentazione di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	

Parametro	Tipo	Descrizione
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	Restituisce l'oggetto di discovery del trigger nella proprietà `triggerDiscovery`. Gli oggetti di discovery del trigger collegano il trigger a un prototipo di trigger dal quale è stato creato. Questa query è **deprecata**; utilizzare invece `selectDiscoveryData`.

Valori restituiti

`(integer/array)` Restituisce:

- un array di oggetti;
- il numero di oggetti recuperati, se è stato utilizzato il parametro `countOutput`.

Esempi

Recupero dei dati tramite ID del trigger

Recupera tutti i dati e le funzioni utilizzate nel trigger "14062".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Recupero dei trigger nello stato di problema

Recupera l'ID, il nome e la gravità di tutti i trigger nello stato di problema e li ordina per gravità in ordine decrescente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```

Recupero di uno specifico trigger con tag

Recupera uno specifico trigger con tag.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
```

```
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Vedi anche

- Regola di individuazione
- Item
- Host
- Gruppo host
- Gruppo template

Fonte

CTrigger::get() in *ui/include/classes/api/services/CTrigger.php*.

trigger.update

Descrizione

`object trigger.update(object/array triggers)`

Questo metodo consente di aggiornare i trigger esistenti.

> **Note:**
> Questo metodo è disponibile solo per i tipi di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del trigger da aggiornare.

La proprietà `triggerid` deve essere definita per ciascun trigger, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

In aggiunta alle proprietà standard del trigger, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
dependencies	array	Trigger da cui il trigger dipende. I trigger devono avere definita solo la proprietà `triggerid`.
tags	array	Tag del trigger.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID dei trigger aggiornati sotto la proprietà `triggerids`.

Esempi

Abilitazione di un trigger

Abilitare un trigger, cioè impostarne lo stato a "0".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Sostituzione dei tag del trigger

Sostituisce i tag per il trigger.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Sostituzione delle dipendenze

Sostituisce le dipendenze per il trigger.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::update() in *ui/include/classes/api/services/CTrigger.php*.

Utente

Questa classe è progettata per funzionare con gli utenti.

Riferimenti agli oggetti:

- User
- Media

Metodi disponibili:

- user.checkauthentication - controllare e prolungare le sessioni utente
- user.create - creare nuovi utenti
- user.delete - eliminare utenti

- user.get - recuperare utenti
- user.login - accedere all'API
- user.logout - uscire dall'API
- user.provision - effettuare il provisioning degli utenti LDAP
- user.resettotp - reimpostare i segreti TOTP degli utenti
- user.unblock - sbloccare utenti
- user.update - aggiornare utenti

Oggetto utente

I seguenti oggetti sono direttamente correlati all'API `user`.

Utente

L'oggetto utente ha le seguenti proprietà.

Property	Type	Description
userid	ID	ID dell'utente. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
username	string	Nome dell'utente. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli utenti provisioned se l'utente è collegato a una directory utenti (`userdirectoryid` è impostato su un valore valido diverso da "0"), lo stato di provisioning della directory utenti è abilitato (`provision_status` dell'oggetto User directory è impostato su "1") e lo stato di autenticazione di tutto il provisioning LDAP o SAML è abilitato (`ldap_jit_status` dell'oggetto Authentication è impostato su "Enabled for configured LDAP IdPs" oppure `saml_jit_status` dell'oggetto Authentication è impostato su "Enabled for configured SAML IdPs")
passwd	string	Password dell'utente. Il valore di questo parametro può essere una stringa vuota se l'utente è collegato a una directory utenti. Comportamento della proprietà: - *sola scrittura*
roleid	ID	ID del ruolo dell'utente. Gli utenti senza un ruolo possono accedere a Zabbix solo utilizzando l'autenticazione LDAP o SAML, a condizione che le loro informazioni LDAP/SAML corrispondano alle mappature dei gruppi utenti configurate in Zabbix.
attempt_clock	timestamp	Ora dell'ultimo tentativo di accesso non riuscito. Comportamento della proprietà: - *sola lettura*
attempt_failed	integer	Numero dei tentativi di accesso non riusciti recenti. Comportamento della proprietà: - *sola lettura*
attempt_ip	string	Indirizzo IP da cui è stato effettuato l'ultimo tentativo di accesso non riuscito. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
autologin	integer	Indica se abilitare l'accesso automatico. Valori possibili: 0 - *(predefinito)* accesso automatico disabilitato; 1 - accesso automatico abilitato.
autologout	string	Durata della sessione utente. Accetta un numero di secondi (ad esempio, 1, 60, 300) oppure un valore con un suffisso temporale (ad esempio, 30s, 1m, 2h). Se impostato su 0s, la sessione non scadrà mai. Predefinito: 15m.
default_maintenance_period	string	Periodo di manutenzione predefinito per una manutenzione creata tramite scorciatoia. Accetta secondi o un'unità di tempo con suffisso (300s, 5m, 1h, 1d, 1w). Il valore specificato verrà arrotondato per difetto a minuti interi. Predefinito: 1h.
lang	string	Codice della lingua dell'utente, ad esempio en_US. Predefinito: default - impostazione predefinita del sistema.
name	string	Nome dell'utente.
provisioned	integer	Indica se l'utente è stato provisioned. Valori possibili: 0 - non provisioned; 1 - provisioned. Comportamento della proprietà: - *sola lettura*
refresh	string	Periodo di aggiornamento automatico. Accetta un numero di secondi (ad esempio, 1, 60, 300) oppure un valore con un suffisso temporale (ad esempio, 30s, 1m, 1h). Predefinito: 30s.
rows_per_page	integer	Numero di righe degli oggetti da visualizzare per pagina. Predefinito: 50.
surname	string	Cognome dell'utente.
theme	string	Tema dell'utente. Valori possibili: default - *(predefinito)* impostazione predefinita del sistema; blue-theme - Blu; blue-classic-theme - Blu (classico); dark-theme - Scuro; dark-blue-theme - Blu scuro; dark-classic-theme - Scuro (classico); hc-light - Chiaro ad alto contrasto; hc-dark - Scuro ad alto contrasto.
ts_provisioned	timestamp	Ora in cui è stata eseguita l'ultima operazione di provisioning. Comportamento della proprietà: - *sola lettura* - *supportata* per gli utenti di tipo *Super admin*
url	string	URL della pagina a cui reindirizzare l'utente dopo l'accesso.

Property	Type	Description
userdirectoryid	ID	ID della directory utenti a cui è collegato l'utente. Utilizzato per il provisioning (creazione o aggiornamento) e per consentire l'accesso a un utente collegato a una directory utenti. Per le operazioni di accesso, il valore di questa proprietà avrà la precedenza sulla proprietà `userdirectoryid` dei gruppi utenti a cui appartiene l'utente. Predefinito: 0. Comportamento della proprietà: - *sola lettura* - *supportata* per gli utenti di tipo *Super admin*
timezone	string	Fuso orario dell'utente, ad esempio `Europe/London`, `UTC`. Predefinito: `default` - impostazione predefinita del sistema. Per l'elenco completo dei fusi orari supportati, consultare la documentazione PHP.

Media

L'oggetto media presenta le seguenti proprietà.

Property	Type	Description
mediaid	ID	ID del media dell'utente. Comportamento della proprietà: - *sola lettura*
mediatypeid	ID	ID del tipo di media utilizzato dal media dell'utente. Comportamento della proprietà: - *obbligatoria*
sendto	string/array	Indirizzo, nome utente o altro identificatore del destinatario. Se `type` di Media type è impostato su "Email" o "Push", i valori sono rappresentati come array. Per "Push", l'array può contenere un carattere jolly (`*`) oppure uno o più ID di dispositivi mobili. Se `type` di Media type è impostato su "Script", "SMS" o "Webhook", il valore è rappresentato come stringa. Comportamento della proprietà: - *obbligatoria*
active	integer	Indica se il media è abilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.

Property	Type	Description
severity	integer	Gravità dei trigger per le quali inviare notifiche. Valori bitmap possibili: 1 - Non classificato; 2 - Informazioni; 4 - Avviso; 8 - Media; 16 - Alta; 32 - Disastro. Questo è un campo con maschera di bit; è accettabile qualsiasi somma dei valori bitmap possibili (ad esempio, 48 per Media, Alta e Disastro). Predefinito: 63.
period	string	Orario durante il quale è possibile inviare le notifiche, espresso come periodo di tempo o come macro utente separate da un punto e virgola. Predefinito: 1-7,00:00-24:00.
provisioned	integer	Indica se l'utente è stato provisioned. Valori possibili: 0 - non provisioned; 1 - provisioned.
userdirectory_mediaid	ID	ID della mappatura del media nella directory utenti per il media provisioned. Comportamento della proprietà: - *sola lettura* - *supportata* per gli utenti di tipo *Super admin*

m4rv1n

Descrizione

`string/object user.login(object parameters)`

Questo metodo consente di accedere all'API e generare un token di autenticazione.

Warning:
Quando si utilizza questo metodo, è necessario eseguire anche user.logout per evitare la generazione di un gran numero di record di sessione aperti.

Attention:
Questo metodo è disponibile solo per gli utenti non autenticati che non appartengono a nessun user group con l'autenticazione a più fattori abilitata.

Parametri

`(object)` Parametri contenenti il nome utente e la password.

Il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
password	string	Password dell'utente. Comportamento del parametro: - *obbligatorio*

Parametro	Type	Descrizione
username	string	Nome utente. Comportamento del parametro: - *obbligatorio*
userData	flag	Restituisce informazioni sull'utente autenticato.

Valori restituiti

`(string/object)` Se viene utilizzato il parametro `userData`, restituisce un oggetto contenente informazioni sull'utente autenticato.

Oltre alle proprietà utente standard, vengono restituite le seguenti informazioni:

Proprietà	Tipo	Descrizione
auth_type	integer	Autenticazione predefinita per l'utente. Fare riferimento alla proprietà `authentication_type` dell'oggetto Authentication per un elenco dei valori possibili.
debug_mode	integer	Indica se la modalità di debug è abilitata o disabilitata per l'utente. Fare riferimento alla proprietà `debug_mode` dell'oggetto User group per un elenco dei valori possibili.
deprovisioned	boolean	Indica se l'utente appartiene a un gruppo di utenti deprovisionati.
gui_access	string	Metodo di autenticazione dell'utente al frontend. Fare riferimento alla proprietà `gui_access` dell'oggetto User group per un elenco dei valori possibili.
mfaid	integer	ID del metodo MFA da utilizzare per l'utente durante l'accesso. Restituisce "0" se MFA è disabilitato globalmente o per tutti i gruppi di utenti a cui l'utente appartiene.
secret	string	Stringa casuale di 32 caratteri. Viene generata all'accesso dell'utente.
sessionid	string	Token di autenticazione, che deve essere utilizzato nelle successive richieste API.
type	integer	Tipo di utente. Fare riferimento alla proprietà `type` dell'oggetto Role per un elenco dei valori possibili.
userip	string	Indirizzo IP dell'utente.

Note:
Se un utente è stato autenticato correttamente dopo uno o più tentativi non riusciti, il metodo restituirà i valori correnti delle proprietà `attempt_clock`, `attempt_failed` e `attempt_ip` e quindi le reimposterà.

Se il parametro `userData` non viene utilizzato, il metodo restituisce un token di autenticazione richiesto per l'autenticazione.

Esempi

Autenticazione di un utente

Autenticare un utente.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
```

```
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Richiesta delle informazioni dell'utente autenticato

Autentica l'utente e restituisce informazioni aggiuntive sull'utente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Guarda anche

- user.logout

Fonte

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.checkAuthentication

Descrizione

`object user.checkAuthentication`

Questo metodo verifica e prolunga la sessione utente.

> **Attention:**
> La chiamata del metodo `user.checkAuthentication` utilizzando il parametro `sessionid` prolunga la sessione utente per impostazione predefinita.

Parametri

Il metodo accetta i seguenti parametri.

Parameter	Type	Description
extend	boolean	Indica se prolungare la sessione utente. Valore predefinito: "true". Impostando il valore su "false" è possibile verificare la sessione utente senza prolungarla.
sessionid	string	Comportamento del parametro: - *supported* se `sessionid` è impostato Token di autenticazione dell'utente.
token	string	Comportamento del parametro: - *required* se `token` non è impostato Token API dell'utente.
		Comportamento del parametro: - *required* se `sessionid` non è impostato

Valori restituiti

`(object)` Restituisce un oggetto contenente informazioni sull'utente.

Oltre alle proprietà utente standard, vengono restituite le seguenti informazioni.

Proprietà	Type	Descrizione
auth_type	integer	Autenticazione predefinita per l'utente. Fare riferimento alla proprietà `authentication_type` dell'oggetto Authentication per un elenco dei valori possibili.
debug_mode	integer	Indica se la modalità di debug è abilitata o disabilitata per l'utente. Fare riferimento alla proprietà `debug_mode` dell'oggetto User group per un elenco dei valori possibili.
deprovisioned	boolean	Indica se l'utente appartiene a un gruppo di utenti deprovisioned.
gui_access	string	Metodo di autenticazione dell'utente al frontend. Fare riferimento alla proprietà `gui_access` dell'oggetto User group per un elenco dei valori possibili.
secret	string	Stringa casuale di 32 caratteri. Viene generata al login dell'utente. La proprietà `secret` non viene restituita se la sessione utente viene verificata utilizzando un token API.

Proprietà	Type	Descrizione
sessionid	string	Token di autenticazione, che deve essere utilizzato nelle successive richieste API. La proprietà `sessionid` non viene restituita se la sessione utente viene verificata utilizzando un token API.
type	integer	Tipo di utente. Fare riferimento alla proprietà `type` dell'oggetto Role per un elenco dei valori possibili.
userip	string	Indirizzo IP dell'utente.

Esempi

Verificare l'autenticazione utilizzando il token di autenticazione

Verifica e prolunga una sessione utente utilizzando il token di autenticazione dell'utente e restituisce informazioni aggiuntive sull'utente.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Verificare l'autenticazione utilizzando il token API

Verificare una sessione utente utilizzando il token API dell'utente e restituire informazioni aggiuntive sull'utente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

Risposta:

```
{
    "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
}
```

Fonte

CUser::checkAuthentication() in *ui/include/classes/api/services/CUser.php*.

user.create

Descrizione

`object user.create(object/array users)`

Questo metodo consente di creare nuovi utenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Note:
La robustezza della password dell'utente viene convalidata in base alle regole dei criteri per le password definite dall'API Authentication. Per ulteriori informazioni, vedere oggetto authentication.

Parametri

`(object/array)` Utenti da creare.

Oltre alle proprietà utente standard, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
usrgrps	array	Gruppi di utenti a cui aggiungere l'utente. I gruppi di utenti devono avere definita solo la proprietà `usrgrpid`.
medias	array	Supporti utente da creare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli utenti creati sotto la proprietà `userids`. L'ordine degli ID restituiti corrisponde all'ordine degli utenti passati.

Esempi

Creazione di un utente

Creare un nuovo utente, aggiungerlo a un gruppo di utenti e creare un nuovo media per lui.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
```

```
    "id": 1
}
```

Vedi anche

- Autenticazione
- Media
- Gruppo utenti
- Ruolo

Fonte

CUser::create() in *ui/include/classes/api/services/CUser.php*.

user.delete

Descrizione

`object user.delete(array users)`

Questo metodo consente di eliminare gli utenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID degli utenti da eliminare.

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli utenti eliminati nella proprietà `userids`.

Esempi

Eliminazione di più utenti

Elimina due utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Sorgente

CUser::delete() in *ui/include/classes/api/services/CUser.php*.

user.get

Descrizione

`integer/array user.get(object parameters)`

Il metodo consente di recuperare gli utenti in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

> **Note:**
> Quando si richiedono media utente, permessi o informazioni sul ruolo, gli utenti di tipo *Admin* e *User* possono recuperare dati solo relativi al proprio utente. Per un esempio, vedere Recupero degli utenti come *Admin*.

Parametro	Type	Descrizione
mediaids	ID/array	Restituisce solo gli utenti che utilizzano i media specificati.
mediatypeids	ID/array	Restituisce solo gli utenti che utilizzano i tipi di media specificati.
userids	ID/array	Restituisce solo gli utenti con gli ID specificati.
usrgrpids	ID/array	Restituisce solo gli utenti che appartengono ai gruppi utenti specificati.
getAccess	flag	Aggiunge informazioni aggiuntive sui permessi dell'utente. Aggiunge le seguenti proprietà per ciascun utente: `gui_access` - `(integer)` metodo di autenticazione al frontend dell'utente. Fare riferimento alla proprietà `gui_access` del user group object per un elenco dei valori possibili. `debug_mode` - `(integer)` indica se il debug è abilitato per l'utente. Valori possibili: 0 - debug disabilitato, 1 - debug abilitato. `users_status` - `(integer)` indica se l'utente è disabilitato. Valori possibili: 0 - utente abilitato, 1 - utente disabilitato.
selectMedias	query	Restituisce i media utilizzati dall'utente nella proprietà `medias`.
selectMediatypes	query	Restituisce i tipi di media utilizzati dall'utente nella proprietà `mediatypes`. Vedere `mediatype.get` per le limitazioni basate sul tipo di utente.
selectUsrgrps	query	Restituisce i gruppi utenti a cui l'utente appartiene nella proprietà `usrgrps`. Vedere `usergroup.get` per le limitazioni basate sul tipo di utente.
selectRole	query	Restituisce il ruolo utente nella proprietà `role`.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta le proprietà del data type `text`. Possibili proprietà del User object per utenti di tipo *Admin* e *User* quando richiedono dati sugli utenti del proprio gruppo utenti: `userid`, `name`, `surname`, `username`.

Parametro	Type	Descrizione
output	query	Proprietà del User object da restituire. Gli utenti di tipo *Admin* e *User* possono recuperare solo le seguenti proprietà: - Per il proprio utente: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. - Per gli utenti del proprio gruppo utenti: `userid`, `name`, `surname`, `username`. Predefinito: `extend`.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto, in cui le chiavi sono nomi di proprietà e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Supporta solo le proprietà del data type `string` e `text`. Possibili proprietà del User object per utenti di tipo *Admin* e *User* quando richiedono dati sugli utenti del proprio gruppo utenti: `name`, `surname`, `username`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `userid`, `username`.
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valori restituiti

`(integer/array)` Restituisce o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

Esempi

Recupero degli utenti

Recupera tutti gli utenti configurati.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```


Recupero degli utenti come *Admin*

In qualità di utente di tipo *Admin*, recupera i dati dettagliati del tuo utente e i dati limitati degli utenti appartenenti al tuo gruppo di utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
```

```
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "1",
      "username": "Admin",
      "name": "Zabbix",
      "surname": "Administrator",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
      "attempt_clock": "0",
      "rows_per_page": "50",
      "timezone": "default",
      "roleid": "3",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "0",
      "usrgrps": [
        {
          "usrgrpid": "7",
          "name": "Zabbix administrators",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
```

```
      "role": {
        "roleid": "3",
        "name": "Super admin role",
        "type": "3",
        "readonly": "1"
      }
    },
    {
      "userid": "2",
      "username": "guest",
      "name": "",
      "surname": "",
      "url": "",
      "autologin": "0",
      "autologout": "15m",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
      "attempt_clock": "0",
      "rows_per_page": "50",
      "timezone": "default",
      "roleid": "4",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "1",
      "usrgrps": [
        {
          "usrgrpid": "8",
          "name": "Guests",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "9",
          "name": "Disabled",
          "gui_access": "0",
          "users_status": "1",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
```

```
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Recupero dei dati dell'utente

Recupera i dati dell'utente con ID "2".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
}
```

Risposta:


```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Vedi anche

- Media
- Tipo di supporto
- Gruppo utenti
- Ruolo

Fonte

CUser::get() in *ui/include/classes/api/services/CUser.php*.

user.logout

Descrizione

`string/object user.logout(array)`

Questo metodo consente di disconnettersi dall'API e invalida il token di autenticazione corrente.

> **Note:**
> Questo metodo è disponibile per utenti di qualsiasi tipo. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` Il metodo accetta un array vuoto.

Valori restituiti

`(booleano)` Restituisce `true` se l'utente è stato disconnesso correttamente.

Esempi

Disconnessione

Disconnessione dall'API.

Richiesta:

```
{
"jsonrpc": "2.0",
"method": "user.logout",
"params": [],
"id": 1
}
```

Risposta:

```
{
"jsonrpc": "2.0",
"result": true,
"id": 1
}
```

Vedi anche

- user.login

Sorgente

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.provision

Descrizione

`object user.provision(object/array users)`

Questo metodo consente di effettuare il provisioning degli utenti LDAP.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

"(array)" ID degli utenti da preparare.

Restituisce valori

"(object)" Restituisce un oggetto contenente gli ID degli utenti sottoposti a provisioning nella proprietà "userids".

Esempi

Provisioning di più utenti

Eseguire il provisioning di due utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CUser::provision() in *ui/include/classes/api/services/CUser.php*.

user.resettotp

Descrizione

`object user.resettotp(object/array users)`

Questo metodo consente di reimpostare i segreti TOTP degli utenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(array)` ID degli utenti per i quali reimpostare i segreti TOTP.

> **Note:**
> Anche le sessioni utente degli utenti specificati verranno eliminate (tranne quella dell'utente che invia la richiesta).

Valori restituiti

`(object)` Restituisce un oggetto contenente gli ID degli utenti per i quali i segreti TOTP sono stati reimpostati, nella proprietà `userids`.

Esempi

Reimpostazione dei segreti TOTP per più utenti

Reimpostare i segreti TOTP per due utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
```

```
        "1",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Vedi anche

- Oggetto MFA

Sorgente

CUser::resettotp() in *ui/include/classes/api/services/CUser.php*.

user.unblock

Descrizione

`object user.unblock(array userids)`

Questo metodo consente di sbloccare gli utenti.

> **Note:**
> Questo metodo è disponibile solo per il tipo di utente *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Parametri

`(array)` ID degli utenti da sbloccare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli utenti sbloccati sotto la proprietà `userids`.

Esempi

Sblocco di più utenti

Sblocca due utenti.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CUser::unblock() in *ui/include/classes/api/services/CUser.php*.

user.update

Descrizione

`object user.update(object/array users)`

Questo metodo consente di aggiornare gli utenti esistenti.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Vedere Ruoli utente per ulteriori informazioni.

Note:
La robustezza della password dell'utente viene convalidata in base alle regole dei criteri password definite da Authentication API. Vedere Authentication API per ulteriori informazioni.

Parametri

`(object/array)` Proprietà utente da aggiornare.

La proprietà `userid` deve essere definita per ciascun utente, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà utente standard, il metodo accetta i seguenti parametri.

Parameter	Type	Description
current_passwd	string	Password corrente dell'utente. Il valore di questo parametro può essere una stringa vuota se l'utente è collegato a una directory utenti. Comportamento del parametro: - *sola scrittura* - *obbligatorio* se `passwd` dell'oggetto User è impostato e l'utente modifica la propria password
usrgrps	array	Gruppi utenti per sostituire i gruppi utenti esistenti. I gruppi utenti devono avere definita solo la proprietà `usrgrpid`.
medias	array	Supporti utente per sostituire i supporti esistenti non sottoposti a provisioning. I supporti sottoposti a provisioning possono essere omessi durante l'aggiornamento dei supporti.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID degli utenti aggiornati sotto la proprietà `userids`.

Esempi

Rinominare un utente

Rinominare un utente in John Doe.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

Modifica del ruolo utente

Modificare il ruolo di un utente.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

Guarda anche

- Autenticazione

Fonte

CUser::update() in *ui/include/classes/api/services/CUser.php*.

Appendice 1. Commento di riferimento

Notazione Firma del metodo

Ogni pagina del metodo API contiene la firma del metodo.

Esempio: `object action.create(object/array actions)`

Firma	Descrizione
`object`	Il tipo di dati restituito dal metodo.
`action.create`	Il nome completo del metodo.
`object/array`	Il tipo di dati accettato dal metodo.
`actions`	Un breve riepilogo di ciò che il metodo accetta.

Tipi di dati

L'API di Zabbix supporta i seguenti tipi di dati come input:

Type	Description
ID	Un identificatore univoco utilizzato per fare riferimento a un'entità.
boolean	Un valore booleano (`true` oppure `false`).
flag	Un valore considerato `true` se passato e diverso da `null`; altrimenti, il valore è considerato `false`.
integer	Un numero intero.
float	Un numero in virgola mobile.
string	Una stringa di testo.
text	Una stringa di testo più lunga.
timestamp	Un timestamp Unix.
array	Una sequenza ordinata di valori (un array semplice).
object	Un array associativo.
query	Un valore che definisce i dati da restituire. Il valore può essere definito come un array di nomi di proprietà (per restituire solo proprietà specifiche), oppure come uno dei valori predefiniti: `extend` - restituisce tutte le proprietà dell'oggetto; `count` - restituisce il numero di record recuperati, supportato solo da alcune sotto-selezioni.

Attention:
L'API di Zabbix restituisce sempre valori solo come stringhe o array.

Comportamento delle proprietà

Alcune proprietà dell'oggetto sono contrassegnate con brevi etichette per descriverne il comportamento. Vengono utilizzate le seguenti etichette:

- ***sola lettura*** - il valore della proprietà viene impostato automaticamente e non può essere definito o modificato dall'utente, nemmeno in alcune condizioni specifiche (ad esempio, *sola lettura* per oggetti ereditati o oggetti individuati);
- ***sola scrittura*** - il valore della proprietà può essere impostato, ma non può essere successivamente consultato;
- ***costante*** - il valore della proprietà può essere impostato durante la creazione di un oggetto, ma non può essere modificato in seguito;
- ***supportata*** - non è necessario impostare il valore della proprietà, ma è consentito farlo in alcune condizioni specifiche (ad esempio, *supportata* se `type` è impostato su "Simple check", "External check", "SSH agent", "TELNET agent" o "HTTP agent"); si noti tuttavia che le proprietà *supportate* possono comunque essere impostate ai loro valori predefiniti indipendentemente dalle condizioni;
- ***obbligatoria*** - il valore della proprietà deve essere impostato per tutte le operazioni (eccetto le operazioni di get) oppure in alcune condizioni specifiche (ad esempio, *obbligatoria* per le operazioni di creazione; *obbligatoria* se `operationtype` è impostato su "global script" e `opcommand_hst` non è impostato).

Note:
Per le operazioni di aggiornamento, una proprietà è considerata "impostata" quando viene impostata durante l'operazione di aggiornamento.

Le proprietà non contrassegnate con etichette sono facoltative.

Comportamento dei parametri

Alcuni parametri dell'operazione sono contrassegnati con brevi etichette per descriverne il comportamento per l'operazione. Vengono utilizzate le seguenti etichette:

- ***sola lettura*** - il valore del parametro viene impostato automaticamente e non può essere definito o modificato dall'utente, nemmeno in alcune condizioni specifiche (ad esempio, *sola lettura* per oggetti ereditati o oggetti individuati);
- ***sola scrittura*** - il valore del parametro può essere impostato, ma successivamente non può essere consultato;

- ***supportato*** - non è necessario impostare il valore del parametro, ma è consentito farlo in alcune condizioni specifiche (ad esempio, *supportato* se `operating_mode` dell'oggetto Proxy è impostato su "passive proxy"); si noti tuttavia che i parametri *supportati* possono comunque essere impostati ai loro valori predefiniti indipendentemente dalle condizioni;
- ***obbligatorio*** - è necessario impostare il valore del parametro.

I parametri che non sono contrassegnati con etichette sono facoltativi.

Valore ID riservato "0" Il valore ID riservato "0" può essere utilizzato per filtrare gli elementi e per rimuovere gli oggetti referenziati. Ad esempio, per rimuovere un proxy referenziato da un host, `proxyid` deve essere impostato su 0 (`"proxyid": "0"`), oppure, per filtrare gli host monitorati dal server, l'opzione `proxyids` deve essere impostata su 0 (`"proxyids": "0"`).

Parametri comuni dei metodi "get" I seguenti parametri sono supportati da tutti i metodi `get`:

Parametro	Tipo	Descrizione
countOutput	boolean	Restituisce il numero di record nel risultato invece dei dati effettivi.
editable	boolean	Se impostato su `true`, restituisce solo gli oggetti per i quali l'utente dispone dei permessi di scrittura. Predefinito: `false`.
excludeSearch	boolean	Restituisce i risultati che non corrispondono ai criteri specificati nel parametro `search`.
filter	object	Restituisce solo i risultati che corrispondono esattamente al filtro specificato. Accetta un oggetto, in cui le chiavi sono nomi di proprietà (ad esempio, proprietà dell'oggetto Host in `host.get`, proprietà dell'oggetto Item in `item.get`, ecc.) e i valori sono un singolo valore oppure un array di valori da confrontare. Non supporta proprietà del tipo di dati `text`. Si noti che alcuni metodi hanno funzionalità specifiche per questo parametro, descritte nella pagina del metodo (ad esempio, il parametro `filter` in host.get supporta anche le proprietà dell'interfaccia Host).
limit	integer	Limita il numero di record restituiti.
output	query	Proprietà dell'oggetto da restituire. Si noti che l'ID dell'oggetto (ovvero `hostid`, `itemid`, ecc.) è sempre incluso nella risposta, anche se non è specificato nel parametro `output`. Predefinito: `extend`.
preservekeys	boolean	Usa gli ID come chiavi nell'array risultante.
search	object	Restituisce i risultati che corrispondono al modello specificato (senza distinzione tra maiuscole e minuscole). Accetta un oggetto, in cui le chiavi sono nomi di proprietà (ad esempio, proprietà dell'oggetto Host in `host.get`, proprietà dell'oggetto Item in `item.get`, ecc.) e i valori sono stringhe da cercare. Se non vengono fornite opzioni aggiuntive, verrà eseguita una ricerca `LIKE "%…%"`. Supporta solo proprietà del tipo di dati `string` e `text`. Si noti che alcuni metodi hanno funzionalità specifiche per questo parametro, descritte nella pagina del metodo (ad esempio, il parametro `search` in host.get supporta anche le proprietà dell'interfaccia Host).
searchByAny	boolean	Se impostato su `true`, restituisce i risultati che corrispondono a uno qualsiasi dei criteri specificati nel parametro `filter` o `search`, invece che a tutti. Predefinito: `false`.
searchWildcardsEnabled	boolean	Se impostato su `true`, abilita l'uso di "*" come carattere jolly nel parametro `search`. Predefinito: `false`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Fare riferimento alla descrizione di uno specifico metodo API get per un elenco delle proprietà che possono essere usate per l'ordinamento. Le macro non vengono espanse prima dell'ordinamento. Se non viene specificato alcun valore, i dati verranno restituiti senza ordinamento.

Parametro	Tipo	Descrizione
sortorder	string/array	Ordine di ordinamento. Se viene passato un array, ciascun valore verrà associato alla proprietà corrispondente specificata nel parametro `sortfield`. Valori possibili: `ASC` - *(predefinito)* crescente; `DESC` - decrescente.
startSearch	boolean	Il parametro `search` confronterà l'inizio dei campi, ovvero eseguirà invece una ricerca `LIKE "...%"`. Ignorato se `searchWildcardsEnabled` è impostato su `true`.

Flag di origine dell'entità I metodi get restituiscono una proprietà `flags` per le entità correlate al low-level discovery (regola LLD/prototipo di regola LLD, item/prototipo di item, ecc.). Questa proprietà è utile per indicare se l'entità è stata rilevata oppure no, poiché la modifica delle entità rilevate è limitata.

La proprietà `flags` restituisce un risultato basato su una combinazione (operazione "+") di questi valori:

Value	Description
0	Entità di base (item, trigger, graph, host)
1	Regola di low-level discovery
2	Qualsiasi prototipo (prototipo di item, prototipo di trigger, prototipo di regola LLD, ecc.)
4	Entità rilevata (item, trigger, graph, host, regola LLD rilevati)

Il valore **combinato** restituito dalla proprietà `flags` può essere:

Value	Combination of	Description
0	0	Entità semplice (item, trigger, graph, host).
2	2	Prototipo di entità (prototipo di item, prototipo di trigger, ecc.).
6	2+4	item, trigger, graph, host rilevati (convertiti da prototipo).
1	1	Regola di low-level discovery.
3	1+2	Prototipo di regola di low-level discovery.
5	1+4	Regola di low-level discovery rilevata (convertita da prototipo).
7	1+2+4	Prototipo di regola di low-level discovery rilevato.

Esempi Verifica dei permessi utente

L'utente ha il permesso di scrivere negli host i cui nomi iniziano con "MySQL" o "Linux" ?

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": "0",
```

```
    "id": 1
}
```

Note:
Un risultato pari a zero significa che non ci sono host con permessi di lettura/scrittura.

Conteggio delle mancate corrispondenze

Conta il numero di host i cui nomi non contengono la sottostringa ”ubuntu”

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

Ricerca di host utilizzando caratteri jolly

Trova gli host il cui nome contiene la parola ”server” e che hanno porte di interfaccia ”10050” oppure ”10071”. Ordina il risultato per nome host in ordine decrescente e limitalo a 5 host.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "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
}
```

Ricerca di host utilizzando caratteri jolly con ”preservekeys”

Se si aggiunge il parametro ”preservekeys” alla richiesta precedente, il risultato viene restituito come array associativo, in cui le chiavi sono gli ID degli oggetti.

Richiesta:

```
{
    "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
}
```

Risposta:

```
{
    "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
}
```

Appendice 2. Modifiche dalla 7.4 alla 8.0

Modifiche incompatibili

Altre modifiche e correzioni di bug

Appendice 3. Modifiche nella versione 8.0

8.0.1 Questa versione non è ancora stata rilasciata.

Estensioni

Panoramica Sebbene Zabbix offra una molteplicità di funzionalità, c'è sempre spazio per funzionalità aggiuntive. Le estensioni sono un modo pratico per modificare e migliorare le capacità di monitoraggio di Zabbix senza cambiarne il codice sorgente.

È possibile estendere le funzionalità di Zabbix sia utilizzando le opzioni di estensione integrate (item trapper, parametri utente, ecc.) sia utilizzando o creando estensioni personalizzate (moduli caricabili, plugin, ecc.).

Questa sezione fornisce una panoramica con riferimenti a tutte le opzioni per estendere Zabbix.

Raccolta dati con comandi personalizzati Item trapper

Gli item trapper sono item che accettano dati in ingresso invece di interrogarli. Gli item trapper sono utili per inviare dati specifici a server o proxy Zabbix, ad esempio dati periodici di disponibilità e prestazioni nel caso di script utente a lunga esecuzione. L'invio dei dati a server o proxy Zabbix è possibile usando l'utility Zabbix sender oppure il protocollo di Zabbix sender. L'invio dei dati a server Zabbix è inoltre possibile usando il metodo API `history.push`.

Controlli esterni

Un controllo esterno è un item per eseguire controlli avviando un file eseguibile, ad esempio uno script shell o un binario.

I controlli esterni vengono eseguiti da Zabbix server o proxy (quando l'host è monitorato dal proxy) e non richiedono un agent in esecuzione sull'host monitorato.

Parametri utente

Un parametro utente è un comando definito dall'utente (associato a una chiave definita dall'utente) che, quando eseguito, può recuperare i dati necessari dal host su cui è in esecuzione Zabbix agent. I parametri utente sono utili per configurare item di agent o agent 2 che non sono predefiniti in Zabbix.

Item `system.run[]` dell'agent Zabbix

`system.run[]` è un item dell'agent Zabbix per un comando definito dall'utente (associato a una chiave predefinita `system.run[]`, ad esempio `system.run[myscript.sh]`) che può essere eseguito sul host in cui è in esecuzione l'agent Zabbix.

> **Note:**
> Gli item `system.run[]` sono disabilitati per impostazione predefinita e, se utilizzati, devono essere abilitati (consentiti) e definiti nel file di configurazione dell'agent Zabbix o dell'agent Zabbix 2 (parametro di configurazione `AllowKey` o `AllowKeyRegexp`).

Attention:
I comandi definiti dall'utente negli item come controlli esterni, parametri utente e item `system.run[]` di Zabbix agent vengono eseguiti dall'utente del sistema operativo utilizzato per eseguire i componenti Zabbix. Per eseguire questi comandi, questo utente deve disporre delle autorizzazioni necessarie.

item HTTP agent

L'item HTTP agent consente di eseguire richieste di dati tramite HTTP/HTTPS. Gli item HTTP agent sono utili per inviare richieste agli endpoint HTTP e recuperare dati da servizi come *Elasticsearch* e *OpenWeatherMap*, per verificare lo stato dell'API di Zabbix o lo stato del web server Apache o NGINX, ecc. Gli item HTTP agent (con il trapping abilitato) possono anche funzionare come item trapper.

Item script

Uno script item è un item per eseguire codice JavaScript definito dall'utente che recupera dati tramite HTTP/HTTPS. Gli item script sono utili quando la funzionalità fornita dagli item HTTP agent non è sufficiente. Ad esempio, in scenari di raccolta dati complessi che richiedono più passaggi o logica articolata, un item script può essere configurato per effettuare una chiamata HTTP, quindi elaborare i dati ricevuti, e poi passare il valore trasformato a una seconda chiamata HTTP.

Note:
Gli elementi dell'agente HTTP e gli elementi dello script sono supportati dal server e dal proxy Zabbix, e non richiedono un agente in esecuzione sull'host monitorato.

Estensioni avanzate Moduli caricabili

I moduli caricabili, scritti in C, sono un'opzione versatile e orientata alle prestazioni per estendere le funzionalità dei componenti di Zabbix (server, proxy, agent) sulle piattaforme UNIX. Un modulo caricabile è sostanzialmente una libreria condivisa utilizzata dal demone Zabbix e caricata all'avvio. La libreria deve contenere determinate funzioni, in modo che un processo Zabbix possa rilevare che il file è effettivamente un modulo che può caricare e utilizzare.

I moduli caricabili offrono numerosi vantaggi, tra cui la possibilità di aggiungere nuove metriche o implementare qualsiasi altra logica (ad esempio, l'esportazione dei dati di storico di Zabbix), prestazioni elevate e la possibilità di sviluppare, utilizzare e condividere le funzionalità che forniscono. Ciò contribuisce a una manutenzione senza problemi e aiuta a fornire nuove funzionalità più facilmente e indipendentemente dal codice di base di Zabbix.

I moduli caricabili sono particolarmente utili in una configurazione di monitoraggio complessa. Quando si monitorano sistemi embedded, si dispone di un numero elevato di parametri monitorati oppure si utilizzano script pesanti con logica complessa o tempi di avvio lunghi, estensioni come i parametri utente, gli item `system.run[]` di Zabbix agent e i controlli esterni avranno un impatto sulle prestazioni. I moduli caricabili offrono un modo per estendere le funzionalità di Zabbix senza sacrificare le prestazioni.

Plugin

I plugin estendono le capacità di monitoraggio di Zabbix agent 2. Scritti nel linguaggio di programmazione Go, forniscono un'alternativa ai moduli caricabili (scritti in C).

È possibile sviluppare un proprio plugin (collettore di dati indipendente) per Zabbix agent 2. Vedere Developer center.

Personalizzazione degli avvisi Webhook

Un webhook è un tipo di supporto di Zabbix che offre un'opzione per estendere le funzionalità di avviso di Zabbix a software esterni come sistemi di helpdesk, chat o messenger. Analogamente agli item script, i webhook sono utili per effettuare chiamate HTTP utilizzando codice JavaScript personalizzato, ad esempio per inviare notifiche a diverse piattaforme come Microsoft Teams, Discord e Jira. È anche possibile restituire alcuni dati (per esempio, sui ticket helpdesk creati) che vengono poi visualizzati in Zabbix.

I webhook esistenti sono disponibili nel repository Git di Zabbix. Per lo sviluppo di webhook personalizzati, vedere le Linee guida per lo sviluppo dei webhook.

Script di avviso

Uno script di avviso è un tipo di supporto di Zabbix che offre la possibilità di creare un modo alternativo (script) per gestire gli avvisi di Zabbix. Gli script di avviso sono utili se non si è soddisfatti dei tipi di supporto esistenti per l'invio degli avvisi in Zabbix.

Personalizzazione del frontend Temi personalizzati

È possibile modificare l'aspetto visivo del frontend di Zabbix utilizzando temi personalizzati. Consulta le istruzioni sulla creazione e l'applicazione dei tuoi temi.

Moduli del frontend

I moduli del frontend offrono la possibilità di estendere le funzionalità del frontend di Zabbix aggiungendo moduli di terze parti oppure sviluppandone di propri. Con i moduli del frontend è possibile aggiungere nuove voci di menu, le rispettive viste, azioni, ecc.

Script globali Uno script globale è un insieme di comandi definito dall'utente che può essere eseguito su una destinazione di monitoraggio (tramite l'interprete shell (/bin/sh)), a seconda dell'ambito configurato e dei permessi dell'utente. Gli script globali possono essere configurati per le seguenti azioni:

- Operazione di un'azione
- Azione manuale sull'host
- Azione manuale sull'evento

Gli script globali sono utili in molti casi. Ad esempio, se configurati per le operazioni delle azioni o per le azioni manuali sull'host, è possibile usare gli script globali per eseguire automaticamente o manualmente comandi remoti come il riavvio di un'applicazione (web server, middleware, CRM, ecc.) oppure la liberazione di spazio su disco (rimozione di file meno recenti, pulizia di `/tmp`, ecc.). Oppure, un altro esempio: se configurati per le azioni manuali sugli eventi, è possibile usare gli script globali per gestire i ticket dei problemi in sistemi esterni.

Gli script globali possono essere eseguiti da Zabbix server, proxy o agent.

> **Attention:**
> I comandi definiti dall'utente vengono eseguiti dall'utente del sistema operativo utilizzato per eseguire i componenti Zabbix. Per eseguire questi comandi, questo utente deve disporre delle autorizzazioni necessarie.

API di Zabbix Zabbix API è un'API basata su HTTP che fa parte del frontend di Zabbix. Con Zabbix API, è possibile eseguire una qualsiasi delle seguenti operazioni:

- Recuperare e modificare programmaticamente la configurazione di Zabbix.
- Importare ed esportare la configurazione di Zabbix.
- Accedere ai dati storici e di trend di Zabbix.
- Configurare le applicazioni affinché funzionino con Zabbix.
- Integrare Zabbix con software di terze parti.
- Automatizzare le attività di routine.

Zabbix API è costituita da una molteplicità di metodi raggruppati nominalmente in API separate. Ogni metodo esegue un'attività specifica. Per i metodi disponibili, nonché per una panoramica delle funzioni fornite da Zabbix API, vedere il Riferimento ai metodi di Zabbix API.

Moduli caricabili

Panoramica

I moduli caricabili offrono un'opzione orientata alle prestazioni per estendere le funzionalità di Zabbix.

È possibile estendere le funzionalità di Zabbix in molti modi, ad esempio con i parametri utente, i controlli esterni e gli item Zabbix agent system.run. Questi metodi funzionano molto bene, ma presentano un importante svantaggio, vale a dire fork(). Zabbix deve eseguire il fork di un nuovo processo ogni volta che gestisce una metrica utente, e questo non è positivo per le prestazioni. Normalmente non è un grosso problema, tuttavia può diventare una questione seria nel monitoraggio di sistemi embedded, in presenza di un numero elevato di parametri monitorati oppure di script pesanti con logica complessa o tempi di avvio lunghi.

Il supporto dei moduli caricabili offre modi per estendere Zabbix agent, server e proxy senza sacrificare le prestazioni.

Un modulo caricabile è fondamentalmente una libreria condivisa utilizzata dal demone Zabbix e caricata all'avvio. La libreria deve contenere determinate funzioni, in modo che un processo Zabbix possa rilevare che il file è effettivamente un modulo che può caricare e utilizzare.

I moduli caricabili offrono numerosi vantaggi. Le prestazioni elevate e la possibilità di implementare qualsiasi logica sono molto importanti, ma forse il vantaggio più importante è la possibilità di sviluppare, utilizzare e condividere moduli Zabbix. Ciò contribuisce a una manutenzione senza problemi e aiuta a fornire nuove funzionalità più facilmente e indipendentemente dal codice core di Zabbix.

La licenza dei moduli e la distribuzione in forma binaria sono regolate dalla licenza AGPL-3.0 (i moduli si collegano a Zabbix in fase di esecuzione e utilizzano gli header di Zabbix; l'intero codice di Zabbix è concesso in licenza AGPL-3.0 a partire da Zabbix 7.0). La compatibilità binaria non è garantita da Zabbix.

La stabilità della Module API è garantita durante un ciclo di rilascio Zabbix LTS (Long Term Support). La stabilità della Zabbix API non è garantita (tecnicamente è possibile chiamare funzioni interne di Zabbix da un modulo, ma non vi è alcuna garanzia che tali moduli funzionino).

API del modulo

Affinché una libreria condivisa venga trattata come un modulo Zabbix, deve implementare ed esportare diverse funzioni. Attualmente nell'API dei moduli Zabbix sono presenti sei funzioni, di cui solo una è obbligatoria e le altre cinque sono facoltative.

Interfaccia obbligatoria

L'unica funzione obbligatoria è `zbx_module_api_version()`:

```
int zbx_module_api_version(void);
```

Questa funzione deve restituire la versione dell'API implementata da questo modulo e, affinché il modulo venga caricato, questa versione deve corrispondere alla versione dell'API del modulo supportata da Zabbix. La versione dell'API del modulo supportata da Zabbix è `ZBX_MODULE_API_VERSION`. Quindi questa funzione deve restituire questa costante. La vecchia costante `ZBX_MODULE_API_VERSION_ONE`, usata per questo scopo, è ora definita come uguale a `ZBX_MODULE_API_VERSION` per preservare la compatibilità del codice sorgente, ma il suo utilizzo non è consigliato.

Interfaccia opzionale

Le funzioni opzionali sono `zbx_module_init()`, `zbx_module_item_list()`, `zbx_module_item_timeout()`, `zbx_module_history_w` e `zbx_module_uninit()`:

```
int zbx_module_init(void);
```

Questa funzione deve eseguire l'inizializzazione necessaria per il modulo (se presente). Se ha esito positivo, deve restituire `ZBX_MODULE_OK`. In caso contrario, deve restituire `ZBX_MODULE_FAIL`. In quest'ultimo caso Zabbix non si avvierà.

```
ZBX_METRIC  *zbx_module_item_list(void);
```

Questa funzione deve restituire un elenco di item supportati dal modulo. Ogni item è definito in una struttura `ZBX_METRIC`, vedere la sezione seguente per i dettagli. L'elenco termina con una struttura `ZBX_METRIC` con il campo "key" impostato a NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Se il modulo esporta `zbx_module_item_list()`, questa funzione viene usata da Zabbix per specificare le impostazioni di timeout nel file di configurazione di Zabbix che i controlli degli item implementati dal modulo devono rispettare. Qui, il parametro "timeout" è espresso in secondi.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Questa funzione deve restituire le funzioni di callback che Zabbix server userà per esportare la cronologia dei diversi tipi di dati. Le funzioni di callback sono fornite come campi della struttura `ZBX_HISTORY_WRITE_CBS`; i campi possono essere NULL se il modulo non è interessato alla cronologia di un determinato tipo.

```
int zbx_module_uninit(void);
```

Questa funzione deve eseguire la necessaria deinizializzazione (se presente), come liberare le risorse allocate, chiudere i descrittori di file, ecc.

Tutte le funzioni vengono chiamate una volta all'avvio di Zabbix quando il modulo viene caricato, ad eccezione di `zbx_module_uninit()`, che viene chiamata una volta all'arresto di Zabbix quando il modulo viene scaricato.

Definizione degli item

Ogni item è definito in una struttura `ZBX_METRIC`:

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

Qui, **key** è la chiave dell'item (ad esempio, "dummy.random"), **flags** è `CF_HAVEPARAMS` oppure 0 (a seconda che l'item accetti o meno parametri), **function** è una funzione C che implementa l'item (ad esempio, `zbx_module_dummy_random`), e **test_param** è l'elenco dei parametri da usare quando Zabbix agent viene avviato con il flag `-p` (ad esempio, "1,1000", può essere NULL). Una definizione di esempio potrebbe essere la seguente:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Ogni funzione che implementa un item deve accettare due parametri puntatore: il primo di tipo AGENT_REQUEST e il secondo di tipo 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;
}
```

Queste funzioni devono restituire SYSINFO_RET_OK se il valore dell'item è stato ottenuto correttamente. In caso contrario, devono restituire SYSINFO_RET_FAIL. Vedere l'esempio del modulo "dummy" qui sotto per i dettagli su come ottenere informazioni da AGENT_REQUEST e su come impostare informazioni in AGENT_RESULT.

Fornire callback per l'esportazione della history

Attention:
L'esportazione della history tramite modulo non è più supportata da Zabbix proxy.

Il modulo può specificare funzioni per esportare i dati di history per tipo: Numeric (float), Numeric (unsigned), Character, Text e 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;
```

Ognuna di esse deve accettare come argomenti l'array history di history_num elementi. A seconda del tipo di dati di history da esportare, history è un array delle seguenti strutture, rispettivamente:

```
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;
```

I callback saranno utilizzati dai processi history syncer di Zabbix server al termine della procedura di sincronizzazione della history, dopo che i dati sono stati scritti nel database di Zabbix e salvati nella value cache.

Attention:
In caso di errore interno nel modulo di esportazione della history, si raccomanda che il modulo sia scritto in modo tale da non bloccare l'intero monitoraggio fino al ripristino, ma da scartare i dati e consentire a Zabbix server di continuare a funzionare.

Compilazione dei moduli

I moduli sono attualmente pensati per essere compilati all'interno dell'albero dei sorgenti di Zabbix, perché l'API dei moduli dipende da alcune strutture dati definite negli header di Zabbix.

L'header più importante per i moduli caricabili è `include/module.h`, che definisce queste strutture dati. Gli altri header di sistema necessari per far funzionare correttamente `include/module.h` sono `stdlib.h` e `stdint.h`.

Con queste informazioni, tutto è pronto per compilare il modulo. Il modulo dovrebbe includere `stdlib.h`, `stdint.h` e `module.h`, e lo script di compilazione dovrebbe assicurarsi che questi file siano nel percorso di inclusione. Per i dettagli, vedere il modulo di esempio "dummy" qui sotto.

Un altro header utile è `include/zbxcommon.h`, che definisce la funzione `zabbix_log()`, utilizzabile per scopi di logging e debug.

Parametri di configurazione

Zabbix agent, server e proxy supportano due parametri per gestire i moduli:

- `LoadModulePath` - percorso completo della posizione dei moduli caricabili
- `LoadModule` - modulo o moduli da caricare all'avvio.

I moduli devono trovarsi in una directory specificata da `LoadModulePath` oppure il percorso deve precedere il nome del modulo. Se il percorso precedente è assoluto (inizia con /), `LoadModulePath` viene ignorato. Sono consentiti più parametri `LoadModule`.

Ad esempio, per estendere Zabbix agent è possibile aggiungere i seguenti parametri:

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

All'avvio dell'agent, i moduli mariadb.so, apache.so e kernel.so verranno caricati dalla directory `/usr/local/lib/zabbix/agent`, mentre dummy.so verrà caricato da `/usr/local/lib/zabbix`. L'agent non si avvierà se manca un modulo, in caso di autorizzazioni errate o se una libreria condivisa non è un modulo Zabbix.

Configurazione del frontend

I moduli caricabili sono supportati da Zabbix agent, server e proxy. Pertanto, il tipo di item nel frontend di Zabbix dipende da dove viene caricato il modulo. Se il modulo viene caricato nell'agent, allora il tipo di item dovrebbe essere "Zabbix agent" o "Zabbix agent (active)". Se il modulo viene caricato nel server o nel proxy, allora il tipo di item dovrebbe essere "Simple check".

L'esportazione della history tramite i moduli Zabbix non richiede alcuna configurazione del frontend. Se il modulo viene caricato correttamente dal server e fornisce la funzione `zbx_module_history_write_cbs()` che restituisce almeno una funzione callback non NULL, allora l'esportazione della history verrà abilitata automaticamente.

Modulo dummy

Zabbix include un modulo di esempio scritto in linguaggio C. Il modulo si trova in `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
```

Il modulo è ben documentato e può essere usato come template per i tuoi moduli.

Dopo aver eseguito `./configure` nella directory radice del tree dei sorgenti di Zabbix, come descritto sopra, esegui semplicemente `make` per compilare `dummy.so`.

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* la variabile mantiene l'impostazione del timeout per l'elaborazione degli item */
static int  item_timeout = 0;

/* il modulo DOVREBBE definire le funzioni interne come static e usare un pattern di denominazione diverso
/* interni di Zabbix (zbx_*) e dalle funzioni API dei moduli caricabili (zbx_module_*) per evitare conflit
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)
    {
```

```
        /* imposta un messaggio di errore opzionale */
        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)
    {
        /* imposta un messaggio di errore opzionale */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* non c'è una validazione rigorosa dei parametri e dei tipi per semplicità */
    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)
{
    /* inizializzazione per dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
```

```
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
```

```
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

Il modulo esporta tre nuovi item:

- `dummy.ping` - restituisce sempre '1'
- `dummy.echo[param1]` - restituisce il primo parametro così com'è, ad esempio `dummy.echo[ABC]` restituirà ABC
- `dummy.random[param1, param2]` - restituisce un numero casuale nell'intervallo tra param1 e param2, ad esempio `dummy.random[1,1000000]`

Limitazioni

Il supporto dei moduli caricabili è implementato solo per la piattaforma Unix. Ciò significa che non funziona per gli agent Windows.

In alcuni casi, un modulo potrebbe dover leggere i parametri di configurazione relativi al modulo da `zabbix_agentd.conf`. Al momento non è supportato. Se il tuo modulo deve utilizzare alcuni parametri di configurazione, probabilmente dovresti implementare l'analisi di un file di configurazione specifico del modulo.

Moduli del frontend

Panoramica

È possibile migliorare la funzionalità del frontend di Zabbix aggiungendo moduli di terze parti oppure sviluppando moduli personalizzati, senza dover modificare il codice sorgente di Zabbix.

Tieni presente che il codice del modulo verrà eseguito con gli stessi privilegi del codice sorgente di Zabbix. Questo significa:

- I moduli di terze parti possono essere dannosi. Devi fidarti dei moduli che installi.
- Errori nel codice di un modulo di terze parti possono causare il crash del frontend. In tal caso, è sufficiente rimuovere il codice del modulo dal frontend. Non appena ricarichi il frontend di Zabbix, vedrai una nota che indica che alcuni moduli sono assenti. Vai a Amministrazione moduli (in *Amministrazione* > *Generale* > *Moduli*) e fai di nuovo clic su *Scansiona directory* per rimuovere dal database i moduli inesistenti.

Installazione

Leggere sempre il manuale di installazione per un modulo specifico. Si consiglia di installare i nuovi moduli uno alla volta per individuare facilmente eventuali errori.

Prima di installare un modulo:

- Assicurarsi di aver scaricato il modulo da una fonte attendibile. L'installazione di codice dannoso può portare a conseguenze, come la perdita di dati.
- Versioni diverse dello stesso modulo (stesso ID) possono essere installate in parallelo, ma solo una singola versione può essere abilitata alla volta.

Passaggi per installare un modulo:

- Decomprimere il modulo nella propria cartella all'interno della directory `modules` dell'installazione del frontend di Zabbix (ad esempio, `zabbix/ui/modules`).
- Verificare che la cartella del modulo contenga almeno il file `manifest.json`.
- Andare a Amministrazione moduli e fare clic sul pulsante *Scan directory*.
- Il nuovo modulo apparirà nell'elenco insieme alla sua versione, autore, descrizione e stato.
- Abilitare il modulo facendo clic sul suo stato.

Risoluzione dei problemi:

Problema	Soluzione
Il modulo non è apparso nell'elenco	Se il modulo non è apparso nell'elenco, assicurarsi che `manifest.json` esista in `modules/your-module/`. Se non esiste, è possibile che il modulo sia stato decompresso nella directory sbagliata. Se invece esiste, il modulo potrebbe non essere compatibile con la versione di Zabbix in uso. Inoltre, verificare che l'utente del server web abbia almeno accesso in lettura ed esecuzione (`r-x`) alla cartella del modulo e a tutte le sottodirectory, e accesso in lettura (`r--`) a tutti i file al suo interno.
Il frontend è andato in crash	Il codice del modulo non è compatibile con la versione corrente di Zabbix o con la configurazione del server. Eliminare i file del modulo e ricaricare il frontend. Verrà visualizzato un avviso che indica che alcuni moduli sono assenti. Andare a Amministrazione moduli e fare di nuovo clic su *Scan directory* per rimuovere dal database i moduli inesistenti.
Appare un messaggio di errore relativo a namespace, ID o azioni identici	Il nuovo modulo ha tentato di registrare un namespace, un ID o azioni già registrati da altri moduli abilitati. Disabilitare il modulo in conflitto (menzionato nel messaggio di errore) prima di abilitare il nuovo modulo.
Appaiono messaggi di errore tecnici	Segnalare gli errori allo sviluppatore del modulo.

Sviluppo di moduli

Per informazioni sullo sviluppo di moduli personalizzati, vedere il Centro sviluppatori.

Appendici

Panoramica Questa sezione contiene materiali di riferimento, parametri di configurazione, specifiche tecniche e guide procedurali che supportano il manuale principale.

Riferimento essenziale

- Installazione e configurazione - creazione del database, installazione e configurazione di server e agent.
- Configurazione dei processi - parametri di configurazione per Zabbix server, proxy, agent e altro.
- Item - riferimento alle chiavi degli item e ai tipi di dati supportati.
- Macro - sintassi e utilizzo delle macro integrate e utente.

Installazione e configurazione

Questa sezione illustra la creazione del database, la configurazione della sicurezza, le procedure di configurazione specifiche della piattaforma e la configurazione delle funzionalità.

Utilizza la barra laterale per trovare la procedura pertinente al tuo ambiente.

Creazione del database

Panoramica

Un database Zabbix deve essere creato durante l'installazione di Zabbix server o proxy.

Questa sezione fornisce istruzioni per creare un database Zabbix. Per ciascun database supportato è disponibile un insieme separato di istruzioni.

Note:
Per migliorare la sicurezza del database creando ruoli/utenti del database con privilegi minimi, vedere le best practice per la creazione del database per ciascun database supportato:

- MySQL/MariaDB
- PostgreSQL/TimescaleDB

Per configurare connessioni TLS sicure, vedere Secure connection to the database.

UTF-8 è l'unica codifica supportata da Zabbix. È noto che funziona senza alcun problema di sicurezza. Gli utenti devono essere consapevoli che esistono problemi di sicurezza noti se si utilizzano alcune delle altre codifiche. Per passare a UTF-8, vedere Repairing Zabbix database character set and collation. Vedere anche Limits of filtering with utf8mb4 collations.

Note:
Se si esegue l'installazione da Zabbix Git repository, è necessario eseguire il seguente comando prima di procedere ai passaggi successivi:

 `make dbschema`

MySQL/MariaDB

Il set di caratteri utf8mb4 con la collation utf8mb4_bin è supportato affinché il server/proxy Zabbix funzioni correttamente con i database MySQL/MariaDB.
 Per evitare problemi futuri, è fortemente consigliato utilizzare il set di caratteri utf8mb4 con la collation utf8mb4_bin sia per le nuove installazioni sia per quelle esistenti. Per convertire i database che utilizzano il set di caratteri utf8mb3 e la collation utf8mb3_bin nel set di caratteri utf8mb4 e nella collation utf8mb4_bin, seguire le istruzioni in Riparazione del set di caratteri e della collation del database Zabbix.

I trigger deterministici devono essere creati durante l'importazione dello schema. Su MySQL e MariaDB, ciò richiede che `GLOBAL log_bin_trust_function_creators = 1` sia impostato se la registrazione binaria è abilitata, non si dispone dei privilegi di superuser e `log_bin_trust_function_creators = 1` non è impostato nel file di configurazione di MySQL.

Se si installa Zabbix dai pacchetti o dai container, consultare le istruzioni corrispondenti.

Se si installa Zabbix dai sorgenti:

- Creare e configurare un database e un utente. Per motivi di sicurezza, utilizzare una password complessa e univoca per l'utente del database Zabbix.

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- Importare i dati nel database e impostare il set di caratteri utf8mb4 come predefinito (supponendo di trovarsi nella directory principale dei sorgenti Zabbix). Per un database proxy Zabbix, deve essere importato solo `schema.sql` (non `images.sql` né `data.sql`).

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### fermarsi qui se si sta creando un database per il proxy Zabbix
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` può essere disabilitato dopo che lo schema è stato importato correttamente:

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

È necessario disporre di un utente del database con i permessi per creare oggetti del database.

Se stai installando Zabbix da pacchetti o container, consulta le rispettive istruzioni.

Se stai installando Zabbix dai sorgenti:

- Crea un utente del database.

Il seguente comando shell creerà l'utente `zabbix`. Specifica una password quando richiesto e ripeti la password (nota: potresti prima essere richiesto di inserire la password di `sudo`):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Crea un database.

Il seguente comando shell creerà il database `zabbix` (ultimo parametro) con l'utente creato in precedenza come proprietario (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importa lo schema e i dati iniziali (supponendo che tu sia nella directory root dei sorgenti di Zabbix). Per un database di Zabbix proxy, deve essere importato solo `schema.sql` (non `images.sql` né `data.sql`).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### fermati qui se stai creando il database per Zabbix proxy
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
I comandi sopra sono forniti come esempio e funzioneranno nella maggior parte delle installazioni GNU/Linux. Puoi usare comandi diversi a seconda di come è configurato il tuo sistema/database, ad esempio:

 `psql -U <username>`

 Se hai problemi nella configurazione del database, consulta l'amministratore del database.

TimescaleDB

Le istruzioni per creare e configurare TimescaleDB sono fornite in una sezione separata.

Elasticsearch e ClickHouse

Zabbix può anche memorizzare i dati storici in Elasticsearch o ClickHouse. Per le istruzioni di configurazione, vedere le pagine collegate.

SQLite

L'uso di SQLite è supportato solo per **Zabbix proxy**!

Il database verrà creato automaticamente se non esiste.

Torna alla sezione di installazione.

Riparazione del set di caratteri e della collazione del database Zabbix

MySQL/MariaDB

Storicamente, MySQL e i suoi derivati usavano 'utf8' come alias di utf8mb3 - l'implementazione a 3 byte di MySQL dello standard UTF8, che è a 4 byte. A partire da MySQL 8.0.28 e MariaDB 10.6.1, il set di caratteri 'utf8mb3' è deprecato e, a un certo punto, il suo supporto verrà rimosso, mentre 'utf8' diventerà un riferimento a 'utf8mb4'. A partire da Zabbix 6.0, 'utf8mb4' è supportato. Per evitare problemi futuri, è fortemente consigliato usare 'utf8mb4'. Un altro vantaggio del passaggio a 'utf8mb4' è il supporto dei caratteri Unicode supplementari.

Warning:
Poiché le versioni precedenti a Zabbix 6.0 non conoscono utf8mb4, assicurati di aggiornare prima Zabbix server e lo schema DB alla versione 6.0.x o successiva prima di eseguire la conversione a utf8mb4.

1. Verifica il set di caratteri e la collation del database.

Ad esempio:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2_general_ci    |
+--------------------------+----------------------+
```

Oppure:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Come si vede, il set di caratteri qui non è 'utf8mb4' e la collation non è 'utf8mb4_bin', quindi dobbiamo correggerli.

2. Arresta Zabbix.

3. Crea una copia di backup del database!

4. Correggi il set di caratteri e la collation a livello di database:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Valori corretti:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Carica lo script per correggere il set di caratteri e la collation a livello di tabella e colonna:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Esegui lo script:

```
               SET @ZABBIX_DATABASE = '<your DB name>';
If MariaDB →   set innodb_strict_mode = OFF;
               CALL zbx_convert_utf8();
If MariaDB →   set innodb_strict_mode = ON;
               drop procedure zbx_convert_utf8;
```

Tieni presente che 'utf8mb4' dovrebbe consumare leggermente più spazio su disco.

7. Se non ci sono errori, potresti voler creare una copia di backup del database con il database corretto.

8. Avvia Zabbix.

Aggiornamento del database alle chiavi primarie

Panoramica

Questa sezione fornisce istruzioni per aggiornare manualmente le tabelle nelle installazioni esistenti alle chiavi primarie.

L'aggiornamento alle chiavi primarie ottimizza il modo in cui i dati vengono indicizzati e accessibili, il che può velocizzare le query e risparmiare spazio. Migliora inoltre la gestione dei dati e la sincronizzazione nelle configurazioni in cluster, aiutando la scalabilità e garantendo che il sistema rimanga affidabile anche in caso di guasto di alcuni server.

> **Attention:**
> Le istruzioni fornite in questa pagina sono pensate per utenti esperti e potrebbero dover essere adattate alla configurazione specifica. L'aggiornamento alle chiavi primarie può richiedere molto tempo e molte risorse. Assicurarsi che sia disponibile spazio libero su disco sufficiente; a seconda delle dimensioni del database e dei dati memorizzati, il processo può richiedere fino a 2,5 volte lo spazio attualmente utilizzato dalle tabelle history.

Le chiavi primarie vengono utilizzate per tutte le tabelle nelle nuove installazioni a partire da Zabbix 6.0.

Non esiste un aggiornamento automatico del database alle chiavi primarie; tuttavia, le installazioni esistenti possono essere aggiornate manualmente **dopo** aver aggiornato il server Zabbix alla versione 6.0 o successiva.

Attention:
A partire da Zabbix 7.0, l'aggiornamento delle tabelle alle chiavi primarie aggiorna anche le tabelle per utilizzare tipi di dati a doppia precisione.

 Se si utilizza Zabbix 7.0 (o versioni successive), le tabelle usano già la doppia precisione. Tuttavia, le istruzioni in questa pagina possono comunque essere seguite per aggiornare le tabelle alle chiavi primarie senza influire sulle tabelle che già usano la doppia precisione.

 Se si utilizza Zabbix 6.4 (o versioni precedenti), si consiglia di aggiornare prima le tabelle alla doppia precisione. Per ulteriori informazioni, vedere Aggiornamento ai valori numerici di intervallo esteso nella documentazione di Zabbix 7.0.

Le istruzioni sono disponibili per:

- MariaDB/MySQL
- PostgreSQL
- TimescaleDB

Note importanti

Per eseguire l'aggiornamento del database:

1. Arrestare Zabbix server.

Si consiglia vivamente di arrestare Zabbix server per la durata dell'aggiornamento. Tuttavia, se strettamente necessario, è possibile eseguire l'aggiornamento mentre il server è in esecuzione (solo per MySQL, MariaDB e PostgreSQL senza TimescaleDB).

2. Eseguire un backup del database.
3. Installare il pacchetto zabbix-sql-scripts più recente compatibile con la propria versione di Zabbix (ad esempio, per RHEL: `dnf install zabbix-sql-scripts`).
4. Eseguire gli script per il proprio database.
5. Avviare Zabbix server.

Warning:
Eseguire gli script solo per il database del server. Il proxy non trarrà beneficio da questo aggiornamento.

Se il database utilizza partizioni, contattare l'amministratore DB o il Supporto Zabbix per assistenza.

I file CSV possono essere rimossi dopo un aggiornamento riuscito alle chiavi primarie.

Facoltativamente, Zabbix frontend può essere impostato in modalità di manutenzione.

MariaDB/MySQL

L'esportazione e l'importazione devono essere eseguite in tmux/screen per garantire che la sessione non venga interrotta.

Vedi anche: Note importanti

MySQL 8.4+ con mysqlsh

Questo metodo può essere utilizzato con un server Zabbix in esecuzione, ma si consiglia di arrestare il server per la durata dell'aggiornamento. MySQL Shell (*mysqlsh*) deve essere installato e in grado di connettersi al DB.

- Accedere alla console MySQL come root (consigliato) oppure come qualsiasi utente con privilegi FILE.
- Avviare MySQL con la variabile local_infile abilitata.
- Rinominare le vecchie tabelle e creare le nuove eseguendo `history_upgrade_prepare.sql`.

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- Esportare e importare i dati.

Connettersi tramite mysqlsh. Se si utilizza una connessione socket, potrebbe essere necessario specificare il percorso.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Passare alla modalità JavaScript usando:

```
\js
```

Quindi eseguire il codice seguente (CSVPATH può essere modificato secondo necessità):

```
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" });
```

Se viene visualizzato il messaggio "JavaScript is not supported", l'installazione di MySQL Shell non include il supporto JS. In tal caso, installare il pacchetto ufficiale MySQL Shell package di Oracle (oppure compilarlo dai sorgenti) in modo che la modalità JavaScript sia abilitata.

- Seguire le istruzioni post-migrazione per eliminare le vecchie tabelle.

MariaDB/MySQL 8.4+ senza mysqlsh

Questo metodo di upgrade richiede più tempo e deve essere usato solo se un upgrade con *mysqlsh* non è possibile.

Upgrade delle tabelle

- Accedi alla console MySQL come root (consigliato) oppure come qualsiasi utente con privilegi FILE.
- Se esegui la migrazione con un server Zabbix in esecuzione, avvia MySQL con la variabile local_infile abilitata.
- Rinomina le vecchie tabelle e crea le nuove eseguendo `history_upgrade_prepare.sql`:

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Migrazione con server arrestato

`max_execution_time` (in MySQL) o `max_statement_time` (in MariaDB) devono essere disabilitati prima della migrazione dei dati per evitare timeout durante la migrazione.

Per MySQL:

```
SET @@max_execution_time=0;
```

Per 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;
```

Segui le istruzioni post-migrazione per eliminare le vecchie tabelle.

Migrazione con server in esecuzione

Verifica per quali percorsi l'importazione/esportazione è abilitata:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Se il valore di *secure_file_priv* è un percorso di directory, l'esportazione/importazione verrà eseguita per i file in quella directory. In questo caso, modifica di conseguenza i percorsi dei file nelle query oppure imposta il valore di *secure_file_priv* su una stringa vuota per il tempo dell'upgrade.

Se il valore di *secure_file_priv* è vuoto, l'esportazione/importazione può essere eseguita da qualsiasi percorso.

Se il valore di *secure_file_priv* è NULL, impostalo sul percorso che contiene i dati esportati delle tabelle ('/var/lib/mysql-files/' nell'esempio sopra).

Per ulteriori informazioni, consulta la documentazione MySQL o la documentazione MariaDB.

`max_execution_time` (in MySQL) o `max_statement_time` (in MariaDB) devono essere disabilitati prima dell'esportazione dei dati per evitare timeout durante l'esportazione.

Per MySQL:

```
SET @@max_execution_time=0;
```

Per 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
```

Segui le istruzioni post-migrazione per eliminare le vecchie tabelle.

PostgreSQL

L'esportazione e l'importazione devono essere eseguite in tmux/screen per garantire che la sessione non venga interrotta. Per le installazioni con TimescaleDB, saltare questa sezione e procedere a PostgreSQL + TimescaleDB.

Vedi anche: Note importanti

Aggiornamento delle tabelle

- Rinominare le tabelle usando `history_upgrade_prepare.sql`:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Migrazione con server arrestato

- Esportare la cronologia corrente, importarla nella tabella temporanea, quindi inserire i dati nelle nuove tabelle ignorando i duplicati:

```
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;
```

Vedi i suggerimenti per migliorare le prestazioni di INSERT: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Seguire le istruzioni post-migrazione per eliminare le vecchie tabelle.

Migrazione con server in esecuzione

- Esporta la cronologia corrente, importala nella tabella temporanea, quindi inserisci i dati nelle nuove tabelle ignorando i duplicati:

```
\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;
```

- Segui le istruzioni post-migrazione per eliminare le vecchie tabelle.

PostgreSQL + TimescaleDB

L'esportazione e l'importazione devono essere eseguite in tmux/screen per garantire che la sessione non venga interrotta. Il server Zabbix deve essere arrestato durante l'aggiornamento.

Vedi anche: Note importanti

- Rinominare le tabelle usando `history_upgrade_prepare.sql`.
  - Se la compressione è abilitata (nell'installazione predefinita), eseguire lo script da `/usr/share/zabbix/sql-scripts/postgr`

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - Se la compressione è disabilitata, eseguire lo script da `/usr/share/zabbix/sql-scripts/postgresql/timescaledb/opti`

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
```

- Eseguire gli script di migrazione delle hypertable di TimescaleDB in base alle impostazioni di compressione:
  - Se la compressione è abilitata (nell'installazione predefinita), eseguire gli script da `/usr/share/zabbix/sql-scripts/postgr`

```
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
```

  - Se la compressione è disabilitata, eseguire gli script da `/usr/share/zabbix/sql-scripts/postgresql/timescaledb/opt`

```
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
```

Vedi anche: Suggerimenti per migliorare le prestazioni di INSERT.

- Seguire le istruzioni post-migrazione per eliminare le vecchie tabelle.

Post-migrazione

Per tutti i database, una volta completata la migrazione, procedere come segue:

- Verificare che tutto funzioni come previsto.
- Eliminare le vecchie tabelle:

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- Per TimescaleDB, eliminare anche la seguente vecchia tabella:

```
DROP TABLE trends_old;
```

Vedi anche

- Preparazione della tabella auditlog per il partizionamento

Preparazione della tabella auditlog per il partizionamento

Panoramica

Alcuni database (ad esempio MySQL) richiedono che la colonna di partizionamento faccia parte del vincolo univoco della tabella. Pertanto, per partizionare la tabella `auditlog` in base al tempo, la chiave primaria deve essere modificata da `auditid` a una chiave composta `auditid` + `clock`.

Questa sezione fornisce istruzioni per modificare la chiave primaria della tabella `auditlog`.

> **Attention:**
> Le istruzioni fornite in questa pagina sono pensate per utenti avanzati. Si noti che queste istruzioni potrebbero dover essere adattate alla configurazione specifica in uso. La modifica della chiave primaria può inoltre essere incompatibile con future patch di aggiornamento, quindi potrebbe essere necessario gestire manualmente i futuri aggiornamenti.

 La modifica della chiave primaria può essere un'operazione che richiede molte risorse e molto tempo, a seconda delle dimensioni della tabella `auditlog`. Si consiglia di arrestare Zabbix server e di impostare Zabbix frontend in modalità di manutenzione per la durata della modifica. Tuttavia, se strettamente necessario, esiste un modo per modificare la chiave primaria senza tempi di inattività (vedere sotto).

Il partizionamento della tabella `auditlog` può migliorare, ad esempio, l'housekeeping in configurazioni di grandi dimensioni. Sebbene l'housekeeping di Zabbix attualmente non possa sfruttare le tabelle partizionate (tranne che per TimescaleDB), è possibile disabilitare l'housekeeping di Zabbix ed eliminare le partizioni utilizzando script.

A partire da Zabbix 7.0, la tabella auditlog per TimescaleDB è stata convertita in una hypertable, il che consente al processo di housekeeping di eliminare i dati per chunk. Per aggiornare la tabella auditlog esistente a una hypertable, vedere Aggiornamento dello schema TimescaleDB.

MySQL

Note importanti sulla ricostruzione degli indici

MySQL ricostruisce automaticamente gli indici per la chiave primaria durante l'operazione ALTER TABLE. Tuttavia, è fortemente consigliato ricostruire manualmente anche gli indici con l'istruzione OPTIMIZE TABLE per garantire prestazioni ottimali del database.

La ricostruzione degli indici può richiedere temporaneamente una quantità di spazio su disco aggiuntivo pari a quella utilizzata dalla tabella stessa. Per ottenere le dimensioni correnti dei dati e degli indici, è possibile eseguire le seguenti istruzioni:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Se lo spazio disponibile su disco è un problema, seguire le istruzioni di Alterazione della chiave primaria senza tempi di inattività. Sono disponibili anche altre opzioni:

- Aumentare il parametro MySQL sort_buffer_size può aiutare a ridurre l'uso dello spazio su disco quando si ricostruiscono manualmente gli indici. Tuttavia, la modifica di questa variabile può influire sull'utilizzo complessivo della memoria del database.
- Valutare la possibilità di liberare spazio eliminando dati potenzialmente non necessari.
- Valutare la possibilità di ridurre il parametro *Periodo di conservazione dei dati* housekeeper prima di eseguire l'housekeeper.

Modifica della chiave primaria con downtime

1. Rimuovere la chiave primaria corrente della tabella auditlog e aggiungere la nuova chiave primaria.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Ricostruire gli indici (opzionale ma fortemente consigliato, vedere Note importanti sulla ricostruzione degli indici).

```
OPTIMIZE TABLE auditlog;
```

Modifica della chiave primaria senza downtime

Qui è descritto il metodo manuale per modificare la chiave primaria. In alternativa, puoi usare il toolkit pt-online-schema-change di Percona. Questo toolkit esegue automaticamente le seguenti azioni, riducendo al minimo anche lo spazio utilizzato per la modifica della tabella auditlog.

1. Crea una nuova tabella con la nuova chiave primaria e crea gli indici.

```
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. Scambia le tabelle.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Copia i dati dalla vecchia tabella alla nuova tabella.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Questo può essere fatto a blocchi (più istruzioni `INSERT INTO` con clausole `WHERE clock` secondo necessità) per evitare un uso eccessivo delle risorse.

4. Elimina la vecchia tabella.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Note importanti sulla ricostruzione degli indici

PostgreSQL ricostruisce automaticamente gli indici per la chiave primaria durante l'operazione `ALTER TABLE`. Tuttavia, è fortemente consigliato ricostruire manualmente anche gli indici con l'istruzione `REINDEX TABLE CONCURRENTLY` per garantire prestazioni ottimali del database.

La ricostruzione degli indici può richiedere temporaneamente fino a tre volte lo spazio su disco attualmente utilizzato dagli indici. Per ottenere la dimensione corrente degli indici, è possibile eseguire la seguente query:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Se lo spazio disponibile su disco è un problema, seguire le istruzioni in Modifica della chiave primaria senza tempi di inattività. Sono disponibili anche altre opzioni:

- Aumentare il parametro PostgreSQL `maintenance_work_mem` può aiutare a ridurre l'uso dello spazio su disco quando si ricostruiscono manualmente gli indici. Tuttavia, la modifica di questa variabile può influire sull'uso complessivo della memoria del database.
- Se si dispone di un altro disco o tablespace con più spazio disponibile, si può prendere in considerazione la modifica della posizione di archiviazione temporanea per la ricostruzione dell'indice. È possibile impostare il parametro PostgreSQL `temp_tablespaces` per specificare un tablespace diverso per gli oggetti temporanei.
- Valutare la possibilità di liberare spazio eliminando dati potenzialmente non necessari.
- Valutare la possibilità di ridurre il parametro *Periodo di conservazione dei dati* housekeeper prima di eseguire l'housekeeper.

Modifica della chiave primaria con downtime

1. Rimuovere la chiave primaria corrente della tabella `auditlog` e aggiungere la nuova chiave primaria.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Ricostruire gli indici (opzionale ma altamente consigliato, vedere Note importanti sulla ricostruzione degli indici).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Modifica della chiave primaria senza downtime

Qui è descritto il metodo manuale per modificare la chiave primaria. In alternativa, l'estensione `pg_repack` può essere presa in considerazione per creare una nuova tabella, copiare i dati e scambiare le tabelle.

1. Creare una nuova tabella con la nuova chiave primaria e creare gli indici.

```
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. Scambiare le tabelle.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Copiare i dati dalla vecchia tabella alla nuova tabella.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Questo può essere fatto in blocchi (più istruzioni `INSERT INTO` con clausole `WHERE clock` secondo necessità) per evitare un uso eccessivo delle risorse.

4. Eliminare la vecchia tabella.

```
DROP TABLE auditlog_old;
```

Vedi anche

- Aggiornamento del database alle chiavi primarie

Configurazione di TimescaleDB

Panoramica

Zabbix supporta TimescaleDB, una soluzione di database basata su PostgreSQL che partiziona automaticamente i dati in blocchi basati sul tempo per supportare prestazioni più elevate su larga scala.

> **Warning:**
> Attualmente, TimescaleDB non è supportato da Zabbix proxy.

Le istruzioni in questa pagina possono essere utilizzate per i seguenti scenari:

- Creazione di un database TimescaleDB o migrazione dalle tabelle PostgreSQL esistenti a TimescaleDB (vedere Configurazione).
- Aggiornamento dello schema di un database TimescaleDB esistente durante l'aggiornamento di Zabbix (vedere Aggiornamento dello schema TimescaleDB).

Configurazione

Prerequisiti: estensione TimescaleDB di una versione supportata installata sul server del database. Per le istruzioni di installazione, vedere la documentazione di TimescaleDB.

> **Warning:**
> Prima di installare TimescaleDB, installare una release PostgreSQL supportata dal repository ufficiale di PostgreSQL.

Abilitare l'estensione TimescaleDB per il database specifico eseguendo:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

L'esecuzione di questo comando richiede privilegi di amministratore del database.

> **Note:**
> Se si utilizza uno schema del database diverso da 'public', è necessario aggiungere una clausola SCHEMA al comando sopra. Ad es.:
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Quindi eseguire lo script `postgresql/timescaledb/schema.sql`. Per le nuove installazioni, lo script deve essere eseguito dopo che il normale database PostgreSQL è stato creato con lo schema/dati iniziali (vedere creazione del database).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Ignorare i messaggi di avviso che informano che le best practice non vengono seguite durante l'esecuzione dello script `schema.sql` su TimescaleDB versione 2.9.0 e successive. Nonostante questo avviso, la configurazione verrà completata correttamente.

La migrazione dei dati esistenti di history, trend e audit log può richiedere molto tempo. Zabbix server e frontend devono essere arrestati per tutta la durata della migrazione.

Lo script `schema.sql` imposta i seguenti parametri di housekeeping:

- Override del periodo di history degli item
- Override del periodo di trend degli item

Per utilizzare l'housekeeping partizionato per history e trend, entrambe queste opzioni devono essere abilitate. È anche possibile abilitare l'override singolarmente solo per history o solo per trend.

Per PostgreSQL e TimescaleDB, lo script `postgresql/timescaledb/schema.sql` imposta due parametri aggiuntivi:

- Abilita la compressione
- Comprimi i record più vecchi di 7 giorni

Per rimuovere correttamente i dati compressi tramite housekeeper, devono essere abilitate entrambe le opzioni *Override item history period* e *Override item trend period*. Se l'override è disabilitato e le tabelle contengono chunk compressi, housekeeper non rimuoverà i dati da queste tabelle e verranno visualizzati avvisi di configurazione non corretta nelle sezioni *Housekeeping* e *System information*.

Tutti questi parametri possono essere modificati in *Administration* > *Housekeeping* dopo l'installazione.

Note:
Potresti voler eseguire lo strumento timescaledb-tune fornito da TimescaleDB per ottimizzare i parametri di configurazione di PostgreSQL nel tuo `postgresql.conf`.

Aggiornamento dello schema TimescaleDB

Quando si aggiorna Zabbix a una versione che contiene nuove hypertable di TimescaleDB, Zabbix server non configura automaticamente tali hypertable (ad esempio, durante l'aggiornamento da Zabbix 6.4 a 8.0, poiché le versioni 7.0.0 e 7.0.2 hanno introdotto nuove hypertable).

Per configurare le nuove hypertable di TimescaleDB, seguire questi passaggi:

1. Avviare Zabbix server; in questo modo viene aggiornato il database esistente.
2. Verificare nel file di log del server che l'aggiornamento del database sia stato completato; una volta completato, arrestare Zabbix server. Si noti che il server registra un avviso se tenta di abilitare la compressione per una tabella che non è una hypertable.
3. Eseguire lo script `postgresql/timescaledb/schema.sql`; questo configura le nuove hypertable di TimescaleDB. Si noti che, a partire da Zabbix 7.0.0, il percorso e il nome dello script sono cambiati da `postgresql/timescaledb.sql` a `postgresql/timescaledb/schema.sql`.

Attention:
Ignorare i messaggi di avviso che indicano che le best practice non vengono seguite durante l'esecuzione dello script `schema.sql` su TimescaleDB versione 2.9.0 e successive. Nonostante questo avviso, la configurazione verrà completata correttamente.

Compressione TimescaleDB

La compressione nativa di TimescaleDB è supportata per tutte le tabelle Zabbix che sono hypertable di TimescaleDB. Durante l'aggiornamento o la migrazione a TimescaleDB, la compressione iniziale delle tabelle di grandi dimensioni può richiedere molto tempo.

Nota che la compressione è supportata con la licenza Timescale Community "timescale" e non è supportata con la licenza Apache 2.0 "apache". Se Zabbix rileva che la compressione non è supportata, viene scritto un messaggio di avviso nel log del server Zabbix e gli utenti non possono abilitare la compressione nel frontend.

Note:
Si consiglia agli utenti di familiarizzare con la compressione nella documentazione di TimescaleDB prima di utilizzarla.

Nota che esistono alcune limitazioni imposte dalla compressione, in particolare:

- Le modifiche ai chunk compressi (insert, delete, update) non sono consentite
- Le modifiche dello schema per le tabelle compresse non sono consentite.

Le impostazioni di compressione possono essere modificate nel blocco *Compressione di history, trend e log di audit* nella sezione *Amministrazione* > *Housekeeping* del frontend di Zabbix.

Parametro	Predefinito	Commenti
Abilita compressione	Abilitato	Selezionare o deselezionare la casella non attiva/disattiva la compressione immediatamente. Poiché la compressione è gestita dall'Housekeeper, le modifiche avranno effetto entro un massimo di 2 volte `HousekeepingFrequency` ore (impostato in `zabbix_server.conf`) Dopo la disabilitazione della compressione, i nuovi chunk che rientrano nel periodo di compressione non verranno compressi. Tuttavia, tutti i dati già compressi rimarranno compressi. Per decomprimere i chunk precedentemente compressi, seguire le istruzioni nella documentazione di TimescaleDB. Durante l'aggiornamento da versioni precedenti di Zabbix con supporto TimescaleDB, la compressione non sarà abilitata per impostazione predefinita.
Comprimi record più vecchi di	7d	Questo parametro non può essere inferiore a 7 giorni. A causa dell'immutabilità dei chunk compressi, tutti i dati in ritardo (ad esempio, dati ritardati da un proxy) più vecchi di questo valore verranno scartati.

Per migliorare le prestazioni di aggiornamento dei trend, potrebbe essere utile ridurre il valore di "chunk_time_interval" per le tabelle `trends` e `trends_uint` da 30 giorni a 7 giorni o meno, a seconda di quanti item utilizzano i trend. Lo scopo di questa impostazione è aderire alle best practice di TimescaleDB e garantire che la dimensione del chunk rimanga entro le risorse disponibili del sistema.

Configurazione di Elasticsearch

Zabbix può memorizzare i dati storici in Elasticsearch come alternativa a un database relazionale.

Attention:
Il supporto di Elasticsearch è attualmente sperimentale.

Questa guida illustra la configurazione delle versioni supportate di Elasticsearch. Se stai usando una versione diversa, alcune funzionalità potrebbero non funzionare come previsto.

Elasticsearch può memorizzare i seguenti tipi di valore:

Tipo di valore dell'item	Tabella del database	Tipo Elasticsearch
Numerico (senza segno)	history_uint	uint
Numerico (float)	history	dbl
Carattere	history_str	str
Log	history_log	log
Testo	history_text	text
Binario	history_bin	non supportato da Zabbix
JSON	history_json	json

Note:
Elasticsearch non accetta array JSON. Un valore JSON deve essere un singolo oggetto oppure un insieme di oggetti.

Note importanti

- Elasticsearch richiede libcurl. Vedi requirements per i dettagli.
- Il housekeeper non elimina i dati da Elasticsearch. Per controllare per quanto tempo i dati vengono conservati, vedi Index Lifecycle Management.
- Zabbix **non** calcola né memorizza le trend in Elasticsearch. Valuta di estendere il periodo di conservazione della history per preservare i dati più vecchi.
- Quando viene usato Elasticsearch, le query di intervallo che recuperano valori dal database sono limitate dal timestamp del periodo di conservazione dei dati.

- Elasticsearch non è supportato per i proxy Zabbix.

Se Elasticsearch non è ancora installato, consulta la guida ufficiale all'installazione prima di procedere.

Configurazione di Elasticsearch

Per memorizzare i dati di storico in Elasticsearch, è necessario:

- Creare un indice per ogni tipo di valore che si desidera memorizzare: è qui che Elasticsearch archivia i dati, in modo simile a una tabella in un database relazionale.
- Definire una mappatura per ogni indice: questa definisce la struttura dei dati, in modo simile allo schema di una tabella.
- Configurare una pipeline di ingest per elaborare i valori prima della memorizzazione (necessaria per i valori JSON e per gli indici basati sulla data).

Elasticsearch può memorizzare i dati in un singolo indice per tipo di valore oppure in più indici basati sulla data. Entrambi gli approcci sono descritti di seguito.

Archiviazione della cronologia in un singolo indice

In questo approccio, tutti i dati di cronologia per un determinato tipo di valore vengono scritti in un singolo indice (ad esempio, `uint` o `text`).

Per creare un indice per il tipo di valore *Numeric (unsigned)*, invia la seguente richiesta (con `/uint` nell'URL) alla tua istanza Elasticsearch:

```
curl -X PUT \
 http://localhost:9200/uint \
 -H 'content-type:application/json' \
 -d '{
     "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": { "type": "long" }
      }
   }
}'
```

Elasticsearch risponderà con una conferma che l'indice è stato creato:

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Richieste simili devono essere inviate per ogni ulteriore tipo di valore che si desidera memorizzare in Elasticsearch.

> **Note:**
> Le mappature per tutti i tipi di valore sono disponibili nel repository sorgente di Zabbix.

Ad esempio, per creare un indice per il tipo di valore *Text*:

```
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"
         }
      }
   }
}'
```

Tipo di valore JSON

A differenza degli altri tipi di valore, i valori JSON richiedono un'elaborazione aggiuntiva prima dell'archiviazione.

L'indice seguente utilizza campi separati per i valori analizzati e grezzi, quindi è necessaria una pipeline di ingest per analizzare ogni valore come JSON e memorizzarlo nel campo corretto.

Per creare un indice per il tipo di valore *JSON*, invia la seguente richiesta (con `/json` nell'URL) alla tua istanza Elasticsearch.

```
curl -X PUT \
 http://localhost:9200/json \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
    }
  }
}'
```

Quindi, crea la pipeline di ingest:

```
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 risponderà con una conferma della creazione della pipeline di ingest:

```
{"acknowledged": true}
```

Archiviazione della history in indici basati sulla data

Invece di scrivere tutti i dati della history in un unico indice (ad esempio, `uint`), Elasticsearch può distribuire questi dati su più indici basati sulla data (ad esempio, `uint-2026-01-01`, `uint-2026-01-02`). Questo semplifica la gestione del volume dei dati e della retention nel tempo.

Per abilitarlo, è necessario:

- Creare un index template per ogni tipo di valore che si desidera archiviare: questo indica a Elasticsearch quale mapping applicare quando crea automaticamente un nuovo indice basato sulla data.
- Creare un ingest pipeline per ogni tipo di valore: elabora ogni valore in ingresso e lo indirizza al corretto indice basato sulla data.

Template di indice

Per creare un template per l'indice `text`, invia una richiesta con i seguenti dettagli:

- Usa `_template/text_template` nell'URL della tua istanza Elasticsearch.
- Usa `"text*"` nel campo `"index_patterns"` per corrispondere al nome dell'indice.
- Usa una mappatura per il tipo di valore `text` (vedi le mappature nel repository sorgente di Zabbix).

```
curl -X PUT \
 http://localhost:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "text*" ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Template per l'indice `json`:

```
curl -X PUT \
 http://localhost:9200/_template/json_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "json*" ],
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
```

```
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
      }
   }
}'
```

Pipeline di ingestione

Per creare una pipeline di ingestione per l'indice `text`:

- Usa `_ingest/pipeline/text-pipeline` nell'URL della tua istanza Elasticsearch.
- Includi un processore `date_index_name` per instradare ogni valore all'indice basato sulla data corretto in base al relativo timestamp.

```
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"
         }
      }
   ]
}'
```

Per l'indice `json`, la pipeline deve anche analizzare il valore JSON prima di instradarlo all'indice corretto:

```
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"
         }
      }
```

```
    ]
}'
```

Configurazione di Zabbix server

Nel file di configurazione di Zabbix server (`zabbix_server.conf`), impostare il parametro `HistoryProvider`.

Ad esempio, per archiviare i valori di tipo *Character*, *Log*, *Text* e *JSON* in Elasticsearch (mantenendo i valori *Numeric* in un database):

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

Se si utilizzano indici basati sulla data, aggiungere `date_index=1` al parametro:

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

Dopo aver apportato le modifiche, riavviare Zabbix server:

```
systemctl restart zabbix-server
```

Configurazione del frontend Zabbix

Nel file di configurazione del frontend di Zabbix (`zabbix.conf.php`), impostare la variabile `$HISTORY_PROVIDERS` in modo che corrisponda alla configurazione del server:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

Risoluzione dei problemi

I seguenti passaggi possono aiutarti a risolvere i problemi con la configurazione di Elasticsearch:

1. Controlla i log di Elasticsearch o del server Zabbix per individuare eventuali errori.
2. Per identificare le query lente, usa l'opzione `log_slow_queries` nel parametro di configurazione del server Zabbix `HistoryProvider`.
3. Verifica che Elasticsearch consenta l'accesso dal server Zabbix e dal frontend Zabbix.
4. Verifica che `auto_create_index` sia abilitato:

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"

#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

Per abilitarlo, invia la seguente richiesta:

```
curl -X PUT \
 http://localhost:9200/_cluster/settings \
 -H 'content-type:application/json' \
 -d '{
   "persistent": {
      "action.auto_create_index": "true"
   }
}'

#### {"acknowledged": true, "persistent": {"action": {"auto_create_index": "true"} }, "transient": {} }
```

5. Verifica che mapping, template e ingest pipeline siano corretti inviando richieste `GET` ai rispettivi URL:

```
curl -X GET http://localhost:9200/json
curl -X GET http://localhost:9200/_template/json*
curl -X GET http://localhost:9200/_ingest/pipeline/json*
```

Puoi confrontare le risposte ricevute con le risposte attese nella documentazione API di Elasticsearch.

6. Controlla se qualche shard si trova in stato di errore; il riavvio di Elasticsearch potrebbe risolvere il problema.
7. Interroga Elasticsearch per verificare se i dati raccolti da Zabbix sono stati memorizzati, ad esempio:

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Se devi reimpostare la configurazione di Elasticsearch e ricominciare da zero, puoi eliminare tutti gli indici, i template e le ingest pipeline:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

Configurazione di ClickHouse

Zabbix può archiviare i dati della cronologia in ClickHouse come alternativa a un database relazionale.

> **Note:**
> ClickHouse può essere utilizzato anche come origine dati di telemetria per il monitoraggio delle prestazioni delle applicazioni (APM); per i dettagli, vedere *Amministrazione* > *Origine dati*.

Questa guida descrive la configurazione per le versioni supportate di ClickHouse. Se si utilizza una versione diversa, alcune funzionalità potrebbero non funzionare come previsto.

ClickHouse può archiviare i seguenti tipi di valore:

Tipo di valore dell'item	Tabella del database	Tipo ClickHouse
Numerico (senza segno)	history_uint	uint
Numerico (virgola mobile)	history	dbl
Carattere	history_str	str
Log	history_log	log
Testo	history_text	text
Binario	history_bin	non supportato da Zabbix
JSON	history_json	json

> **Note:**
> ClickHouse non accetta array JSON. Un valore JSON deve essere un singolo oggetto oppure un insieme di oggetti. Inoltre, ClickHouse gestisce le chiavi JSON con NULL allo stesso modo delle chiavi mancanti.

Note importanti

- Il housekeeper non elimina i dati da ClickHouse. Per controllare per quanto tempo vengono conservati i dati, configurare il periodo di conservazione dei dati di ClickHouse in ClickHouse.
- Zabbix **non** calcola né memorizza i trend in ClickHouse. Valutare la possibilità di estendere il periodo di conservazione della cronologia per preservare i dati meno recenti.
- Se si desidera migrare i dati della cronologia da un database Zabbix esistente (MySQL o PostgreSQL) a ClickHouse, consultare lo schema di ClickHouse e gli script di migrazione della cronologia.
- ClickHouse non è supportato come archivio della cronologia per i proxy Zabbix, ma può essere utilizzato come origine dei dati di telemetria per un proxy Zabbix con APM abilitato.
- Se si dispone di un cluster ClickHouse, consultare Configurazione dei cluster ClickHouse.

Configurazione di ClickHouse

È necessario creare e configurare un database e un utente Zabbix, quindi importare lo schema del database.

Questa guida fornisce istruzioni per installazioni di ClickHouse tramite Docker o pacchetto.

Docker

1. Creare e configurare il database e l'utente Zabbix quando si esegue il container ClickHouse:

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. Confermare che ClickHouse sia in esecuzione e che sia possibile connettersi ad esso:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"

#### 26.4.4.38
```

3. Importare lo schema del database usando lo script `history_all.sh` dalla directory di Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

> **Note:**
> Il periodo di conservazione dei dati (Time-To-Live, o TTL) per ClickHouse è, per impostazione predefinita, di 31 giorni. Per modificarlo, usare l'opzione `--ttl` durante l'importazione dello schema del database (ad esempio, `--ttl 604800` per 7 giorni), oppure configurarlo in un secondo momento.

4. Verificare che le tabelle siano state create:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Pacchetti

1. Avvia ClickHouse:

```
sudo service clickhouse-server start
```

2. Crea e configura il database e l'utente Zabbix:

```
clickhouse-client --user default

localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Importa lo schema del database usando lo script `history_all.sh` dalla tua directory Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
Il periodo di conservazione dei dati (Time-To-Live, o TTL) per ClickHouse è, per impostazione predefinita, di 31 giorni. Per modificarlo, usa l'opzione `--ttl` durante l'importazione dello schema del database (ad esempio, `--ttl 604800` per 7 giorni), oppure configuralo in un secondo momento.

4. Verifica che le tabelle siano state create:

```
clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Configurazione di Zabbix server

Nel file di configurazione di Zabbix server (`zabbix_server.conf`), impostare il parametro `HistoryProvider`.

Ad esempio, per memorizzare tutti i tipi di valori supportati in ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Dopo aver apportato le modifiche, riavviare Zabbix server:

```
systemctl restart zabbix-server
```

Configurazione del frontend di Zabbix

Nel tuo file di configurazione del frontend di Zabbix (`zabbix.conf.php`), imposta la variabile `$HISTORY_PROVIDERS` in modo che corrisponda alla configurazione del server:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

Configurazione aggiuntiva

I passaggi seguenti sono opzionali. Non sono necessari per una configurazione di base.

Configurazione del periodo di conservazione dei dati di ClickHouse

Il periodo di conservazione dei dati (Time-To-Live, o TTL) per ClickHouse è, per impostazione predefinita, di 31 giorni. Per modificarlo, esegui i comandi riportati di seguito.

Gli esempi seguenti usano Docker. Se hai installato ClickHouse usando i pacchetti, esegui le query direttamente nel client di ClickHouse.

1. Modifica la tabella (sostituisci `history_json` e 3600 con i valori necessari):

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. Applica immediatamente le modifiche:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. Verifica che il periodo di conservazione dei dati sia cambiato:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"

#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. Riavvia il server Zabbix per aggiornare il periodo di conservazione dei dati in *Administration* > *Housekeeping*:

```
systemctl restart zabbix-server
```

Configurazione dei cluster ClickHouse

Questa sezione fornisce i passaggi di configurazione per i cluster ClickHouse con replica e sharding.

I passaggi seguenti usano Docker e il ClickHouse `cluster_2S_2R` dalla documentazione di ClickHouse (con ClickHouse 26.4.4.38 e ClickHouse Keeper in esecuzione).

Prima di iniziare, è consigliabile inserire un load balancer davanti al cluster ClickHouse per Zabbix. La configurazione di server e frontend di Zabbix supporta solo un singolo URL ClickHouse, quindi può puntare a un solo nodo. Se quel nodo non è disponibile, un load balancer reindirizzerà Zabbix verso un nodo integro, tramite lo stesso URL.

1. Creare il database Zabbix:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

> **Attention:**
> La clausola `ON CLUSTER` raggiunge solo i nodi che fanno parte del cluster al momento dell'esecuzione del comando. Se in seguito aggiungi nodi al cluster, esegui di nuovo questo comando.

2. Creare e configurare l'utente Zabbix:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
```

```
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Importare lo schema del database usando lo script `history_all.sh` dalla directory di Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
Il periodo di conservazione dei dati (Time-To-Live, o TTL) per ClickHouse è, per impostazione predefinita, di 31 giorni. Per modificarlo, usa l'opzione `--ttl` durante l'importazione dello schema del database (ad esempio, `--ttl 604800` per 7 giorni), oppure configuralo in un secondo momento.

Note:
Se il tuo cluster ClickHouse richiede percorsi espliciti per le repliche (ad esempio, se la configurazione del server ClickHouse non definisce percorsi predefiniti per le tabelle replicate), esegui singolarmente gli script dello schema (`history_schema.sh`, `history_uint_schema.sh`, ecc.) e imposta un percorso diverso per ogni tabella:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

Per ulteriori dettagli sulla replica in ClickHouse, consulta Replicating data e Replicated* table engines nella documentazione di ClickHouse.

4. Configurare lo sharding per ogni tipo di tabella. Rinomina le tabelle e crea una tabella `Distributed` con il nome originale, puntando alla tabella rinominata. Questa tabella `Distributed` è quella da cui Zabbix legge e su cui scrive effettivamente; la tabella distribuisce automaticamente i dati tra gli shard. Ad esempio, per la tabella `history`:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
Per ulteriori dettagli sullo sharding in ClickHouse, consulta Distributed table engine nella documentazione di ClickHouse.

5. Configurare il server Zabbix.

6. Configurare il frontend Zabbix.

7. Verificare che i dati di Zabbix siano distribuiti tra gli shard. Il numero di righe in `zabbix.history_local` può cambiare tra controlli ripetuti, poiché il server Zabbix inserisce continuamente nuovi dati. Esegui questo comando su ogni nodo:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. Verificare che la replica sia sincronizzata (`active_replicas` deve corrispondere a `total_replicas`):

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

Risoluzione dei problemi

I seguenti passaggi possono aiutarti a risolvere i problemi con la configurazione di ClickHouse:

1. Controlla i log di ClickHouse o del server Zabbix per individuare eventuali errori.
2. Per identificare le query lente, usa l'opzione `log_slow_queries` nel parametro di configurazione del server Zabbix `HistoryProvider`.
3. Verifica che ClickHouse consenta l'accesso dal server Zabbix e dal frontend Zabbix.
4. Interroga ClickHouse per verificare se i dati raccolti da Zabbix vengono archiviati, ad esempio:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Note specifiche per la distribuzione sulla configurazione di NGINX per Zabbix

RHEL

Su RHEL 8, 9 e 10, NGINX è incluso in AppStream: non sono necessari repository aggiuntivi. È sufficiente abilitarlo e installarlo con:

```
dnf module list nginx
dnf -y install nginx
```

Su RHEL 7, NGINX può essere installato dal repository CentOS 7 Extras (se si utilizza CentOS) tramite:

```
yum -y install nginx
```

Se si preferiscono le build upstream più recenti, configurare il repository ufficiale NGINX in `/etc/yum.repos.d/nginx.repo`:

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

In SUSE Linux Enterprise Server 15 è necessario configurare `php-fpm` (il percorso del file di configurazione può variare leggermente a seconda del service pack):

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

Configurazione SAML con Microsoft Entra ID

Panoramica

Questa sezione fornisce linee guida per configurare il single sign-on e il provisioning degli utenti in Zabbix da Microsoft Entra ID (in precedenza Microsoft Azure Active Directory) utilizzando l'autenticazione SAML 2.0.

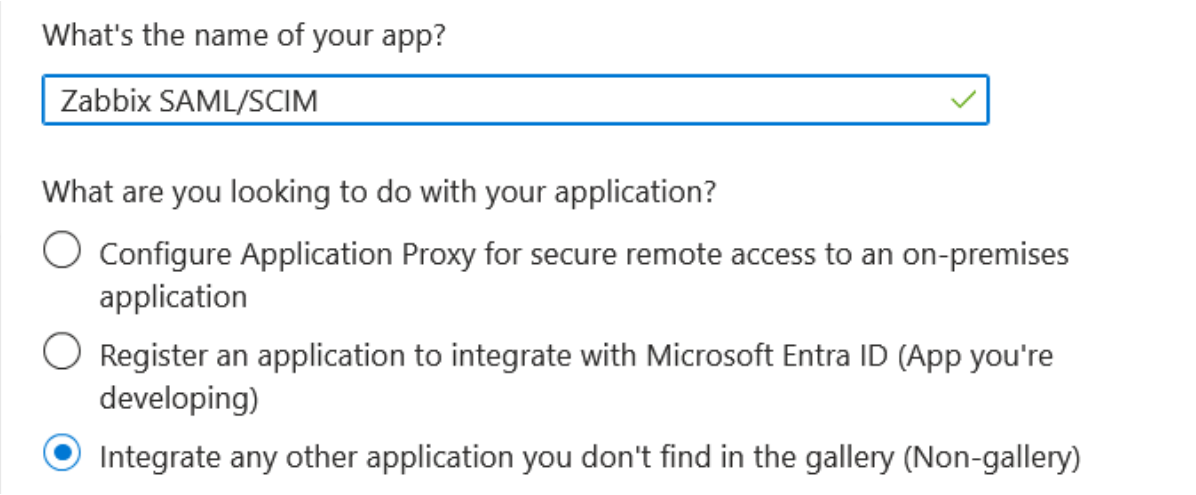
Configurazione di Microsoft Entra ID

Creazione dell'applicazione

1. Accedi a Microsoft Entra admin center su Microsoft Entra ID. Per scopi di test, puoi creare un account di prova gratuito in Microsoft Entra ID.

2. In Microsoft Entra admin center seleziona *Applications* > *Enterprise applications* > *New application* > *Create your own application*.

3. Aggiungi il nome della tua app e seleziona l'opzione *Integrate any other application...*. Dopodiché, fai clic su *Create*.

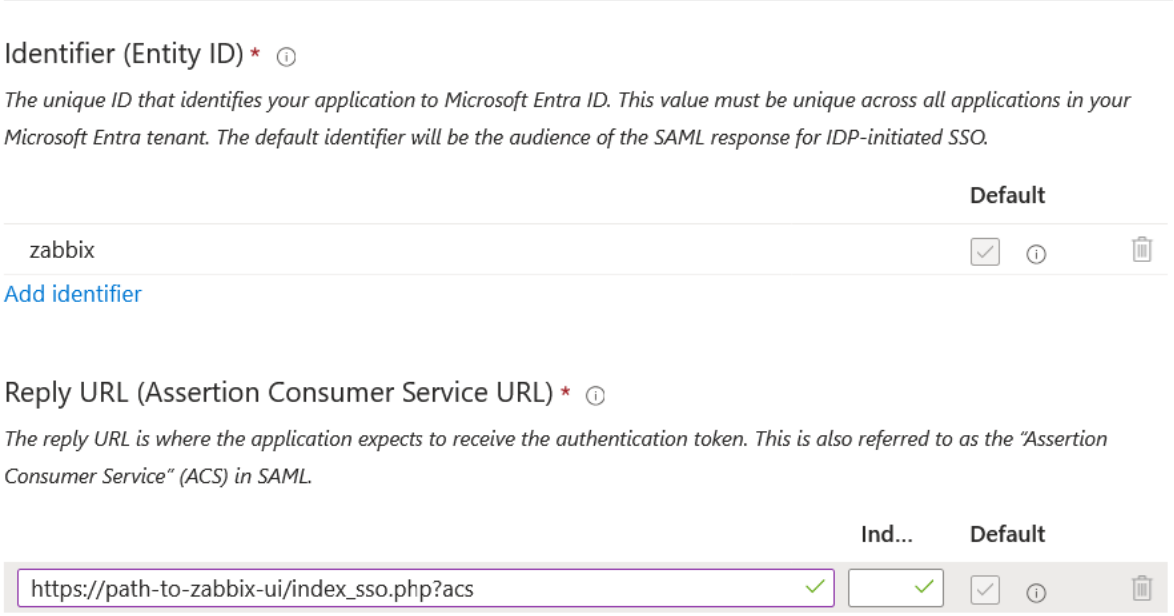


Configurazione del single sign-on

1. Nella pagina dell'applicazione, accedere a *Set up single sign on* e fare clic su *Get started*. Selezionare quindi *SAML*.

2. Modificare *Basic SAML Configuration*:

   - In *Identifier (Entity ID)*, impostare un nome univoco per identificare l'app in Microsoft Entra ID, ad esempio `zabbix`;
   - In *Reply URL (Assertion Consumer Service URL)*, impostare l'endpoint del single sign-on di Zabbix: `https://<path-to-zabbix-ui>/i`



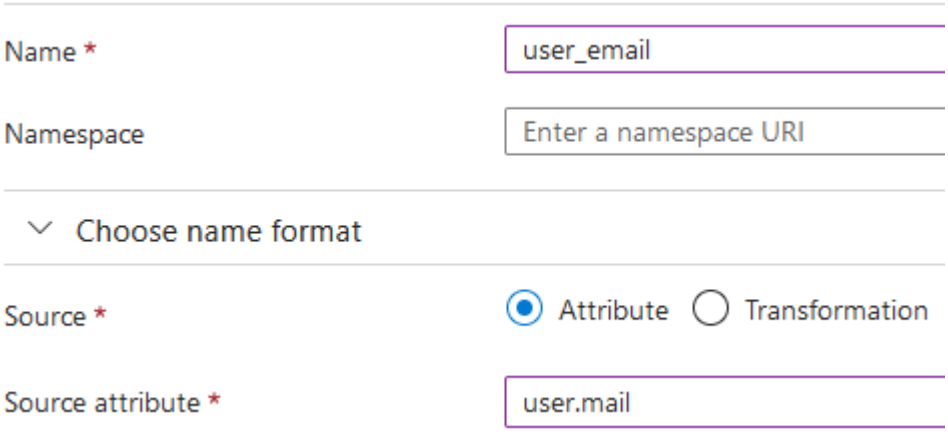
Tenere presente che `https` è obbligatorio. Per utilizzare questa configurazione con Zabbix, aggiungere la seguente riga al file di configurazione del frontend:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

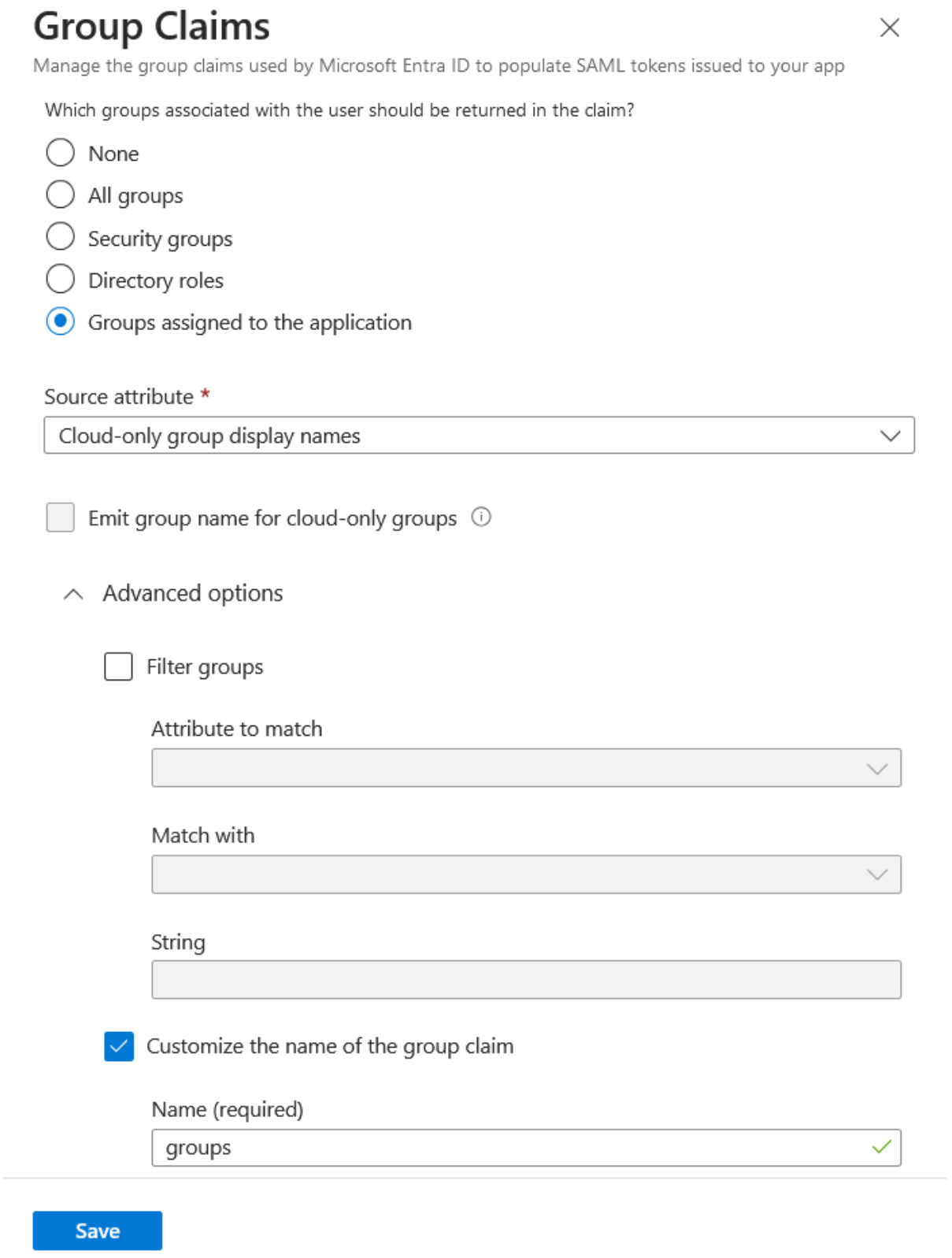
3. Modificare *Attributes & Claims*. È necessario aggiungere tutti gli attributi che si desidera passare a Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

I nomi degli attributi sono arbitrari. È possibile utilizzare nomi di attributi diversi, tuttavia è necessario che corrispondano al rispettivo valore del campo nelle impostazioni SAML di Zabbix.

- Fare clic su *Add new claim* per aggiungere un attributo:



- Fare clic su *Add a group claim* per aggiungere un attributo per passare i gruppi a Zabbix:



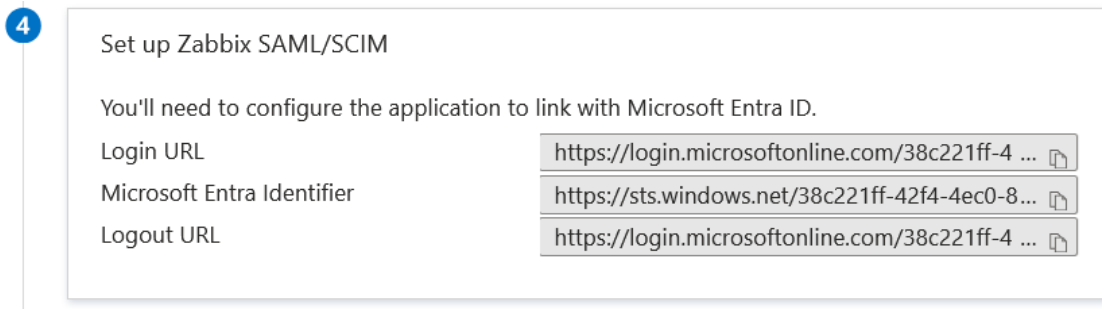
È importante che, in questo claim, i nomi dei gruppi (anziché gli ID dei gruppi) vengano passati a Zabbix tramite l'attributo *Source attribute* selezionato. In caso contrario, il provisioning JIT degli utenti non funzionerà correttamente.

4. In *SAML Certificates*, scaricare il certificato Base64 fornito da Entra ID e inserirlo nella directory `conf/certs` dell'installazione del frontend Zabbix (il percorso può variare in base al sistema operativo o alla sorgente dell'installazione di Zabbix).

Impostare i permessi 644 eseguendo:

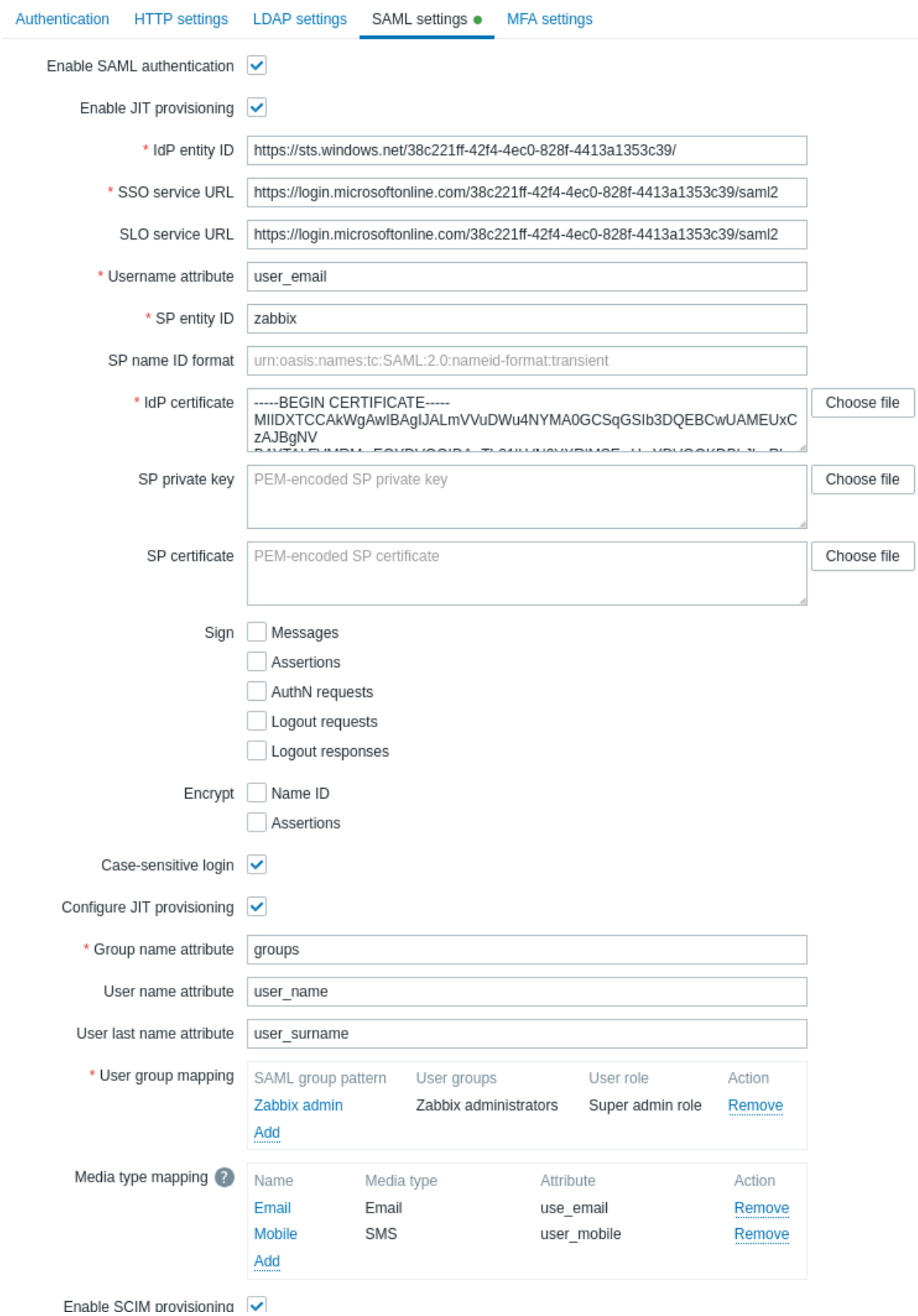
```
chmod 644 entra.cer
```

5. Utilizzare i valori di *Set up <your app name>* in Entra ID per configurare l'autenticazione SAML di Zabbix (vedere la sezione successiva):



Configurazione di Zabbix

1. In Zabbix, accedere alle impostazioni SAML e compilare le opzioni di configurazione in base alla configurazione di Entra ID:



Campo Zabbix	Campo di configurazione in Entra ID	Valore di esempio
ID entità IdP	Identificatore Microsoft Entra	
URL del servizio SSO	URL di accesso	
URL del servizio SLO	URL di disconnessione	
ID entità SP	Identificatore (ID entità)	
Attributo del nome utente	Attributo personalizzato (claim)	`user_email`
Attributo del nome del gruppo	Attributo personalizzato (claim)	`groups`
Attributo del nome	Attributo personalizzato (claim)	`user_name`

Campo Zabbix	Campo di configurazione in Entra ID	Valore di esempio
Attributo del cognome	Attributo personalizzato (claim)	`user_lastname`

2. Aggiungere il certificato Base64 fornito da Entra ID.

Se nel file di configurazione del frontend (`zabbix.conf.php`) è impostato `$SSO['CERT_STORAGE'] = 'database'`, è possibile incollare il testo del certificato o caricare il file del certificato nel frontend durante la configurazione SAML; non sono necessari file nel filesystem.

Se nel file di configurazione è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere disponibile nel filesystem (per impostazione predefinita in `ui/conf/certs` o nel percorso configurato nel file di configurazione) e il frontend non memorizzerà i certificati nel database. Se `$SSO['CERT_STORAGE']` non è impostato o è commentato, viene utilizzata la memorizzazione su file e i certificati vengono letti da `ui/conf/certs`.

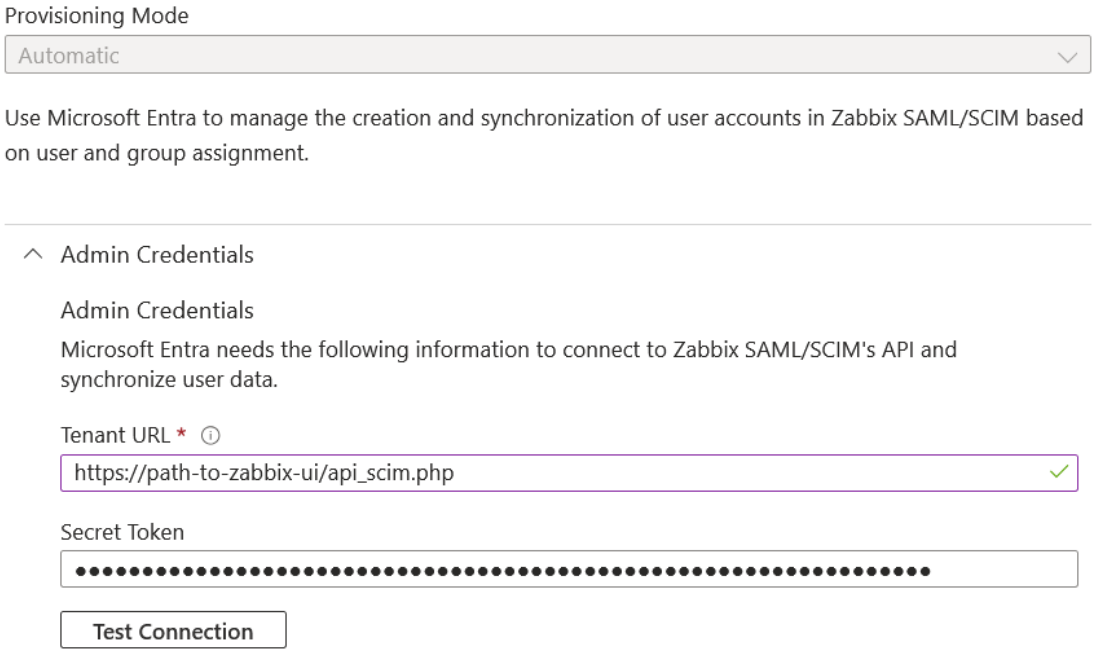
È inoltre necessario configurare la mappatura dei gruppi di utenti. La mappatura dei supporti è facoltativa.

3. Fare clic sul pulsante *Aggiorna* per salvare queste impostazioni.

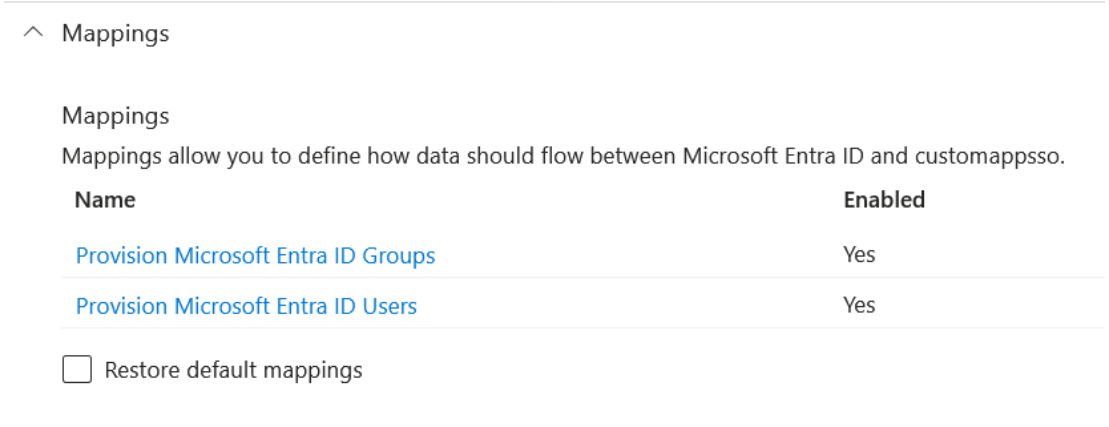
Provisioning utenti SCIM

1. Nella pagina dell'applicazione Entra ID, dal menu principale apri la pagina Provisioning. Fai clic su *Get started* e poi seleziona la modalità Automatic provisioning:

- In *Tenant URL*, imposta il seguente valore: `https://<path-to-zabbix-ui>/api_scim.php`
- In *Secret token*, inserisci un token API di Zabbix con permessi Super admin.
- Fai clic su *Test connection* per verificare se la connessione è stabilita.

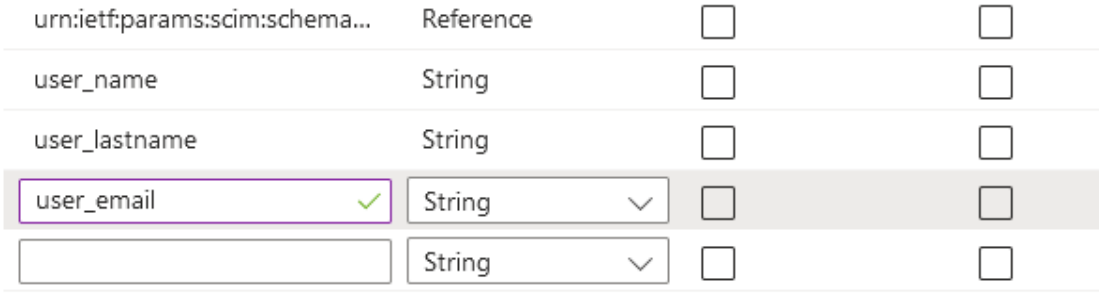


2. Ora puoi aggiungere tutti gli attributi che verranno passati con SCIM a Zabbix. Per farlo, fai clic su *Mappings* e poi su *Provision Microsoft Entra ID Users*.



In fondo all'elenco Attribute Mapping, abilita *Show advanced options*, quindi fai clic su *Edit attribute list for customappsso*.

In fondo all'elenco degli attributi, aggiungi i tuoi attributi con tipo 'String':



Salva l'elenco.

3. Ora puoi aggiungere le mappature per gli attributi aggiunti. In fondo all'elenco Attribute Mapping, fai clic su *Add New Mapping* e crea le mappature come mostrato di seguito:



Quando tutte le mappature sono state aggiunte, salva l'elenco delle mappature.



4. Come prerequisito per il provisioning degli utenti in Zabbix, devi avere utenti e gruppi configurati in Entra ID.

Per farlo, vai al *Microsoft Entra admin center* e aggiungi utenti/gruppi nelle rispettive pagine Users e Groups.

5. Quando utenti e gruppi sono stati creati in Entra ID, puoi andare al menu *Users and groups* della tua applicazione e aggiungerli all'app.

6. Vai al menu *Provisioning* della tua app e fai clic su *Start provisioning* per eseguire il provisioning degli utenti in Zabbix.

Nota che la richiesta PATCH Users in Entra ID non supporta modifiche in media.

Firma delle richieste di autenticazione

È possibile configurare Entra ID per convalidare la firma delle richieste di autenticazione firmate.

Per farlo, creare le chiavi pubblica e privata:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Assegnare le autorizzazioni:

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Aggiornare la configurazione del frontend di Zabbix aggiungendo:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Risoluzione dei problemi

Possono verificarsi problemi di autenticazione con i browser Microsoft Edge quando un utente che tenta di accedere a Zabbix tramite SAML ha già effettuato l'accesso con il profilo Microsoft Edge. Un segnale di questo problema è che l'utente potrebbe riuscire ad accedere a Zabbix utilizzando Microsoft Edge in modalità privata.

Per evitare problemi di autenticazione in questo caso, potrebbe essere necessario impostare `requestedAuthnContext` su `false` nel file di configurazione del frontend di Zabbix.

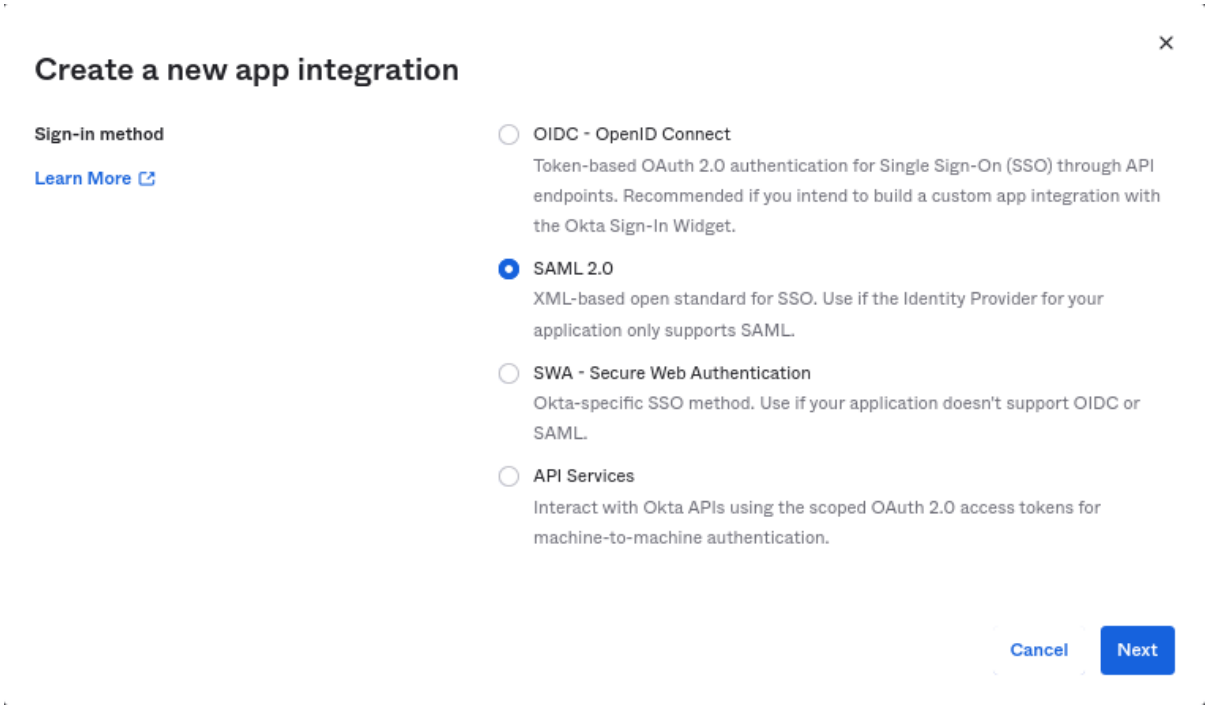
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

Configurazione SAML con Okta

Questa sezione fornisce linee guida per configurare Okta per abilitare l'autenticazione SAML 2.0 e il provisioning degli utenti per Zabbix.

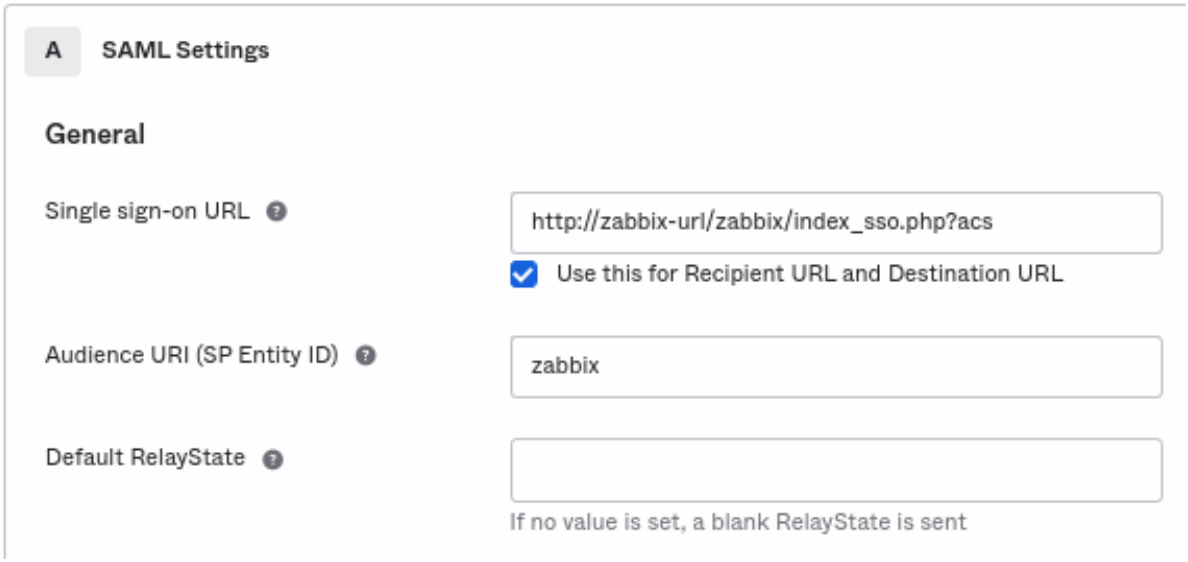
Configurazione di Okta

1. Vai su https://developer.okta.com/signup/ e registrati/accedi al tuo account.
2. Nell'interfaccia web di Okta, vai su *Applications* > *Applications*.
3. Fai clic su *Create App Integration*.

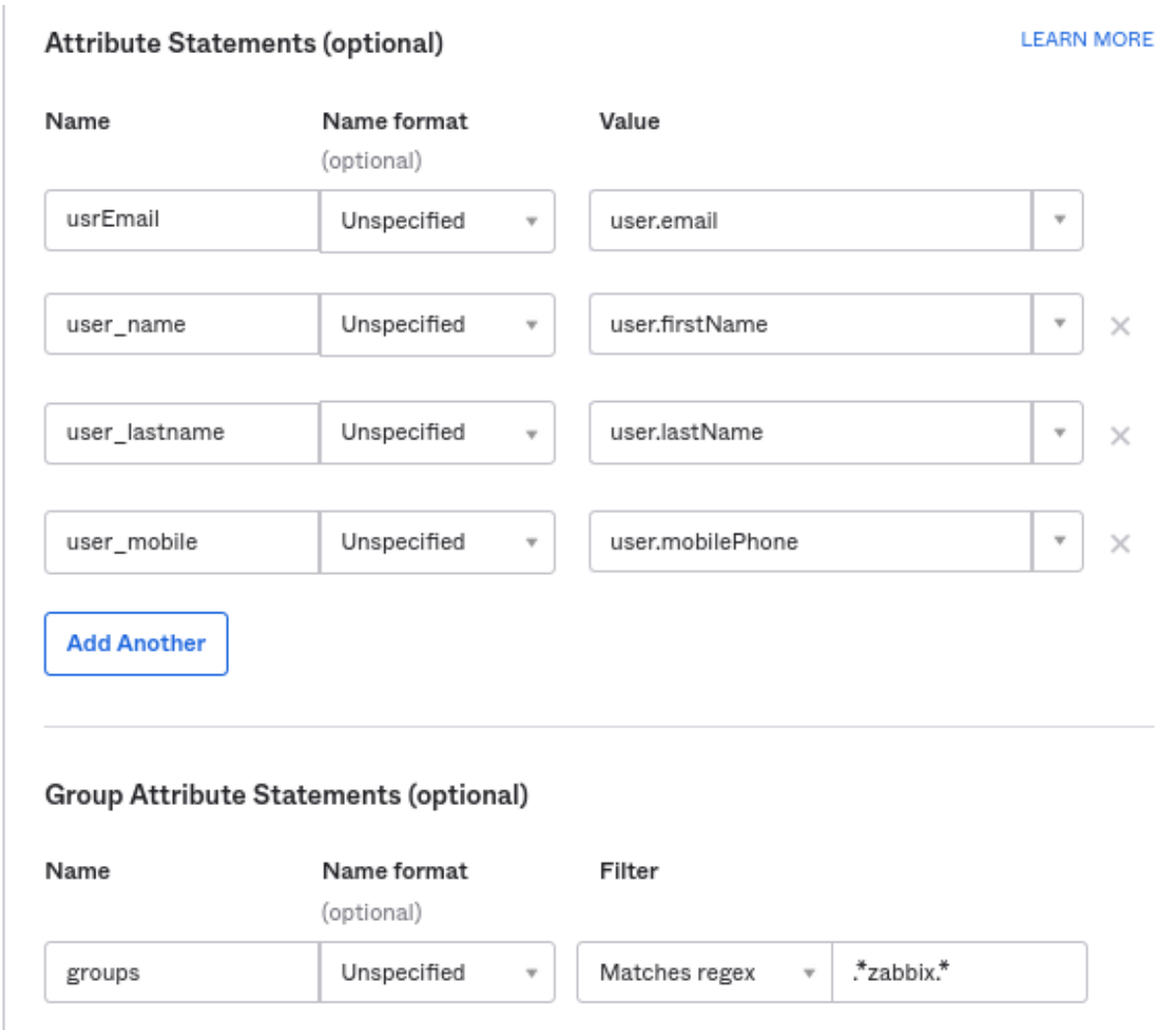


Seleziona "SAML 2.0" come metodo di accesso e fai clic su *Next*.

4. Nelle impostazioni generali, inserisci il nome dell'app e fai clic su *Next*.
5. Nella configurazione SAML, inserisci i valori indicati di seguito, quindi fai clic su *Next*.



- In **General** aggiungi:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` Nota l'uso di "http" e non di "https", in modo che il parametro `acs` non venga rimosso dalla richiesta. Deve essere selezionata anche la casella *Use this for Recipient URL and Destination URL*.
  - *Audience URI (SP Entity ID)*: `zabbix<br>` Tieni presente che questo valore verrà usato all'interno dell'asserzione SAML come identificatore univoco del service provider (se non corrisponde, l'operazione verrà rifiutata). In questo campo è possibile specificare un URL o qualsiasi stringa di dati.
  - *Default RelayState*:
 Lascia questo campo vuoto; se è necessario un reindirizzamento personalizzato, può essere aggiunto in Zabbix nelle impostazioni *Users* > *Users*.
  - Compila gli altri campi in base alle tue preferenze.
- In **Attribute Statements/Group Attribute Statements** aggiungi:



Queste dichiarazioni di attributo vengono inserite nelle asserzioni SAML condivise con Zabbix.

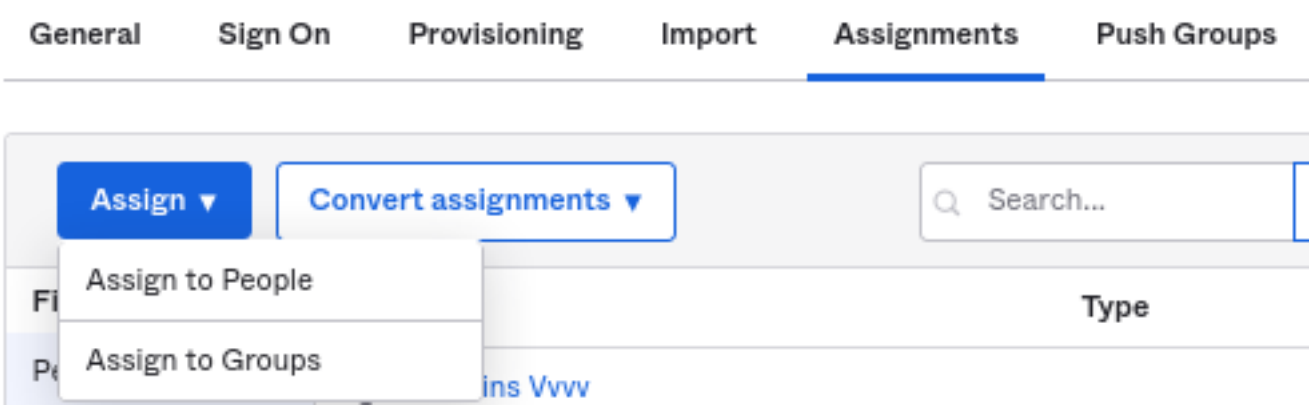
I nomi degli attributi usati qui sono esempi arbitrari. Puoi usare nomi di attributo diversi, tuttavia è necessario che corrispondano al rispettivo valore del campo nelle impostazioni SAML di Zabbix.

Se vuoi configurare l'accesso SAML a Zabbix *senza* il provisioning utenti JIT, allora è richiesto solo l'attributo email.

Note:
Se prevedi di usare una connessione crittografata, genera i certificati di crittografia privato e pubblico, quindi carica il certificato pubblico su Okta. Il modulo per il caricamento del certificato appare quando *Assertion Encryption* è impostato su "Encrypted" (fai clic su *Show Advanced Settings* per trovare questo parametro).

6. Nella scheda successiva, seleziona "I'm a software vendor. I'd like to integrate my app with Okta" e fai clic su *Finish*.

7. Vai alla scheda "Assignments" della nuova applicazione creata e fai clic sul pulsante *Assign*, quindi seleziona "Assign to People" dal menu a discesa.



8. Nella finestra popup che appare, assegna l'app alle persone che utilizzeranno SAML 2.0 per autenticarsi con Zabbix, quindi fai clic su *Save and go back*.

9. Scarica il certificato IdP. Imposta i permessi 644 eseguendo:

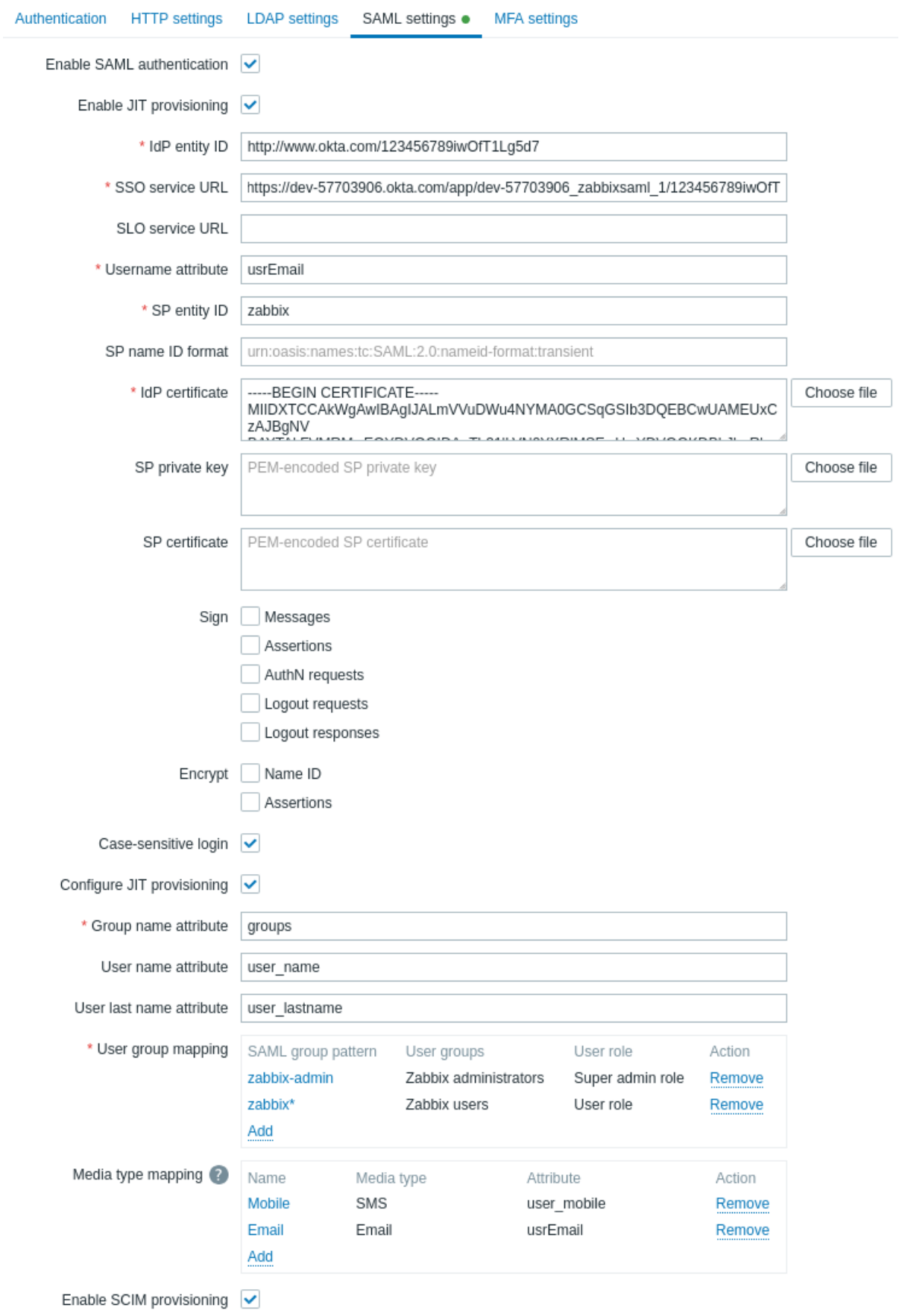
```
chmod 644 idp.crt
```

10. Vai alla scheda "Sign On" e fai clic sul pulsante *View Setup Instructions*.

Le **istruzioni** di configurazione si apriranno in una nuova scheda; lascia questa scheda aperta mentre configuri Zabbix.

Configurazione di Zabbix

1. In Zabbix, accedere alle impostazioni SAML e compilare le opzioni di configurazione in base alle istruzioni di configurazione di Okta:



Campo di Zabbix	Campo di configurazione in Okta	Valore di esempio
IdP entity ID	Identity Provider Issuer	
SSO service URL	Identity Provider Single Sign-On URL	
Username attribute	Attribute name	`usrEmail`
SP entity ID	Audience URI	`zabbix`
Group name attribute	Attribute name	`groups`

Campo di Zabbix	Campo di configurazione in Okta	Valore di esempio
User name attribute	Attribute name	`user_name`
User last name attribute	Attribute name	`user_lastname`

È inoltre necessario configurare la mappatura dei gruppi di utenti e dei media.

2. Aggiungere il certificato Base64 fornito nelle istruzioni di configurazione SAML di Okta.

Se nel file di configurazione del frontend (`zabbix.conf.php`) è impostato `$SSO['CERT_STORAGE'] = 'database'`, è possibile incollare il testo del certificato o caricare il file del certificato nel frontend durante la configurazione SAML; non sono necessari file nel filesystem.

Se nel file di configurazione è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere disponibile nel filesystem (per impostazione predefinita in `ui/conf/certs` o nel percorso configurato nel file di configurazione) e il frontend non memorizzerà i certificati nel database. Se `$SSO['CERT_STORAGE']` non è impostato o è commentato, viene utilizzata la memorizzazione su file e i certificati vengono letti da `ui/conf/certs`.

3. Se in Okta *Assertion Encryption* è impostato su "Encrypted", anche in Zabbix deve essere selezionata la casella di controllo "Assertions" del parametro *Encrypt*.

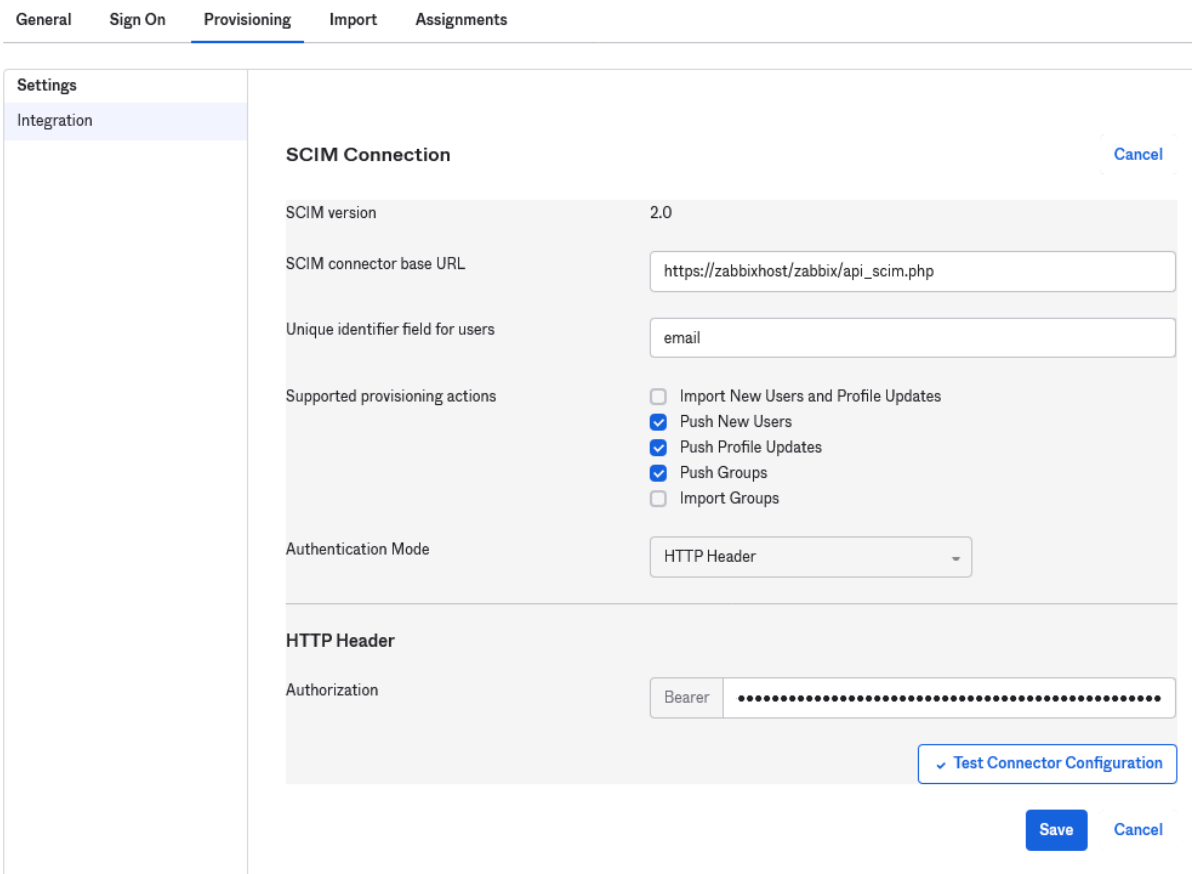
4. Fare clic sul pulsante *Update* per salvare queste impostazioni.

Provisioning SCIM

1. Per abilitare il provisioning SCIM, accedere a *General* > *App Settings* dell'applicazione in Okta.

Selezionare la casella di controllo *Enable SCIM provisioning*. Di conseguenza, viene visualizzata una nuova scheda *Provisioning*.

2. Accedere alla scheda "Provisioning" per configurare una connessione SCIM: - In *SCIM connector base URL*, specificare il percorso del frontend Zabbix e aggiungere `api_scim.php`, ovvero:
 `https://<your-zabbix-url>/zabbix/api_scim.php` - *Unique identifier field for users*: `email` - *Authentication mode*: `HTTP header` - In *Authorization*, inserire un token API valido con diritti di Super admin.



Attention:
In caso di problemi di autenticazione, consultare Authorization header forwarding.

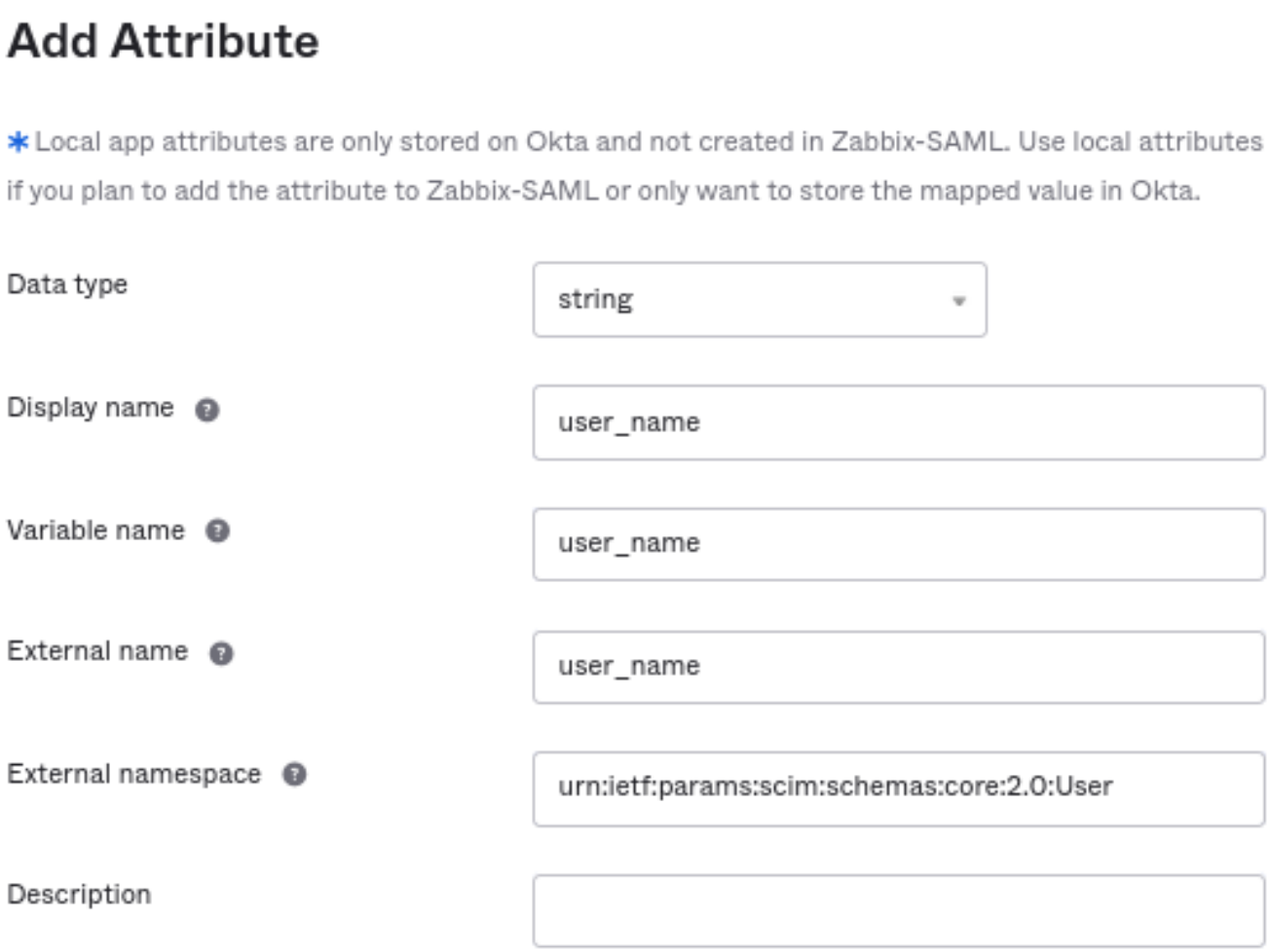
3. Fare clic su *Test Connector Configuration* per verificare la connessione. Se tutto è corretto, verrà visualizzato un messaggio di conferma.

4. In *Provisioning* > *To App*, assicurarsi di selezionare le seguenti caselle di controllo: - Create Users - Update User Attributes - Deactivate Users

In questo modo, questi tipi di richieste verranno inviati a Zabbix.

5. Assicurarsi che tutti gli attributi definiti in SAML siano definiti anche in SCIM. È possibile accedere all'editor del profilo dell'applicazione in *Provisioning* > *To App*, facendo clic su *Go to Profile Editor*.

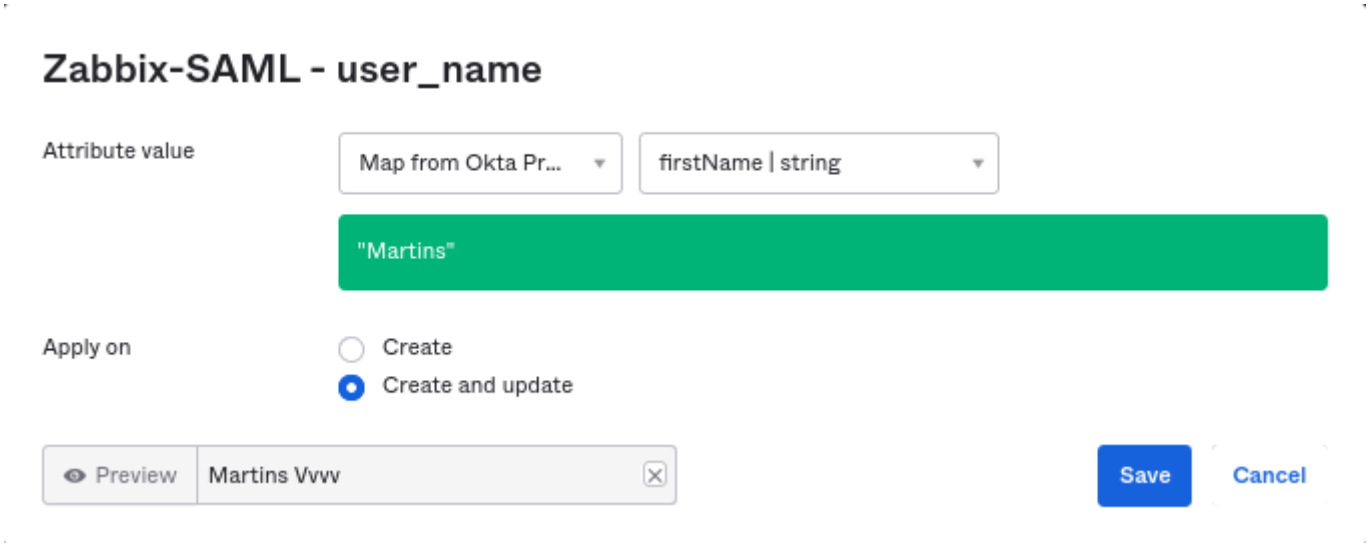
Fare clic su *Add Attribute*. Compilare i valori per *Display name*, *Variable name*, *External name* con il nome dell'attributo SAML, ad esempio `user_name`.



External namespace deve corrispondere allo schema utente: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Accedere a *Provisioning* > *To App* > *Attribute Mappings* dell'applicazione. Fare clic su *Show Unmapped Attributes* in fondo alla pagina. Vengono visualizzati gli attributi aggiunti di recente.

7. Mappare ogni attributo aggiunto.



8. Aggiungere gli utenti nella scheda ”Assignments”. Gli utenti devono essere stati precedentemente aggiunti in *Directory* > *People*. Tutte queste assegnazioni verranno inviate come richieste a Zabbix.

9. Aggiungere i gruppi nella scheda ”Push Groups”. Il modello di mappatura dei gruppi utente nelle impostazioni SAML di Zabbix deve corrispondere a un gruppo specificato qui. Se non viene trovata alcuna corrispondenza, l’utente non può essere creato in Zabbix.

Le informazioni sui membri dei gruppi vengono inviate ogni volta che viene apportata una modifica.

Configurazione SAML con OneLogin

Panoramica

Questa sezione fornisce linee guida per configurare il single sign-on e il provisioning degli utenti in Zabbix da OneLogin utilizzando l’autenticazione SAML 2.0.

Configurazione di OneLogin

Creazione dell’applicazione

1. Accedi al tuo account su OneLogin. Per scopi di test, puoi creare un account sviluppatore gratuito in OneLogin.

2. Nell’interfaccia web di OneLogin, vai su *Applications* > *Applications*.

3. Fai clic su ”Add App” e cerca l’app appropriata. Le linee guida in questa pagina si basano sull’esempio dell’app *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

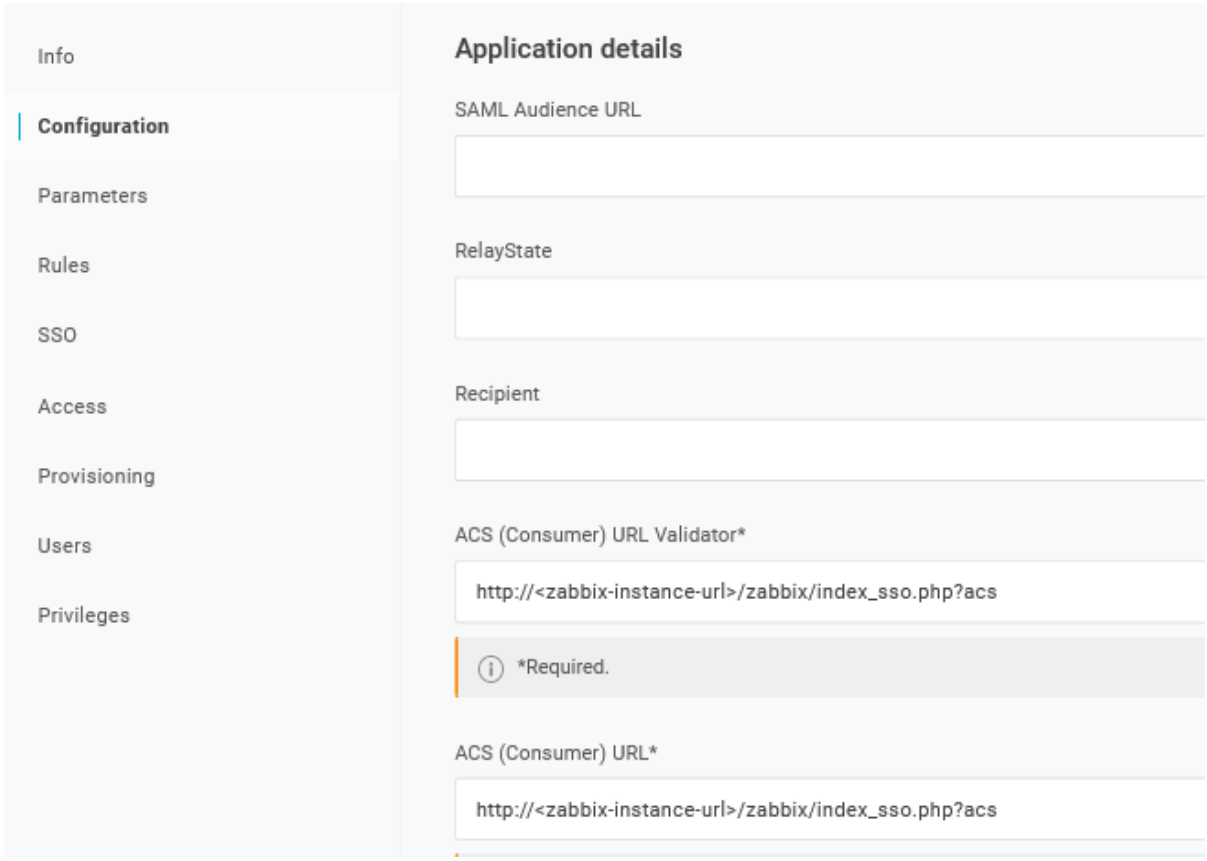
4. Per iniziare, potresti voler personalizzare il nome visualizzato della tua app. Potresti anche voler aggiungere l’icona e i dettagli dell’app. Dopodiché, fai clic su *Save*.

Configurazione del provisioning SSO/SCIM

1. In *Configuration* > *Application details*, impostare l’endpoint di single sign-on di Zabbix `http://<zabbix-instance-url>/zabbix/inde` come valore dei seguenti campi:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Notare l’uso di ”http”, e non di ”https”, affinché il parametro `acs` non venga rimosso dalla richiesta.



È anche possibile usare "https". Per utilizzarlo con Zabbix, aggiungere la seguente riga al file di configurazione del frontend (`zabbix.conf.php`):

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Lasciare le altre opzioni con i valori predefiniti.

2. In *Configuration* > *API connection*, impostare i seguenti valori:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: deve contenere tutti gli attributi personalizzati che si desidera passare a Zabbix tramite SCIM, come `user_name`, `user_lastname`, `user_email` e `user_mobile`:

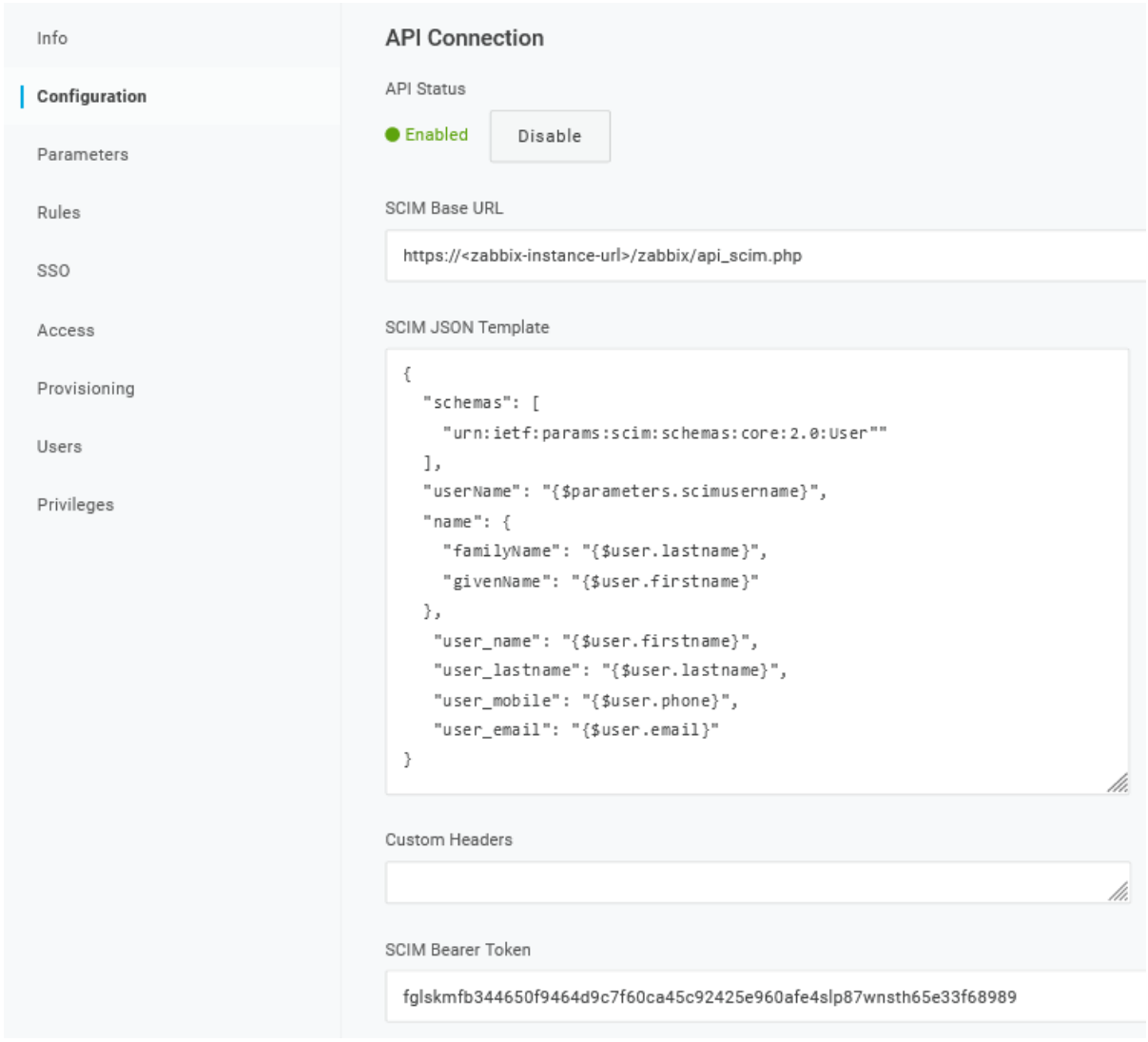
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

I nomi degli attributi sono arbitrari. È possibile usare nomi di attributi diversi, tuttavia è necessario che corrispondano al rispettivo valore del campo nelle impostazioni SAML di Zabbix.

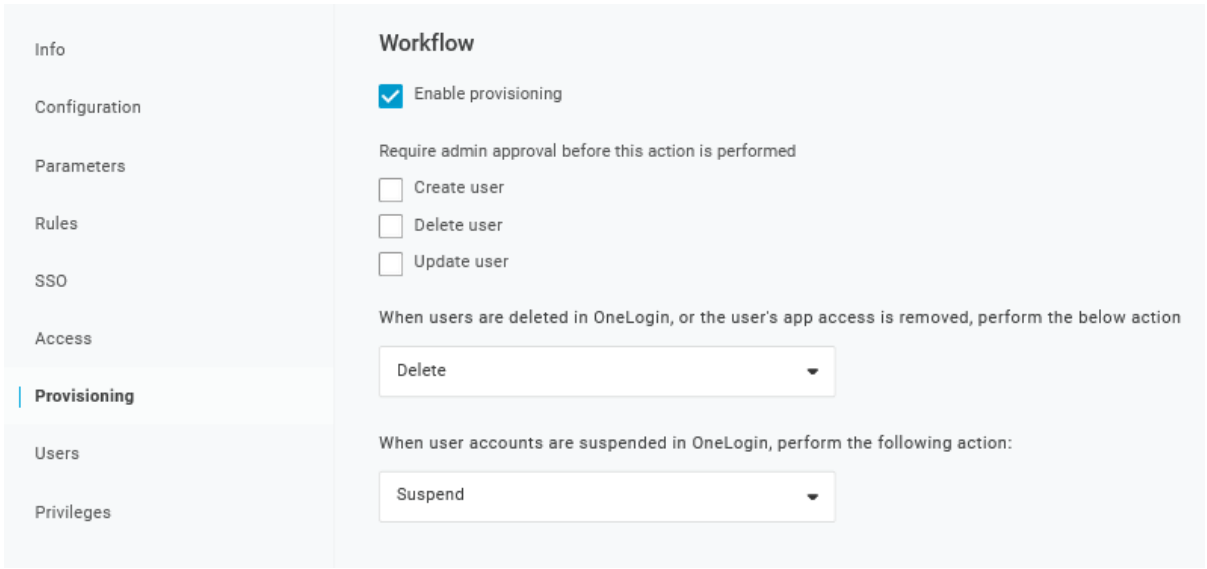
Perché il provisioning degli utenti funzioni, OneLogin deve ricevere nella risposta un attributo 'name' con 'givenName' e 'familyName', anche se non era richiesto dal service provider. Pertanto, è necessario specificarlo nello schema nella sezione di configurazione dell'applicazione.

- *SCIM Bearer Token*: inserire un token API di Zabbix con autorizzazioni di Super admin.

Fare clic su *Enable* per attivare la connessione.

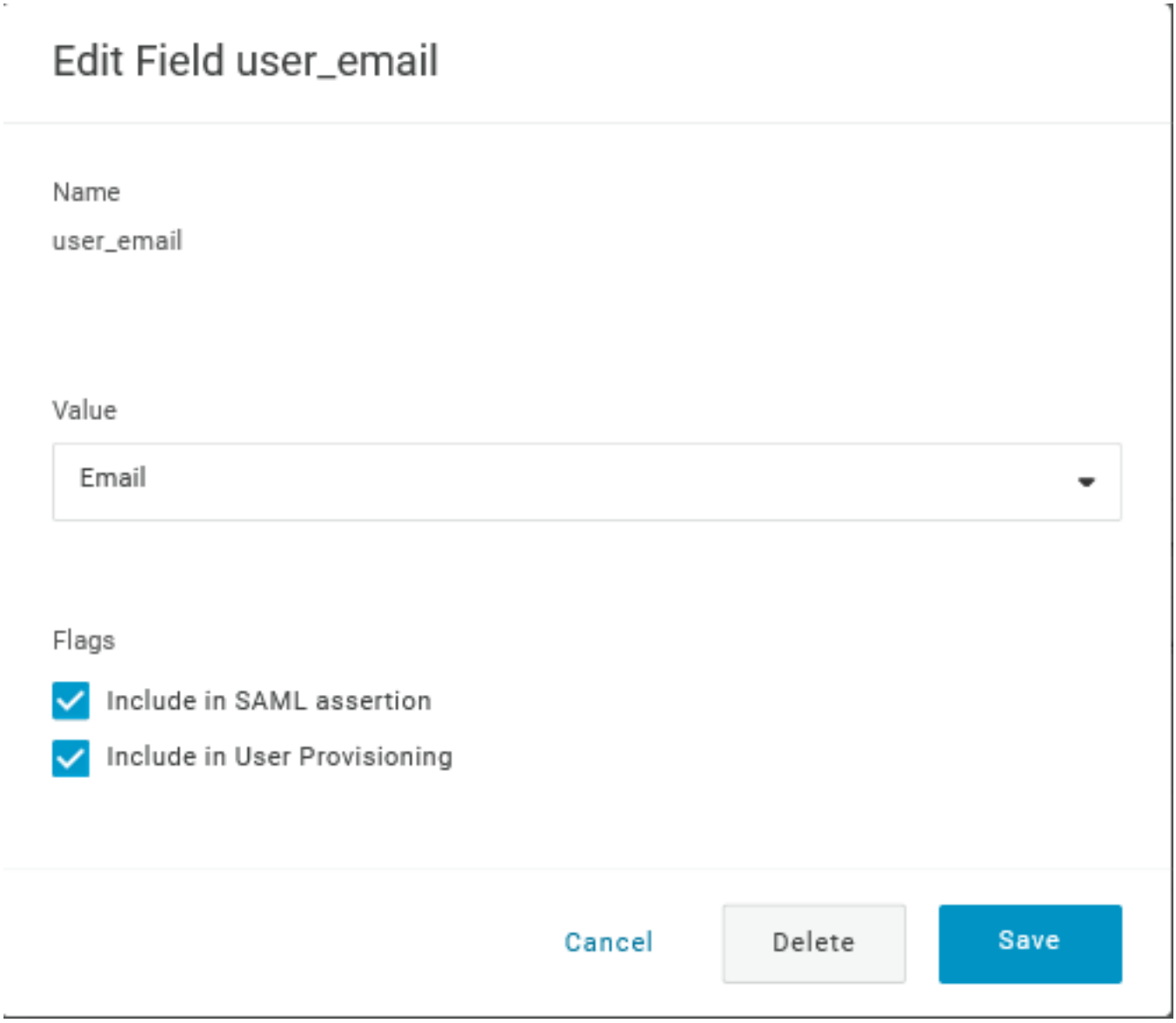


3. Nella pagina *Provisioning*, abilitare l'opzione Provisioning:



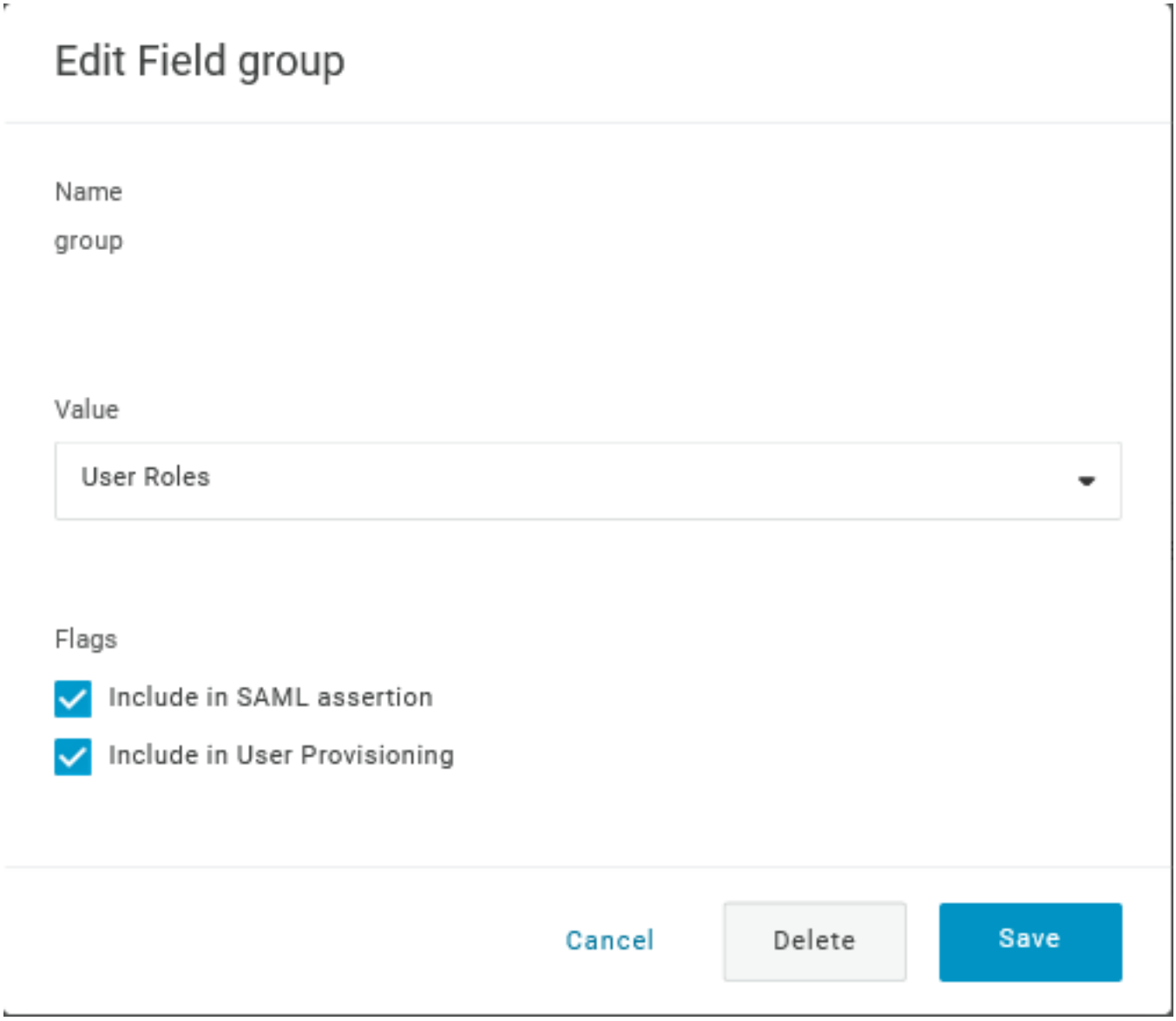
4. La pagina *Parameters* contiene un elenco di parametri predefiniti:

- Assicurarsi che 'scimusername' corrisponda al valore di accesso dell'utente in OneLogin (ad esempio, l'e-mail);
- Selezionare l'opzione *Include in User Provisioning* per il parametro 'Groups';
- Fare clic su "+" per creare i parametri personalizzati richiesti per le asserzioni SAML e il provisioning degli utenti, come `user_name`, `user_lastname`, `user_email` e `user_mobile`:

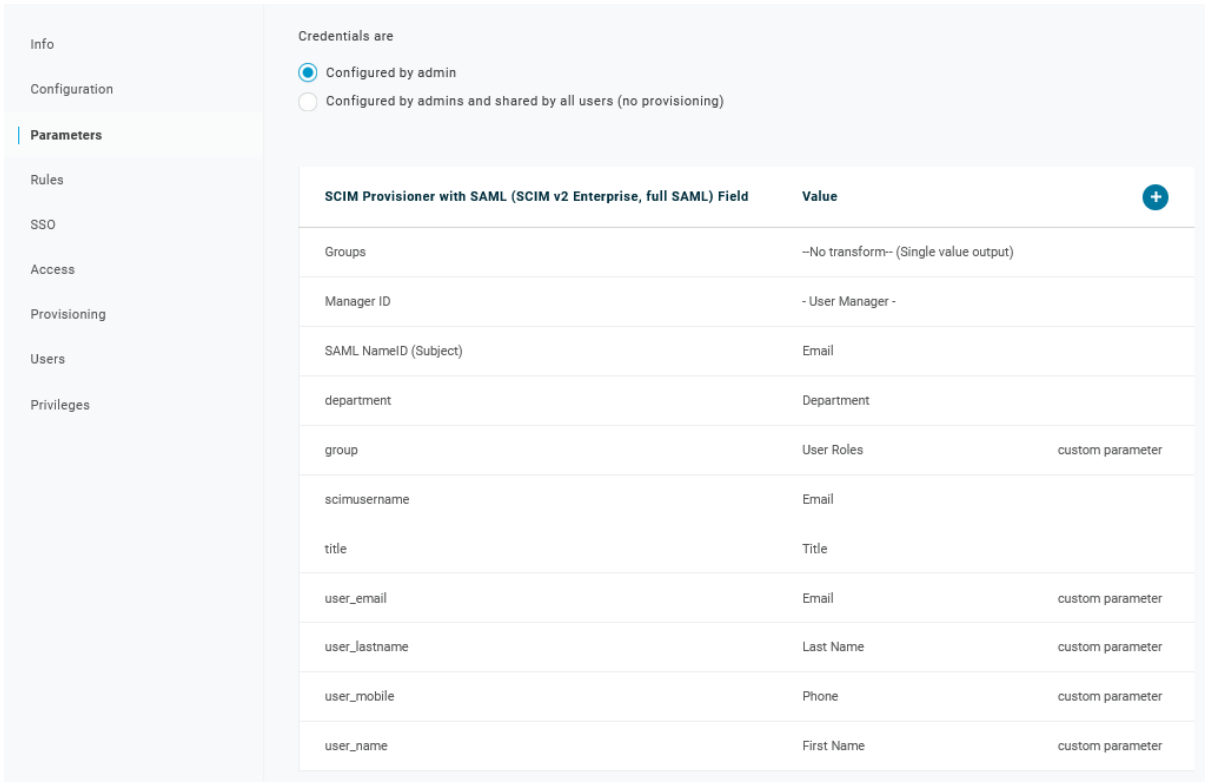


Quando si aggiunge un parametro, assicurarsi di selezionare sia l'opzione *Include in SAML assertion* sia l'opzione *Include in User Provisioning*.

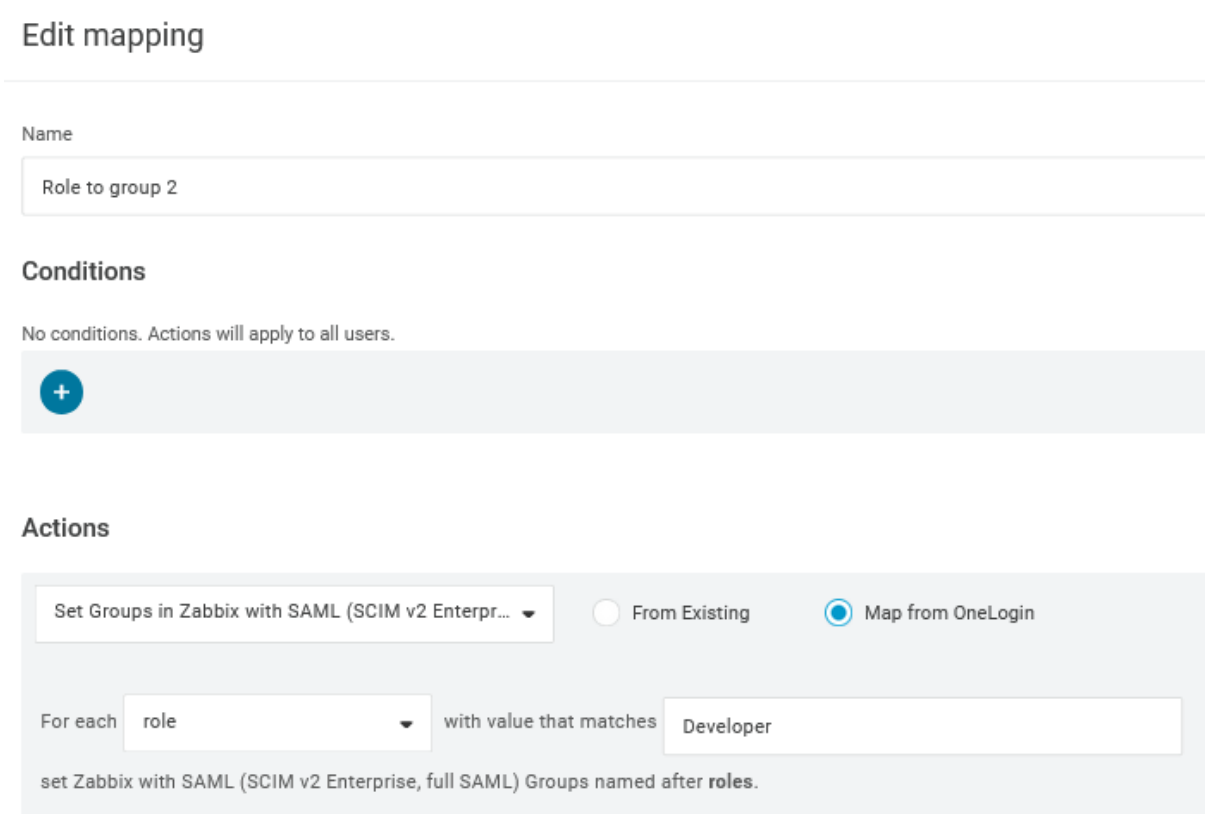
- Aggiungere un parametro 'group' che corrisponda ai ruoli degli utenti in OneLogin. I ruoli degli utenti verranno passati come stringa, separati da un punto e virgola `;`. I ruoli degli utenti OneLogin verranno utilizzati per creare i gruppi di utenti in Zabbix:



Verificare l’elenco dei parametri:



5. Nella pagina *Rules*, creare le mappature dei ruoli utente al parametro Groups predefinito.



È possibile usare un'espressione regolare per passare ruoli specifici come gruppi. I nomi dei ruoli non devono contenere `;`, poiché OneLogin lo usa come separatore quando invia un attributo con più ruoli.

6. Scaricare il certificato IdP. Impostare i permessi 644 eseguendo:

```
chmod 644 idp.crt
```

Configurazione di Zabbix

1. In Zabbix, accedere alle impostazioni SAML e compilare le opzioni di configurazione in base alla configurazione di OneLogin:



Campo Zabbix	Campo di configurazione in OneLogin	Valore di esempio
ID entità IdP	URL dell'emittente (vedere la scheda SSO dell'applicazione in OneLogin)	

Campo Zabbix	Campo di configurazione in OneLogin	Valore di esempio
URL del servizio SSO	Endpoint SAML 2.0 (HTTP) (vedere la scheda SSO dell'applicazione in OneLogin)	
URL del servizio SLO	Endpoint SLO (HTTP) (vedere la scheda SSO dell'applicazione in OneLogin)	
Attributo del nome utente	Parametro personalizzato	`user_email`
Attributo del nome del gruppo	Parametro personalizzato	`group`
Attributo del nome dell'utente	Parametro personalizzato	`user_name`
Attributo del cognome dell'utente	Parametro personalizzato	`user_lastname`

È inoltre necessario configurare la mappatura dei gruppi di utenti. La mappatura dei media è facoltativa. Fare clic su *Aggiorna* per salvare queste impostazioni.

2. Aggiungere il certificato Base64 fornito da OneLogin.

Se nel file di configurazione del frontend (`zabbix.conf.php`) è impostato `$SSO['CERT_STORAGE'] = 'database'`, è possibile incollare il testo del certificato o caricare il file del certificato nel frontend durante la configurazione SAML; non sono necessari file nel filesystem.

Se nel file di configurazione è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere disponibile nel filesystem (per impostazione predefinita in `ui/conf/certs` o nel percorso configurato nel file di configurazione) e il frontend non memorizzerà i certificati nel database. Se `$SSO['CERT_STORAGE']` non è impostato o è commentato, viene utilizzata la memorizzazione su file e i certificati vengono letti da `ui/conf/certs`.

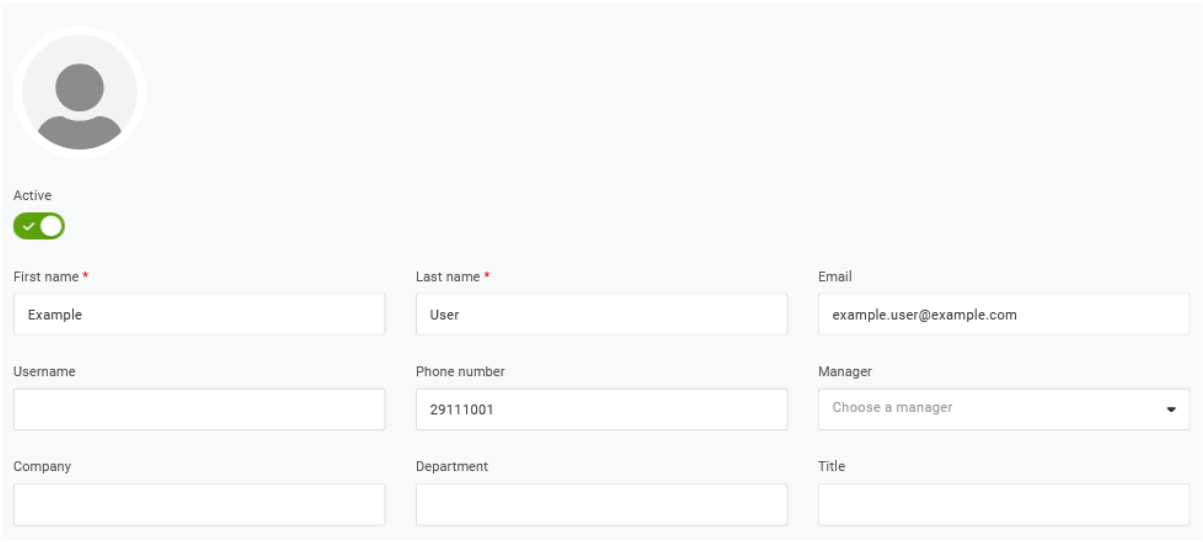
È possibile accedere al download del certificato in OneLogin selezionando *Applications* > *SSO* > facendo clic su *View details* sotto il certificato corrente.

3. Fare clic sul pulsante *Aggiorna* per salvare queste impostazioni.

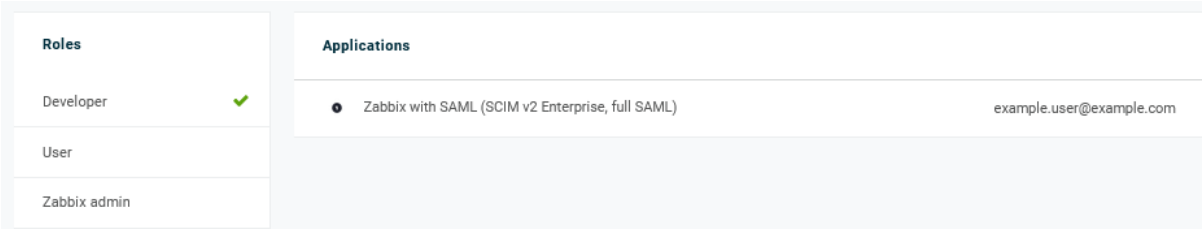
Provisioning utenti SCIM

Con il provisioning utenti abilitato, ora è possibile aggiungere/aggiornare gli utenti e i loro ruoli in OneLogin e farli provisionare immediatamente in Zabbix.

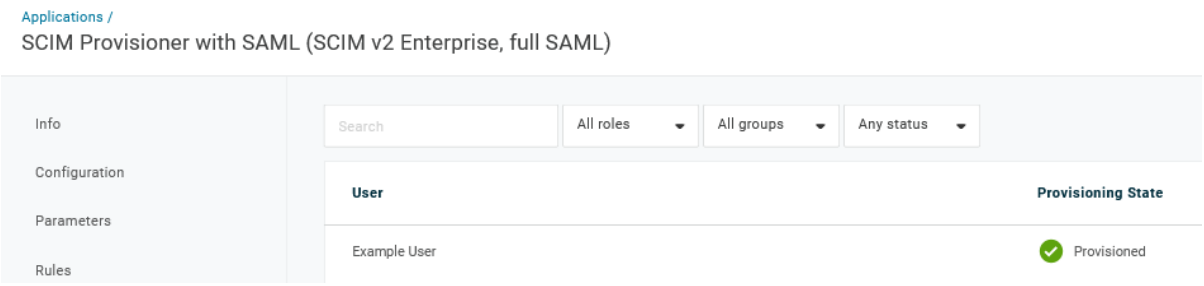
Ad esempio, è possibile creare un nuovo utente:



Aggiungerlo a un ruolo utente e all'applicazione che eseguirà il provisioning dell'utente:



Salvando l'utente, questo verrà provisionato in Zabbix. In *Application* > *Users* è possibile verificare lo stato di provisioning degli utenti correnti dell'applicazione:



Se il provisioning ha esito positivo, l'utente sarà visibile nell'elenco utenti di Zabbix.



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2023-04-18 21:11:43)	Ok	System default
example.user@example.com	Example	User	Admin role	Zabbix administrators	No	Ok	SAML

Configurazione dei report pianificati

Panoramica

Questa sezione fornisce istruzioni sull'installazione del servizio web di Zabbix e sulla configurazione di Zabbix per abilitare la generazione di report pianificati.

Installazione

Per abilitare la generazione di report pianificati, è necessario installare un nuovo processo Zabbix web service e il browser Google Chrome. Il web service può essere installato sulla stessa macchina su cui è installato il server Zabbix oppure su una macchina diversa. Il browser Google Chrome deve essere installato sulla stessa macchina su cui è installato il web service.

Il pacchetto ufficiale zabbix-web-service è disponibile nel repository di Zabbix. Il browser Google Chrome non è incluso in questi pacchetti e deve essere installato separatamente.

Per compilare il servizio web di Zabbix dai sorgenti, vedere Installing Zabbix web service.

Dopo l'installazione, eseguire `zabbix_web_service` sulla macchina in cui è installato il servizio web:

```
zabbix_web_service
```

Configurazione

Per garantire una comunicazione corretta tra tutti gli elementi coinvolti, assicurarsi che i parametri di configurazione del server e configurazione del frontend siano configurati correttamente.

Zabbix server

È necessario aggiornare i seguenti parametri nel file di configurazione di Zabbix server: `WebServiceURL` e `StartReportWriters`.

WebServiceURL

Questo parametro è necessario per abilitare la comunicazione con il servizio web. L'URL deve avere il formato `http[s]://host:port/repor`

- Per impostazione predefinita, il servizio web è in ascolto sulla porta 10053. È possibile specificare una porta diversa nel file di configurazione del servizio web.
- La specifica del percorso `/report` è obbligatoria (il percorso è codificato e non può essere modificato).

Esempio:

```
WebServiceURL=http://localhost:10053/report
```

> **Attention:**
> Si consiglia vivamente di configurare la crittografia tra Zabbix server e il servizio web di Zabbix utilizzando certificati. Per impostazione predefinita, i dati trasmessi tra Zabbix server e il servizio web di Zabbix non sono crittografati, il che può consentire accessi non autorizzati.

StartReportWriters

Questo parametro determina quanti processi di generazione dei report devono essere avviati. Se non è impostato o è uguale a 0, la generazione dei report è disabilitata. In base al numero e alla frequenza dei report richiesti, è possibile abilitare da 1 a 100 processi di generazione dei report.

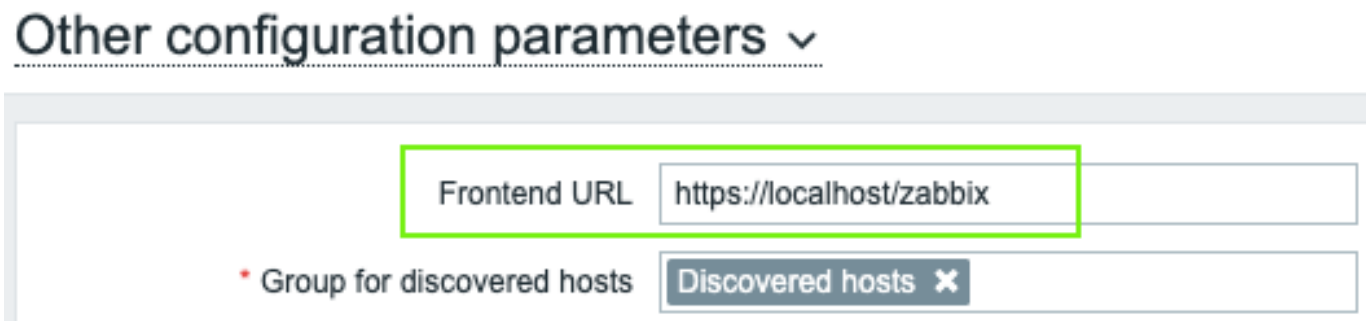
Esempio:

```
StartReportWriters=3
```

Frontend di Zabbix

Per abilitare la comunicazione tra il frontend di Zabbix e il servizio web di Zabbix, è necessario impostare il parametro *Frontend URL*:

- Andare alla sezione del menu frontend *Administration > General > Other*
- Specificare l'URL completo dell'interfaccia web di Zabbix nel parametro *Frontend URL*



Note:
Una volta completata la procedura di configurazione, è possibile configurare e inviare un report di prova per assicurarsi che tutto funzioni correttamente.

Item

Questa sezione contiene informazioni di riferimento e note tecniche sugli item, inclusi i dettagli dei parametri per specifiche chiavi di item, dettagli specifici della piattaforma e note su configurazioni speciali degli item.

Utilizza la barra laterale per trovare le informazioni pertinenti al tuo caso d'uso.

Parametri di vm.memory.size

Panoramica

Questa sezione fornisce alcuni dettagli sui parametri per l'item agent vm.memory.size[<mode>].

Parametri

Per questo item sono disponibili i seguenti parametri:

- **active** - memoria attualmente in uso o usata molto di recente, e quindi presente nella RAM
- **anon** - memoria non associata a un file (non può essere riletta da esso)
- **available** - memoria disponibile, calcolata in modo diverso a seconda della piattaforma (vedere la tabella sotto)
- **buffers** - cache per elementi come i metadati del file system
- **cached** - cache per vari elementi
- **exec** - codice eseguibile, tipicamente proveniente da un file (di programma)
- **file** - cache per il contenuto dei file a cui si è avuto accesso di recente
- **free** - memoria immediatamente disponibile per qualsiasi entità che richieda memoria
- **inactive** - memoria contrassegnata come non utilizzata
- **pavailable** - memoria 'available' come percentuale di 'total' (calcolata come `available/total*100`)
- **pinned** - uguale a 'wired'
- **pused** - memoria 'used' come percentuale di 'total' (calcolata come `used/total*100`)
- **shared** - memoria a cui possono accedere simultaneamente più processi
- **slab** - quantità totale di memoria usata dal kernel per memorizzare nella cache strutture dati per il proprio utilizzo
- **total** - memoria fisica totale disponibile
- **used** - memoria utilizzata, calcolata in modo diverso a seconda della piattaforma (vedere la tabella sotto)

- **wired** - memoria contrassegnata per rimanere sempre nella RAM. Non viene mai spostata su disco.

Warning:
Alcuni di questi parametri sono specifici della piattaforma e potrebbero non essere disponibili sulla propria piattaforma. Vedere Zabbix agent items per i dettagli.

Calcolo specifico per piattaforma di **available** e **used**:

Platform	**"available"**	**"used"**
AIX	free + cached	memoria reale in uso
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (anche retroportato alla 3.10 su RHEL 7)	/proc/meminfo, vedere "MemAvailable" nella documentazione del kernel Linux per i dettagli. Si noti che free + buffers + cached non è più uguale a 'available' perché non tutta la page cache può essere liberata e nel calcolo viene usata la low watermark.	total - free
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:
La somma di *vm.memory.size[used]* e *vm.memory.size[available]* non è necessariamente uguale al totale. Per esempio, su FreeBSD:
 * Le memorie active, inactive, wired e cached sono considerate usate, perché memorizzano alcune informazioni utili.
 * Allo stesso tempo, le memorie inactive, cached e free sono considerate disponibili, perché questi tipi di memoria possono essere assegnati immediatamente ai processi che richiedono più memoria.
Quindi la memoria inactive è contemporaneamente sia usata sia disponibile. Per questo motivo, l'item *vm.memory.size[used]* è progettato solo a scopo informativo, mentre *vm.memory.size[available]* è progettato per essere usato nei trigger.

Guarda anche

1. Ulteriori dettagli sul calcolo della memoria in diversi OS

Livello minimo di autorizzazione per gli item dell'agent Windows

Panoramica

Quando si monitorano sistemi utilizzando un agent, una buona pratica consiste nell'ottenere le metriche dal host su cui l'agent è installato. Per utilizzare il principio del privilegio minimo, è necessario determinare quali metriche vengono ottenute dall'agent.

La tabella in questo documento consente di selezionare i diritti minimi per garantire il corretto funzionamento di Zabbix agent.

Se per il funzionamento dell'agent viene selezionato un utente diverso da 'LocalSystem', allora, affinché l'agent funzioni come servizio Windows, il nuovo utente deve disporre del diritto "Accedi come servizio" in "Criteri locali→Assegnazione diritti utente" e del diritto di creare, scrivere ed eliminare il file di log di Zabbix agent. Un utente Active Directory deve essere aggiunto al gruppo *Performance Monitor Users*.

Note:
Quando si lavora con i diritti di un agent basati sul gruppo "minimo tecnicamente accettabile", è necessario concedere preventivamente i diritti agli oggetti da monitorare.

Item agent comuni supportati su Windows

Item key	Gruppo di utenti	
	Consigliato	Minimo tecnicamente accettabile (la funzionalità è limitata)

Item key	Gruppo di utenti	
agent.hostname	Guest	Guest
agent.ping	Guest	Guest
agent.variant	Guest	Guest
agent.version	Guest	Guest
log	Amministratori	Guest
log.count	Amministratori	Guest
logrt	Amministratori	Guest
logrt.count	Amministratori	Guest
net.dns	Guest	Guest
net.dns.perf	Guest	Guest
net.dns.record	Guest	Guest
net.if.discovery	Guest	Guest
net.if.in	Guest	Guest
net.if.out	Guest	Guest
net.if.total	Guest	Guest
net.tcp.listen	Guest	Guest
net.tcp.port	Guest	Guest
net.tcp.service	Guest	Guest
net.tcp.service.perf	Guest	Guest
net.udp.service	Guest	Guest
net.udp.service.perf	Guest	Guest
proc.num	Amministratori	Guest
system.cpu.discovery	Utenti Monitoraggio prestazioni	Utenti Monitoraggio prestazioni
system.cpu.load	Utenti Monitoraggio prestazioni	Utenti Monitoraggio prestazioni
system.cpu.num	Guest	Guest
system.cpu.util	Utenti Monitoraggio prestazioni	Utenti Monitoraggio prestazioni
system.hostname	Guest	Guest
system.localtime	Guest	Guest
system.run	Amministratori	Guest
system.sw.arch	Guest	Guest
system.swap.size	Guest	Guest
system.uname	Guest	Guest
system.uptime	Utenti Monitoraggio prestazioni	Utenti Monitoraggio prestazioni
vfs.dir.count	Amministratori	Guest
vfs.dir.get	Amministratori	Guest
vfs.dir.size	Amministratori	Guest
vfs.file.cksum	Amministratori	Guest
vfs.file.contents	Amministratori	Guest
vfs.file.exists	Amministratori	Guest
vfs.file.md5sum	Amministratori	Guest
vfs.file.regexp	Amministratori	Guest
vfs.file.regmatch	Amministratori	Guest
vfs.file.size	Amministratori	Guest
vfs.file.time	Amministratori	Guest
vfs.fs.discovery	Amministratori	Guest
vfs.fs.size	Amministratori	Guest
vfs.fs.get	Amministratori	Guest
vm.memory.size	Guest	Guest
web.page.get	Guest	Guest
web.page.perf	Guest	Guest
web.page.regexp	Guest	Guest
zabbix.stats	Guest	Guest

Chiavi item specifiche per Windows

Item key	Gruppo di utenti	
	Consigliato	Minimo tecnicamente accettabile (la funzionalità è limitata)
eventlog	Lettori del registro eventi	Ospiti
net.if.list	Ospiti	Ospiti

Item key	Gruppo di utenti	
perf_counter	Utenti Monitoraggio prestazioni	Utenti Monitoraggio prestazioni
proc_info	Amministratori	Ospiti
service.discovery	Ospiti	Ospiti
service.info	Ospiti	Ospiti
services	Ospiti	Ospiti
wmi.get	Amministratori	Ospiti
vm.vmemory.size	Ospiti	Ospiti

Codifica dei valori restituiti

Il server Zabbix si aspetta che ogni valore di testo restituito sia codificato in UTF8. Questo vale per qualsiasi tipo di controllo: Zabbix agent, SSH, Telnet, ecc.

Diversi sistemi/dispositivi monitorati e controlli possono restituire caratteri non ASCII nel valore. In questi casi, quasi tutte le possibili chiavi **zabbix** contengono un parametro aggiuntivo della chiave item - **<encoding>**. Questo parametro della chiave è facoltativo, ma deve essere specificato se il valore restituito non è codificato in UTF8 e contiene caratteri non ASCII. In caso contrario, il risultato può essere inatteso e imprevedibile.

Segue una descrizione del comportamento con diversi backend di database in tali casi.

MySQL

Se un valore contiene un carattere non ASCII in una codifica non UTF8, questo carattere e quelli successivi verranno scartati quando il database memorizza questo valore. Nessun messaggio di avviso verrà scritto nel file `zabbix_server.log`.
 Rilevante almeno per la versione 5.1.61 di MySQL.

PostgreSQL

Se un valore contiene un carattere non ASCII in una codifica non UTF8, ciò porterà al fallimento della query SQL (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) e i dati non verranno memorizzati. Un messaggio di avviso appropriato verrà scritto nel file `zabbix_server.log`.
 Rilevante almeno per la versione 9.1.3 di PostgreSQL.

Supporto per file di grandi dimensioni

Il supporto per file di grandi dimensioni, spesso abbreviato in LFS, è il termine usato per indicare la possibilità di lavorare con file più grandi di 2 GB sui sistemi operativi a 32 bit. Il supporto per file di grandi dimensioni influisce almeno sul monitoraggio dei file di log e su tutti gli item vfs.file.*. Il supporto per file di grandi dimensioni dipende dalle funzionalità di un sistema al momento della compilazione di Zabbix, ma è completamente disabilitato su Solaris a 32 bit a causa della sua incompatibilità con procfs e swapctl.

Sensore

Panoramica

Ogni chip sensore ha una propria directory nell'albero sysfs `/sys/devices`. Per trovare tutti i chip sensore, è più semplice seguire i collegamenti simbolici dei dispositivi da `/sys/class/hwmon/hwmon*`, dove `*` è un numero reale (0,1,2,...).

Le letture dei sensori si trovano nella directory `/sys/class/hwmon/hwmon*/` per i dispositivi virtuali, oppure nella directory `/sys/class/hwmon/hwmon*/device` per i dispositivi non virtuali. Un file denominato `name`, situato all'interno delle directory `hwmon*` o `hwmon*/device`, contiene il nome del chip, che corrisponde al nome del driver del kernel utilizzato dal chip sensore.

Esiste un solo valore di lettura del sensore per file. Lo schema comune per la denominazione dei file che contengono le letture dei sensori all'interno di una qualsiasi delle directory sopra menzionate è: `<type><number>_<item>`, dove

- **type** - per i chip sensore è "in" (tensione), "temp" (temperatura), "fan" (ventola), ecc.
- **item** - "input" (valore misurato), "max" (soglia alta), "min" (soglia bassa), ecc.
- **number** - viene sempre utilizzato per gli elementi che possono essere presenti più di una volta (di solito inizia da 1, tranne che per le tensioni che iniziano da 0). Se i file non si riferiscono a un elemento specifico, hanno un nome semplice senza numero.

Le informazioni relative ai sensori disponibili sul host possono essere ottenute utilizzando gli strumenti **sensors-detect** e **sensors** (pacchetto lm-sensors). **Sensors-detect** aiuta a determinare quali moduli sono necessari per i sensori disponibili. Quando i moduli sono caricati, il programma **sensors** può essere utilizzato per mostrare le letture di tutti i chip sensore. L'etichettatura delle letture dei sensori, utilizzata da questo programma, può essere diversa dallo schema di denominazione comune (`<type><number>_<item>`):

- se esiste un file chiamato <type><number>_label, allora l'etichetta all'interno di questo file verrà utilizzata al posto del nome <type><number><item>;
- se non esiste un file <type><number>_label, allora il programma cerca all'interno di /etc/sensors.conf (può essere anche /etc/sensors3.conf, o diverso) la sostituzione del nome.

Questa etichettatura consente all'utente di determinare quale tipo di hardware viene utilizzato. Se non esiste né il file <type><number>_label né un'etichetta all'interno del file di configurazione, il tipo di hardware può essere determinato tramite l'attributo name (hwmon*/device/name). I nomi effettivi dei sensori, accettati da zabbix_agent, possono essere ottenuti eseguendo il programma **sensors** con il parametro -u (sensors -u).

Nel programma **sensors** i sensori disponibili sono separati in base al tipo di bus (adattatore ISA, adattatore PCI, adattatore SPI, dispositivo virtuale, interfaccia ACPI, adattatore HID).

Ottenere gli identificatori dei sensori

Le etichette dei sensori, come stampate dal comando sensors, non possono sempre essere usate direttamente perché la denominazione varia tra i diversi produttori di chip. Ad esempio, l'output di sensors potrebbe contenere le seguenti righe:

```
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)
```

Tra questi, solo un nome visualizzato corrisponde direttamente a un identificatore interno:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Il tentativo di usare altre etichette visualizzate (come Vcore o +12V) non funzionerà:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Per trovare l'identificatore interno del sensore accettato da Zabbix, eseguire sensors -u. Nell'output si può osservare quanto segue:

```
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
...
```

Quindi l'etichetta Vcore corrisponde all'identificatore in1, e +12V corrisponde a in4. Secondo la specifica hwmon sysfs del kernel, queste sono tensioni sui pin del chip e, in generale, potrebbero richiedere un ridimensionamento.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix può recuperare non solo letture di tensione (in), ma anche di corrente (curr), temperatura (temp) e velocità della ventola (fan).

Compatibilità

Distribuzioni legacy

Le letture dei sensori vengono ottenute dalla directory /proc/sys/dev/sensors:

- **device** - nome del dispositivo (se viene usato <mode>, è un'espressione regolare)
- **sensor** - nome del sensore (se viene usato <mode>, è un'espressione regolare)
- **mode** - valori possibili: avg, max, min (se questo parametro viene omesso, device e sensor vengono trattati letteralmente)

Esempio di key: `sensor[w83781d-i2c-0-2d,temp1]`.

Distribuzioni correnti

Le letture dei sensori vengono ottenute dalla directory `/sys/class/hwmon`:

- **device** - nome del dispositivo (espressione non regolare). Il nome del dispositivo può essere il nome effettivo del dispositivo (ad es. `0000:00:18.3`) oppure il nome ottenuto tramite il programma sensors (ad es. `k8temp-pci-00c3`). Spetta all'utente scegliere quale nome utilizzare.
- **sensor** - nome del sensore (espressione non regolare).
- **mode** - valori possibili: `avg`, `max`, `min` (se questo parametro viene omesso, device e sensor vengono trattati alla lettera).

Esempi di chiavi:

`sensor[k8temp-pci-00c3,temp,max]` oppure `sensor[0000:00:18.3,temp1]`.

`sensor[smsc47b397-isa-0880,in,avg]` oppure `sensor[smsc47b397.2176,in1]`.

Note sul parametro memtype negli item proc.mem

Panoramica

Il parametro **memtype** è supportato sulle piattaforme Linux, AIX, FreeBSD e Solaris.

Tre valori comuni di `memtype` sono supportati su tutte queste piattaforme: `pmem`, `rss` e `vsize`. Inoltre, su alcune piattaforme sono supportati valori di `memtype` specifici della piattaforma.

AIX

Nella tabella sono indicati i valori supportati per il parametro `memtype` su AIX.

Valore supportato	Descrizione	Origine nella struttura procentry64	Tenta di essere compatibile con
`vsize` [1]	Dimensione della memoria virtuale	`pi_size`	
`pmem`	Percentuale di memoria reale	`pi_prm`	`ps -o pmem`
`rss`	Dimensione del set residente	`pi_trss + pi_drss`	`ps -o rssize`
`size`	Dimensione del processo (codice + dati)	`pi_dvm`	Colonna SIZE di `ps gvw`
`dsize`	Dimensione dei dati	`pi_dsize`	
`tsize`	Dimensione del testo (codice)	`pi_tsize`	Colonna TSIZ di `ps gvw`
`sdsize`	Dimensione dei dati dalla libreria condivisa	`pi_sdsize`	
`drss`	Dimensione del set residente dei dati	`pi_drss`	
`trss`	Dimensione del set residente del testo	`pi_trss`	

Note per AIX:

1. Quando si scelgono i parametri per la chiave dell'item `proc.mem[]` su AIX, cercare di specificare criteri di selezione dei processi restrittivi. In caso contrario, esiste il rischio che processi indesiderati vengano conteggiati nel risultato di `proc.mem[]`.

Esempio:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Questo esempio mostra che specificare solo il parametro della riga di comando (espressione regolare da confrontare) comporta il conteggio automatico dell'agent Zabbix, probabilmente non desiderato.

2. Non utilizzare `ps -ef` per visualizzare i processi: mostra solo i processi non kernel. Utilizzare `ps -Af` per visualizzare tutti i processi che saranno rilevati dall'agent Zabbix.
3. Esaminiamo l'esempio di `topasrec` per vedere come l'agent Zabbix seleziona i processi con `proc.mem[]`.

```
$ ps -Af | grep topasrec
root 10747984        1   0   Mar 16      -  0:00 /usr/bin/topasrec  -L -s 300 -R 1 -r 6 -o /var/perf daily
```

`proc.mem[]` presenta i seguenti argomenti:

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

dove: - Il primo criterio è il nome di un processo (argomento <name>). Nel nostro esempio, Zabbix agent lo vedrà come topasrec. Per trovare una corrispondenza, è necessario specificare topasrec oppure lasciarlo vuoto. - Il secondo criterio è il nome di un utente (argomento <user>). Per trovare una corrispondenza, è necessario specificare root oppure lasciarlo vuoto. - Il terzo criterio utilizzato nella selezione del processo è un argomento <cmdline>. Zabbix agent ne vedrà il valore come /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. Per trovare una corrispondenza, è necessario specificare un'espressione regolare che corrisponda a questa stringa oppure lasciarla vuota.

Gli argomenti <mode> e <memtype> vengono applicati dopo aver utilizzato i tre criteri indicati sopra.

FreeBSD

Vedere nella tabella i valori supportati per il parametro memtype su FreeBSD.

Valore supportato	Descrizione	Origine nella struttura kinfo_proc	Tenta di essere compatibile con
vsize	Dimensione della memoria virtuale	kp_eproc.e_vm.vm_map_size o ki_size	ps -o vsz
pmem	Percentuale di memoria reale	calcolata da rss	ps -o pmem
rss	Dimensione del set residente	kp_eproc.e_vm.vm_rssize o ki_rssize	ps -o rss
size [1]	Dimensione del processo (codice + dati + stack)	tsize + dsize + ssize	
tsize	Dimensione del testo (codice)	kp_eproc.e_vm.vm_tsize o ki_tsize	ps -o tsiz
dsize	Dimensione dei dati	kp_eproc.e_vm.vm_dsize o ki_dsize	ps -o dsiz
ssize	Dimensione dello stack	kp_eproc.e_vm.vm_ssize o ki_ssize	ps -o ssiz

Linux

Vedere nella tabella i valori supportati per il parametro memtype su Linux.

Valore supportato	Descrizione	Origine nel file /proc/<pid>/status
vsize [1]	Dimensione della memoria virtuale	VmSize
pmem	Percentuale di memoria reale	(VmRSS/total_memory) * 100
rss	Dimensione del set residente	VmRSS
data	Dimensione del segmento dati	VmData
exe	Dimensione del segmento di codice	VmExe
hwm	Dimensione massima del set residente	VmHWM
lck	Dimensione della memoria bloccata	VmLck
lib	Dimensione delle librerie condivise	VmLib
peak	Dimensione massima della memoria virtuale	VmPeak
pin	Dimensione delle pagine fissate	VmPin
pte	Dimensione delle voci della tabella delle pagine	VmPTE
size	Dimensione dei segmenti di codice + dati + stack del processo	VmExe + VmData + VmStk
stk	Dimensione del segmento stack	VmStk
swap	Dimensione dello spazio di swap utilizzato	VmSwap

Note per Linux:

1. Non tutti i valori di memtype sono supportati dai kernel Linux meno recenti. Ad esempio, i kernel Linux 2.4 non supportano i valori hwm, pin, peak, pte e swap.
2. Abbiamo osservato che il monitoraggio autonomo del processo di controllo attivo dell'agent Zabbix con proc.mem[...,...,...,...,
mostra un valore di 4 kB superiore a quello riportato dalla riga VmData nel file /proc/<pid>/status dell'agent. Al momento della misurazione autonoma, il segmento dati dell'agent aumenta di 4 kB e poi torna alla dimensione precedente.

Solaris

Nella tabella sono indicati i valori supportati per il parametro `memtype` su Solaris.

Valore supportato	Descrizione	Origine nella struttura psinfo	Tenta di essere compatibile con
`vsize` [1]	Dimensione dell'immagine del processo	`pr_size`	`ps -o vsz`
`pmem`	Percentuale di memoria reale	`pr_pctmem`	`ps -o pmem`
`rss`	Dimensione del resident set Potrebbe essere sottostimata (vedere la descrizione di `rss` in `man ps`).	`pr_rssize`	`ps -o rss`

Note a piè di pagina

[1] Valore predefinito.

Note sulla selezione dei processi negli item proc.mem e proc.num

Processi che modificano la propria command line

Alcuni programmi usano la modifica della propria command line come metodo per visualizzare la loro attività corrente. Un utente può vedere l'attività eseguendo i comandi `ps` e `top`. Esempi di tali programmi includono *PostgreSQL*, *Sendmail*, *Zabbix*.

Vediamo un esempio da Linux. Supponiamo di voler monitorare un numero di processi di Zabbix agent.

Il comando `ps` mostra i processi di interesse come

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/zabbix_agentd.conf
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

La selezione dei processi per nome e utente funziona:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Ora rinominiamo l'eseguibile `zabbix_agentd` in `zabbix_agentd_30` e riavviamolo.

`ps` ora mostra

```
$ 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]
...
```

Ora la selezione dei processi per nome e utente produce un risultato errato:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Perché una semplice rinomina dell'eseguibile con un nome più lungo porta a un risultato così diverso?

Zabbix agent inizia controllando il nome del processo. Viene aperto il file `/proc/<pid>/status` e viene controllata la riga `Name`.
Nel nostro caso le righe `Name` sono:

```
$ 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
```

Il nome del processo nel file `status` viene troncato a 15 caratteri.

Un risultato simile si può vedere con il comando ps:

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

Ovviamente, questo non è uguale al valore del parametro `name` `zabbix_agentd_30` del nostro `proc.num[]`. Non riuscendo a far corrispondere il nome del processo dal file `status`, Zabbix agent passa al file `/proc/<pid>/cmdline`.

Come l'agent vede il file "cmdline" può essere illustrato eseguendo il comando

```
$ 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
```

Nel nostro caso i file `/proc/<pid>/cmdline` contengono byte null invisibili e non stampabili, usati per terminare le stringhe nel linguaggio *C*. In questo esempio i byte null sono mostrati come "<NUL>".

Zabbix agent controlla "cmdline" per il processo principale e prende `zabbix_agentd_30`, che corrisponde al valore del nostro parametro `name` `zabbix_agentd_30`. Quindi il processo principale viene conteggiato dall'item `proc.num[zabbix_agentd_30,zabbix]`.

Quando controlla il processo successivo, l'agent prende `zabbix_agentd_30: collector [idle 1 sec]` dal file `cmdline` e questo non corrisponde al nostro parametro `name` `zabbix_agentd_30`. Quindi viene conteggiato solo il processo principale, che non modifica la propria command line. Gli altri processi agent modificano la propria command line e vengono ignorati.

Questo esempio mostra che il parametro `name` non può essere usato in `proc.mem[]` e `proc.num[]` per selezionare i processi in questo caso.

Note:
Per l'item `proc.get[]`, quando Zabbix agent controlla "cmdline" per il nome del processo, userà solo la parte del nome che inizia dall'ultima barra e termina al primo spazio o ai due punti. Il nome del processo ricevuto dal file cmdline verrà usato solo se il suo inizio corrisponde completamente al nome abbreviato del processo nel file `status`. L'algoritmo è lo stesso sia per il nome del processo nel filtro sia per quello nell'output JSON.

L'uso del parametro `cmdline` con un'espressione regolare appropriata produce un risultato corretto:

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Prestare attenzione quando si usano gli item `proc.get[]`, `proc.mem[]` e `proc.num[]` per monitorare programmi che modificano le proprie command line.

Prima di inserire i parametri `name` e `cmdline` negli item `proc.get[]`, `proc.mem[]` e `proc.num[]`, può essere utile testare i parametri usando l'item `proc.num[]` e il comando ps.

Thread del kernel Linux

I thread non possono essere selezionati con il parametro `cmdline` negli item `proc.get[]`, `proc.mem[]` e `proc.num[]`

Prendiamo come esempio uno dei thread del kernel:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

Può essere selezionato con il parametro name del processo:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Ma la selezione tramite il parametro cmdline del processo non funziona:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

Il motivo è che Zabbix agent prende l'espressione regolare specificata nel parametro cmdline e la applica al contenuto di /proc/<pid>/cmdline del processo. Per i thread del kernel, i file /proc/<pid>/cmdline sono vuoti. Pertanto, il parametro cmdline non trova mai corrispondenza.

Conteggio dei thread negli item proc.mem[] e proc.num[]

I thread del kernel Linux vengono conteggiati dall'item proc.num[], ma non riportano memoria nell'item proc.mem[]. Per esempio:

```
$ 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.
```

Ma cosa succede se esiste un processo utente con lo stesso nome di un thread del kernel? In tal caso potrebbe apparire così:

```
$ 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[] ha conteggiato sia il thread del kernel sia il processo utente. proc.mem[] riporta la memoria solo per il processo utente e considera la memoria del thread del kernel come se fosse 0. Questo è diverso dal caso precedente, in cui veniva riportato ZBX_NOTSUPPORTED.

Prestare attenzione quando si utilizzano gli item proc.mem[] e proc.num[] se il nome del programma coincide con quello di uno dei thread.

Prima di inserire parametri negli item proc.mem[] e proc.num[], si consiglia di testare i parametri usando l'item proc.num[] e il comando ps.

Dettagli di implementazione dei controlli net.tcp.service e net.udp.service

L'implementazione dei controlli net.tcp.service e net.udp.service è descritta in questa pagina per i vari servizi specificati nel parametro service.

Parametri dell'item net.tcp.service

ftp

Crea una connessione TCP e si aspetta che i primi 4 caratteri della risposta siano "220 ", quindi invia "QUIT\r\n". Se non specificata, viene utilizzata la porta predefinita 21.

http

Crea una connessione TCP senza aspettarsi né inviare nulla. Se non specificata, viene utilizzata la porta predefinita 80.

https

Utilizza libcurl (e funziona solo con essa), non verifica l'autenticità del certificato, non verifica il nome del host nel certificato SSL, recupera solo l'header della risposta (richiesta HEAD). Se non specificata, viene utilizzata la porta predefinita 443.

imap

Crea una connessione TCP e si aspetta che i primi 4 caratteri della risposta siano ”* OK”, quindi invia ”a1 LOGOUT\r\n”. Se non specificata, viene utilizzata la porta predefinita 143.

ldap

Apre una connessione a un server LDAP ed esegue un'operazione di ricerca LDAP con il filtro impostato su (objectClass=*). Si aspetta il recupero corretto del primo attributo della prima voce. Se non specificata, viene utilizzata la porta predefinita 389.

nntp

Crea una connessione TCP e si aspetta che i primi 3 caratteri della risposta siano ”200” o ”201”, quindi invia ”QUIT\r\n”. Se non specificata, viene utilizzata la porta predefinita 119.

pop

Crea una connessione TCP e si aspetta che i primi 3 caratteri della risposta siano ”+OK”, quindi invia ”QUIT\r\n”. Se non specificata, viene utilizzata la porta predefinita 110.

smtp

Crea una connessione TCP e si aspetta che i primi 3 caratteri della risposta siano ”220”, seguiti da uno spazio, dalla fine riga o da un trattino. Le righe contenenti un trattino appartengono a una risposta multilinea e la risposta verrà riletta finché non viene ricevuta una riga senza trattino. Quindi invia ”QUIT\r\n”. Se non specificata, viene utilizzata la porta predefinita 25.

ssh

Crea una connessione TCP. Se la connessione è stata stabilita, entrambe le parti si scambiano una stringa di identificazione (SSH-major.minor-XXXX), dove major e minor sono le versioni del protocollo e XXXX è una stringa. Zabbix verifica se viene trovata una stringa corrispondente alla specifica e quindi invia in risposta la stringa ”SSH-major.minor-zabbix_agent\r\n” oppure ”0\n” in caso di mancata corrispondenza. Se non specificata, viene utilizzata la porta predefinita 22.

tcp

Crea una connessione TCP senza aspettarsi né inviare nulla. A differenza degli altri controlli, richiede che il parametro della porta sia specificato.

telnet

Crea una connessione TCP e si aspetta un prompt di login (':' alla fine). Se non specificata, viene utilizzata la porta predefinita 23.

Parametri dell'item net.udp.service

ntp

Invia un pacchetto SNTP tramite UDP e convalida la risposta in base a RFC 4330, sezione 5. Se non specificata, viene utilizzata la porta predefinita 123.

Parametri di proc.get

Panoramica

L'item **proc.get**[<name>,<user>,<cmdline>,<mode>] è supportato su Linux, Windows, FreeBSD, OpenBSD e NetBSD.

L'elenco dei parametri di processo restituiti dall'item varia a seconda del sistema operativo e del valore dell'argomento 'mode'.

Linux

I seguenti parametri del processo vengono restituiti in Linux per ciascuna modalità:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nome del processo
ppid: PID del processo padre	ppid: PID del processo padre	processes: numero di processi
name: nome del processo	name: nome del processo	vsize: dimensione della memoria virtuale
cmdline: comando con argomenti	user: utente (reale) con cui viene eseguito il processo	pmem: percentuale della memoria reale
user: utente (reale) con cui viene eseguito il processo	group: gruppo (reale) con cui viene eseguito il processo	rss: dimensione del resident set
group: gruppo (reale) con cui viene eseguito il processo	uid: ID utente	data: dimensione del segmento dati
uid: ID utente	gid: ID del gruppo con cui viene eseguito il processo	exe: dimensione del segmento di codice

mode=process	mode=thread	mode=summary
gid: ID del gruppo con cui viene eseguito il processo	tid: ID del thread	lib: dimensione delle librerie condivise
vsize: dimensione della memoria virtuale	tname: nome del thread	lck: dimensione della memoria bloccata
pmem: percentuale della memoria reale	cputime_user: tempo CPU totale trascorso in modalità utente (il valore è espresso in tick di clock; dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)	pin: dimensione delle pagine bloccate
rss: dimensione del resident set	cputime_system: tempo CPU totale trascorso in modalità sistema (il valore è espresso in tick di clock; dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)	pte: dimensione delle voci della tabella delle pagine
data: dimensione del segmento dati	state: stato del thread	size: dimensione dei segmenti di codice + dati + stack del processo
exe: dimensione del segmento di codice	ctx_switches: numero di cambi di contesto	stk: dimensione del segmento stack
hwm: dimensione massima del resident set	page_faults: numero di page fault	swap: dimensione dello spazio di swap utilizzato
lck: dimensione della memoria bloccata		cputime_user: tempo CPU totale trascorso in modalità utente (il valore è espresso in tick di clock; dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)
lib: dimensione delle librerie condivise		cputime_system: tempo CPU totale trascorso in modalità sistema (il valore è espresso in tick di clock; dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)
peak: dimensione massima della memoria virtuale		ctx_switches: numero di cambi di contesto
pin: dimensione delle pagine bloccate		threads: numero di thread
pte: dimensione delle voci della tabella delle pagine		page_faults: numero di page fault
size: dimensione dei segmenti di codice + dati + stack del processo		pss: memoria del proportional set size
stk: dimensione del segmento stack		
swap: dimensione dello spazio di swap utilizzato		
cputime_user: tempo CPU totale trascorso in modalità utente (il valore è espresso in tick di clock; dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)		
cputime_system: tempo CPU totale trascorso in modalità sistema (il valore è espresso in tick di clock; dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)		
state: stato del processo (recuperato in modo trasparente da procfs, formato esteso)		
ctx_switches: numero di cambi di contesto		

mode=process	mode=thread	mode=summary
threads: numero di thread page_faults: numero di page fault pss: memoria del proportional set size		

OS basati su BSD

I seguenti parametri di processo vengono restituiti su FreeBSD, OpenBSD e NetBSD per ciascuna modalità:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nome del processo
ppid: PID padre	ppid: PID padre	processes: numero di processi
jid: ID della jail (solo FreeBSD)	jid: ID della jail (solo FreeBSD)	vsize: dimensione della memoria virtuale
jname: nome della jail (solo FreeBSD)	jname: nome della jail (solo FreeBSD)	pmem: percentuale della memoria reale (solo FreeBSD)
name: nome del processo	name: nome del processo	rss: dimensione del resident set
cmdline: comando con argomenti	user: utente (reale) con cui viene eseguito il processo	size: dimensione del processo (codice + dati + stack)
user: utente (reale) con cui viene eseguito il processo	group: gruppo (reale) con cui viene eseguito il processo	tsize: dimensione del testo (codice)
group: gruppo (reale) con cui viene eseguito il processo	uid: ID utente	dsize: dimensione dei dati
uid: ID utente	gid: ID del gruppo con cui viene eseguito il processo	ssize: dimensione dello stack
gid: ID del gruppo con cui viene eseguito il processo	tid: ID del thread	cputime_user: tempo totale di CPU trascorso in modalità utente (riportato in secondi)
vsize: dimensione della memoria virtuale	tname: nome del thread	cputime_system: tempo totale di CPU trascorso in modalità sistema (riportato in secondi)
pmem: percentuale della memoria reale (solo FreeBSD)	cputime_user: tempo totale di CPU trascorso in modalità utente (riportato in secondi)	ctx_switches: numero di cambi di contesto
rss: dimensione del resident set	cputime_system: tempo totale di CPU trascorso in modalità sistema (riportato in secondi)	threads: numero di thread (non supportato per NetBSD)
size: dimensione del processo (codice + dati + stack)	state: stato del thread	stk: dimensione del segmento di stack
tsize: dimensione del testo (codice)	ctx_switches: numero di cambi di contesto	page_faults: numero di page fault
dsize: dimensione dei dati	io_read_op: numero di volte in cui il sistema ha dovuto eseguire input	fds: numero di descrittori di file (solo OpenBSD)
ssize: dimensione dello stack	io_write_op: numero di volte in cui il sistema ha dovuto eseguire output	swap: dimensione dello spazio di swap utilizzato
cputime_user: tempo totale di CPU trascorso in modalità utente (riportato in secondi)		io_read_op: numero di volte in cui il sistema ha dovuto eseguire input
cputime_system: tempo totale di CPU trascorso in modalità sistema (riportato in secondi)		io_write_op: numero di volte in cui il sistema ha dovuto eseguire output
state: stato del processo (attesa disco/in esecuzione/in sospensione/arresto per tracing/zombie/altro)		
ctx_switches: numero di cambi di contesto		
threads: numero di thread (non supportato per NetBSD)		
page_faults: numero di page fault		

mode=process	mode=thread	mode=summary
fds: numero di descrittori di file (solo OpenBSD) swap: dimensione dello spazio di swap utilizzato io_read_op: numero di volte in cui il sistema ha dovuto eseguire input io_write_op: numero di volte in cui il sistema ha dovuto eseguire output		

Windows

Su Windows, per ciascuna modalità vengono restituiti i seguenti parametri di processo:

mode=process	mode=thread	mode=summary
pid: PID ppid: PID padre name: nome del processo user: utente con cui viene eseguito il processo sid: SID dell'utente vmsize: dimensione della memoria virtuale wkset: dimensione del working set del processo cputime_user: tempo totale di CPU trascorso in modalità utente (riportato in millisecondi) cputime_system: tempo totale di CPU trascorso in modalità sistema (riportato in millisecondi) threads: numero di thread page_faults: numero di page fault handles: numero di handle io_read_b: byte IO letti io_write_b: byte IO scritti io_read_op: operazioni di lettura IO io_write_op: operazioni di scrittura IO io_other_b: byte IO trasferiti, diversi dalle operazioni di lettura e scrittura io_other_op: operazioni IO diverse dalle operazioni di lettura e scrittura	pid: PID ppid: PID padre name: nome del processo user: utente con cui viene eseguito il processo sid: SID dell'utente tid: ID del thread	name: nome del processo processes: numero di processi vmsize: dimensione della memoria virtuale wkset: dimensione del working set del processo cputime_user: tempo totale di CPU trascorso in modalità utente (riportato in millisecondi) cputime_system: tempo totale di CPU trascorso in modalità sistema (riportato in millisecondi) threads: numero di thread page_faults: numero di page fault handles: numero di handle io_read_b: byte IO letti io_write_b: byte IO scritti io_read_op: operazioni di lettura IO io_write_op: operazioni di scrittura IO io_other_b: byte IO trasferiti, diversi dalle operazioni di lettura e scrittura io_other_op: operazioni IO diverse dalle operazioni di lettura e scrittura

Impostazioni dell'interfaccia host irraggiungibile/non disponibile

Panoramica

Diversi parametri di configurazione definiscono come il server Zabbix deve comportarsi quando un controllo agent (Zabbix, SNMP, IPMI, JMX) fallisce e un'interfaccia host diventa irraggiungibile.

Interfaccia non raggiungibile

Un'interfaccia di host viene considerata non raggiungibile dopo un controllo fallito (errore di rete, timeout) da parte di Zabbix, SNMP, IPMI o degli agent. A partire da Zabbix 6.2.0, anche i controlli attivi di Zabbix agent influiscono sulla disponibilità dell'interfaccia. Quando i controlli attivi diventano non disponibili, contribuiscono allo stato complessivo di disponibilità dell'interfaccia dell'agent.

Dal momento in cui un'interfaccia diventa non raggiungibile, **UnreachableDelay** definisce con quale frequenza viene ricontrollata usando uno degli item (incluse le regole LLD). Questi ricontrolli vengono eseguiti dagli unreachable pollers (o dagli IPMI pollers per i controlli IPMI). Per impostazione predefinita, l'intervallo tra controlli consecutivi di raggiungibilità è di 15 secondi.

Attention:
I controlli eseguiti dagli asynchronous pollers non vengono spostati sugli unreachable pollers.

Nel log del server Zabbix la non raggiungibilità è indicata da messaggi come:

```
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
```

I messaggi di log specificano l'esatto item che è fallito e il suo tipo (Zabbix agent).

Note:
Il parametro *Timeout* influisce anche su quanto presto un'interfaccia viene ricontrollata durante la non raggiungibilità. Se Timeout è impostato a 20 secondi e UnreachableDelay a 30 secondi, il controllo successivo avverrà dopo 50 secondi dal primo tentativo.

Il parametro **UnreachablePeriod** definisce la durata totale del periodo di non raggiungibilità. Per impostazione predefinita, UnreachablePeriod è 45 secondi. Questo valore dovrebbe essere diverse volte maggiore di UnreachableDelay per garantire che un'interfaccia venga ricontrollata più volte prima di essere contrassegnata come non disponibile.

Un item interno, zabbix[host,active_agent,available], consente di monitorare la disponibilità dei controlli attivi in scenari di non raggiungibilità.

Ripristino dell'interfaccia allo stato disponibile

Quando il periodo di irraggiungibilità termina, l'interfaccia viene nuovamente interrogata, riducendo la priorità per l'item che ha portato l'interfaccia nello stato di irraggiungibilità. Se l'interfaccia irraggiungibile ricompare, il monitoraggio torna automaticamente alla normalità:

```
ripresa dei controlli Zabbix agent sull'host "New host": connessione ripristinata
```

Note:
Una volta che l'interfaccia torna disponibile, l'host non interroga immediatamente tutti i suoi item per due motivi:

- Potrebbe sovraccaricare l'host.
- Il momento di ripristino dell'interfaccia non coincide sempre con il tempo pianificato per l'interrogazione degli item.

Quindi, dopo che l'interfaccia torna disponibile, gli item non vengono interrogati immediatamente, ma vengono riprogrammati per il successivo ciclo di interrogazione.

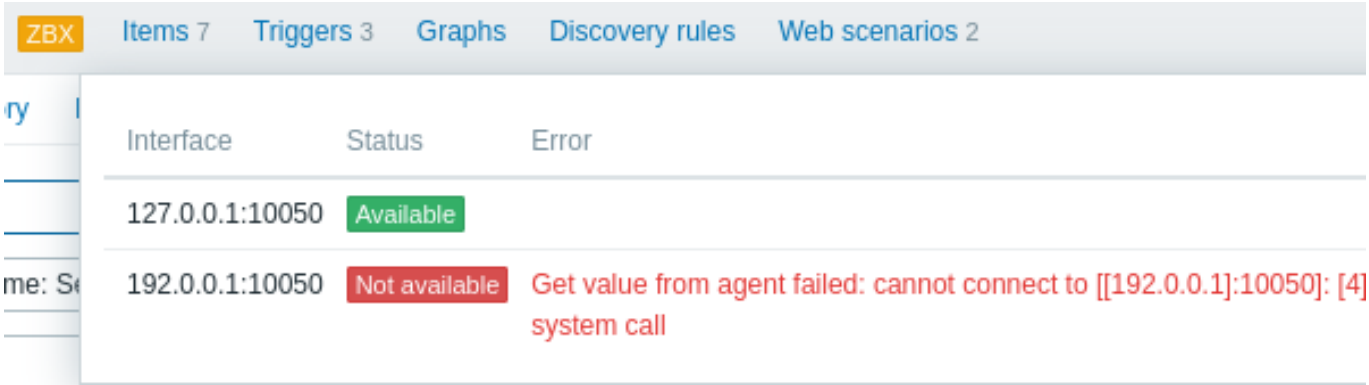
Interfaccia non disponibile

Dopo la fine di UnreachablePeriod, se l'interfaccia non è ricomparsa, l'interfaccia viene considerata non disponibile.

Nel log del server ciò è indicato da messaggi come questi:

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

e nel frontend l'icona di disponibilità del host passa da verde/grigio a giallo/rosso (i dettagli dell'interfaccia non disponibile possono essere visualizzati nel riquadro di suggerimento che viene mostrato quando il puntatore del mouse viene posizionato sull'icona di disponibilità del host):



Il parametro **UnavailableDelay** definisce con quale frequenza un'interfaccia viene controllata durante il periodo di non disponibilità dell'interfaccia.

Per impostazione predefinita è di 60 secondi (quindi in questo caso ”temporarily disabling”, dal messaggio di log sopra, significherà disabilitare i controlli per un minuto).

Quando la connessione all'interfaccia viene ripristinata, anche il monitoraggio torna automaticamente alla normalità:

```
enabling Zabbix agent checks on host "New host": interface became available
```

Monitoraggio remoto delle statistiche di Zabbix

Panoramica

È possibile rendere accessibili da remoto alcune metriche interne di Zabbix server e proxy tramite un'altra istanza di Zabbix o uno strumento di terze parti. Questo può essere utile affinché i fornitori di supporto/servizi possano monitorare da remoto i server/proxy Zabbix dei propri clienti oppure, nelle organizzazioni in cui Zabbix non è lo strumento di monitoraggio principale, affinché le metriche interne di Zabbix possano essere monitorate da un sistema di terze parti in una configurazione di monitoraggio centralizzato.

Le statistiche interne di Zabbix sono esposte a un insieme configurabile di indirizzi elencati nel nuovo parametro 'StatsAllowedIP' server/proxy. Le richieste saranno accettate solo da questi indirizzi.

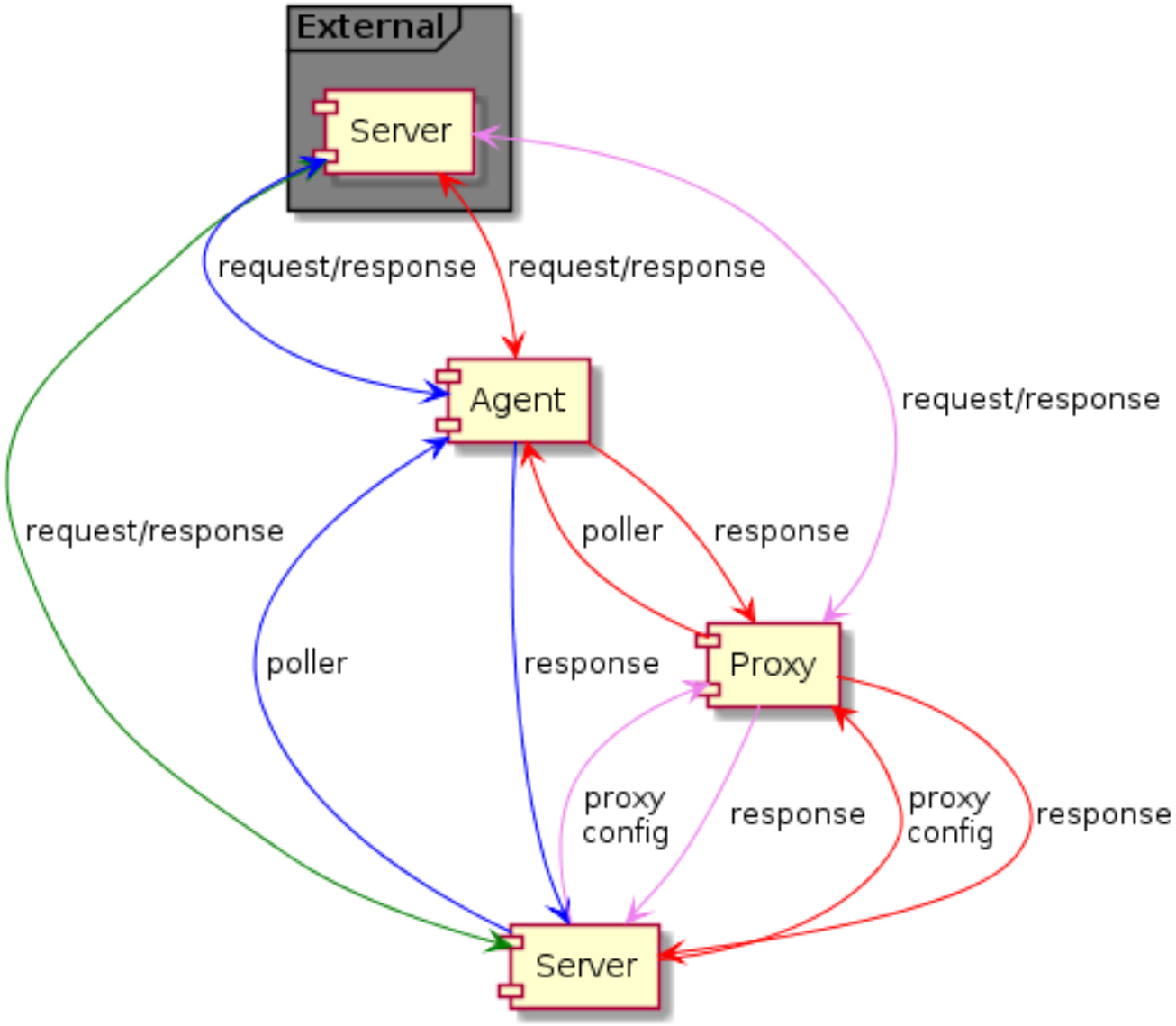
Item

Per configurare l'interrogazione delle statistiche interne su un'altra istanza Zabbix, è possibile utilizzare due item:

- item interno `zabbix[stats,<ip>,<port>]` - per interrogazioni remote dirette di Zabbix server/proxy. <ip> e <port> vengono utilizzati per identificare l'istanza di destinazione.
- item agent `zabbix.stats[<ip>,<port>]` - per interrogazioni remote di Zabbix server/proxy basate su agent. <ip> e <port> vengono utilizzati per identificare l'istanza di destinazione.

Vedi anche: Item interni, item Zabbix agent

Il seguente diagramma illustra l'uso di uno dei due item a seconda del contesto.



- - Server → istanza Zabbix esterna (`zabbix[stats,<ip>,<port>]`)
- - Server → proxy → istanza Zabbix esterna (`zabbix[stats,<ip>,<port>]`)
- - Server → agent → istanza Zabbix esterna (`zabbix.stats[<ip>,<port>]`)
- - Server → proxy → agent → istanza Zabbix esterna (`zabbix.stats[<ip>,<port>]`)

Per assicurarsi che l'istanza di destinazione consenta di essere interrogata dall'istanza esterna, elencare l'indirizzo dell'istanza esterna nel parametro 'StatsAllowedIP' sull'istanza di destinazione.

Metriche esposte

Gli item di statistiche raccolgono le statistiche in blocco e restituiscono un JSON, che costituisce la base da cui gli item dipendenti ottengono i propri dati. Le seguenti metriche interne sono restituite da uno dei due item:

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (solo server)
- `zabbix[preprocessing_queue]` (solo server)
- `zabbix[process,<type>,<mode>,<state>]` (solo statistiche basate sul tipo di processo)
- `zabbix[rcache,<cache>,<mode>]`
- `zabbix[requiredperformance]`
- `zabbix[triggers]` (solo server)
- `zabbix[uptime]`
- `zabbix[vcache,buffer,<mode>]` (solo server)

- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] (tipo di cache 'trends' solo server)

Template

Le template sono disponibili per il monitoraggio remoto delle metriche interne di Zabbix server o proxy da un'istanza esterna:

- Stato di salute remoto di Zabbix server
- Stato di salute remoto di Zabbix proxy

Si noti che, per utilizzare un template per il monitoraggio remoto di più istanze esterne, è richiesto un host separato per il monitoraggio di ciascuna istanza esterna.

Processo trapper

La ricezione delle richieste di metriche interne da un'istanza Zabbix esterna è gestita dal processo trapper, che convalida la richiesta, raccoglie le metriche, crea il buffer di dati JSON e invia il JSON preparato di ritorno, ad esempio, dal 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
      }
    }
  }
}
```


item della coda interna

Esistono anche altri due item che consentono specificamente di eseguire query remote sulle statistiche della coda interna di un'altra istanza Zabbix:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` item interno - per query dirette della coda interna al server/proxy Zabbix remoto
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` item agent - per query basate su agent della coda interna al server/proxy Zabbix remoto

Vedere anche: Item interni, item Zabbix agent

Configurazione di Kerberos con Zabbix

Panoramica

L'autenticazione Kerberos può essere utilizzata nel monitoraggio web e negli item HTTP in Zabbix.

Questa pagina descrive un esempio di configurazione di Kerberos per il server Zabbix, in modo da eseguire il monitoraggio web di `www.example.com` con un principal Kerberos per il processo Zabbix su Debian/Ubuntu.

Configurazione

1. Installare KDC e le utilità client:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Durante la configurazione del pacchetto, rispondere alle richieste, ad esempio:

```
Realm Kerberos predefinito versione 5: EXAMPLE.COM
Server Kerberos per il realm: localhost (o il proprio FQDN)
Server amministrativo per il realm Kerberos: localhost (o il proprio FQDN)
```

2. Mappare un nome host intuitivo (facoltativo, per i test locali).

Modificare /etc/hosts e aggiungere una voce per il DC e il webserver se non si dispone del DNS:

```
sudo vi /etc/hosts
```

Esempio di riga da aggiungere:

```
192.168.1.100  dc01.example.com dc01
```

3. Configurare il client Kerberos e il realm KDC:

```
sudo vi /etc/krb5.conf
```

Impostazioni di esempio:

```
[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
```

Se si prevede di utilizzare `.localdomain` o altri nomi non pubblici, aggiungere mappature esplicite dominio→realm affinché la mappatura nome host→realm funzioni. Le discrepanze in questa configurazione causano errori `Server not found in Kerberos database`.

4. Inizializzare il database Kerberos (una sola volta, sull'host KDC). Quando richiesto, impostare una password master sicura:

```
sudo krb5_newrealm
```

5. Creare il principal `HTTP/host.fqdn@REALM` utilizzando il nome host esatto che verrà usato dai client; preferire le lettere minuscole (ad esempio, `HTTP/web.example.com@EXAMPLE.COM`). Una discrepanza tra maiuscole/minuscole o tra nomi causa l'errore `Server not found in Kerberos database`.

```
sudo kadmin.local
```

All'interno di `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM     # principal amministrativo
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Spostare il keytab sull'host web (oppure mantenerlo localmente se si tratta della stessa macchina) e impostare autorizzazioni utilizzabili da Apache:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verifica
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Installare e abilitare il modulo GSSAPI di Apache:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Non tutte le versioni di mod_auth_gssapi supportano ogni direttiva `Gssapi*`. Se Apache restituisce l'errore `Invalid command 'GssapiCredStore'`, rimuovere la direttiva non supportata oppure aggiornare il modulo.

7. Configurare un VirtualHost (adattare `DocumentRoot` / il percorso alla propria interfaccia Zabbix):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

All'interno di `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>
```

Riavviare Apache:

```
sudo systemctl restart apache2
```

8. Abilitare/avviare i servizi KDC e verificare le porte in ascolto (host KDC):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # oppure: sudo netstat -tnlp | grep :80
```

9. Ottenere un TGT per i test (eseguire come l'utente che utilizzerà il ticket).

Nell'elenco dei ticket dovrebbe essere visualizzato `krbtgt/EXAMPLE.COM@EXAMPLE.COM`. Eseguire `kinit` come lo stesso utente

del sistema operativo che necessita del ticket (ad esempio, zabbix per i controlli web oppure www-data/Apache per i test SSO interattivi nel browser). I ticket rilasciati a un altro utente del sistema operativo non saranno visibili, a meno che non vengano modificati KRB5CCNAME e le autorizzazioni.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Testare lo scambio SPNEGO con curl (da un client con un TGT valido). Una risposta 200 OK (o un reindirizzamento all'applicazione) indica che SPNEGO è riuscito:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Facoltativamente, se l'interfaccia Zabbix deve accettare accessi autenticati tramite HTTP, abilitare l'autenticazione HTTP nel frontend Zabbix.

Nel file di configurazione del frontend (zabbix.conf.php) impostare:

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

Quindi configurare l'autenticazione HTTP nell'interfaccia web.

12. Configurazione del browser (Firefox viene utilizzato come esempio): impostare network.negotiate-auth.trusted-uris sugli host che eseguono Negotiate (dc01.example.com), affinché il browser invii automaticamente i token Kerberos.

All'interno di about:config:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Visitando ora http://dc01.example.com, si dovrebbe accedere direttamente a Zabbix senza visualizzare il modulo.

13. Mantenere aggiornati chiavi e ticket. La durata predefinita di un ticket Kerberos è di circa 10 ore. Aggiungere un timer cron/systemd per evitare le scadenze:

```
####per il servizio web
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####per l'utente di monitoraggio
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Controlli di manutenzione:

- klist -k /etc/apache2/http.keytab - verificare che il principal del servizio sia presente nel keytab.
- sudo tail -f /var/log/apache2/error.log - monitorare gli errori GSSAPI (gss_acquire_cred[_from]() failed to get server creds indica un problema con il keytab/le autorizzazioni oppure l'assenza del principal).
- La restituzione di 401/403 da parte di curl --negotiate indica spesso un principal errato, l'assenza di un ticket, una mancata corrispondenza dell'header host o un problema di autorizzazioni del file system; controllare i log e le mappature del dominio in /etc/krb5.conf.

Note sulla sicurezza e sui permessi dei file

I file keytab devono essere leggibili solo dall'account che ne ha bisogno. Esempi di permessi: 0400, con proprietario zabbix:zabbix, per un keytab dell'utente zabbix, oppure 0440 e root:www-data per un keytab di Apache.

Evita di memorizzare password in chiaro a lunga durata sul host. Quando possibile, usa keytab o principal di macchine aggiunte al dominio.

Quando esegui test o script che impostano KRB5CCNAME o copiano keytab, ricontrolla il proprietario e i permessi dopo l'operazione: se un webserver rifiuta le credenziali, spesso il problema riguarda i permessi dei file.

Parametri di modbus.get

Panoramica

La tabella seguente presenta i dettagli dei parametri dell'item modbus.get.

Parametri

Parameter	Description	Defaults	Example
endpoint	Protocollo e indirizzo dell'endpoint, definiti come `protocol://connection_string` Valori di protocollo possibili: *rtu*, *ascii* (solo Agent 2), *tcp* Formato della stringa di connessione: con *tcp* - `address:port` con linea seriale: *rtu*, *ascii* - `port_name:speed:params` dove 'speed' - 1200, 9600 ecc. 'params' - bit di dati (5,6,7 o 8), parità (n,e o o per nessuna/pari/dispari), bit di stop (1 o 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	Indirizzo Modbus del dispositivo a cui è destinato (da 1 a 247), vedere MODBUS Messaging Implementation Guide (pagina 23) il dispositivo tcp (non GW) ignorerà il campo	serial: 1 tcp: 255 (0xFF)	2
function	Vuoto oppure valore di una funzione supportata: 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	empty	3
address	Indirizzo del primo registro, coil o input. Se 'function' è vuoto, allora 'address' deve rientrare nell'intervallo di: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 Se 'function' non è vuoto, il campo 'address' andrà da 0 a 65535 e verrà usato senza modifiche (PDU)	empty function: 00001 non-empty function: 0	9999
count	Numero di 'type' sequenziali che verranno letti dal dispositivo, dove: per Coil o Discrete input il 'type' = 1 bit negli altri casi: (count*sizeof(type))/2 = numero reale di registri da leggere Se 'offset' non è 0, il valore verrà aggiunto al 'real count' L'intervallo accettabile per 'real count' è 1:65535	1	2
type	Tipo di dato: per Read Coil e Read Discrete Input - *bit* per Read Holding Registers e Read Input Registers: *int8* - 8bit *uint8* - 8bit (senza segno) *int16* - 16bit *uint16* - 16bit (senza segno) *int32* - 32bit *uint32* - 32bit (senza segno) *float* - 32bit *uint64* - 64bit (senza segno) *double* - 64bit	bit uint16	uint64

Parameter	Description	Defaults	Example
endianness	Tipo di endianness: *be* - Big Endian *le* - Little Endian *mbe* - Mid-Big Endian *mle* - Mid-Little Endian Limitazioni: per 1 bit - be per 8 bit - be,le per 16 bit - be,le	be	le
offset	Numero di registri, a partire da 'address', il cui risultato verrà scartato. La dimensione di ciascun registro è 16bit (necessario per supportare apparecchiature che non supportano l'accesso in lettura casuale).	0	4

Creazione di nomi personalizzati per i contatori di prestazioni per VMware

Panoramica

Il percorso del contatore di prestazioni VMware ha il formato `group/counter[rollup]` in cui:

- `group` - il gruppo del contatore di prestazioni, ad esempio, *cpu*
- `counter` - il nome del contatore di prestazioni, ad esempio, *usagemhz*
- `rollup` - il tipo di rollup del contatore di prestazioni, ad esempio, *average*

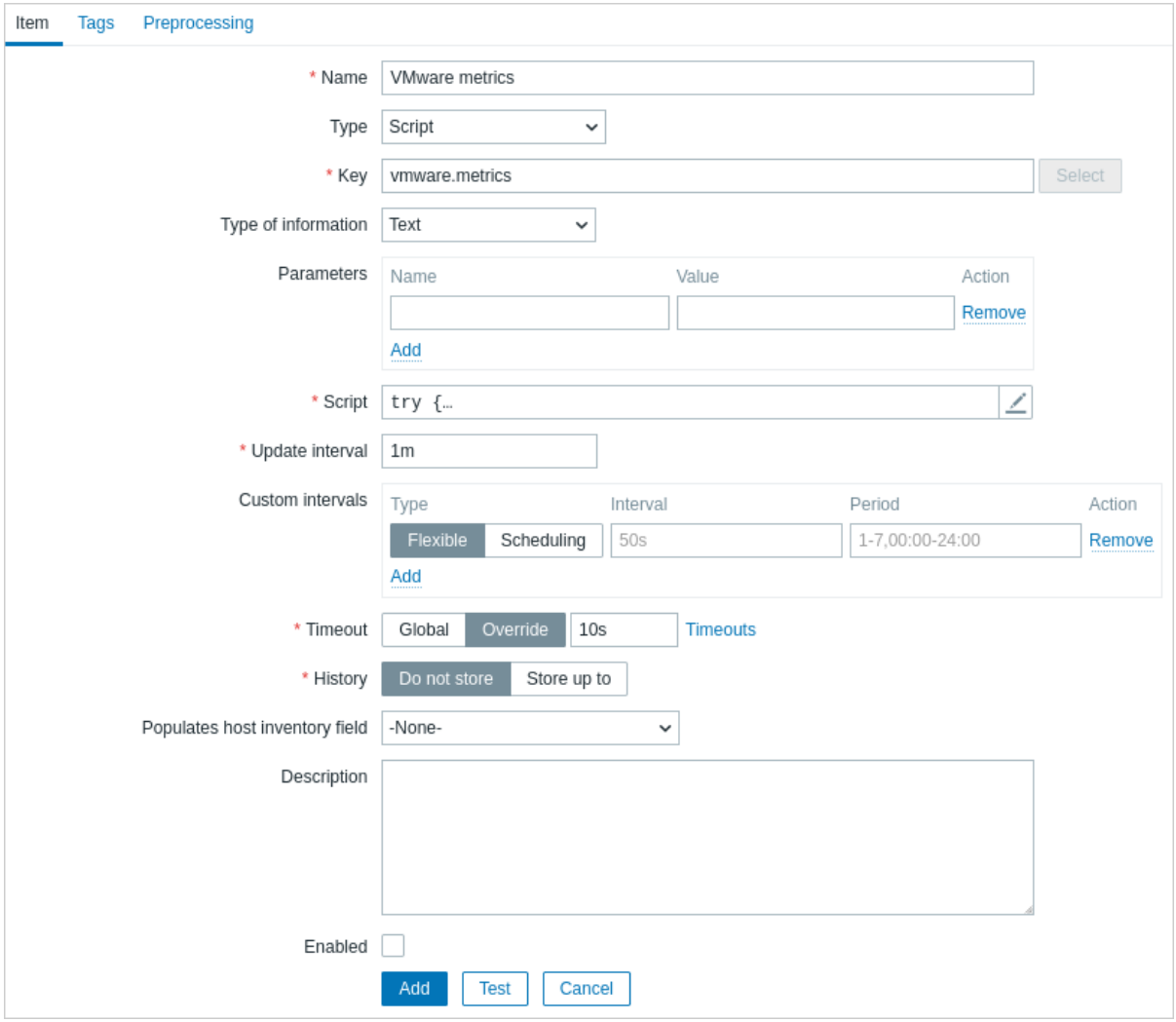
Quindi l'esempio sopra fornirebbe il seguente percorso del contatore: `cpu/usagemhz[average]`

Le descrizioni dei gruppi di contatori di prestazioni, i nomi dei contatori e i tipi di rollup possono essere trovati nella documentazione VMware.

È possibile ottenere i nomi interni e creare nomi personalizzati dei contatori di prestazioni utilizzando un item di tipo script in Zabbix.

Configurazione

1. Creare un item Script disabilitato sull'host VMware principale (dove è presente l'item **eventlog[]**) con i seguenti parametri:



- *Name*: VMware metrics
- *Type*: Script
- *Key*: vmware.metrics
- *Type of information*: Text
- *Script*: copia e incolla lo script riportato di seguito
- *Timeout*: 10
- *History*: Do not store
- *Enabled*: non selezionato

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;
```

Una volta configurato l'item, fare clic sul pulsante *Test*, quindi su *Get value*.



Copiare l'XML ricevuto in un qualsiasi formatter XML e individuare la metrica desiderata.

Un esempio di XML per una metrica:

```
<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>
```

Usare XPath per estrarre il percorso del counter dall'XML ricevuto. Per l'esempio sopra, l'XPath sarà:

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Il percorso risultante del performance counter in questo caso è: `cpu/usagemhz[average]`

Valori di ritorno per system.sw.packages.get

Panoramica

Questa sezione fornisce i dettagli sui valori restituiti per l'item Zabbix agent `system.sw.packages.get`.

Dettagli

L'output di questo item è un array di oggetti, ciascuno contenente le seguenti proprietà:

- **name** - nome del pacchetto
- **manager** - gestore di pacchetti che ha riportato questi dati (`rpm`, `dpkg`, `pacman`, `pkgtool` o `portage`)
- **version** - versione del pacchetto
- **size** - dimensione del pacchetto non compresso in byte (se non disponibile, impostata a 0)
- **arch** - architettura del pacchetto
- **buildtime** - un oggetto con 2 voci:
  - **timestamp** - timestamp UNIX relativo al momento in cui il pacchetto è stato compilato (se non disponibile, impostato a 0)
  - **value** - data e ora in formato leggibile relative al momento in cui il pacchetto è stato compilato (se non disponibili, impostate a stringa vuota)
- **installtime** - un oggetto con 2 voci:
  - **timestamp** - timestamp UNIX relativo al momento in cui il pacchetto è stato installato (se non disponibile, impostato a 0)
  - **value** - data e ora in formato leggibile relative al momento in cui il pacchetto è stato installato (se non disponibili, impostate a stringa vuota)

Esempio:

```
[
    {
        "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": ""
        }
    }
]
```

Valori di ritorno per net.dns.get

Panoramica

Questa sezione fornisce i dettagli sui valori restituiti per l'item `net.dns.get` di Zabbix agent 2.

Dettagli

L'output di questo item è un oggetto contenente informazioni sui record DNS in base ai parametri forniti nella chiave dell'item.

Ad esempio, l'item `net.dns.get[,example.com]` può restituire il seguente JSON di una query rifiutata:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

Specificando l'indirizzo IP del DNS server, l'item `net.dns.get[192.0.2.1,example.com]` può restituire il seguente JSON:

```
{
    "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
}
```


Se si verifica un problema di connessione, l'item `net.dns.get[192.0.2.1,example.com]` può restituire un errore:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```


Sono possibili i seguenti tipi di codici di errore:

Scenario	"zbx_error_code"	"zbx_error_msg"
Nessun errore e la risposta DNS è stata ricevuta e analizzata.	0	
DNS non disponibile.	-1	"Communication error"
Si verifica un errore durante l'analisi JSON	-2	"Received unexpected response"

Con parametri aggiuntivi, l'item `net.dns.get[192.0.2.1,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,nsid,edns0,aaf` può restituire il seguente JSON:


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
                ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
```

```
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "a": "192.0.2.1"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Vedi anche

Per ulteriori informazioni sui record DNS, vedere:

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

Valori di ritorno per net.if.get

Panoramica

Questa sezione fornisce i dettagli sui valori restituiti per l'item `net.if.get`. L'item è supportato su Linux da Zabbix agent e Zabbix agent 2.

Dettagli

Questo item restituisce informazioni dettagliate sulle interfacce di rete i cui nomi corrispondono alla data espressione regolare. L'output è strutturato come un oggetto JSON contenente gli array `config` e `values`.

L'output contiene le seguenti proprietà:

- **name** - nome dell'interfaccia, letto da `/proc/net/dev`;
- **ifalias** - alias dell'interfaccia, letto da `/sys/class/net/<name>/ifalias`;
- **mac** - indirizzo MAC dell'interfaccia, letto da `/sys/class/net/<name>/address`;
- **type** - tipo di interfaccia (`physical`, `virtual`, `loopback`);
- **speed** - velocità del collegamento in Mbit/sec, specifica per Ethernet, letta da `/sys/class/net/<name>/speed`;
- **duplex** - valore duplex più recente o corrente dell'interfaccia (`half`, `full`), specifico per Ethernet, letto da `/sys/class/net/<name>/`
- **administrative_state** - stato configurato (`down`, `up`), letto da `/sys/class/net/<name>/flags`;
- **operational_state** - stato operativo RFC2863 dell'interfaccia (`unknown`, `notpresent`, `down`, `lowerlayerdown`, `testing`, `dormant`, `up`), letto da `sys/class/net/<name>/operstate`;
- **carrier** - presenza del collegamento fisico (0 - il collegamento fisico è inattivo, 1 - il collegamento fisico è attivo), letto da `/sys/class/net/<name>/carrier`;
- **carrier_changes** - numero totale di volte in cui lo stato del collegamento è passato da attivo a inattivo e viceversa, letto da `/sys/class/net/<name>/carrier_changes`;
- **carrier_up_count** - numero di volte in cui il collegamento è diventato attivo, letto da `/sys/class/net/<name>/carrier_up_count`
- **carrier_down_count** - numero di volte in cui il collegamento è diventato inattivo, letto da `/sys/class/net/<name>/carrier_down`
- **in** - traffico in ingresso, letto da `/proc/net/dev`. Contiene i contatori dell'interfaccia:
  - **bytes** - numero di byte validi;
  - **packets** - numero di pacchetti validi;
  - **errors** - numero totale di pacchetti non validi;
  - **dropped** - numero di pacchetti ricevuti ma non elaborati + conteggio dei pacchetti persi dal host;
  - **overruns** - numero di eventi di overflow;
  - **frame** - numero di pacchetti con errori di framing;
  - **compressed** - numero di pacchetti compressi ricevuti correttamente;
  - **multicast** - pacchetti multicast ricevuti.
- **out** - traffico in uscita, letto da `/proc/net/dev`. Contiene i contatori dell'interfaccia:
  - **bytes** - numero di byte validi;
  - **packets** - numero di pacchetti validi;
  - **errors** - numero totale di pacchetti non validi;
  - **dropped** - numero di pacchetti ricevuti ma non elaborati + conteggio dei pacchetti persi dal host;
  - **overruns** - numero di eventi di overflow;
  - **collisions** - numero di collisioni durante la trasmissione dei pacchetti;
  - **carrier** - numero di errori di trasmissione dei frame dovuti alla perdita del carrier durante la trasmissione;
  - **compressed** - numero di pacchetti compressi trasmessi.

Le metriche non disponibili sono escluse dall'output JSON. Nell'esempio seguente, non è configurato alcun alias per l'interfaccia "lo", quindi la proprietà `ifalias` è esclusa.

Esempio:


```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
            "carrier": 0
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
```

```
            "type": "physical",
            "administrative_state": "up",
            "operational_state": "up",
            "carrier": 1
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
```

```
            }
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


Vedi anche

Per ulteriori informazioni sull'ABI sysfs per i dispositivi di rete, vedere:

- Documentazione del kernel Linux

Valori di ritorno per vfs.dev.get

Panoramica

Questa sezione fornisce i dettagli sul valore restituito per l'item `vfs.dev.get`. L'item è supportato su Linux da Zabbix agent e Zabbix agent 2, e su AIX da Zabbix agent.

Dettagli

Questo item restituisce informazioni dettagliate sui dispositivi a blocchi i cui nomi corrispondono alla regular expression specificata.

L'output è strutturato come un oggetto JSON contenente gli array `config` e, a seconda del parametro <mode>, `values`.

L'array `config` contiene le proprietà disponibili per la modalità selezionata. L'array `values` viene restituito solo per le modalità che forniscono statistiche sui dispositivi a blocchi.

Le metriche non disponibili sono escluse dall'output JSON.

Le sezioni seguenti descrivono le proprietà restituite per ciascuna modalità.

modalità dischi

`vfs.dev.get[disks,*]` restituisce i dati di configurazione dei dispositivi a blocchi.

Proprietà:

- **name** - nome del dispositivo a blocchi;
- **devid** - identificatore del dispositivo a blocchi, ad esempio un ID dispositivo basato su WWN o generato dal kernel;
- **type** - tipo di dispositivo a blocchi (`disk`, `partition`, `rom`);

- **path** - percorso del nodo del dispositivo a blocchi;
- **model** - modello del dispositivo;
- **serial** - numero di serie del dispositivo;
- **wwn** - World Wide Name (WWN) del dispositivo;
- **size_bytes** - dimensione del dispositivo in byte;
- **logical_block_size** - dimensione del blocco logico in byte;
- **physical_block_size** - dimensione del blocco fisico in byte.

Esempio per Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
    }
  ]
}
```


Esempio per AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
```

```
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```


modalità dispositivi

`vfs.dev.get[devices,.*]` restituisce i dati di configurazione dei dispositivi a blocchi, incluse le partizioni.

Proprietà:

- **name** - nome del dispositivo a blocchi.
- **devid** - identificatore del dispositivo a blocchi.
- **type** - tipo di dispositivo a blocchi (`disk`, `partition`, `rom`).
- **partitions** - partizioni sul dispositivo. Contiene i nomi delle partizioni e le loro dimensioni in byte.

Esempio per Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
    }
  ]
}
```


modalità disk_stats

`vfs.dev.get[disk_stats,.*]` restituisce i dati di configurazione dei dispositivi a blocchi e le statistiche del disco.

Proprietà:

- **name** - nome del dispositivo a blocchi.
- **devid** - identificatore del dispositivo a blocchi.
- **type** - tipo di dispositivo a blocchi (`disk`, `partition`, `rom`).
- **size_bytes** - dimensione del dispositivo in byte.
- **logical_block_size** - dimensione del blocco logico in byte.
- **stats** - statistiche I/O del dispositivo a blocchi. Contiene:
  - **reads_completed** - numero di operazioni di lettura completate;
  - **writes_completed** - numero di operazioni di scrittura completate;
  - **bytes_read** - numero di byte letti;
  - **bytes_written** - numero di byte scritti;
  - **io_time_ms** - tempo totale trascorso per eseguire I/O, in millisecondi.

Esempio per Linux


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
```

```
      }
    }
  ]
}
```

Esempio per AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
        "bytes_written": 0,
        "io_time_ms": 0
      }
```

```
    }
  ]
}
```

modalità device_stats

`vfs.dev.get[device_stats,.*]` restituisce i dati di configurazione dei dispositivi a blocchi e le statistiche dei dispositivi.

Proprietà:

- **name** - nome del dispositivo a blocchi.
- **devid** - identificatore del dispositivo a blocchi.
- **type** - tipo di dispositivo a blocchi (`disk`, `partition`, `rom`).
- **size_bytes** - dimensione del dispositivo in byte.
- **stats** - statistiche I/O del dispositivo a blocchi. Contiene:
  - **reads_completed** - numero di operazioni di lettura completate;
  - **writes_completed** - numero di operazioni di scrittura completate;
  - **bytes_read** - numero di byte letti;
  - **bytes_written** - numero di byte scritti;
  - **io_time_ms** - tempo totale trascorso per l'I/O, in millisecondi.

Esempio per Linux:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
```

```
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sda1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sda2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
```

```
    }
  },
  {
    "name": "sdb",
    "stats": {
      "reads_completed": 51017,
      "writes_completed": 10682,
      "bytes_read": 10514627,
      "bytes_written": 1530472,
      "io_time_ms": 284988
    }
  },
  {
    "name": "sdb1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb2",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb3",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "nvme0n1",
    "stats": {
      "reads_completed": 1259,
      "writes_completed": 7,
      "bytes_read": 49698,
      "bytes_written": 6,
      "io_time_ms": 243
    }
  },
  {
    "name": "nvme0n1p1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
```

```
    {
      "name": "nvme0n1p2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```

Note sugli item system.cpu.util su Windows

L'item [system.cpu.util] fornisce la percentuale di utilizzo della CPU.

Quando su Zabbix agent per Windows viene avviato un processo collector, viene inizializzato un buffer per N item per N processori logici (thread). I valori vengono aggiornati periodicamente nel buffer utilizzando i contatori delle prestazioni di Windows. Questi valori vengono restituiti quando viene richiesto `system.cpu.util[n]`, dove 0 <= n < N.

Un ulteriore item nel buffer viene utilizzato per `system.cpu.util[total]`.

Zabbix seleziona automaticamente contatori delle prestazioni diversi per supportare sia i sistemi NUMA sia le versioni meno recenti di Windows senza supporto per i gruppi di processori.

Se il numero di processori logici (thread) è minore o uguale a 64 e il numero di gruppi di processori è uguale a 1, allora:

```
N è il numero di processori logici (thread).

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

Altrimenti:

```
G è il numero di gruppi di processori e N è il numero di processori logici (thread) nel gruppo.

\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
```

Supporto per valori JSON di grandi dimensioni

Per garantire una gestione affidabile di valori di grandi dimensioni con il tipo di dato JSON data type (1 MiB o più), esaminare e applicare le seguenti modifiche di configurazione, se necessario:

1. Se si utilizza MySQL o MariaDB, aumentare la dimensione massima consentita del pacchetto in `/etc/mysql/my.cnf` (su alcuni sistemi potrebbe essere impostata per impostazione predefinita a 1 MB):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Regolare, se necessario, ulteriori server system variables di MySQL/MariaDB e InnoDB system variables. Ad esempio:

```
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. Aumentare `HistoryCacheSize` a `2G` nel file di configurazione di Zabbix server.

4. Se si utilizza Zabbix proxy, aumentare `HistoryCacheSize` a `2G` e `ProxyMemoryBufferSize` a `2G` nel file di configurazione del proxy.

5. Se si utilizza Zabbix agent, aumentare i timeout di comunicazione tra Zabbix server/proxy e agent (ad esempio, i item timeouts oppure il valore del parametro `Timeout` nei file di configurazione di Zabbix server/proxy e agent). In caso contrario, i messaggi tra questi componenti potrebbero non completarsi e potrebbero comparire errori come *"message length does not match expected length"*. Inoltre, rivedere altri timeout, se necessario, come il timeout dello script timeout o il timeout del item test.

Funzioni supportate

Vedere Espressione e funzioni dei trigger.

Macro

È possibile utilizzare macro supportate predefinite e macro utente supportate per posizione.

Macro supportate dalla posizione

Panoramica

La tabella macro supportate in base alla posizione contiene il riferimento di tutte le macro (integrate, di espressione e definibili dall'utente) supportate nella posizione specifica.

Se tutte le macro di un gruppo correlato sono supportate, viene indicato solo il nome del gruppo. Fare clic su ciascun nome di macro o nome di gruppo per visualizzare i dettagli completi delle macro.

L’elenco dettagli delle macro contiene un elenco dettagliato di tutte le macro **integrate**, raggruppate per area di applicazione. Nota che per personalizzare i valori delle macro (ad esempio, abbreviare o estrarre sottostringhe specifiche), puoi usare le funzioni macro.

Macro supportate per la posizione

Posizione	Macro supportate	Commenti
Action		
Condizione del periodo di tempo, Durata operazione predefinita, Durata passo	macro definibili dall’utente (solo globali)	Un singolo macro deve occupare l’intero campo; non sono supportati più macro o macro mescolati con testo
Notifiche e co-mandi		
Basato su trigger	macro Action macro Data/ora {ESC.HISTORY} macro Event, macro Evento causa/sintomo Macro di aggiornamento evento: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Macro di espressione: {?EXPRESSION} {FUNCTION.VALUE} Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro host di destinazione (solo comandi) macro inventario host macro Item (eccetto {ITEM.STATE}, {ITEM.STATE.ERROR}) macro Proxy Macro trigger: {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} macro definibili dall’utente	Vedi anche: Uso delle macro nei messaggi

Posizione	Macro supportate	Commenti
Aggiornamento problema	macro Action macro Data/ora {ESC.HISTORY} macro Event, macro Evento causa/sintomo, macro Evento di recupero macro Aggiornamento evento: tutte (eccetto {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Macro di espressione: {?EXPRESSION} macro Function Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro host di destinazione (solo comandi) macro inventario host macro Item (eccetto {ITEM.STATE}, {ITEM.STATE.ERROR}) macro Proxy Macro trigger: {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} macro definibili dall'utente	
Ripristino problema	macro Evento di recupero macro Event {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Discovery	macro Action macro Data/ora macro Discovery Macro evento: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} macro host di destinazione (solo comandi) macro Proxy	
Autoregistrazione	macro Action macro Data/ora Macro evento: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Macro host: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} macro host di destinazione (solo comandi) macro Proxy	
Basato su servizio	macro Action macro Data/ora {ESC.HISTORY} Macro evento: tutte eccetto {EVENT.ACK.STATUS}, {EVENT.OPDATA} macro Service macro definibili dall'utente	
Ripristino servizio	macro Event (eccetto {EVENT.ACK.STATUS}, {EVENT.OPDATA}), macro Evento di recupero	

Posizione	Macro supportate	Commenti
Aggiornamento servizio	macro Action macro Data/ora {ESC.HISTORY} macro Event (eccetto {EVENT.ACK.STATUS}, {EVENT.OPDATA}) macro di aggiornamento evento: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} macro Service macro definibili dall'utente	
Notifiche interne	macro Action macro Data/ora {ESC.HISTORY} macro Event (eccetto {EVENT.NSEVERITY}, {EVENT.SEVERITY}), macro Evento di recupero Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host Macro item: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} macro Proxy	
Basato su item	Macro item: {ITEM.STATE}, {ITEM.STATE.ERROR}	
Basato su regola LLD	macro notifica di discovery di basso livello	
Basato su trigger	Macro trigger: {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} macro definibili dall'utente	
Parametri script di allarme	macro Alert macro definibili dall'utente (solo globali)	
Connector		
Campo URL, Username, Password, Bearer token, HTTP proxy, certificato SSL, file chiave SSL, password chiave SSL	macro definibili dall'utente (solo globali)	
Campo Timeout	macro definibili dall'utente (solo globali)	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.

Posizione	Macro supportate	Commenti
Nome grafico	Macro di espressione: {?EXPRESSION}	In questa posizione, in questo macro è supportata solo una singola funzione **avg**, **last**, **max** o **min** con i secondi come parametro (si possono usare i suffissi temporali). La macro {HOST.HOST<1-9>} può essere usata come host all'interno del macro (vedi anche macro indicizzate). Esempio di utilizzo: {?avg(/{HOST.HOST}/item.key,1h)} Uso non supportato: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Host, prototipo host		
IP/DNS interfaccia	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} macro definibili dall'utente [1]	[1] Per l'indirizzo IP, un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo
Porta interfaccia	macro definibili dall'utente	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Community SNMP *Nome contesto SNMPv3, Nome sicurezza, Passphrase autenticazione, Passphrase privacy* *Nome utente IPMI, password*	macro definibili dall'utente	
Item, prototipo item, regola LLD		
Nome	macro definibili dall'utente	Non supportato nei nomi delle regole LLD.
Descrizione	macro definibili dall'utente	

Posizione	Macro supportate	Commenti
Parametri chiave	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro definibili dall'utente	Le macro {HOST.*} supportate verranno risolte nell'interfaccia selezionata per l'item. Se usate in item senza interfacce, verranno risolte nell'interfaccia Zabbix agent, SNMP, JMX o IPMI dell'host in questo ordine di priorità, oppure in 'UNKNOWN' se l'host non ha alcuna interfaccia. Le macro non sono supportate nella chiave item degli item Zabbix trapper.
Intervallo di aggiornamento, Intervalli personalizzati, Timeout (disponibile per tipi di item supportati), Memorizza fino a (per history/trends)	macro definibili dall'utente	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Elimina risorse perse, Disabilita risorse perse	macro definibili dall'utente	Questi campi sono supportati solo per le regole LLD. Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Parametri del passo di preprocessing (inclusi script personalizzati), *parametri personalizzati di gestione errori* (campi *Imposta valore a* e *Imposta errore a*)	macro definibili dall'utente	
Espressioni regolari del filtro	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro definibili dall'utente	Questi campi sono supportati solo per le regole LLD.
Espressioni regolari del filtro di override	macro definibili dall'utente	Questi campi sono supportati solo per le regole LLD.
Operazioni di override: *Intervallo di aggiornamento*, *Periodo di conservazione della history*, *Periodo di conservazione dei trend* (per i prototipi item)	macro definibili dall'utente	Questi campi sono supportati solo per le regole LLD. Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.

Posizione	Macro supportate	Commenti
Tipi di item		
Browser item *nomi e valori dei parametri*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host Macro item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} macro definibili dall'utente	
Script item *nomi e valori dei parametri*		
Calcolato/aggregato formula item (*costanti di espressione*, *parametri di funzione*, *parametri chiave item*, (solo item aggregato) *condizioni di filtro* (nome del gruppo host e nome del tag))	macro definibili dall'utente	
Monitor database item *campi Username, Password, SQL query*	macro definibili dall'utente	
HTTP agent item *campi URL, campi Query, corpo della richiesta, intestazioni (nomi e valori), file certificato SSL, file chiave SSL*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} macro definibili dall'utente	Gli URL che contengono una macro segreta non funzioneranno, poiché la macro in essi verrà risolta come "******"
HTTP agent item *campo HTTP Proxy*	Macro item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} macro definibili dall'utente	
HTTP agent item *campi Required status codes, HTTP authentication username/password, SSL key password*	macro definibili dall'utente	
HTTP agent item *campo Allowed hosts*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} macro definibili dall'utente	
JMX agent item *campo Endpoint*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} macro definibili dall'utente	
Trapper item *campo Allowed hosts*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP} macro definibili dall'utente	
SNMP agent item *campo SNMP OID*	macro definibili dall'utente	
SSH and Telnet item *script*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro definibili dall'utente	
SSH agent item *campi Username, Public key file, Private key file, Password*	macro definibili dall'utente	
Telnet agent item *campi Username, Password*	macro definibili dall'utente	

Posizione	Macro supportate	Commenti
Mappa		
URL, nome URL	Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} macro inventario host macro Mappa {TRIGGER.ID}	
Etichetta elemento	Macro di espressione: {?EXPRESSION} [1] Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} macro inventario host macro Mappa Macro trigger: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] In questa posizione, in questo macro è supportata solo una singola funzione **avg**, **last**, **max** o **min** con i secondi come parametro (si possono usare i suffissi temporali). La macro {HOST.HOST<1-9>} può essere usata come host all'interno del macro. Esempio di utilizzo: {?avg(/{HOST.HOST}/item.key,1h)} Uso non supportato: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Etichetta collegamento	Macro di espressione: {?EXPRESSION} [1]	
Etichetta forma	Macro di espressione: {?EXPRESSION} [1]	
Campo testo nelle forme della mappa	{MAP.NAME}	
Rilevamento di rete		
Campo Intervallo di aggiornamento	macro definibili dall'utente (solo globali)	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
campi Community SNMP, SNMP OID campi Nome contesto SNMPv3, Nome sicurezza, Passphrase autenticazione, Passphrase privacy	macro definibili dall'utente (solo globali)	
Proxy		
Address for active agents > *Port* (se il proxy appartiene a un gruppo) Per la modalità proxy passiva: campi *Address* e *Port* dell'interfaccia campi *Timeouts for item types* in modalità override	macro definibili dall'utente (solo globali)	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Gruppo proxy		

Posizione	Macro supportate	Commenti
Failover period, Minimum number of proxies	macro definibili dall'utente (solo globali)	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Script		
Comandi, testo di conferma (Tipo: script, **host action** manuale)	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host [2] {MANUALINPUT} macro Username macro definibili dall'utente	[1] Solo testo di conferma [2] Supportato per Zabbix server e Zabbix proxy
Comandi (Tipo: script, **event action** manuale)	macro Data/ora macro Event, macro Evento causa/sintomo, macro Evento di recupero (se il recupero ha avuto luogo) macro di aggiornamento evento: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host macro Item (eccetto {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} macro Proxy Macro trigger: {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} macro Username macro definibili dall'utente	
Testo di conferma (Tipo: script, **event action** manuale)	Macro evento: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host {MANUALINPUT} macro Username macro definibili dall'utente	
URL, testo di conferma (Tipo: URL, **host/event action** manuale)		
Nome tag e valore	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host Macro item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macro trigger: {TRIGGER.ID} (solo valore del tag trigger) macro definibili dall'utente	Le macro del nome e del valore del tag vengono risolte solo durante il processo di generazione dell'evento.
Trigger		
Nome	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macro/riferimenti posizionali: $1...$9 macro definibili dall'utente	

Posizione	Macro supportate	Commenti
Nome evento	macro Data/ora: {TIME}, {TIMESTAMP} Macro di espressione: {?EXPRESSION} macro Function Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macro trigger: {TRIGGER.EXPRESSION.EXPLAIN}	
Dati operativi	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} macro definibili dall'utente	
Espressione	{TRIGGER.VALUE} macro definibili dall'utente [1]	[1] Solo nelle costanti e nei parametri di funzione; le macro segrete non sono supportate
Campi Tag per corrispondenza, Nome voce di menu, URL voce di menu	macro definibili dall'utente	Gli URL che contengono una macro segreta non funzioneranno, poiché la macro in essi verrà risolta come "******".
Descrizione	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} macro definibili dall'utente	
URL	{EVENT.ID} Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Comandi parametro utente	macro/riferimenti posizionali: `$1...$9`	
Scenario web		
Campi Nome, Agent, HTTP proxy	macro definibili dall'utente	
Campo Intervallo di aggiornamento	macro definibili dall'utente	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Campi Variabili, Intestazioni, certificato SSL, file chiave SSL	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro definibili dall'utente	
Campi User, Password, SSL key password	macro definibili dall'utente	
Passo dello scenario web *Campi Nome, Variabili (solo valori)*	macro definibili dall'utente	
Passo dello scenario web *Campi Timeout, Required status codes*	macro definibili dall'utente	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.

Posizione	Macro supportate	Commenti
Passo dello scenario web *Campi URL, Intestazioni (nomi e valori), Required string*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro definibili dall'utente	
Passo dello scenario web *Campo Post*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Tipo di media webhook		
Valori parametro	macro Alert Tutte le macro supportate nelle notifiche di problema basate su trigger (vedi sopra)	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Script JavaScript, nome parametro script *JavaScript*	macro definibili dall'utente (solo globali)	
Nome voce di menu, URL voce di menu	{EVENT.TAGS.<tag name>}	
Widget		
Gauge widget *Campo Descrizione*	Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host Macro item: {ITEM.MACROS} (eccetto {ITEM.STATE.ERROR}) macro definibili dall'utente (solo globali)	
Honeycomb widget *Campo etichetta primaria/secondaria* **Item value** widget *Campo Descrizione*		
Top hosts widget *Colonna Dati testuali*	Macro host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro inventario host macro definibili dall'utente (solo globali)	
URL (Dynamic) widget *Campo URL*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macro definibili dall'utente	
Colonna **Custom text***		
Monitoring > Problems	macro Host (eccetto {HOST.METADATA}) macro inventario host macro Item (eccetto {ITEM.STATE.ERROR}) Macro trigger: {TRIGGER.ID} Macro evento: {EVENT.ID} macro definibili dall'utente	
Monitoring > Hosts	macro Host (eccetto {HOST.METADATA}) macro inventario host macro definibili dall'utente	
Monitoring > Latest data	macro Host (eccetto {HOST.METADATA}) macro inventario host macro Item (eccetto {ITEM.STATE.ERROR}) macro definibili dall'utente	

Posizione	Macro supportate	Commenti
Data collection > Templates *Data collection > Hosts*	Macro host: {HOST.DESCRIPTION}, {HOST.HOST}, {HOST.ID}, {HOST.NAME} macro Host (eccetto {HOST.METADATA}) macro inventario host macro definibili dall'utente	
Altre posizioni		
Campo Orario di lavoro	macro definibili dall'utente (solo globali)	Un singolo macro deve occupare l'intero campo; non sono supportati più macro o macro mescolati con testo.
Campo Timeouts for item types (impostazione globale)		
Media utente *Campo When active*	macro definibili dall'utente (solo globali)	
Tipo di media Email *campi Username, Password*	macro definibili dall'utente (solo globali)	Si raccomandano le macro segrete.
Report pianificati *campi Subject, Message*	macro Data/ora: {TIME}, {TIMESTAMP}	

Dettagli delle macro

Questo elenco contiene i dettagli di ciascuna macro integrata, raggruppata per area applicativa:

- Macro delle azioni
- Macro degli avvisi
- Macro di data/ora
- Macro di individuazione
- Macro degli eventi
  - Macro di aggiornamento evento
  - Macro evento causa/sintomo
  - Macro evento di ripristino
- Macro di escalation
- Macro di funzione
- Macro host
  - Macro host di destinazione
- Macro dei gruppi host
- Macro dell'inventario host
- Macro item
- Macro di notifica di low-level discovery
- Macro delle mappe
- Macro proxy
- Macro degli script
- Macro dei servizi
- Macro trigger
- Macro del nome utente

Macro delle azioni

{ACTION.ID}

L'ID numerico dell'azione attivata.

{ACTION.NAME}

Il nome dell'azione attivata.

Macro di avviso

{ALERT.MESSAGE}

Il valore del *messaggio predefinito* dalla configurazione dell'azione.

{ALERT.SENDTO}

Il valore *Send to* dalla configurazione dei supporti utente.

{ALERT.SUBJECT}

Il valore di *Default subject* dalla configurazione dell'azione.

Macro di data e ora

{DATE}

La data corrente nel formato `yyyy.mm.dd` .

{TIME}

L'ora corrente nel formato `hh:mm:ss`.

{TIMESTAMP}

L'ora corrente nel formato timestamp UNIX.

Macro di discovery

{DISCOVERY.DEVICE.IPADDRESS}

L'indirizzo IP del dispositivo rilevato.
 Sempre disponibile, non dipende dall'aggiunta dell'host.

{DISCOVERY.DEVICE.DNS}

Il nome DNS del dispositivo individuato.
 Sempre disponibile, non dipende dall'aggiunta del host.

{DISCOVERY.DEVICE.STATUS}

Lo stato del dispositivo rilevato (UP/DOWN).

{DISCOVERY.DEVICE.UPTIME}

Il tempo trascorso dall'ultima modifica dello stato di discovery per un determinato dispositivo.
 Ad esempio: 1h 29m 01s.
 Per i dispositivi con stato DOWN, questo è il periodo di inattività.

{DISCOVERY.RULE.NAME}

Il nome della regola di individuazione che ha rilevato la presenza o l'assenza del dispositivo/servizio.

{DISCOVERY.SERVICE.NAME}

Il nome del servizio individuato.
 Per esempio: HTTP.

{DISCOVERY.SERVICE.PORT}

La porta del servizio rilevato.
 Ad esempio: 80.

{DISCOVERY.SERVICE.STATUS}

Lo stato del servizio individuato (UP/DOWN).

{DISCOVERY.SERVICE.UPTIME}

Il tempo trascorso dall'ultima modifica dello stato di discovery per un particolare servizio.
 Ad esempio: 1h 29m 01s.
 Per i servizi con stato DOWN, questo è il periodo della loro indisponibilità.

Macro di escalation

{ESC.HISTORY}

Il log delle notifiche inviate in precedenza, del relativo passo di escalation e del loro stato (*inviato*, *in corso* o *fallito*).

Macro evento

{EVENT.ACK.STATUS}

Lo stato di riconoscimento dell'evento (Sì/No).

{EVENT.AGE}

L'età dell'evento che ha attivato un'azione, con precisione fino al secondo.
 Utile nei messaggi di escalation. Per esempio: 9m 13s.

{EVENT.DATE}

La data dell'evento che ha attivato un'azione nel formato `yyyy.mm.dd`.
 Per esempio: 2025.04.14.

{EVENT.DURATION}

La durata dell'evento (differenza di tempo tra gli eventi di problema e di ripristino), con precisione fino al secondo.
 Utile nei messaggi di ripristino del problema.

{EVENT.ID}

L'ID numerico dell'evento che ha attivato un'azione.

{EVENT.NAME}

Il nome dell'evento che ha attivato un'azione.

{EVENT.NSEVERITY}

Il valore numerico della gravità dell'evento.
 Valori possibili: *0* - Non classificato, *1* - Informazione, *2* - Avviso, *3* - Medio, *4* - Alto, *5* - Disastro.

{EVENT.OBJECT}

Il valore numerico dell'oggetto evento.
 Valori possibili: *0* - Trigger, *1* - host scoperto, *2* - servizio scoperto, *3* - autoregistrazione, *4* - Item, *5* - regola di low-level discovery.

{EVENT.OPDATA}

I dati operativi del trigger sottostante del problema.

{EVENT.SEVERITY}

Il nome della gravità dell'evento.

{EVENT.SOURCE}

Il valore numerico della sorgente dell'evento.
 Valori possibili: *0* - Trigger, *1* - Discovery, *2* - Registrazione automatica, *3* - Interno, *4* - Servizio.

{EVENT.STATUS}

Il valore testuale dell'evento che ha attivato un'azione.

{EVENT.TAGS}

Un elenco di tag evento separati da virgole.
Espanso in una stringa vuota se non esistono tag.

{EVENT.TAGSJSON}

Un array JSON contenente oggetti tag dell'evento.
Espanso in un array vuoto se non esistono tag.

{EVENT.TAGS.<tag name>}

Il valore del tag dell'evento a cui si fa riferimento tramite il nome del tag.
 Un nome di tag contenente caratteri non alfanumerici (inclusi caratteri multibyte UTF non inglesi) deve essere racchiuso tra virgolette doppie. Le virgolette e le barre rovesciate all'interno di un nome di tag tra virgolette devono essere escape con una barra rovesciata.

{EVENT.TIME}

L'ora dell'evento che ha attivato un'azione nel formato `hh:mm:ss`.
 Per esempio: 12:57:53.

{EVENT.TIMESTAMP}

Il timestamp UNIX dell'evento che ha attivato un'azione.

{EVENT.VALUE}

Il valore numerico dell'evento che ha attivato un'azione.
 Valori possibili: *1* - problema, *0* - ripristino.

Macro evento di causa/sintomo

{EVENT.CAUSE.ACK.STATUS}

Lo stato di riconoscimento dell'evento causa (Sì/No).
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.AGE}

L'età dell'evento causa, con precisione fino al secondo.
 Utile nei messaggi di escalation.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.DATE}

La data dell'evento causa nel formato `yyyy.mm.dd`.
 Utilizzata nel contesto di un evento sintomo.

{EVENT.CAUSE.DURATION}

La durata dell'evento causa (differenza temporale tra gli eventi di problema e di ripristino), con precisione fino al secondo.
 Utile nei messaggi di ripristino del problema.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.ID}

L'ID numerico dell'evento causa.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.NAME}

Il nome dell'evento causa.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.NSEVERITY}

Il valore numerico della gravità dell'evento causa.
 Valori possibili: *0* - Non classificato, *1* - Informazione, *2* - Avviso, *3* - Medio, *4* - Alto, *5* - Disastro.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.OBJECT}

Il valore numerico dell'oggetto dell'evento causa.
 Valori possibili: *0* - Trigger, *1* - host rilevato, *2* - servizio rilevato, *3* - autoregistrazione, *4* - Item, *5* - regola di low-level discovery.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.OPDATA}

I dati operativi del trigger sottostante del problema causa.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.SEVERITY}

Il nome della gravità dell'evento causa.
 Valori possibili: *Non classificato*, *Informazione*, *Avviso*, *Media*, *Alta*, *Disastro*.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.SOURCE}

Il valore numerico della sorgente dell'evento causa.
 Valori possibili: *0* - Trigger, *1* - Discovery, *2* - Autoregistration, *3* - Internal.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.STATUS}

Il valore testuale dell'evento causa.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.TAGS}

Un elenco di tag dell'evento causa separati da virgole.
Espanso in una stringa vuota se non esistono tag.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.TAGSJSON}

Un array JSON contenente oggetti di tag dell'evento causa.
Espanso in un array vuoto se non esistono tag.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.TAGS.<tag name>}

Il valore del tag dell'evento causa a cui si fa riferimento tramite il nome del tag.
 Un nome di tag contenente caratteri non alfanumerici (inclusi caratteri multibyte UTF non inglesi) deve essere racchiuso tra virgolette doppie. Le virgolette e le barre rovesciate all'interno di un nome di tag tra virgolette devono essere escape con una barra rovesciata.
 Usato nel contesto di un evento sintomo.

{EVENT.CAUSE.TIME}

L'ora dell'evento causa nel formato `hh:mm:ss`.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.TIMESTAMP}

Il timestamp UNIX dell'evento causa.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.UPDATE.HISTORY}

Il log degli aggiornamenti del problema causa (riconoscimenti, ecc).
 Utilizzato nel contesto di un evento sintomo.

{EVENT.CAUSE.VALUE}

Il valore numerico dell'evento causa.
 Valori possibili: *1* - Problema, *0* - In recupero.
 Utilizzato nel contesto di un evento sintomo.

{EVENT.SYMPTOMS}

L'elenco degli eventi sintomo. Include i seguenti dettagli: nome dell'host, nome dell'evento, gravità, età, tag del servizio e valori.
 Questa macro viene usata nel contesto dell'evento causa e restituisce informazioni sugli eventi sintomo.

Macro evento di ripristino

{EVENT.RECOVERY.DATE}

La data dell'evento di ripristino nel formato `yyyy.mm.dd`.

{EVENT.RECOVERY.ID}

L'ID numerico dell'evento di ripristino.

{EVENT.RECOVERY.NAME}

Il nome dell'evento di ripristino.

{EVENT.RECOVERY.STATUS}

Il valore testuale dell'evento di ripristino.

{EVENT.RECOVERY.TAGS}

Un elenco di tag dell'evento di ripristino separati da virgole. Espanso in una stringa vuota se non esistono tag.

{EVENT.RECOVERY.TAGSJSON}

Un array JSON contenente gli oggetti dei tag dell'evento di ripristino. Espanso in un array vuoto se non esistono tag.

{EVENT.RECOVERY.TIME}

L'ora dell'evento di ripristino nel formato `hh:mm:ss`.

{EVENT.RECOVERY.TIMESTAMP}

Il timestamp UNIX dell'evento di ripristino.

{EVENT.RECOVERY.VALUE}

Il valore numerico dell'evento di ripristino.

Macro di aggiornamento evento

{EVENT.UPDATE.ACTION}

Nome leggibile dell'azione o delle azioni eseguite durante l'aggiornamento del problema.
 Si risolve nei seguenti valori: *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* e *closed* (a seconda di quante azioni vengono eseguite in un singolo aggiornamento).

{EVENT.UPDATE.ACTIONJSON}

Un array JSON contenente i dettagli dell'azione o delle azioni eseguite durante l'aggiornamento del problema.
 Possibili valori delle proprietà JSON:
- true (per le proprietà `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` e `symptom`);
- <stringa del messaggio> (per la proprietà `message`);
- timestamp (per le proprietà `suppress_until` e `timestamp`) oppure 0 (per la proprietà `suppress_until` se la soppressione è indefinita);
- 0, 1, 2, 3, 4, 5 (per le proprietà di severità `old` e `new`).

 Ad esempio: `{"acknowledge":true,"message":"Manutenzione mensile.","severity":{"old":2,"new":1},"`

{EVENT.UPDATE.DATE}

La data dell'aggiornamento dell'evento (conferma, ecc.) nel formato `yyyy.mm.dd`.

{EVENT.UPDATE.HISTORY}

Il log degli aggiornamenti del problema (acknowledgments, ecc.).

{EVENT.UPDATE.MESSAGE}

Il messaggio di aggiornamento del problema.

{EVENT.UPDATE.NSEVERITY}

Il valore numerico della nuova severity dell'evento impostata durante l'operazione di aggiornamento del problema.

{EVENT.UPDATE.SEVERITY}

Il nome della nuova gravità dell'evento impostata durante l'operazione di aggiornamento del problema.

{EVENT.UPDATE.STATUS}

Il valore numerico dello stato di aggiornamento del problema.
 Valori possibili: *0* - Il webhook è stato chiamato a causa di un evento di problema/ripristino, *1* - Operazione di aggiornamento.

{EVENT.UPDATE.TIME}

L'ora dell'aggiornamento dell'evento (conferma, ecc.) nel formato `hh:mm:ss`.

{EVENT.UPDATE.TIMESTAMP}

Il timestamp UNIX dell'aggiornamento dell'evento (conferma, ecc).

Macro di funzione

{FUNCTION.VALUE}

Il valore della funzione basata su item N-esima nell'espressione del trigger al momento dell'evento.
 Vengono conteggiate solo le funzioni con `/host/key` come primo parametro.

 Questa macro può essere usata con un indice numerico, ad esempio {FUNCTION.VALUE<1-9>} per indicare la prima, seconda, terza, ecc. funzione in un'espressione del trigger. Vedi indexed macros.

{FUNCTION.RECOVERY.VALUE}

Il valore della N-esima funzione basata su item nell'espressione di recupero al momento dell'evento.*
 Vengono conteggiate solo le funzioni con `/host/key` come primo parametro.

 Questa macro può essere usata con un indice numerico, ad esempio {FUNCTION.RECOVERY.VALUE<1-9>} per indicare la prima, seconda, terza, ecc. funzione in un'espressione di trigger. Vedi indexed macros.

Macro host

{HOST.CONN}

L'indirizzo IP o il nome DNS dell'host, a seconda delle impostazioni dell'host.
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web, la macro verrà risolta nell'interfaccia agent principale. Se non è definita un'interfaccia agent, verrà utilizzata l'interfaccia SNMP principale. Se non è definita nemmeno un'interfaccia SNMP, verrà utilizzata l'interfaccia JMX principale. Se non è definita neppure un'interfaccia JMX, verrà utilizzata l'interfaccia IPMI principale. Se l'host non ha alcuna interfaccia, la macro viene risolta in 'UNKNOWN'.

 Questa macro può essere usata con un indice numerico come {HOST.CONN<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{HOST.DESCRIPTION}

La descrizione dell'host.

 Questa macro può essere utilizzata con un indice numerico come {HOST.DESCRIPTION<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{HOST.DNS}

Il nome DNS dell'host.
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web, la macro verrà risolta nell'interfaccia agent principale. Se non è definita un'interfaccia agent, verrà utilizzata l'interfaccia SNMP principale. Se non è definita neppure un'interfaccia SNMP, verrà utilizzata l'interfaccia JMX principale. Se non è definita un'interfaccia JMX, verrà utilizzata l'interfaccia IPMI principale. Se l'host non ha alcuna interfaccia, la macro viene risolta in 'UNKNOWN'.

 Questa macro può essere usata con un indice numerico come {HOST.DNS<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione trigger. Vedere indexed macros.

{HOST.HOST}

Il nome tecnico dell'host.
 Le funzioni macro non sono supportate per questa macro se viene usata come segnaposto nel primo parametro di una funzione di cronologia, ad esempio, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Questa macro può essere usata con un indice numerico come {HOST.HOST<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere le macro indicizzate.

{HOST.ID}

L'ID dell'host.

 Questa macro può essere usata con un indice numerico come {HOST.ID<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{HOST.IP}

L'indirizzo IP dell'host.
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web, la macro verrà risolta in base all'interfaccia agent principale. Se non è definita un'interfaccia agent, verrà utilizzata l'interfaccia SNMP principale. Se non è definita neppure un'interfaccia SNMP, verrà utilizzata l'interfaccia JMX principale. Se non è definita nemmeno un'interfaccia JMX, verrà utilizzata l'interfaccia IPMI principale. Se l'host non dispone di alcuna interfaccia, la macro viene risolta in 'UNKNOWN'.

 Questa macro può essere usata con un indice numerico come {HOST.IP<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{HOST.METADATA}

I metadati dell'host.

 Utilizzati solo per l'autoregistrazione attiva dell'agent.

{HOST.NAME}

Il nome host visibile.

 Questa macro può essere usata con un indice numerico come {HOST.NAME<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{HOST.PORT}

La porta dell'host (agent).
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web, la macro verrà risolta in base all'interfaccia agent principale. Se non è definita un'interfaccia agent, verrà utilizzata l'interfaccia SNMP principale. Se non è definita neanche un'interfaccia SNMP, verrà utilizzata l'interfaccia JMX principale. Se non è definita nemmeno un'interfaccia JMX, verrà utilizzata l'interfaccia IPMI principale. Se l'host non ha alcuna interfaccia, la macro viene risolta in 'UNKNOWN'.

 Questa macro può essere usata con un indice numerico come {HOST.PORT<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

Macro host di destinazione

{HOST.TARGET.CONN}

L'indirizzo IP o il nome DNS del host di destinazione.

{HOST.TARGET.DNS}

Il nome DNS del host di destinazione.

{HOST.TARGET.HOST}

Il nome tecnico del host di destinazione.

{HOST.TARGET.IP}

L'indirizzo IP del host di destinazione.

{HOST.TARGET.NAME}

Il nome visibile del host di destinazione.

Macro dei gruppi host

{HOSTGROUP.ID}

L'ID del gruppo host.

Macro dell'inventario host

{INVENTORY.ALIAS}

Il campo *Alias* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.ALIAS<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.ASSET.TAG}

Il campo *Asset tag* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.ASSET.TAG<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.CHASSIS}

Il campo *Chassis* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.CHASSIS<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.CONTACT}

Il campo *Contact* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.CONTACT<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.CONTRACT.NUMBER}

Il campo *Numero contratto* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.CONTRACT.NUMBER<1-9>}, per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.DEPLOYMENT.STATUS}

Il campo *Deployment status* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.DEPLOYMENT.STATUS<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.HARDWARE}

Il campo *Hardware* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HARDWARE<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.HARDWARE.FULL}

Il campo *Hardware (dettagli completi)* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HARDWARE.FULL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.HOST.NETMASK}

Il campo *Subnet mask dell'host* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HOST.NETMASK<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.HOST.NETWORKS}

Il campo *Host networks* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HOST.NETWORKS<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.HOST.ROUTER}

Il campo *Host router* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HOST.ROUTER<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.HW.ARCH}

Il campo *Architettura hardware* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HW.ARCH<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.HW.DATE.DECOMM}

Il campo *Date hardware decommissioned* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HW.DATE.DECOMM<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.HW.DATE.EXPIRY}

Il campo *Data di scadenza della manutenzione hardware* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HW.DATE.EXPIRY<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.HW.DATE.INSTALL}

Il campo *Data di installazione dell'hardware* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HW.DATE.INSTALL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.HW.DATE.PURCHASE}

Il campo *Data di acquisto dell'hardware* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.HW.DATE.PURCHASE<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.INSTALLER.NAME}

Il campo *Installer name* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.INSTALLER.NAME<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.LOCATION}

Il campo *Location* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.LOCATION<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.LOCATION.LAT}

Il campo *Latitudine della posizione* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.LOCATION.LAT<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.LOCATION.LON}

Il campo *Longitudine della posizione* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.LOCATION.LON<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.MACADDRESS.A}

Il campo *MAC address A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.MACADDRESS.A<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.MACADDRESS.B}

Il campo *MAC address B* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.MACADDRESS.B<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.MODEL}

Il campo *Model* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.MODEL<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.NAME}

Il campo *Name* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.NAME<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.NOTES}

Il campo *Notes* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.NOTES<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.OOB.IP}

Il campo *Indirizzo IP OOB* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.OOB.IP<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.OOB.NETMASK}

Il campo *maschera di sottorete OOB* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.OOB.NETMASK<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.OOB.ROUTER}

Il campo *OOB router* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.OOB.ROUTER<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.OS}

Il campo *OS* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.OS<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.OS.FULL}

Il campo *OS (dettagli completi)* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.OS.FULL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.OS.SHORT}

Il campo *OS (breve)* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.OS.SHORT<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.POC.PRIMARY.CELL}

Il campo *Primary POC cell* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.CELL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.POC.PRIMARY.EMAIL}

Il campo *Primary POC email* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.EMAIL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.POC.PRIMARY.NAME}

Il campo *Nome del referente principale* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.NAME<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.NOTES}

Il campo *Note del POC principale* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.NOTES<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.PHONE.A}

Il campo *Primary POC phone A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}, per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.POC.PRIMARY.PHONE.B}

Il campo *Primary POC phone B* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.PHONE.B<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.POC.PRIMARY.SCREEN}

Il campo *Nome schermata POC principale* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.SCREEN<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.CELL}

Il campo *Cellulare POC secondario* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.CELL<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.POC.SECONDARY.EMAIL}

Il campo *Email secondaria del referente* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.EMAIL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.NAME}

Il campo *Nome del referente secondario* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.NAME<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.POC.SECONDARY.NOTES}

Il campo *Note del referente secondario* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.NOTES<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.PHONE.A}

Il campo *Telefono secondario del referente A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.PHONE.A<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.PHONE.B}

Il campo *Telefono secondario del referente B* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.PHONE.B<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.POC.SECONDARY.SCREEN}

Il campo *Nome dello schermo del POC secondario* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.SCREEN<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SERIALNO.A}

Il campo *Numero di serie A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SERIALNO.A<1-9>}, per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.SERIALNO.B}

Il campo *Numero di serie B* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SERIALNO.B<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.SITE.ADDRESS.A}

Il campo *Indirizzo sito A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.ADDRESS.A<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ADDRESS.B}

Il campo *Indirizzo sito B* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.ADDRESS.B<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ADDRESS.C}

Il campo *Site address C* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.ADDRESS.C<1-9>}, per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.SITE.CITY}

Il campo *Città del sito* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.CITY<1-9>}, per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.SITE.COUNTRY}

Il campo *Paese del sito* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.COUNTRY<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.NOTES}

Il campo *Note del sito* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.NOTES<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.RACK}

Il campo *Posizione del rack del sito* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.RACK<1-9>}, per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.SITE.STATE}

Il campo *Stato/provincia del sito* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.STATE<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ZIP}

Il campo *CAP/postale della sede* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SITE.ZIP<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE}

Il campo *Software* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.SOFTWARE.APP.A}

Il campo *Software application A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.A<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.SOFTWARE.APP.B}

Il campo *Software application B* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.B<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.SOFTWARE.APP.C}

Il campo *Software application C* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.C<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.SOFTWARE.APP.D}

Il campo *Software application D* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.D<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.SOFTWARE.APP.E}

Il campo *Software application E* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.E<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.SOFTWARE.FULL}

Il campo *Software (dettagli completi)* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.FULL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.TAG}

Il campo *Tag* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.TAG<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.TYPE}

Il campo *Type* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.TYPE<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

{INVENTORY.TYPE.FULL}

Il campo *Tipo (dettagli completi)* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.TYPE.FULL<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.URL.A}

Il campo *URL A* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.URL.A<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.URL.B}

Il campo *URL B* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.URL.B<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedi macro indicizzate.

{INVENTORY.URL.C}

Il campo *URL C* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.URL.C<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere indexed macros.

{INVENTORY.VENDOR}

Il campo *Vendor* nell'inventario dell'host.

 Questa macro può essere usata con un indice numerico, ad esempio {INVENTORY.VENDOR<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedi indexed macros.

Macro degli item

{ITEM.DESCRIPTION}

La descrizione dell'Nth item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere usata con un indice numerico come {ITEM.DESCRIPTION<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione del trigger. Vedere indexed macros.

{ITEM.DESCRIPTION.ORIG}

La descrizione originale (con macro non risolte) dell'N-esimo item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere usata con un indice numerico come {ITEM.DESCRIPTION.ORIG<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedi indexed macros.

{ITEM.ID}

L'ID numerico dell'Nth item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere usata con un indice numerico come {ITEM.ID<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedi indexed macros.

{ITEM.KEY}

La chiave dell'Nth item nell'espressione del trigger che ha causato una notifica.
 Le funzioni macro non sono supportate per questa macro se viene usata come segnaposto nel primo parametro di una funzione di cronologia, ad esempio, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Questa macro può essere usata con un indice numerico come {ITEM.KEY<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedi macro indicizzate.

{ITEM.KEY.ORIG}

La chiave originale (con i macro non risolti) dell'N-esimo item nell'espressione del trigger che ha causato una notifica.

 Questo macro può essere usato con un indice numerico come {ITEM.KEY.ORIG<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedi indexed macros.

{ITEM.LASTVALUE}

L'ultimo valore del N-esimo item nell'espressione del trigger che ha causato una notifica.
 Si risolverà in *UNKNOWN* nel frontend se l'ultimo valore della cronologia è stato raccolto più di *Max history display period* tempo fa (impostato nella sezione di menu *Administration > General*).
 Quando viene usato nel nome del problema, la macro non si risolverà nell'ultimo valore dell'item quando si visualizzano gli eventi del problema; invece, manterrà il valore dell'item dal momento in cui si è verificato il problema.
Quando viene usata nelle notifiche, in alcuni casi la macro potrebbe non risolversi nell'ultimo valore dell'item al momento in cui il trigger è scattato. Ad esempio, se un item riceve rapidamente due valori, "A" e "B", e il trigger scatta per "A", le notifiche potrebbero mostrare "B" come ultimo valore a causa di un lieve ritardo di elaborazione - l'ultimo valore dell'item è cambiato tra il momento in cui il trigger è scattato e quando la notifica è stata creata. Per evitare questo, puoi usare la macro {ITEM.VALUE}, che si risolve nel valore al momento in cui il trigger scatta, garantendo che nella notifica venga usato il valore corretto.
 È un alias di `last(/{HOST.HOST}/{ITEM.KEY})`.
 Il valore risolto per gli item di tipo text/log viene troncato a 20 caratteri dal frontend nelle seguenti posizioni:
- Dati operativi;
- Descrizione del trigger;
- URL del trigger;
- Etichette URL del trigger;
- Descrizione del widget del valore dell'item.
Per risolvere a un valore completo, puoi usare macro functions, poiché nessun valore viene troncato dal server. Ad esempio: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LASTVALUE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedi indexed macros.

{ITEM.LASTVALUE.AGE}

Il tempo trascorso tra l'ultima raccolta del valore dell'item e la valutazione della macro.
 Visualizzato in un formato leggibile dall'uomo (ad esempio 1m 45s).

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LASTVALUE.AGE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.LASTVALUE.DATE}

La data in cui è stato raccolto l'ultimo valore dell'item.
 Visualizzata nel formato `YYYYMMDD`.
 Il fuso orario del server viene utilizzato nei nomi dei trigger (nell'elenco *Monitoring* > *Problems*), nei nomi degli eventi e nei nomi e valori dei tag. In tutti gli altri casi viene utilizzato il fuso orario dell'utente.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LASTVALUE.DATE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.LASTVALUE.TIME}

L'ora in cui è stato raccolto l'ultimo valore dell'item.
 Visualizzato nel formato `HHMMSS`.
 Il fuso orario del server viene utilizzato nei nomi dei trigger nell'elenco *Monitoring* > *Problems*, nei nomi degli eventi e nei nomi e valori dei tag. In tutti gli altri casi viene utilizzato il fuso orario dell'utente.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LASTVALUE.TIME<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere indexed macros.

{ITEM.LASTVALUE.TIMESTAMP}

Il timestamp UNIX in cui è stato raccolto l'ultimo valore dell'item.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LASTVALUE.TIMESTAMP<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.LOG.AGE}

L’età dell’evento di log.
Con precisione fino al secondo.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.AGE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedere indexed macros.

{ITEM.LOG.DATE}

La data in cui la voce di log è stata scritta nel log.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.DATE<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un’espressione di trigger. Vedi indexed macros.

{ITEM.LOG.EVENTID}

L’ID dell’evento nel registro eventi.
 Solo per il monitoraggio del registro eventi di Windows.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.EVENTID<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedi indexed macros.

{ITEM.LOG.NSEVERITY}

La gravità numerica dell’evento nel registro eventi.
 Solo per il monitoraggio del registro eventi di Windows.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.NSEVERITY<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un’espressione di trigger. Vedere indexed macros.

{ITEM.LOG.SEVERITY}

La gravità verbale dell’evento nel registro eventi.
 Solo per il monitoraggio del registro eventi di Windows.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.SEVERITY<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedere indexed macros.

{ITEM.LOG.SOURCE}

La sorgente dell’evento nel registro eventi.
 Solo per il monitoraggio del registro eventi di Windows.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.SOURCE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedi indexed macros.

{ITEM.LOG.TIME}

L’ora in cui la voce di log è stata scritta nel log.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.LOG.TIME<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedi indexed macros.

{ITEM.LOG.TIMESTAMP}

Il timestamp UNIX in cui la voce di log è stata scritta nel log.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.TIMESTAMP<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un’espressione di trigger. Vedere indexed macros.

{ITEM.NAME}

Il nome dell’item con tutte le macro risolte.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.NAME<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedere indexed macros.

{ITEM.NAME.ORIG}

Il nome originale (con i macro non risolti) dell’item.

 Questo macro può essere usato con un indice numerico, ad esempio {ITEM.NAME.ORIG<1-9>} per indicare il primo, secondo, terzo, ecc. item in un’espressione di trigger. Vedi indexed macros.

{ITEM.STATE}

Lo stato più recente dell’Nth item nell’espressione del trigger che ha causato una notifica.
 Valori possibili: *Not supported*, *Normal*.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.STATE<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un’espressione del trigger. Vedere indexed macros.

{ITEM.STATE.ERROR}

Il messaggio di errore con i dettagli del motivo per cui un item è diventato non supportato.
 Se un item passa allo stato non supportato e poi torna immediatamente a essere supportato, il campo dell’errore può essere vuoto.

{ITEM.VALUE}

Si risolve in uno dei seguenti modi:

- Il valore storico (al momento dell’evento) del N-esimo item nell’espressione del trigger, se usato nel contesto della modifica dello stato del trigger, ad esempio durante la visualizzazione degli eventi o l’invio delle notifiche;
- L’ultimo valore del N-esimo item nell’espressione del trigger, se usato senza il contesto della modifica dello stato del trigger, ad esempio durante la visualizzazione di un elenco di trigger in una finestra di selezione a comparsa. In questo caso funziona allo stesso modo di {ITEM.LASTVALUE}.

Nel primo caso si risolverà in *UNKNOWN* se il valore storico è già stato eliminato o non è mai stato memorizzato.
Nel secondo caso, e solo nel frontend, si risolverà in *UNKNOWN* se l'ultimo valore storico è stato raccolto più di *Max history display period* tempo fa (impostato nella sezione di menu *Administration > General*).
Il valore risolto per gli item di tipo text/log viene troncato a 20 caratteri dal frontend nelle seguenti posizioni:
- Dati operativi;
- Descrizione del trigger;
- URL del trigger;
- Etichette URL del trigger;
- Descrizione del widget del valore dell'item.
Per ottenere il valore completo, è possibile usare macro functions, poiché nessun valore viene troncato dal server. Ad esempio: `{{ITEM.VALUE}.regsub("(.*)", \1)}`.

Questa macro può essere usata con un indice numerico, ad esempio {ITEM.VALUE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere indexed macros.

{ITEM.VALUE.AGE}

Il tempo trascorso tra la raccolta del valore dell'item e la valutazione della macro.
 Visualizzato in un formato leggibile dall'utente (ad es. 1m 45s).

 Questa macro può essere usata con un indice numerico, ad es. {ITEM.VALUE.AGE<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.VALUE.DATE}

La data in cui è stato raccolto il valore dell'item.
 Visualizzata nel formato `YYYYMMDD`.
 Il fuso orario del server viene utilizzato nei nomi dei trigger (nell'elenco *Monitoring* > *Problems*), nei nomi degli eventi e nei nomi e valori dei tag. In tutti gli altri casi viene utilizzato il fuso orario dell'utente.

Questa macro può essere usata con un indice numerico, ad esempio {ITEM.VALUE.DATE<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.VALUE.TIME}

L'ora in cui è stato raccolto il valore dell'item.
 Visualizzato nel formato `HHMMSS`.
 Il fuso orario del server viene utilizzato nei nomi dei trigger (nell'elenco *Monitoring* > *Problems*), nei nomi degli eventi e nei nomi e valori dei tag. In tutti gli altri casi viene utilizzato il fuso orario dell'utente.

Questa macro può essere usata con un indice numerico, ad esempio {ITEM.VALUE.TIME<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.VALUE.TIMESTAMP}

Il timestamp UNIX in cui è stato raccolto il valore dell'item.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.VALUE.TIMESTAMP<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un'espressione di trigger. Vedere indexed macros.

{ITEM.VALUETYPE}

Il tipo di valore dell'item N-esimo nell'espressione del trigger che ha causato una notifica.
 Valori possibili: *0* - Numerico float, *1* - Carattere, *2* - Log, *3* - Numerico senza segno, *4* - Testo.

 Questa macro può essere usata con un indice numerico, ad esempio {ITEM.VALUETYPE<1-9>} per indicare il primo, il secondo, il terzo, ecc. item in un'espressione del trigger. Vedere indexed macros.

Macro di notifica del low-level discovery

{LLDRULE.DESCRIPTION}

La descrizione della regola di low-level discovery che ha causato una notifica.

{LLDRULE.DESCRIPTION.ORIG}

La descrizione originale (con le macro non risolte) della regola di low-level discovery che ha causato una notifica.

{LLDRULE.ID}

L'ID numerico della regola di low-level discovery che ha causato una notifica.

{LLDRULE.KEY}

La chiave della regola di low-level discovery che ha causato una notifica.

{LLDRULE.KEY.ORIG}

La chiave originale (con le macro non risolte) della regola di low-level discovery che ha causato una notifica.

{LLDRULE.NAME}

Il nome della regola di low-level discovery che ha causato una notifica.

{LLDRULE.NAME.ORIG}

Il nome originale (con le macro non risolte) della regola di low-level discovery che ha causato una notifica.

{LLDRULE.STATE}

Lo stato più recente della regola di discovery a basso livello.
 Valori possibili: *Not supported*, *Normal*.

 Questa macro può essere usata con un indice numerico, ad esempio {LLDRULE.STATE<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedi indexed macros.

{LLDRULE.STATE.ERROR}

Il messaggio di errore con i dettagli sul motivo per cui la regola di low-level discovery è diventata non supportata.
 Se una regola di low-level discovery entra nello stato non supportato e poi torna immediatamente supportata, il campo dell'errore può essere vuoto.

Macro delle mappe

{MAP.ID}

L'ID della mappa di rete.

{MAP.NAME}

Il nome della mappa di rete.

Macro del proxy

{PROXY.DESCRIPTION}

La descrizione del proxy.
 Si risolve in uno dei seguenti casi:

- Proxy dell'ennesimo item nell'espressione del trigger (nelle notifiche basate su trigger). Puoi usare qui macro indicizzate;
- Proxy che ha eseguito la discovery (nelle notifiche di discovery). Usa qui {PROXY.DESCRIPTION}, senza indicizzazione;
- Proxy a cui un agent attivo si è registrato (nelle notifiche di autoregistrazione). Usa qui {PROXY.DESCRIPTION}, senza indicizzazione.

Questa macro può essere usata con un indice numerico, ad esempio {PROXY.DESCRIPTION<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedi le macro indicizzate.

{PROXY.NAME}

Il nome del proxy.
 Risolto in uno dei seguenti casi:

- Proxy del N-esimo item nell'espressione del trigger (nelle notifiche basate su trigger). Puoi usare qui macro indicizzate;
- Proxy che ha eseguito la discovery (nelle notifiche di discovery). Usa qui {PROXY.NAME}, senza indicizzazione;
- Proxy a cui un agent attivo si è registrato (nelle notifiche di autoregistrazione). Usa qui {PROXY.NAME}, senza indicizzazione.

Questa macro può essere usata con un indice numerico, ad esempio {PROXY.NAME<1-9>} per indicare il primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedi le macro indicizzate.

Macro di script

{MANUALINPUT}

Il valore di input manuale specificato dall'utente al momento dell'esecuzione dello script.

Macro di servizio

{SERVICE.DESCRIPTION}

La descrizione del servizio con le macro risolte.

{SERVICE.ID}

L'ID numerico del servizio che ha attivato un'azione.

{SERVICE.NAME}

Il nome del servizio con le macro risolte.

{SERVICE.ROOTCAUSE}

L'elenco degli eventi problema dei trigger che hanno causato il fallimento di un servizio, ordinato per gravità e nome host.
 Include i seguenti dettagli: nome host, nome evento, gravità, età, tag del servizio e valori.

{SERVICE.TAGS}

Un elenco di tag degli eventi del servizio separati da virgole.
 I tag degli eventi del servizio possono essere definiti nella sezione dei tag della configurazione del servizio. Espanso in una stringa vuota se non esistono tag.

{SERVICE.TAGSJSON}

Un array JSON contenente oggetti tag degli eventi del servizio.
 I tag degli eventi del servizio possono essere definiti nella sezione dei tag della configurazione del servizio. Espanso in un array vuoto se non esistono tag.

{SERVICE.TAGS.<tag name>}

Il valore del tag dell'evento del servizio a cui si fa riferimento tramite il nome del tag.
 I tag dell'evento del servizio possono essere definiti nella sezione dei tag della configurazione del servizio.
 Un nome di tag contenente caratteri non alfanumerici (inclusi caratteri multibyte UTF non inglesi) deve essere racchiuso tra virgolette doppie. Le virgolette e le barre rovesciate all'interno di un nome di tag tra virgolette devono essere escape con una barra rovesciata.

Macro dei trigger

{TRIGGER.DESCRIPTION}

La descrizione del trigger.
 Tutte le macro supportate nella descrizione di un trigger verranno espanse se `{TRIGGER.DESCRIPTION}` viene utilizzato nel testo della notifica.

{TRIGGER.EVENTS.ACK}

Il numero di eventi confermati per un elemento della mappa nelle mappe, oppure per il trigger che ha generato l'evento corrente nelle notifiche.

{TRIGGER.EVENTS.PROBLEM.ACK}

Il numero di eventi problema confermati per tutti i trigger, indipendentemente dal loro stato.

{TRIGGER.EVENTS.PROBLEM.UNACK}

Il numero di eventi di problema non riconosciuti per tutti i trigger, indipendentemente dal loro stato.

{TRIGGER.EVENTS.UNACK}

Il numero di eventi non riconosciuti per un elemento della mappa nelle mappe, oppure per il trigger che ha generato l'evento corrente nelle notifiche.

{TRIGGER.EXPRESSION}

L'espressione del trigger.

{TRIGGER.EXPRESSION.EXPLAIN}

Un'espressione del trigger valutata parzialmente.
 Le funzioni basate su item vengono valutate e sostituite con i risultati al momento della generazione dell'evento, mentre tutte le altre funzioni vengono visualizzate così come sono scritte nell'espressione. Può essere usata per il debug delle espressioni del trigger.

{TRIGGER.EXPRESSION.RECOVERY}

L'espressione di ripristino del trigger se *Generazione eventi OK* nella configurazione del trigger è impostata su "Espressione di ripristino"; altrimenti viene restituita una stringa vuota.

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

Un'espressione di ripristino del trigger valutata parzialmente.
 Le funzioni basate su item vengono valutate e sostituite dai risultati al momento della generazione dell'evento, mentre tutte le altre funzioni vengono visualizzate così come sono scritte nell'espressione. Può essere usata per il debug delle espressioni di ripristino del trigger.

{TRIGGER.HOSTGROUP.NAME}

Un elenco ordinato (in base alla query SQL), separato da virgola e spazio, dei gruppi di host in cui è definito il trigger.

{TRIGGER.ID}

L'ID numerico del trigger che ha attivato questa azione.

{TRIGGER.NAME}

Il nome del trigger con le macro risolte.
 Si noti che `{EVENT.NAME}` può essere utilizzato nelle azioni per visualizzare il nome dell'evento/problema attivato con le macro risolte.

{TRIGGER.NAME.ORIG}

Il nome originale del trigger (con le macro non risolte).

{TRIGGER.NSEVERITY}

La gravità numerica del trigger.
 Valori possibili: *0* - Non classificato, *1* - Informazione, *2* - Avviso, *3* - Medio, *4* - Alto, *5* - Disastro.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

Il numero di eventi di problema confermati per i trigger nello stato di problema.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

Il numero di eventi di problema non confermati per i trigger nello stato di problema.

{TRIGGER.SEVERITY}

Il nome della gravità del trigger.
 Può essere definito in *Administration > General > Trigger displaying options*.

{TRIGGER.STATE}

L'ultimo stato dell'espressione del trigger.
 Valori possibili: *Sconosciuto*, *Normale*.

{TRIGGER.STATE.ERROR}

Il messaggio di errore con i dettagli del motivo per cui un trigger è diventato non supportato.
 Se un trigger entra nello stato non supportato e poi torna immediatamente di nuovo supportato, il campo errore può essere vuoto.

{TRIGGER.STATUS}

Il valore del trigger al momento dell'esecuzione del passo dell'operazione.
 Valori possibili: *Problem*, *OK*.

{TRIGGER.TEMPLATE.NAME}

Un elenco ordinato (in base alla query SQL), separato da virgola e spazio, dei template in cui il trigger è definito, oppure *UNKNOWN* se il trigger è definito in un host.

{TRIGGER.URL}

L'URL del trigger.

{TRIGGER.URL.NAME}

L'etichetta per l'URL del trigger.

{TRIGGER.VALUE}

Il valore numerico corrente del trigger.
 Valori possibili: *0* - il trigger è nello stato OK, *1* - il trigger è nello stato Problema.

{TRIGGERS.ACK}

Il numero di trigger confermati per un elemento della mappa, indipendentemente dallo stato del trigger.
 Un trigger è considerato confermato se tutti i suoi eventi di problema sono confermati.

{TRIGGERS.PROBLEM.ACK}

Il numero di trigger di problema confermati per un elemento della mappa.
 Un trigger è considerato confermato se tutti i suoi eventi di problema sono confermati.

{TRIGGERS.PROBLEM.UNACK}

Il numero di trigger di problema non confermati per un elemento della mappa.
 Un trigger è considerato non confermato se almeno uno dei suoi eventi di problema non è confermato.

{TRIGGERS.UNACK}

Il numero di trigger non riconosciuti per un elemento della mappa, indipendentemente dallo stato del trigger.
 Un trigger è considerato non riconosciuto se almeno uno dei suoi eventi di problema non è riconosciuto.

Macro nome utente

{USER.FULLNAME}

Il nome, il cognome e il nome utente dell'utente che ha aggiunto il riconoscimento dell'evento o avviato lo script.

{USER.NAME}

Il nome dell'utente che ha avviato lo script.

{USER.SURNAME}

Il cognome dell'utente che ha avviato lo script.

{USER.USERNAME}

Il nome utente dell'utente che ha avviato lo script.

Macro indicizzate

La sintassi delle macro indicizzate **{MACRO<1-9>}** può fare riferimento solo all'ennesimo item, funzione o host nel campo *Expression* di un trigger:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** si risolvono negli indirizzi IP del primo, secondo e terzo host nell'espressione del trigger (se presenti).

- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** si risolvono nei valori del primo, secondo e terzo item nell'espressione del trigger al momento dell'evento (se presenti).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** si risolvono nei valori della prima, seconda e terza funzione basata su item al momento dell'evento (se presenti).

Nel contesto dei trigger, le macro indicizzate fanno sempre riferimento al campo *Expression* della configurazione del trigger, non alla *Recovery expression*. Ad esempio, in un evento di recupero, **{ITEM.VALUE2}** si risolverà nel valore del secondo item dell'espressione del problema al momento del recupero.

La macro **{HOST.HOST<1-9>}** è supportata anche all'interno della macro di espressione `{?func(/host/key,param)}` nei **nomi dei graph**. Ad esempio, `{?func(/{HOST.HOST2}/key,param)}` in un nome di graph si risolverà nell'host del secondo item nel graph.

Warning:
Le macro indicizzate non si risolveranno in nessun altro contesto oltre ai casi menzionati qui. Per altri contesti, usa macro senza indice (**{HOST.HOST}, {HOST.IP}**, ecc.).

Macro utente supportate dalla posizione

Panoramica

Questa sezione contiene un elenco di posizioni in cui sono supportate le macro definibili dall'utente.

Note:
Per *Azioni*, *Rilevamento di rete*, *Proxy* e tutte le posizioni elencate nella sezione *Altre posizioni* di questa pagina sono supportate solo le macro utente a livello globale. Nelle posizioni menzionate, le macro a livello di host e di template non verranno risolte.

Note:
Per personalizzare i valori delle macro (ad esempio, abbreviare o estrarre sottostringhe specifiche), è possibile utilizzare le funzioni macro.

Azioni

Nelle azioni, le macro utente possono essere utilizzate nei seguenti campi:

Posizione		Macro multiple/miste con testo[1]
Notifiche e comandi basati su trigger		sì
Notifiche interne basate su trigger		sì
Notifiche di aggiornamento problema		sì
Notifiche e comandi basati su servizio		sì
Notifiche di aggiornamento servizio		sì
Condizione del periodo di tempo		no
Operazioni		
	Durata predefinita del passo dell'operazione	no
	Durata del passo	no

Host/prototipi host

Nella configurazione di un host e di un prototipo host, le macro utente possono essere utilizzate nei seguenti campi:

Posizione		Macro multiple/miste con testo[1]
IP/DNS dell'interfaccia		solo DNS
Porta dell'interfaccia		no
SNMP v1, v2		
	Community SNMP	yes
SNMP v3		
	Nome del contesto	yes
	Nome di sicurezza	yes
	Passphrase di autenticazione	yes
	Passphrase di privacy	yes

Posizione		Macro multiple/miste con testo[1]
IPMI		
	Nome utente	yes
	Password	yes
Tag[2]		
	Nomi dei tag	yes
	Valori dei tag	yes

Item / prototipi di item

Nella configurazione di un item o di un prototipo di item, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Più macro/miscela con testo[1]
Nome dell'item	sì
Parametri della chiave dell'item	sì[4]
Intervallo di aggiornamento	no
Intervalli personalizzati	no
Timeout (disponibile per i tipi di item supportati)	no
Conserva fino a (per cronologia e trend)	no
Descrizione	sì
Item calcolato/aggregato	
Formula (costanti delle espressioni e parametri delle funzioni; parametri della chiave dell'item; (*solo per gli item aggregati*) condizioni di filtro (nome del gruppo di host e nome del tag))	sì

Posizione	Più macro/miscela con testo[1]
Monitoraggio del database	
Nome utente	sì
Password	sì
Query SQL	sì
HTTP agent	
URL[3]	sì
Campi della query	sì
Corpo della richiesta	sì
Header (nomi e valori)	sì
Codici di stato richiesti	sì
Nome utente per l'autenticazione HTTP	sì
Password per l'autenticazione HTTP	sì
File del certificato SSl	sì
File della chiave SSl	sì
Password della chiave SSl	sì
Host consentiti	sì
JMX agent	
Endpoint JMX	sì
Item script	
Nomi e valori dei parametri	sì
Item browser	
Nomi e valori dei parametri	sì
Item query di teleme-tria	
Spostamento temporale	no
Limite di ricerca retrospettiva	no
Granularità	no
SNMP agent	
OID SNMP	sì
SSH agent	
Nome utente	sì
File della chiave pubblica	sì
File della chiave privata	sì
Password	sì
Script	sì
TELNET agent	
Nome utente	sì
Password	sì
Script	sì
Zabbix trap-per	
Host consentiti	sì
Tag[2]	
Nomi dei tag	sì
Valori dei tag	sì

Posizione	Più macro/miscela con testo[1]
Passaggi di pre-pro-cess-ing	
Parametri (inclusi gli script personalizzati)	sì
Parametri personalizzati per la gestione degli errori (campi *Imposta valore su* e *Imposta errore su*)	sì

Low-level discovery

In una regola di low-level discovery rule, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Macro multiple/miste con testo[1]
Parametri della chi-ave	sì
Intervallo di ag-gior-na-mento	no
Intervallo per-son-al-iz-zato	no
Timeout (disponi-bile per i tipi di item sup-por-t-ati)	no
Elimina risorse perse	no
Disabilita risorse perse	no
Descrizione	sì
SNMP agent	
OID SNMP	sì
SSH agent	
Nome utente	sì
File di chiave pubblica	sì
File di chiave privata	sì
Password	sì

Posizione	Macro multiple/miste con testo[1]
Script	sì
TELNET agent	
Nome utente	sì
Password	sì
Script	sì
Zabbix trapper	
Host consentiti	sì
Database monitor	
Nome utente	sì
Password	sì
Query SQL	sì
JMX agent	
Endpoint JMX	sì
HTTP agent	
URL[3]	sì
Campi della query	sì
Corpo della richiesta	sì
Header (nomi e valori)	sì
Codici di stato richiesti	sì
Nome utente per l'autenticazione HTTP	sì
Password per l'autenticazione HTTP	sì
Filtri	
Espressione regolare	sì
Overrides	
Filtri: espressione regolare	sì
Operazioni: intervallo di aggiornamento (per i prototipi di item)	no
Operazioni: periodo di conservazione dello storico (per i prototipi di item)	no
Operazioni: periodo di conservazione dei trend (per i prototipi di item)	no

Individuazione della rete

In una regola di individuazione della rete, le macro utente possono essere utilizzate nei seguenti campi:

Posizione		Macro multiple/miste con testo[1]
Intervallo di aggiornamento		no
SNMP v1, v2		
	Community SNMP	yes
	OID SNMP	yes
SNMP v3		
	Nome del contesto	yes
	Nome di sicurezza	yes
	Passphrase di autenticazione	yes
	Passphrase di privacy	yes
	OID SNMP	yes

Proxy

In una configurazione proxy, le macro utente possono essere utilizzate nei seguenti campi:

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (if the proxy belongs to a group)	no
Interface address and port (for passive proxy)	no
Timeouts for item types	no

Gruppi di proxy

In una configurazione di proxy group, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Più macro/mix con testo[1]
Periodo di failover	no
Numero minimo di proxy	no

Template

Nella configurazione di un template, le macro utente possono essere utilizzate nei seguenti campi:

Posizione		Macro multiple/miste con testo[1]
Tag[2]		
	Nomi dei tag	sì
	Valori dei tag	sì

Trigger

Nella configurazione di un trigger, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Macro multiple/miste con testo[1]
Nome	sì
Dati operativi	sì
Espressione (solo nelle costanti e nei parametri di funzione; le macro segrete non sono supportate)	sì
Tag per la corrispondenza	sì

Posizione	Macro multiple/miste con testo[1]
Nome della voce di menu	sì
URL della voce di menu[3]	sì
Descrizione	sì
Tag[2]	
Nomi dei tag	sì
Valori dei tag	sì

Scenario web

Nella configurazione di uno scenario web, le macro utente possono essere utilizzate nei seguenti campi:

Posizione		Macro multiple/miste con testo[1]
Nome		sì
Intervallo di aggiornamento		no
Agent		sì
Proxy HTTP		sì
Variabili (solo valori)		sì
Intestazioni (nomi e valori)		sì
Passi		
	Nome	sì
	URL[3]	sì
	Variabili (solo valori)	sì
	Intestazioni (nomi e valori)	sì
	Timeout	no
	Stringa richiesta	sì
	Codici di stato richiesti	no
Autenticazione		
	Utente	sì
	Password	sì
	Certificato SSL	sì
	File chiave SSL	sì
	Password della chiave SSL	sì
Tag[2]		
	Nomi dei tag	sì
	Valori dei tag	sì

Altri posizioni

Oltre alle posizioni elencate qui, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Macro multiple/mix con testo[1]
Script globali (URL, script, SSH, Telnet, IPMI), incluso il testo di conferma Webhook	yes
Script JavaScript	no
Nome del parametro dello script JavaScript	no
Valore del parametro dello script JavaScript	yes
Dashboard	
Colonna di tipo dati *Text* nel widget *Top hosts* della dashboard	yes
Parametro *Description* nel widget *Item value* e *Gauge* della dashboard	yes
Parametro *Text* dell'etichetta primaria/secondaria nel widget *Honeycomb* della dashboard	yes
Parametro *URL*[3] nel widget *URL* della dashboard	yes
Monitoring > *Problems*, *Hosts*, *Latest data*	
Colonna *Custom text*	yes
Data collection > *Templates*, *Hosts*	
Colonna *Custom text*	yes
Alerts > *Media types* > *Message templates*	
Oggetto	yes
Messaggio	yes

Posizione	Macro multiple/mix con testo[1]
Alerts > *Media types* > *Script*	
Parametri dello script	yes
Alerts > *Media types* > *Media type*	
Campi *Username* e *Password* per il media type *Email* (quando *Authentication* è impostato su ”Username and password”; macro segrete consigliate)	yes
Users > *Users* > *Media*	
Quando attivo	yes
Administration > *General* > *GUI*	
Orario di lavoro	no
Administration > *General* > *Timeouts*	
Timeout per i tipi di item	no
Administration > *General* > *Connectors*	
URL	yes
Username	yes
Password	yes
Bearer token	yes
Timeout	no
HTTP proxy	yes
File del certificato SSL	yes
File della chiave SSL	yes
Password della chiave SSL	yes

Per un elenco completo di tutte le macro supportate in Zabbix, vedere macro supportate.

Note a piè di pagina

[1] Se più macro in un campo o macro mescolate con testo non sono supportate per la posizione, una singola macro deve occupare l'intero campo.

[2] Le macro usate nei nomi e nei valori dei tag vengono risolte solo durante il processo di generazione dell'evento.

[3] Gli URL che contengono una macro segreta non funzioneranno, poiché la macro in essi contenuta verrà risolta come "******".

[4] Le macro non sono supportate nella chiave dell'item degli item Zabbix trapper.

Compatibilità delle versioni

agent supportati

Per essere compatibile con Zabbix 8.0, Zabbix agent non deve essere precedente alla versione 1.6 né successivo alla versione 8.0.

Potrebbe essere necessario rivedere la configurazione degli agent meno recenti, poiché alcuni parametri sono cambiati, ad esempio i parametri relativi al logging per le versioni precedenti alla 3.0.

Per sfruttare appieno le funzionalità più recenti, le metriche, le prestazioni migliorate e il ridotto utilizzo della memoria, utilizzare l'agent supportato più recente.

Note per Windows XP

- Su Windows XP a 32 bit, non utilizzare Zabbix agent più recenti della versione 6.0.x;
- Su Windows XP/Server 2003, non utilizzare template di agent più recenti di Zabbix 4.0.x. I template più recenti utilizzano contatori delle prestazioni in inglese, supportati solo a partire da Windows Vista/Server 2008.

Agent 2 supportati

I vecchi agent Zabbix 2 dalla versione 4.4 in poi sono compatibili con Zabbix 8.0; Zabbix agent 2 non deve essere più recente della versione 8.0.

Si noti che quando si utilizzano Zabbix agent 2 versioni 4.4 e 5.0, viene usato l'intervallo predefinito di 10 minuti per aggiornare gli item non supportati.

Per sfruttare appieno le funzionalità più recenti, le metriche, le prestazioni migliorate e il ridotto utilizzo della memoria, utilizzare l'ultima versione supportata di agent 2.

Proxy Zabbix supportati

Per una compatibilità completa con Zabbix 8.0, i proxy devono corrispondere alla versione principale del server. Solo i proxy Zabbix 8.0.x sono completamente compatibili con un server Zabbix 8.0.x.

I proxy obsoleti sono supportati parzialmente: possono ancora raccogliere dati ed eseguire script, ma non possono ricevere aggiornamenti di configurazione, come nuovi item.

In relazione a Zabbix server, i proxy possono essere:

- *Correnti* (proxy e server hanno la stessa versione principale);
- *Obsoleti* (la versione del proxy è precedente alla versione del server, ma è parzialmente supportata);
- *Non supportati* (la versione del proxy è precedente alla versione della precedente release LTS del server *oppure* la versione del proxy è successiva alla versione principale del server).

Esempi:

Versione del server	Versione del proxy *corrente*	Versione del proxy *obsoleta*	Versione del proxy *non supportata*
6.4	6.4	6.0, 6.2	Precedente alla 6.0; successiva alla 6.4
7.0	7.0	6.0, 6.2, 6.4	Precedente alla 6.0; successiva alla 7.0
7.2	7.2	7.0	Precedente alla 7.0; successiva alla 7.2
7.4	7.4	7.0	Precedente alla 7.0; successiva alla 7.4
8.0	8.0	7.0, 7.2, 7.4	Precedente alla 7.0; successiva alla 8.0

Funzionalità supportate dai proxy:

Versione del proxy	Aggiornamento dei dati	Aggiornamento della configurazione	Attività
Current	Sì	Sì	Sì
Outdated	Sì	No	Comandi remoti (ad es. script shell); Controlli immediati del valore dell'item (cioè, *Esegui ora*); Nota: i test di preprocessing con un valore reale non sono supportati.
Unsupported	No	No	No

Gli avvisi sull'utilizzo di versioni del demone Zabbix incompatibili sono registrati.

File XML supportati

I file XML non più vecchi della versione 1.8 sono supportati per l'importazione in Zabbix 8.0.

Attention:
Nel formato di esportazione XML, le dipendenze dei trigger sono memorizzate solo per nome. Se ci sono più trigger con lo stesso nome (ad esempio, con severità ed espressioni diverse) tra i quali è stata definita una dipendenza, non è possibile importarli. Tali dipendenze devono essere rimosse manualmente dal file XML e aggiunte nuovamente dopo l'importazione.

Sintassi del periodo di tempo

Panoramica

Per impostare un intervallo di tempo, deve essere utilizzato il seguente formato:

`d-d,hh:mm-hh:mm`

dove i simboli indicano quanto segue:

Symbol	Description
d	Giorno della settimana: 1 - lunedì, 2 - martedì, ... , 7 - domenica
hh	Ore: 00-24
mm	Minuti: 00-59

È possibile specificare più di un intervallo di tempo utilizzando un separatore punto e virgola (;):

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

Lasciare vuoto l'intervallo di tempo equivale a 1-7,00:00-24:00, che è il valore predefinito.

Attention:
Il limite superiore di un intervallo di tempo non è incluso. Pertanto, se si specifica 09:00-18:00, l'ultimo secondo incluso nell'intervallo di tempo è 17:59:59.

Esempi

Orario di lavoro. Da lunedì a venerdì dalle 9:00 alle 18:00:

`1-5,09:00-18:00`

Orario di lavoro più fine settimana. Da lunedì a venerdì dalle 9:00 alle 18:00 e sabato, domenica dalle 10:00 alle 16:00:

`1-5,09:00-18:00;6-7,10:00-16:00`

Simboli delle unità

Panoramica

Lavorare con valori grandi come 86400, 104857600 o 1000000 può essere complicato e può portare a errori. Per semplificare la configurazione e migliorare la leggibilità, Zabbix supporta i simboli di unità (suffissi) che fungono da moltiplicatori di valore.

Espressioni dei trigger senza suffissi:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Espressioni dei trigger con suffissi:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

I suffissi possono anche semplificare la configurazione di altre entità, come item, widget, ecc., e aiutare a visualizzare i valori degli item in un formato leggibile.

> **Note:**
> Per verificare se un campo di configurazione di un'entità supporta i suffissi, consultare sempre la pagina pertinente relativa all'entità che si sta configurando.

Suffissi temporali

Il frontend di Zabbix supporta i seguenti suffissi temporali nella configurazione delle entità:
 s - secondi *(se utilizzato, funziona allo stesso modo del valore grezzo)*;
 m - minuti;
 h - ore;
 d - giorni;
 w - settimane;
 M - mesi *(funzioni di trend solo)*;
 y - anni *(funzioni di trend solo)*.

I suffissi temporali sono supportati solo con numeri interi. Ad esempio, 1h è supportato, ma 1,5h o 1.5h non sono supportati; usa invece 90m.

Suffissi della dimensione della memoria

Zabbix supporta i seguenti suffissi della dimensione della memoria:
 K - kilobyte
 M - megabyte
 G - gigabyte
 T - terabyte

Suffissi del valore dell'item

I suffissi possono essere usati anche per visualizzare i valori numerici degli item in un formato leggibile dall'uomo.

Per abilitarlo, usa uno dei seguenti suffissi nel campo *Units* quando configuri un item:
 B - byte;
 Bps - byte al secondo;
 s - secondi, visualizzati usando fino a tre unità di tempo non nulle più grandi;
 uptime - tempo trascorso in hh:mm:ss oppure N giorni, hh:mm:ss;
 unixtime - timestamp Unix, formattato come yyyy.mm.dd hh:mm:ss.

Si applicano le seguenti regole aggiuntive per l'interpretazione e la visualizzazione di questi suffissi:

- Per B e Bps, Zabbix usa la conversione in base 2 (1K = 1024B), seguendo lo standard JEDEC.
- Per altre unità (come Hz, W, ecc.), Zabbix usa la conversione in base 10 (1K = 1000).
- Per s (secondi):
  - Il formato include yyy mmm ddd hhh mmm sss ms; vengono visualizzate solo fino a tre unità di tempo non nulle più grandi (ad esempio, 1M 10d 4h).
  - Se un'unità è zero e si trova tra due unità non nulle, viene omessa (ad esempio, 10d 56m invece di 10d 0h 56m).

Quando vengono usati *Units*, ai valori degli item vengono applicati automaticamente i seguenti suffissi moltiplicatori:
 K, **M**, **G**, **T** - kilo, mega, giga, tera;
 P, **E**, **Z**, **Y** - peta, exa, zetta, yotta *(questi vengono applicati solo nel frontend).*

Per impedire la conversione delle unità, usa il prefisso ! (ad esempio, !B o !s).

Gli esempi seguenti mostrano come i valori degli item ricevuti vengono convertiti in base alle unità specificate:

```
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:
Prima di Zabbix 4.0, esisteva una stoplist di unità hardcoded composta da `ms`, `rpm`, `RPM`, `%`. Questa stoplist è stata deprecata; quindi il modo corretto per impedire la conversione di tali unità è `!ms`, `!rpm`, `!RPM`, `!%`.

Esecuzione dei comandi

Zabbix utilizza una funzionalità comune per i controlli esterni, i parametri utente, gli item system.run, gli script di allerta personalizzati, i comandi remoti e gli script globali.

Fasi di esecuzione

Note:
Per impostazione predefinita, tutti gli script in Zabbix vengono eseguiti utilizzando la shell *sh*, e non è possibile modificare la shell predefinita. Per utilizzare una shell diversa, è possibile adottare una soluzione alternativa: creare un file di script e richiamarlo durante l'esecuzione del comando.

Il comando/script viene eseguito in modo simile sia sulle piattaforme Unix che Windows:

1. Zabbix (il processo padre) crea una pipe per la comunicazione
2. Zabbix imposta la pipe come output per il processo figlio da creare
3. Zabbix crea il processo figlio (esegue il comando/script)
4. Viene creato un nuovo gruppo di processi (in Unix) o un job (in Windows) per il processo figlio
5. Zabbix legge dalla pipe finché non si verifica il timeout oppure finché nessuno scrive più all'altra estremità (TUTTI gli handle/file descriptor sono stati chiusi). Si noti che il processo figlio può creare altri processi e terminare prima che questi terminino o chiudano l'handle/file descriptor.
6. Se il timeout non è stato raggiunto, Zabbix attende finché il processo figlio iniziale non termina oppure finché non si verifica il timeout
7. Se il processo figlio iniziale è terminato e il timeout non è stato raggiunto, Zabbix controlla il codice di uscita del processo figlio iniziale e lo confronta con 0 (un valore diverso da zero è considerato un errore di esecuzione, solo per gli script di avviso personalizzati, i comandi remoti e gli script utente eseguiti su Zabbix server e Zabbix proxy)
8. A questo punto si presume che tutto sia stato completato e l'intero albero dei processi (cioè il gruppo di processi o il job) viene terminato

Attention:
Zabbix presume che un comando/script abbia completato l'elaborazione quando il processo figlio iniziale è terminato E nessun altro processo mantiene ancora aperto l'handle/file descriptor di output. Quando l'elaborazione è completata, TUTTI i processi creati vengono terminati.

Tutte le virgolette doppie e le barre rovesciate nel comando vengono precedute da una barra rovesciata di escape e il comando viene racchiuso tra virgolette doppie.

Controllo del codice di uscita

I codici di uscita vengono controllati con le seguenti condizioni:

- Solo per script di allarme personalizzati, comandi remoti e script utente eseguiti su Zabbix server e Zabbix proxy.
- Qualsiasi codice di uscita diverso da 0 è considerato un errore di esecuzione.
- Il contenuto di standard error e standard output per le esecuzioni non riuscite viene raccolto ed è disponibile nel frontend (dove viene visualizzato il risultato dell'esecuzione).
- Può essere creato un ulteriore record di log per i comandi remoti eseguiti su Zabbix agent/proxy abilitando il parametro LogRemoteCommands su agent/proxy.

Possibili messaggi del frontend e voci di log per comandi/script non riusciti:

- Il contenuto di standard error e standard output per le esecuzioni non riuscite (se presente).
- "Process exited with code: N." (per output vuoto e codice di uscita non uguale a 0).
- "Process killed by signal: N." (per un processo terminato da un segnale, solo su Linux).
- "Process terminated unexpectedly." (per un processo terminato per motivi sconosciuti).

Vedi anche

- Controlli esterni
- Parametri utente
- item system.run
- Script di avviso personalizzati
- Comandi remoti
- Script globali

Esempi di escaping

Panoramica

Questa pagina fornisce esempi di utilizzo dell'escape corretto quando si usano espressioni regolari in vari contesti.

> **Note:**
> Quando si utilizza il costruttore di espressioni del trigger, l'escape corretto nelle espressioni regolari viene aggiunto automaticamente.

Esempi

Macro utente con contesto

Espressione regolare: `\.+\"[a-z]+<br>` Macro utente con contesto: `{$MACRO:regex:"\.+\\"[a-z]+"}`

Si noti:

- i backslash non vengono sottoposti a escape;
- i segni di virgolette sono sottoposti a escape.

Funzione macro all'interno del parametro della chiave item

Espressione regolare: `.+:(\d+)$<br>` Chiave item: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

Si noti:

- l'espressione regolare all'interno della funzione macro `regsub` è racchiusa tra doppi apici (perché contiene una parentesi chiusa);
- i doppi apici attorno all'espressione regolare sono sottoposti a escape (perché l'intero terzo parametro dell'item è racchiuso tra doppi apici);
- il terzo parametro della chiave item è racchiuso tra doppi apici perché contiene una virgola.

Funzione macro LLD

Espressione regolare: `\.+\"([a-z]+)<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Si noti:

- i backslash non sono sottoposti a escape;
- i segni di virgolette sono sottoposti a escape.

Funzione macro LLD all'interno del contesto della macro utente

Espressione regolare: `\.+\"([a-z]+)<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Macro utente con contesto: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Si noti:

- l'escape dei backslash per LLD non cambia;
- inserendo la macro LLD nel contesto della macro utente, dobbiamo inserirla in una stringa:

1. vengono aggiunti i doppi apici attorno all'espressione macro;
2. i doppi apici vengono sottoposti a escape; in totale, vengono introdotti 3 nuovi backslash.

Parametro stringa di una funzione (qualsiasi)

`concat` è usato come esempio.

Contenuto della stringa: `\.+\"[a-z]+<br>` Espressione: `concat("abc", "\\.\\\"[a-z]+")`

Si noti:

- i parametri stringa richiedono l'escape sia dei backslash sia dei doppi apici.

Funzione macro LLD all'interno del parametro stringa di una funzione

Espressione regolare: `\.+\"([a-z]+)<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Espressione: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Si noti:

- i parametri stringa richiedono l'escape sia dei backslash sia dei doppi apici;
- viene aggiunto un ulteriore livello di escape, perché la macro verrà risolta solo dopo che la stringa sarà stata privata delle virgolette;

Macro utente con contesto all'interno del parametro stringa di una funzione

Espressione regolare: `\.+\"[a-z]+<br>` Macro utente con contesto: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Espressione: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

Si noti:

- come nell'esempio precedente, è necessario un ulteriore livello di escape;
- i backslash e i doppi apici sono sottoposti a escape solo per il livello di escape superiore (in quanto parametro stringa).

Funzione macro LLD all'interno del contesto della macro utente all'interno di una funzione

Espressione regolare: `\.+\"([a-z])+<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Macro utente con contesto: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` Espressione: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

Si notino i tre livelli di escape:

1. Per la funzione macro LLD, senza escape dei backslash;
2. Per la macro utente con contesto, senza escape dei backslash;
3. Per il parametro stringa di una funzione, con escape dei backslash.

Macro utente con contesto semplicemente all'interno di una stringa

Espressione regolare: `\.+\"[a-z]+<br>` Macro utente con contesto: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` All'interno della stringa di una qualche espressione, per esempio: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

Si noti:

- anche le stringhe richiedono l'escape dei backslash;
- anche le stringhe richiedono l'escape dei doppi apici;
- ancora una volta, si tratta di un caso con 2 livelli di escape:

1. Escape per il contesto della macro utente senza escape dei backslash;
2. Escape dovuto al fatto che si tratta di una stringa con escape dei backslash.

Cos'è Zabbix

Zabbix è stato creato da Alexei Vladishev ed è attualmente sviluppato e supportato attivamente da Zabbix SIA.

Zabbix è una soluzione di monitoraggio distribuito open source di livello enterprise.

Zabbix è un software che monitora numerosi parametri di una rete, nonché lo stato e l'integrità di server, macchine virtuali, applicazioni, servizi, database, siti web, cloud e altro ancora. Zabbix utilizza un meccanismo di notifica flessibile che consente agli utenti di configurare avvisi basati sulle email per praticamente qualsiasi evento. Ciò permette di reagire rapidamente ai problemi dei server. Zabbix offre eccellenti funzionalità di reporting e visualizzazione dei dati basate sui dati memorizzati. Questo rende Zabbix ideale per la pianificazione della capacità.

Zabbix supporta sia il polling sia il trapping. Tutti i report e le statistiche di Zabbix, così come i parametri di configurazione, sono accessibili tramite un frontend basato sul web. Un frontend basato sul web garantisce che lo stato della rete e l'integrità dei server possano essere valutati da qualsiasi posizione. Se configurato correttamente, Zabbix può svolgere un ruolo importante nel monitoraggio dell'infrastruttura IT. Ciò vale sia per le piccole organizzazioni con pochi server sia per le grandi aziende con una moltitudine di server.

Zabbix è disponibile gratuitamente. Zabbix è scritto e distribuito con licenza AGPL-3.0. Ciò significa che il suo codice sorgente è distribuito liberamente e disponibile al pubblico.

Il supporto commerciale è disponibile ed è fornito da Zabbix Company e dai suoi partner in tutto il mondo.

Scopri di più sulle funzionalità di Zabbix.

Utenti di Zabbix Molte organizzazioni di diverse dimensioni in tutto il mondo si affidano a Zabbix come piattaforma principale di monitoraggio.

Architettura Zabbix è composto da diversi componenti software principali. Le loro responsabilità sono descritte di seguito.

Server

Zabbix server è il componente centrale a cui gli agent segnalano informazioni e statistiche sulla disponibilità e sull'integrità. Il server è il repository centrale in cui vengono archiviati tutti i dati di configurazione, statistici e operativi.

Archiviazione del database

Tutte le informazioni di configurazione, così come i dati raccolti da Zabbix, vengono memorizzati in un database.

Interfaccia web

Per un facile accesso a Zabbix da qualsiasi luogo e da qualsiasi piattaforma, viene fornita l'interfaccia basata sul web. L'interfaccia fa parte di Zabbix server e di solito (ma non necessariamente) viene eseguita sulla stessa macchina fisica su cui è in esecuzione il server.

Proxy

Zabbix proxy può raccogliere dati sulle prestazioni e sulla disponibilità per conto di Zabbix server. Un proxy è una parte opzionale della distribuzione di Zabbix; tuttavia, può essere molto utile per distribuire il carico di un singolo Zabbix server.

Agent

Gli agent di Zabbix vengono distribuiti sui target di monitoraggio per controllare attivamente le risorse e le applicazioni locali e inviare i dati raccolti a Zabbix server. A partire da Zabbix 4.4, sono disponibili due tipi di agent: Zabbix agent (leggero, supportato su molte piattaforme, scritto in C) e Zabbix agent 2 (estremamente flessibile, facilmente estendibile con plugin, scritto in Go).

Flusso dei dati Inoltre è importante fare un passo indietro e osservare il flusso complessivo dei dati all'interno di Zabbix. Per creare un item che raccolga dati, è necessario prima creare un host. Passando all'altro estremo dello spettro di Zabbix, è necessario prima avere un item per creare un trigger. È necessario avere un trigger per creare un'azione. Quindi, se si desidera ricevere un avviso che il carico della CPU è troppo elevato su *Server X*, è necessario prima creare una voce host per *Server X*, seguita da un item per monitorarne la CPU, quindi un trigger che si attivi se la CPU è troppo alta, seguito da un'azione che invii un'email. Anche se possono sembrare molti passaggi, con l'uso dei template in realtà non lo sono. Tuttavia, grazie a questo design, è possibile creare una configurazione molto flessibile.

Funzionalità Zabbix è una soluzione di monitoraggio di rete altamente integrata, che offre una molteplicità di funzionalità in un unico pacchetto.

Raccolta dati

- controlli di disponibilità e prestazioni
- supporto per SNMP (sia trapping che polling), IPMI, JMX, monitoraggio VMware
- controlli personalizzati
- raccolta dei dati desiderati a intervalli personalizzati
- eseguita da server/proxy e dagli agent

Definizioni flessibili delle soglie

- è possibile definire soglie di problema molto flessibili, chiamate trigger, che fanno riferimento ai valori del database backend

Segnalazione altamente configurabile

- l'invio delle notifiche può essere personalizzato in base al programma di escalation, al destinatario e al tipo di media
- le notifiche possono essere rese significative e utili tramite variabili macro
- le azioni automatiche includono comandi remoti

Grafici in tempo reale

- gli item monitorati vengono immediatamente rappresentati in grafico tramite la funzionalità di grafici integrata

Funzionalità di monitoraggio web

- Zabbix può seguire un percorso di clic simulati del mouse su un sito web e verificare funzionalità e tempo di risposta

Ampie opzioni di visualizzazione

- possibilità di creare grafici personalizzati che possono combinare più item in un'unica vista
- mappe di rete
- presentazioni in stile dashboard
- report
- vista di alto livello (business) delle risorse monitorate

Archiviazione dei dati storici

- dati archiviati in un database
- cronologia configurabile
- procedura di housekeeping integrata

Configurazione semplice

- aggiunta dei dispositivi monitorati come host
- gli host vengono presi in carico per il monitoraggio una volta presenti nel database
- applicazione di template ai dispositivi monitorati

Uso dei template

- raggruppamento dei controlli nei template
- i template possono ereditare altri template

Rilevamento della rete

- rilevamento automatico dei dispositivi di rete
- autoregistrazione dell'agent
- rilevamento di file system, interfacce di rete e OID SNMP

Frontend web veloce

- un frontend basato sul web in PHP
- accessibile da qualsiasi luogo
- è possibile navigare con pochi clic
- registro di audit

API di Zabbix

- l'API di Zabbix fornisce un'interfaccia programmabile a Zabbix per manipolazioni di massa, integrazione con software di terze parti e altri scopi.

Sistema di permessi

- autenticazione sicura degli utenti
- determinati utenti possono essere limitati a determinate viste

agent completo e facilmente estensibile

- distribuito sui target di monitoraggio
- può essere distribuito sia su Linux che su Windows

Daemon binari

- scritti in C, per prestazioni elevate e ridotto consumo di memoria
- facilmente portabili

Pronto per ambienti complessi

- il monitoraggio remoto è reso semplice grazie all'uso di un proxy Zabbix

Glossario In questa sezione è possibile apprendere il significato di alcuni termini comunemente utilizzati in Zabbix.

host

- qualsiasi dispositivo fisico o virtuale, applicazione, servizio o altra raccolta logicamente correlata di parametri monitorati.

host group

- un raggruppamento logico di host. Gli host group vengono utilizzati per assegnare i diritti di accesso agli host a diversi gruppi di utenti.

item

- uno specifico dato che si desidera ricevere da un host, una metrica di dati.

value preprocessing

- una trasformazione del valore della metrica ricevuta prima del salvataggio nel database.

trigger

- un'espressione logica che definisce una soglia di problema e viene utilizzata per "valutare" i dati ricevuti negli item.

Quando i dati ricevuti superano la soglia, i trigger passano dallo stato *Ok* allo stato *Problem*. Quando i dati ricevuti sono inferiori alla soglia, i trigger rimangono o tornano allo stato *Ok*.

template

- un insieme di entità (item, trigger, grafici, regole di low-level discovery, scenari web) pronto per essere applicato a uno o più host.

Lo scopo dei template è velocizzare la distribuzione delle attività di monitoraggio su un host e semplificare l'applicazione di modifiche di massa alle attività di monitoraggio. I template sono collegati direttamente ai singoli host.

template group

- un raggruppamento logico di template. I template group vengono utilizzati per assegnare i diritti di accesso ai template a diversi gruppi di utenti.

event

- una singola occorrenza di qualcosa che richiede attenzione, come il cambiamento di stato di un trigger o l'esecuzione della discovery o dell'autoregistrazione di un agent.

event tag

- un contrassegno predefinito per l'evento. Può essere utilizzato nella correlazione degli eventi, nella definizione granulare delle autorizzazioni e così via.

event correlation

- un metodo per correlare i problemi alla loro risoluzione in modo flessibile e preciso.

Ad esempio, è possibile definire che un problema segnalato da un trigger possa essere risolto da un altro trigger, che può persino utilizzare un metodo di raccolta dati diverso.

problem

- un trigger che si trova nello stato "Problem".

problem update

- opzioni di gestione dei problemi fornite da Zabbix, come l'aggiunta di un commento, la conferma, la modifica della gravità o la chiusura manuale.

action

- un mezzo predefinito per reagire a un evento.

Un'action è costituita da operazioni (ad esempio, l'invio di una notifica) e condizioni (*quando* l'operazione viene eseguita).

escalation

- uno scenario personalizzato per l'esecuzione delle operazioni all'interno di un'action; una sequenza di invio di notifiche o esecuzione di comandi remoti.

macro

- un segnaposto variabile che viene risolto in un valore specifico in base al contesto.

media

- un mezzo per distribuire le notifiche; un canale di distribuzione.

notification

- un messaggio relativo a un evento inviato a un utente tramite il canale media scelto.

remote command

- un comando predefinito che viene eseguito automaticamente su un host monitorato al verificarsi di una determinata condizione.

web scenario

- una o più richieste HTTP per verificare la disponibilità di un sito web.

frontend

- l'interfaccia web fornita con Zabbix.

dashboard

- una sezione personalizzabile dell'interfaccia web che visualizza riepiloghi e rappresentazioni visive di informazioni importanti in unità visive chiamate widget.

widget

- un'unità visiva che visualizza informazioni di un determinato tipo e origine (un riepilogo, una mappa, un grafico, l'orologio e così via), utilizzata nella dashboard.

Zabbix API

- Zabbix API consente di utilizzare il protocollo JSON RPC per creare, aggiornare e recuperare oggetti Zabbix (come host, item, grafici e altri) o eseguire qualsiasi altra attività personalizzata.

Zabbix server

- un processo centrale del software Zabbix che esegue il monitoraggio, interagisce con proxy e agent Zabbix, calcola i trigger, invia notifiche e funge da repository centrale dei dati.

Zabbix proxy

- un processo che può raccogliere dati per conto di Zabbix server, riducendo il carico di elaborazione del server.

Zabbix agent

- un processo distribuito sui target di monitoraggio per monitorare attivamente risorse e applicazioni locali.

Zabbix agent 2

- una nuova generazione di Zabbix agent per il monitoraggio attivo di risorse e applicazioni locali, che consente l'utilizzo di plugin personalizzati per il monitoraggio.

> **Attention:**
> Poiché Zabbix agent 2 condivide gran parte delle funzionalità con Zabbix agent, nella documentazione il termine "Zabbix agent" indica entrambi: Zabbix agent e Zabbix agent 2, quando il comportamento funzionale è lo stesso. Zabbix agent 2 viene indicato specificamente solo quando la sua funzionalità è diversa.

encryption

- il supporto per le comunicazioni crittografate tra i componenti Zabbix (server, proxy, agent, le utilità zabbix_sender e zabbix_get) tramite il protocollo Transport Layer Security (TLS).

agent autoregistration

- un processo automatizzato mediante il quale un Zabbix agent viene registrato autonomamente come host e avvia il monitoraggio.

network discovery

- la discovery automatizzata dei dispositivi di rete.

low-level discovery

- la discovery automatizzata di entità di basso livello su un determinato dispositivo (ad esempio, file system, interfacce di rete e così via).

low-level discovery rule

- un insieme di definizioni per la discovery automatizzata di entità di basso livello su un dispositivo.

item prototype

- *una metrica con determinati parametri utilizzati come variabili, pronta per la low-level discovery.* Dopo la low-level discovery, le variabili vengono sostituite automaticamente con i parametri effettivamente rilevati e la metrica inizia automaticamente a raccogliere dati.

trigger prototype

- *un trigger con determinati parametri utilizzati come variabili, pronto per la low-level discovery.* Dopo la low-level discovery, le variabili vengono sostituite automaticamente con i parametri effettivamente rilevati e il trigger inizia automaticamente a valutare i dati.

Anche i *prototipi* di alcune altre entità Zabbix vengono utilizzati nella low-level discovery: prototipi di grafici, prototipi di host e prototipi di host group.

Zabbix Cloud

Zabbix Cloud

Scopri le opzioni di configurazione disponibili e i controlli di accesso per organizzazioni e nodi in Zabbix Cloud.

Tieni presente che i nodi di Zabbix Cloud eseguono attualmente **Zabbix 7.0**.
 Per informazioni approfondite su Zabbix 7.0, consulta la documentazione ufficiale.

Configurare e gestire Zabbix Cloud

Inizia a usare Zabbix Cloud Istruzioni guidate per mettere rapidamente in funzione la tua istanza Zabbix Cloud, inclusa la configurazione iniziale e il collegamento dei dispositivi monitorati.

Configurazione del nodo Scopri come configurare il tuo nodo Zabbix Cloud, con particolare attenzione all'accesso, alla gestione dei dati e alle impostazioni delle risorse.

Aggiunta di utenti Scopri come aggiungere utenti e assegnare loro ruoli all'interno di un'organizzazione Zabbix Cloud e dei suoi nodi.

Scopri Zabbix Cloud

Zabbix Cloud vs. on-premises Confronta le differenze nella gestione, nella scalabilità e nella manutenzione tra i due tipi di deployment, e scopri quale può supportare il tuo flusso di lavoro nel modo più fluido.

Esplora Zabbix Cloud Ottieni una panoramica di Zabbix Cloud, incluse le sue funzionalità principali, i vantaggi e ciò che è incluso con un abbonamento. Per le domande più comuni su configurazione e utilizzo, consulta le FAQ.

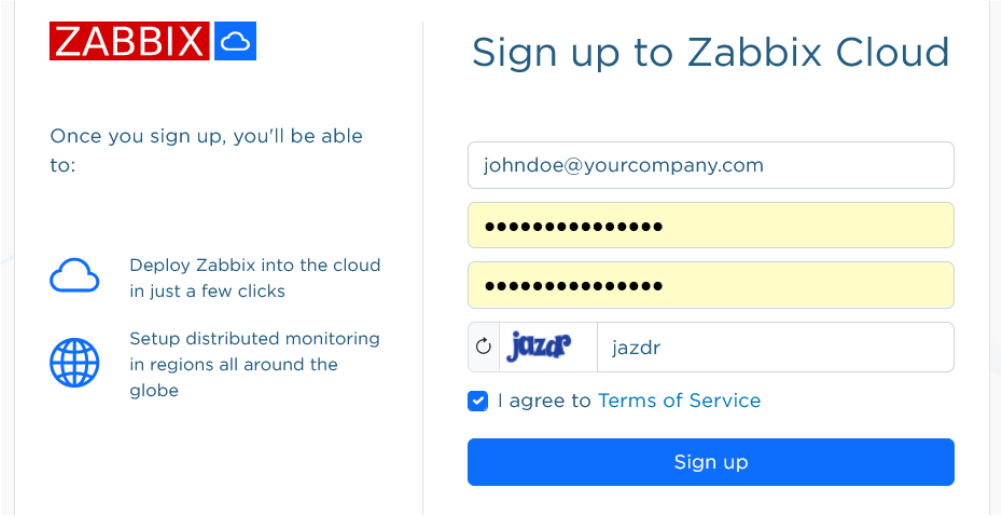
Distribuire Zabbix nel cloud

Per avviare Zabbix nel cloud, segui questa procedura:

1. Accedi o registrati su cloud.zabbix.com.
2. Crea un cloud node.
3. Inizia a utilizzare Zabbix nel cloud.

Accedi o registrati Puoi accedere a cloud.zabbix.com con un account Google, GitHub o Microsoft esistente.

In alternativa, vai alla pagina di registrazione per registrarti.



Dopo la registrazione, riceverai un'email di conferma del nuovo account.

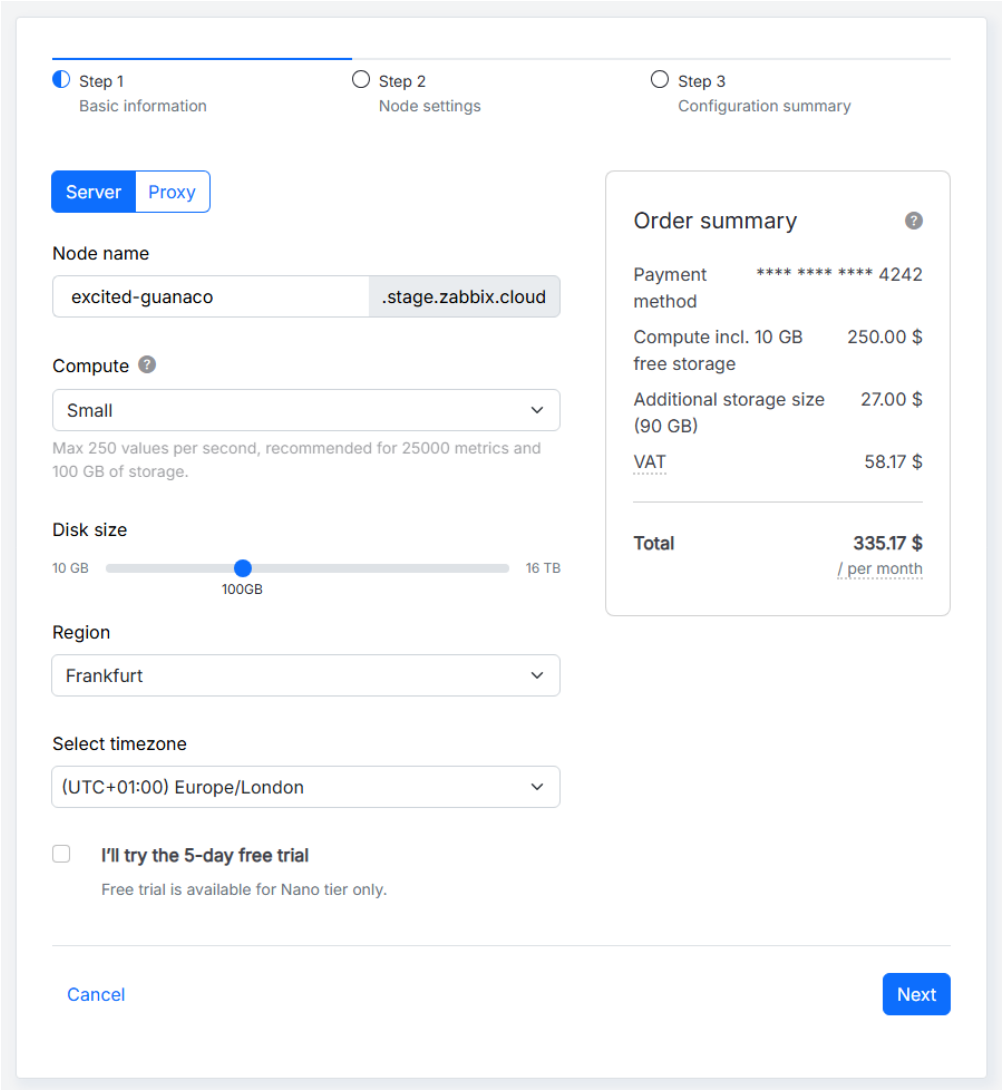
Vai su cloud.zabbix.com per accedere. Durante l'accesso ti verrà richiesta una password monouso. Questa password ti verrà inviata tramite email.

Crea un cloud node Un cloud node è un'istanza di Zabbix distribuita in Zabbix Cloud. Può essere distribuita come server node o proxy node.

Dopo aver effettuato l'accesso, vedrai che per impostazione predefinita esiste già un'organizzazione *Private* per il tuo account. In seguito potrai rinominare l'organizzazione predefinita.

Fai clic su *Create New Node*. Si aprirà un modulo di configurazione del node in tre passaggi.

Nel primo passaggio selezionerai le informazioni di base.



Parametro	Descrizione
Server/Proxy	Seleziona se distribuire un'istanza server Zabbix o un'istanza proxy Zabbix.
Node name	Il nome del node costituisce un sottodominio univoco nel formato `<your-name>.zabbix.cloud`, utilizzato per accedere ai servizi del node.
Compute	Seleziona la potenza di calcolo del node. Per determinare il livello di risorse necessario, consulta i dettagli sui prezzi. Se hai bisogno di maggiore potenza, potrai sempre effettuare un upgrade in seguito. Prima di creare un node non di prova sarà necessario specificare un metodo di fatturazione. Aggiungi le informazioni di fatturazione tramite il menu *Billing*. È disponibile una **prova gratuita** di 5 giorni per provare un node del livello *Nano*. Durante il periodo di prova non ti verrà addebitato alcun costo, ma al termine il node verrà rimosso se non sarà convertito in un node a pagamento.
Disk size	Il dispositivo di scorrimento è impostato per impostazione predefinita su un minimo consigliato in base al livello. Aumentalo se memorizzi grandi volumi di dati storici o trend. Tieni presente che dischi più grandi comportano costi mensili maggiori e che non puoi ridurre autonomamente le dimensioni di un disco: dovrai contattare il supporto per ridurle.
Region	Seleziona la regione in cui verrà distribuito il node.
Select timezone	Seleziona il fuso orario del server Zabbix. Questo fuso orario viene utilizzato dal server Zabbix ed è indipendente dai fusi orari della manutenzione e del profilo utente.

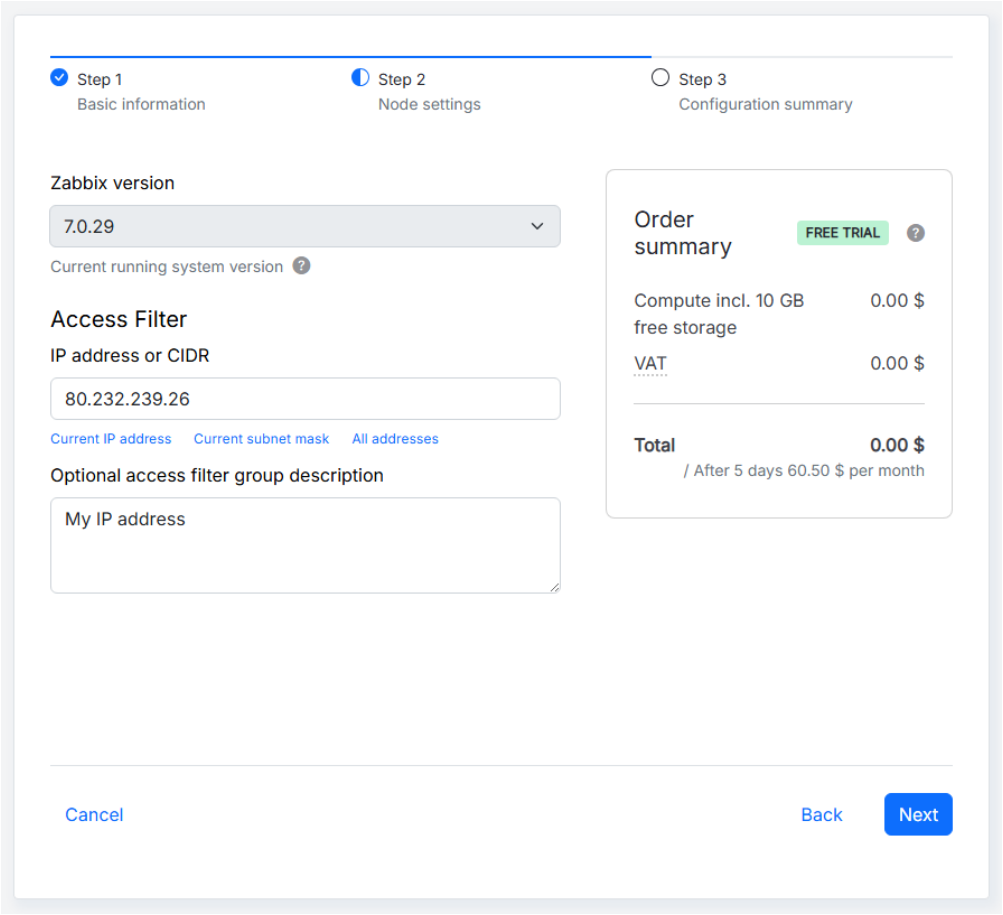
Dopo aver completato la configurazione, fai clic su *Next*.

Nel secondo passaggio selezionerai le impostazioni del node.

Le opzioni di configurazione del secondo passaggio dipendono dal fatto che tu stia creando un server node o un proxy node.

Configura un server node

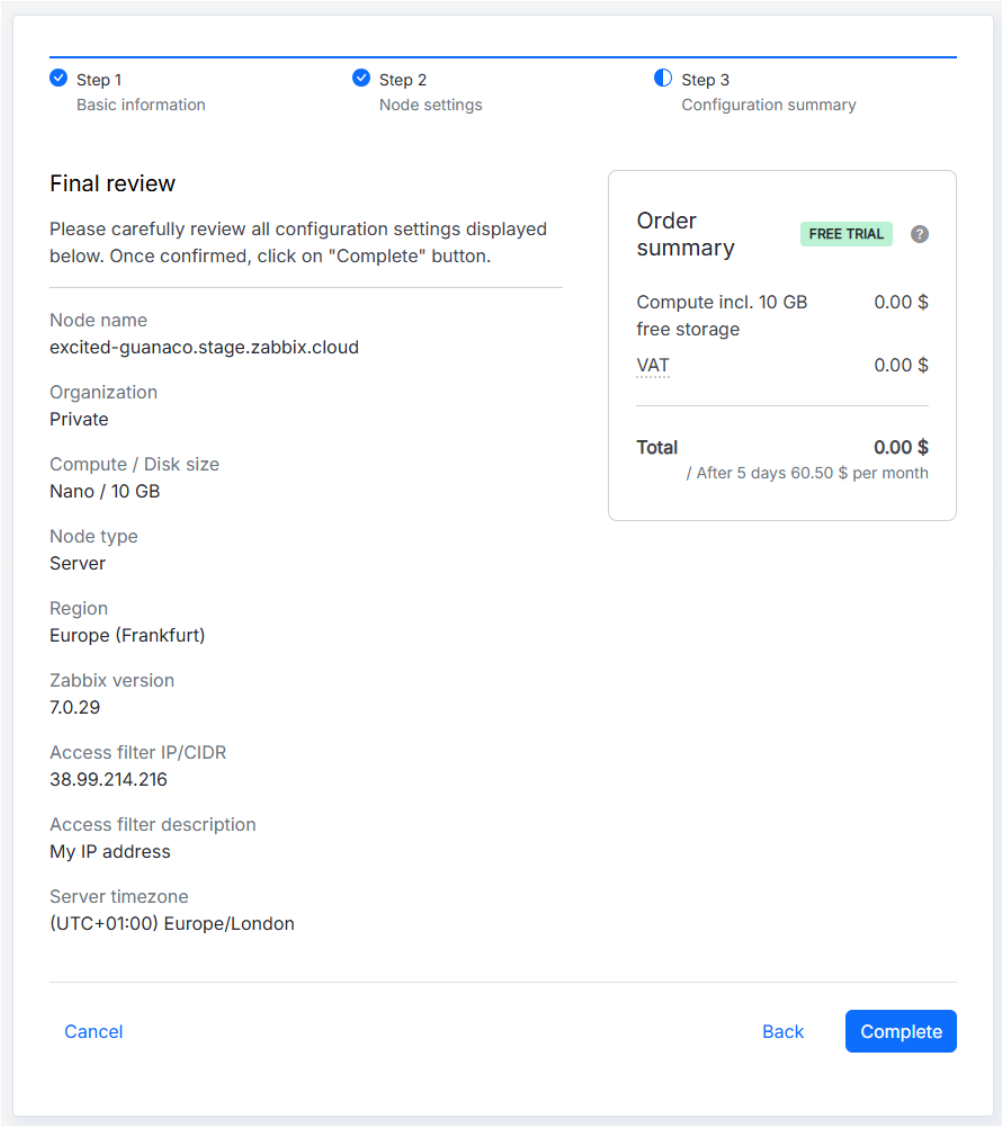
Durante la creazione di un server node sono disponibili i seguenti parametri:



Parametro	Descrizione
Zabbix version	Seleziona la versione di Zabbix. Zabbix Cloud supporta l'ultima versione LTS e applica automaticamente gli aggiornamenti delle versioni minori.
IP address or CIDR	Specifica il filtro di accesso iniziale per il node. Inserisci manualmente l'indirizzo IP o il CIDR oppure compila automaticamente il campo selezionando l'opzione pertinente. Per i server node, questo filtro si applica alle connessioni al server, all'API e al frontend. Il filtro può essere modificato in seguito nella scheda Access filters.
Optional access filter group description	Aggiungi una descrizione facoltativa per il gruppo di filtri di accesso, che può contenere più indirizzi IP o CIDR.

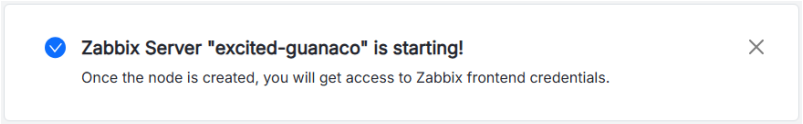
Warning:
Durante la creazione del node, il campo *IP address or CIDR* viene precompilato con l'indirizzo IP utilizzato per accedere a Zabbix Cloud. Puoi modificare questo valore per specificare un filtro di accesso iniziale diverso. Dopo la creazione del node, puoi aggiungere, modificare o eliminare i filtri di accesso dalla scheda *Access filters*.

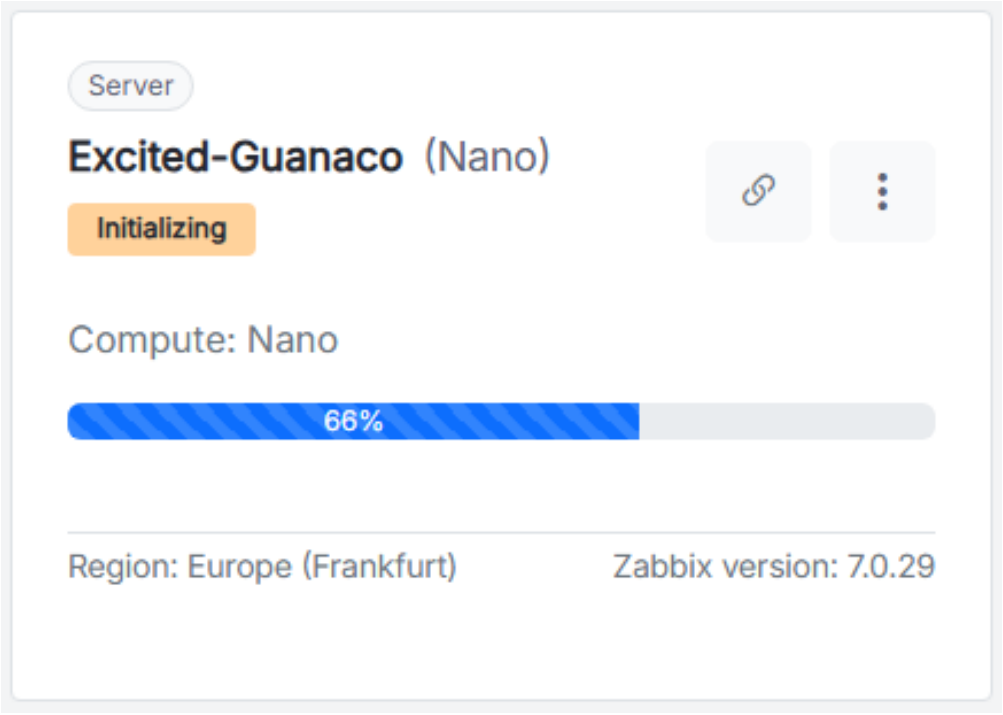
Il terzo passaggio mostra un riepilogo della configurazione del node.



Dopo aver esaminato le impostazioni di configurazione e verificato che tutto sia corretto, fai clic su *Complete* per un node di prova oppure su *Proceed to payment* per un node a pagamento. In caso contrario, torna a un passaggio precedente facendo clic sul pulsante *Back* oppure su *Cancel* per ricominciare dall'inizio.

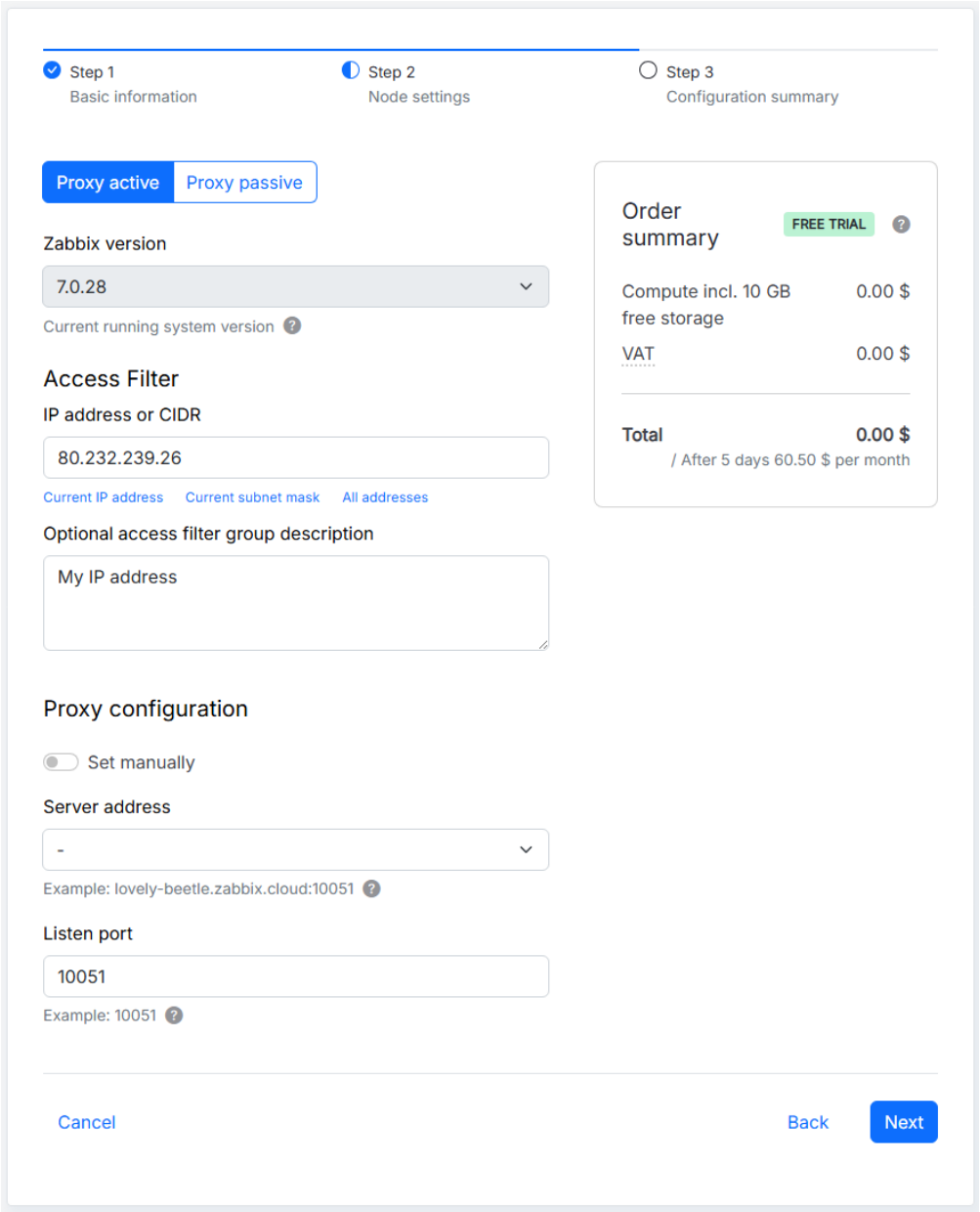
Durante l'inizializzazione del node viene visualizzato un messaggio corrispondente e il node appare già in *Organizations*.





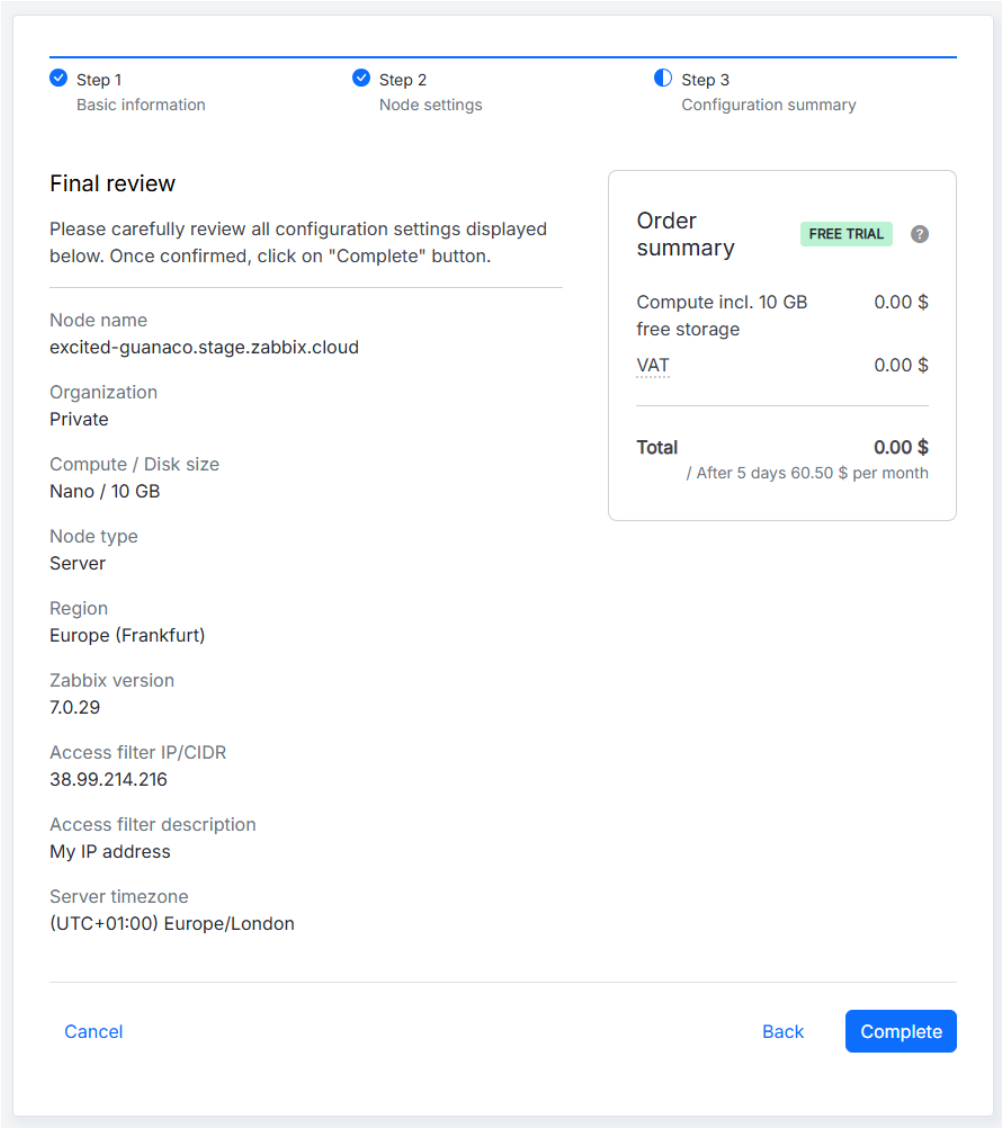
Configura un proxy node

Durante la creazione del proxy node sono disponibili i seguenti parametri:



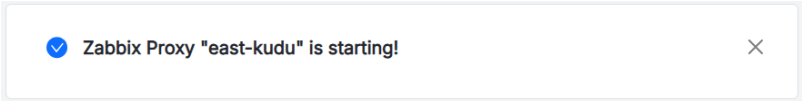
Parametro	Descrizione
Proxy active/Proxy passive	Seleziona la modalità del proxy. In modalità attiva, il proxy avvia le connessioni al server Zabbix; in modalità passiva, il server Zabbix avvia le connessioni al proxy.
Zabbix version	Seleziona la versione di Zabbix. Zabbix Cloud supporta l'ultima versione LTS e applica automaticamente gli aggiornamenti delle versioni minori.
IP address or CIDR	Specifica il filtro di accesso IP in entrata iniziale per il proxy. Inserisci manualmente l'indirizzo IP o il CIDR oppure compila automaticamente il campo selezionando l'opzione pertinente. Il filtro definisce quali richieste di rete possono raggiungere il proxy e può essere modificato in seguito nella scheda Access filters.
Optional access filter group description	Aggiungi una descrizione facoltativa per il gruppo di filtri di accesso, che può contenere più indirizzi IP o CIDR.
Set manually	Seleziona come specificare il server Zabbix. Se disabilitato, seleziona un server node dall'elenco dei server node disponibili. Se abilitato, specifica manualmente gli indirizzi del server.
Server address	Specifica l'indirizzo del server Zabbix. Quando *Set manually* è disabilitato, seleziona un server node dall'elenco. Quando è abilitato, specifica fino a 10 indirizzi del server. Per un proxy attivo, specifica l'indirizzo IP o il nome DNS del server Zabbix. Per un proxy passivo, specifica l'indirizzo IP, il nome DNS o il CIDR del server Zabbix. I filtri di accesso vengono configurati separatamente e non vengono aggiornati automaticamente quando viene specificato un indirizzo del server.
Listen port	Specifica la porta sulla quale il proxy ascolta le connessioni in entrata. La porta predefinita è 10051. La porta di ascolto non può essere modificata dopo la creazione del node.

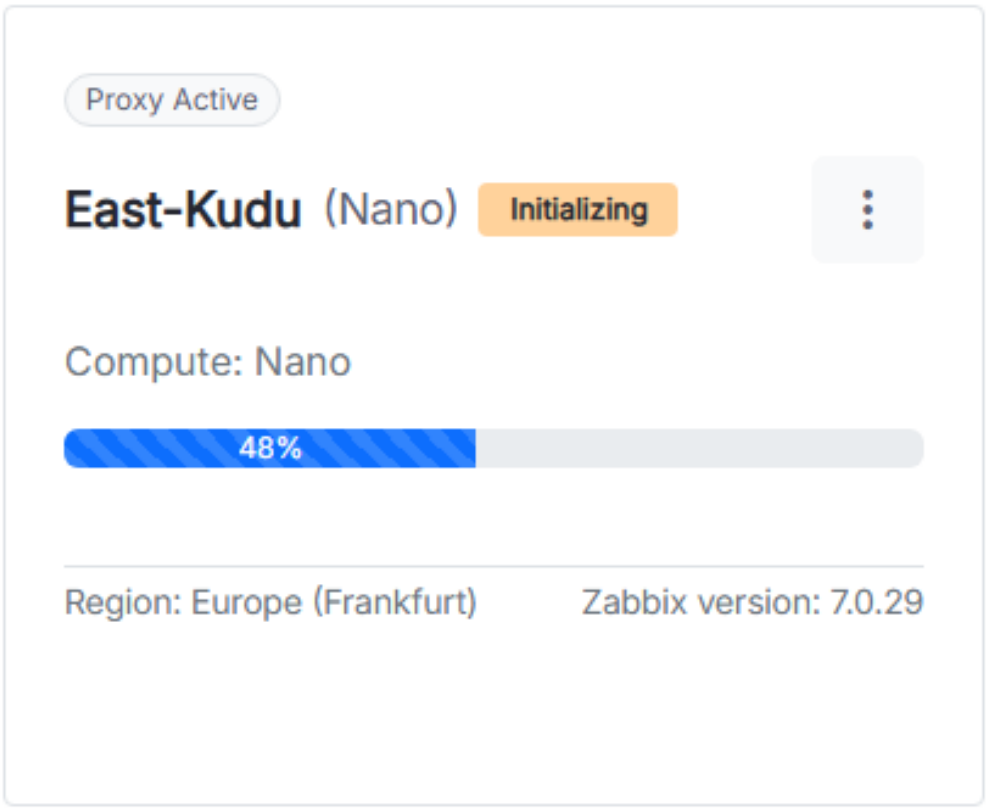
Nel terzo passaggio vedrai il riepilogo della configurazione del node.



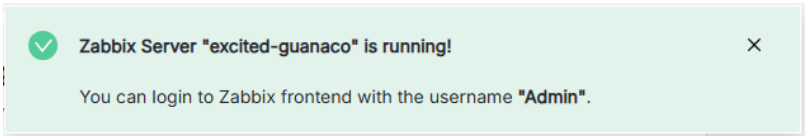
Dopo aver esaminato le impostazioni di configurazione e verificato che tutto sia corretto, fai clic su *Complete* per un node di prova oppure su *Proceed to payment* per un node a pagamento. In caso contrario, torna ai passaggi precedenti facendo clic sul pulsante *Back* oppure su *Cancel* per ricominciare dall'inizio.

Durante l'inizializzazione del node viene visualizzato un messaggio corrispondente e il node appare già in *Organizations*.





Per configurare un proxy Zabbix Cloud, consulta Configura un proxy Zabbix Cloud.

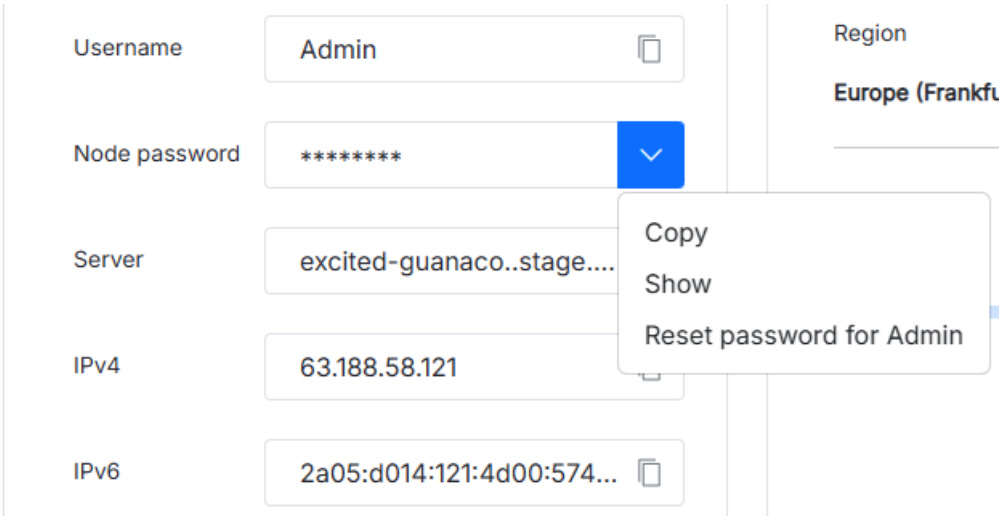


Iniziare a utilizzare Zabbix

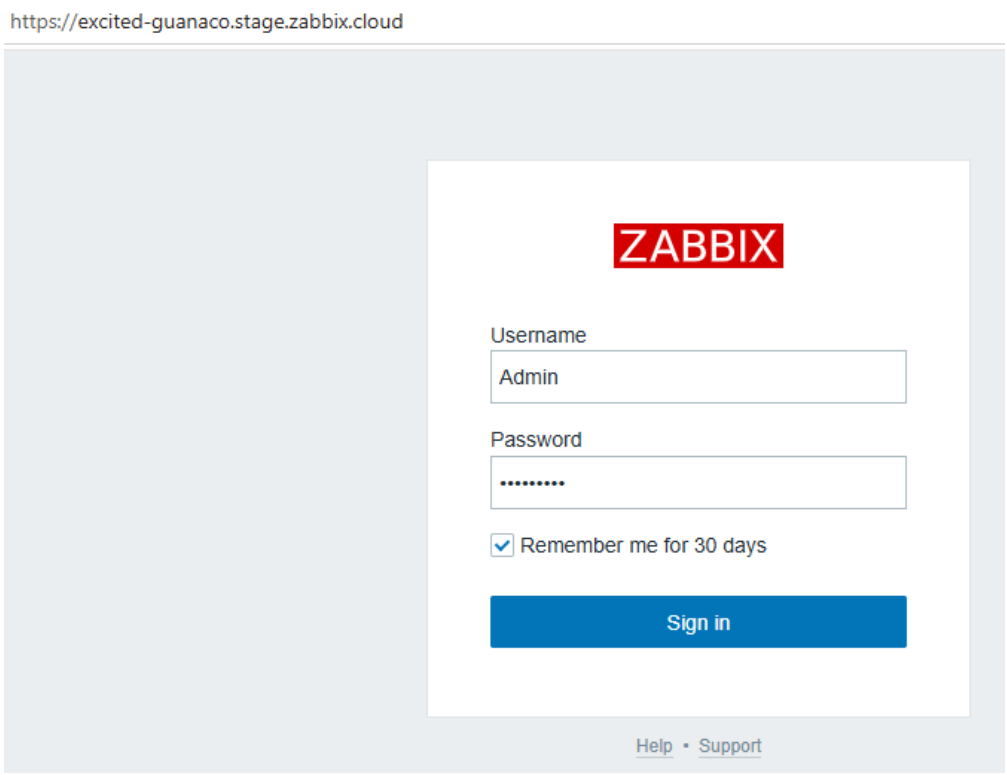
Quando lo stato del nodo server cambia da *Initializing* a *Running*, è possibile accedere al frontend di Zabbix.

Per accedere alla password iniziale del frontend, fare clic sul nodo creato.

Si verrà reindirizzati alle impostazioni del nodo (panoramica, filtri di accesso, crittografia, backup, ecc.). In **Connection info**, individuare il campo *Node password* e selezionare *Reset password for Admin* dall'elenco a discesa. Dopo aver reimpostato la password, fare clic sullo stesso elenco a discesa e selezionare *Copy*.



Fare clic sull'URL del frontend per aprire il frontend di Zabbix (`excited-guanaco.zabbix.cloud`).



Accedere utilizzando le seguenti credenziali:

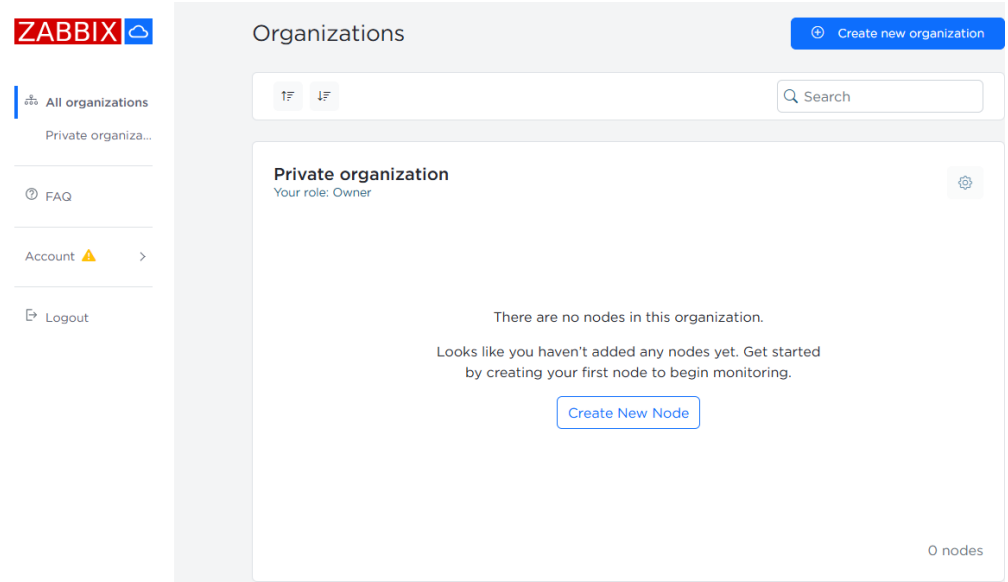
- *Username* - `Admin`
- *Password* - inserire la password copiata

Per iniziare il monitoraggio, è necessario definire i filtri di accesso, ovvero l'indirizzo IP o gli indirizzi IP autorizzati a connettersi al server Zabbix Cloud. È necessario definirne almeno uno.

Gestione dell'organizzazione Un'organizzazione è un contenitore per uno o più nodi. Le organizzazioni consentono inoltre di strutturare le risorse e delegare le responsabilità, ad esempio:

- Gestione dei nodi
- Fatturazione

Quando si accede a Zabbix Cloud, viene visualizzata la pagina *Tutte le organizzazioni*. Per impostazione predefinita, viene creata un'organizzazione privata per il proprio account.

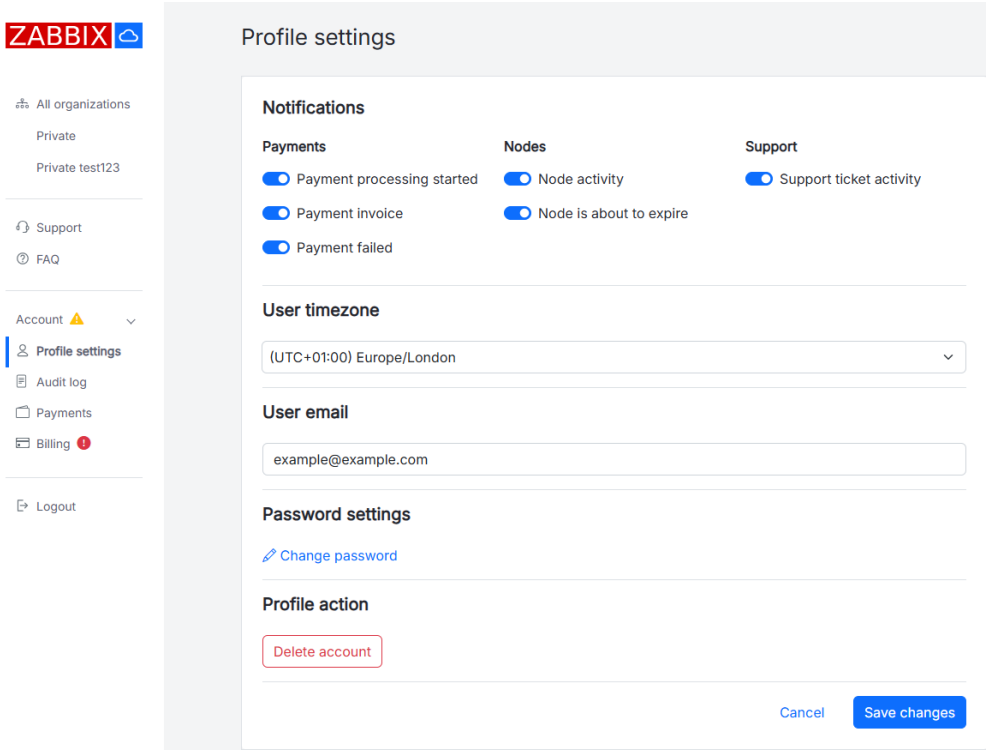


In qualità di proprietario di questa organizzazione, è possibile gestire utenti, amministratori, nodi e nome dell'organizzazione. Fare clic sul pulsante delle impostazioni  per accedere alle opzioni di gestione.

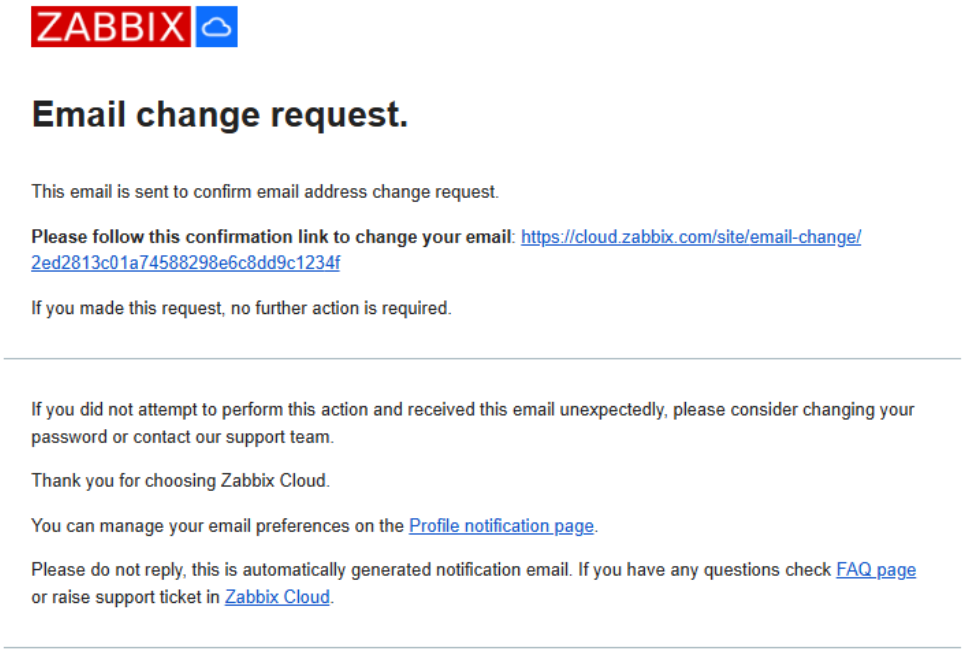
Gestione degli account Modifica dell'indirizzo email dell'account

Per modificare l'indirizzo email dell'account, accedi a *Account* > *Impostazioni del profilo*.

Nel campo *Email dell'utente*, inserisci il nuovo indirizzo email e fai clic su *Salva modifiche*.



All'indirizzo email viene inviato un messaggio di conferma. Segui il link nel messaggio per confermare la modifica dell'indirizzo email.



Il link di conferma apre la pagina *Impostazioni del profilo* con un messaggio di conferma e il nuovo indirizzo email nel campo Email dell'utente.

Dopo aver confermato la modifica, un'email di conferma viene inviata sia al nuovo sia al precedente indirizzo email. Il messaggio inviato al precedente indirizzo include il precedente e il nuovo indirizzo email.

Configurazione del nodo

Panoramica

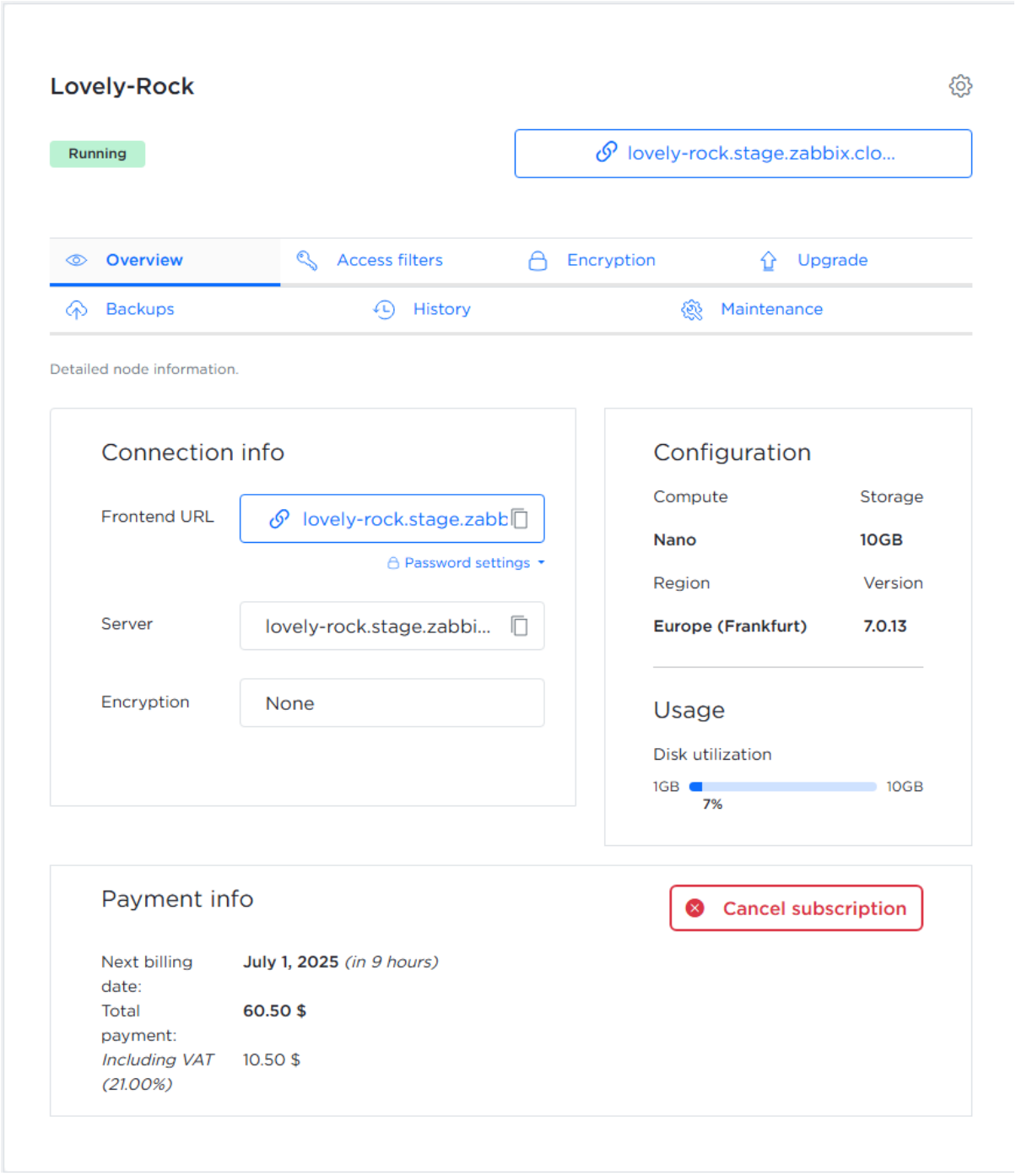
Un **cloud node** è un componente Zabbix distribuito nel cloud. I nodi server includono il server Zabbix, il database e il frontend, mentre i nodi proxy includono il proxy Zabbix e il database. Un cloud node dispone di risorse di calcolo dedicate (CPU, memoria) e di spazio di archiviazione su disco.

Gli utenti possono creare uno o più cloud node. I cloud node possono essere raggruppati in **organizzazioni** (strutturali, regionali, ecc.), creando una configurazione di monitoraggio distribuita.

I cloud node e le organizzazioni vengono creati utilizzando l'interfaccia utente di Zabbix Cloud. Per accedere alla pagina di configurazione del nodo, fare clic sul nodo da *Tutte le organizzazioni*, dal menu dell'organizzazione oppure sul collegamento *Pagina di configurazione del nodo* visualizzato dopo l'avvio del nodo.

Nella vista di configurazione del nodo, ogni scheda consente di gestire un aspetto specifico del nodo:

- Panoramica - visualizzare i dettagli del nodo, aprire il relativo frontend e gestire gli utenti del nodo.
- Filtri di accesso - controllare l'accesso in base all'indirizzo IP o alla subnet.
- Crittografia - configurare TLS caricando certificati, chiavi e CRL facoltativi.
- Backup - creare, ripristinare ed eliminare snapshot del nodo; visualizzare il prossimo backup pianificato.
- Cronologia - impostare i periodi di conservazione per i log di audit, la cronologia e i trend, con unità preimpostate e un limite di aggiornamento di 3 volte al giorno.
- Manutenzione - definire finestre di manutenzione settimanali in base al fuso orario, al giorno e all'ora.
- Aggiornamento - aumentare le risorse di calcolo/archiviazione o gestire lo stato della versione di prova/del pagamento.

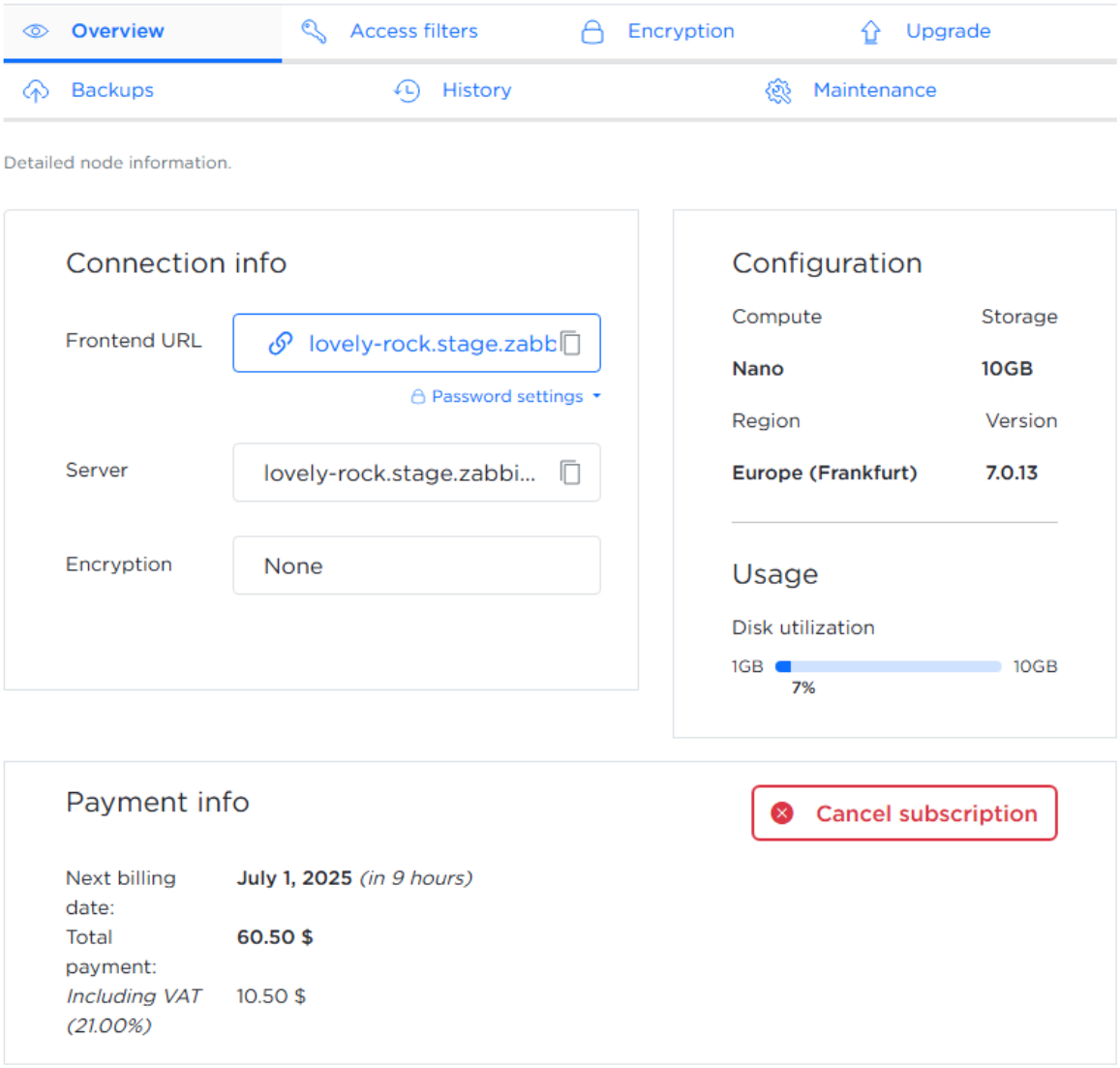


Note:
Alcune schede potrebbero non essere disponibili a seconda del tipo di nodo (server o proxy), del tipo di utente che visualizza la pagina e dello stato del nodo (versione di prova/non di prova).

Scheda Panoramica

La scheda *Panoramica* fornisce informazioni dettagliate sul nodo. Le informazioni visualizzate dipendono dal fatto che il nodo sia un server o un proxy.

Nodo server La scheda *Overview* del nodo server contiene le seguenti sezioni:



Connection info:

- URL frontend cliccabile con un'opzione per copiarla negli appunti
- Nome utente
- Password del nodo e relative impostazioni
- Nome host del server con un'opzione per copiarlo negli appunti
- Indirizzo IPv4 con un'opzione per copiarlo negli appunti
- Indirizzo IPv6 con un'opzione per copiarlo negli appunti
- Metodo di crittografia corrente

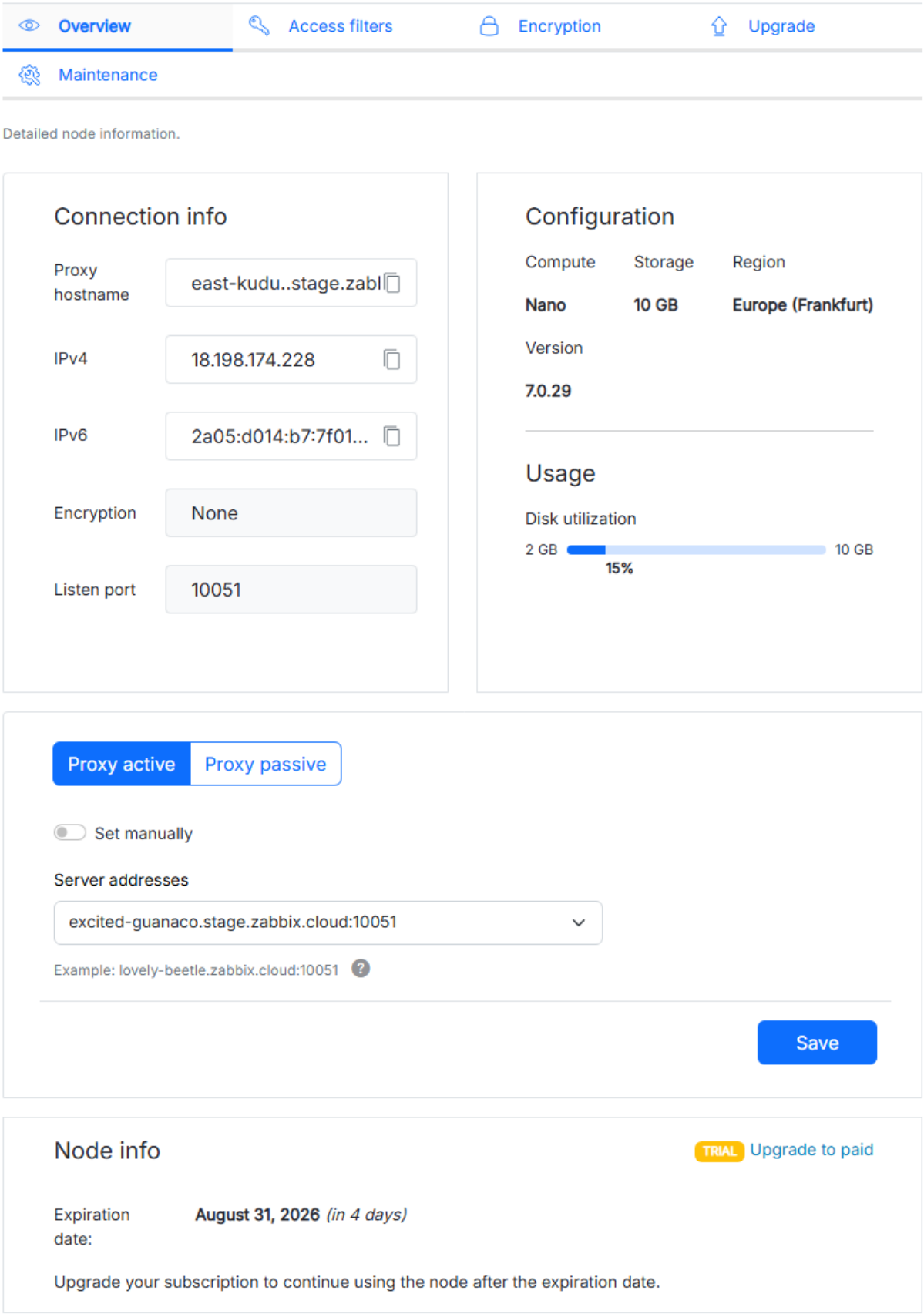
Configuration and usage:

- Livello di elaborazione, spazio di archiviazione massimo, regione
- Versione dei componenti Zabbix
- Utilizzo del disco

Node info/Payment Info:

- Data di scadenza (per i nodi in prova e prepagati)
- Dopo che il nodo completa l'inizializzazione, viene visualizzato un pulsante specifico in base al tipo di nodo: - *Upgrade to paid* - disponibile per i nodi in prova gratuita - *Cancel subscription* - disponibile per i nodi a pagamento con un abbonamento attivo - *Resubscribe to paid* - visualizzato se un abbonamento è stato annullato - Per i nodi prepagati, che hanno una validità prolungata e non possono essere annullati, non viene visualizzato alcun pulsante di azione

Nodo proxy La scheda *Overview* del nodo proxy contiene le seguenti sezioni:



Connection info:

- Nome host del proxy con un'opzione per copiare negli appunti
- Indirizzo IPv4 con un'opzione per copiare negli appunti
- Indirizzo IPv6 con un'opzione per copiare negli appunti
- Metodo di crittografia attualmente in uso
- Porta di ascolto attualmente in uso

Configuration and usage:

- Livello di elaborazione, spazio di archiviazione massimo, regione
- Versione dei componenti Zabbix
- Utilizzo del disco

Proxy mode:

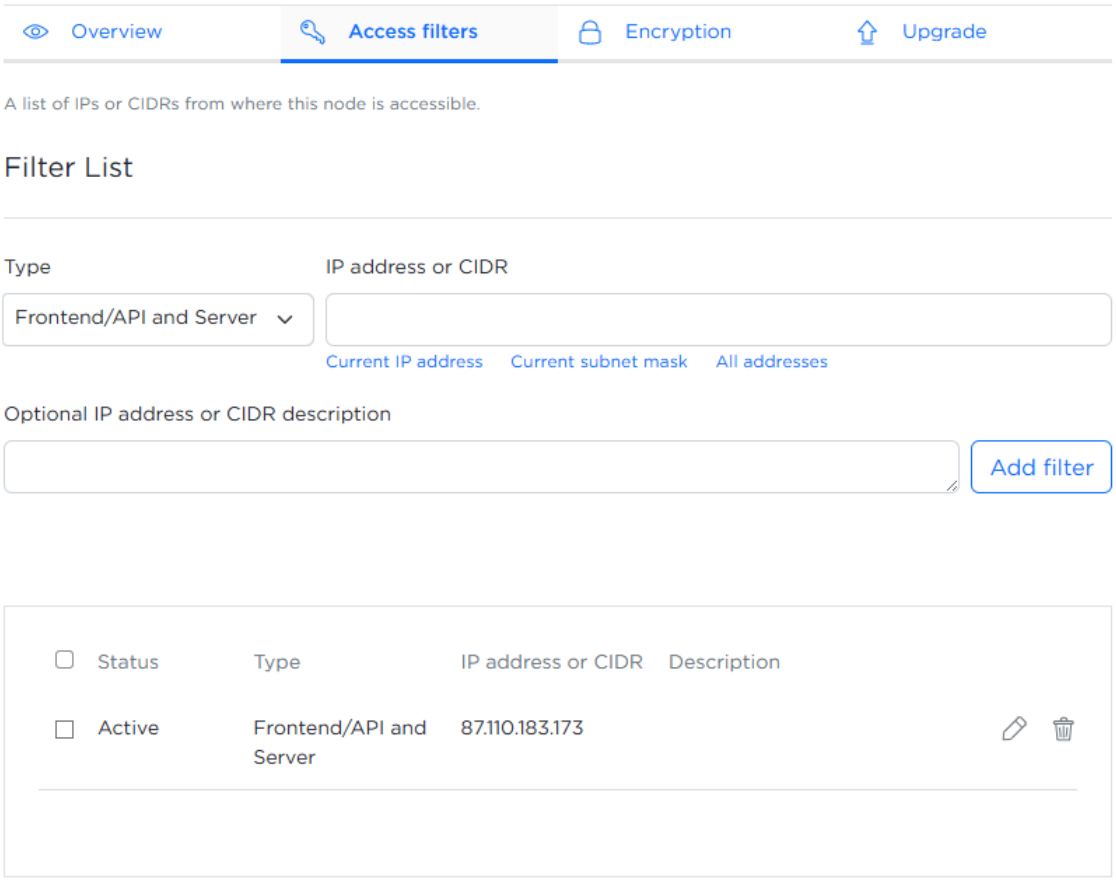
- Passaggio dalla modalità proxy attiva a quella passiva e viceversa
- Selezionare l'indirizzo del server dal menu a discesa *Server addresses* oppure abilitare *Set manually* per inserire manualmente l'indirizzo del server

Node info/Payment Info:

- Data di scadenza (per i nodi di prova e prepagati)
- Dopo il completamento dell'inizializzazione del nodo, viene mostrato un pulsante specifico in base al tipo di nodo: - *Upgrade to paid* - disponibile per i nodi in prova gratuita - *Cancel subscription* - disponibile per i nodi a pagamento con un abbonamento attivo - *Resubscribe to paid* - mostrato se un abbonamento è stato annullato - Per i nodi prepagati, che hanno una validità lunga e non possono essere annullati, non viene visualizzato alcun pulsante di azione

Scheda Filtri di accesso

La scheda *Filtri di accesso* consente di limitare l'accesso al nodo Zabbix in base all'indirizzo IP o alla subnet.



Per configurare un filtro di accesso, compilare i seguenti campi:

- *Tipo* - per i nodi server, utilizzare il menu a discesa per scegliere quale componente Zabbix limitare: solo frontend/API, solo server o entrambi. Per i nodi proxy, è disponibile solo *IP in ingresso*. Definisce quali richieste di rete in ingresso possono raggiungere il proxy.
- *Indirizzo IP o CIDR* - inserire manualmente l'indirizzo IP o il CIDR oppure compilare automaticamente il campo selezionando l'opzione pertinente.
- *Descrizione facoltativa dell'indirizzo IP o del CIDR* - aggiungere una descrizione facoltativa per ogni IP o subnet.

Dopo aver compilato i campi, fare clic su *Aggiungi filtro*. I filtri aggiunti vengono visualizzati nella tabella sottostante, che mostra lo stato, il tipo, l'indirizzo IP o la subnet e la descrizione.

Warning:
Durante l'inizializzazione del nodo, viene creato automaticamente un filtro di accesso per l'indirizzo IP utilizzato. Se in seguito ci si connette da un IP o una rete diversi, è necessario aggiungere tale indirizzo qui, altrimenti l'accesso verrà bloccato.

È possibile:

-  - modificare un filtro.
-  - eliminare un filtro o più filtri utilizzando le caselle di controllo.

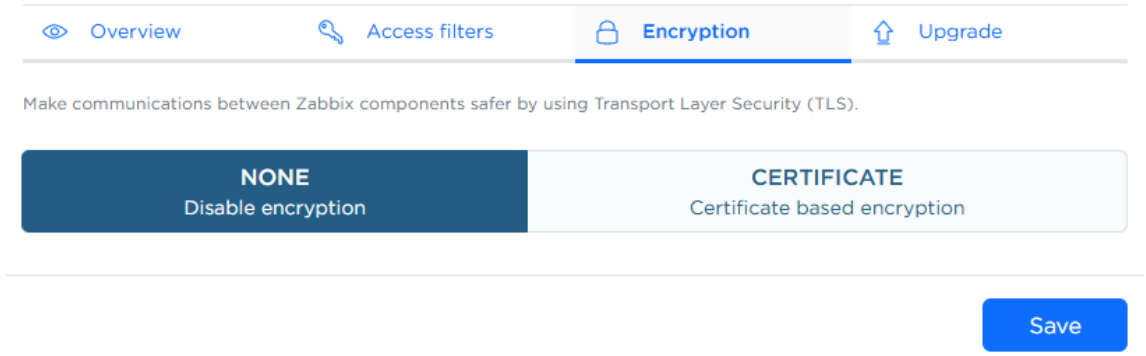
Scheda Crittografia

La scheda *Crittografia* consente di configurare la crittografia TLS per la comunicazione tra i componenti Zabbix caricando i file di certificato richiesti.

Per abilitare la comunicazione sicura tramite certificati, selezionare *CERTIFICATE* e caricare dal computer i seguenti file:

- *Certificati CA radice* - certificati CA attendibili utilizzati per verificare i certificati peer. Fare clic su *Scegli file* e selezionare il file contenente i certificati CA radice attendibili (in genere un file `.crt` o `.pem`, codificato in formato PEM).
- *Catena di certificati* - rappresenta la catena di attendibilità del proprio certificato. Fare clic su *Scegli file* e selezionare il file codificato in formato PEM (ad esempio `.crt` o `.pem`) che include il proprio certificato e gli eventuali certificati CA intermedi.
- *Chiave privata* - la chiave privata corrispondente al proprio certificato. Fare clic su *Scegli file* e selezionare il file `.key` codificato in formato PEM.
- *Certificati revocati* (facoltativo) - se si gestisce una Certificate Revocation List (CRL), fare clic su *Scegli file* e selezionare il file CRL per specificare i certificati che non devono più essere considerati attendibili.

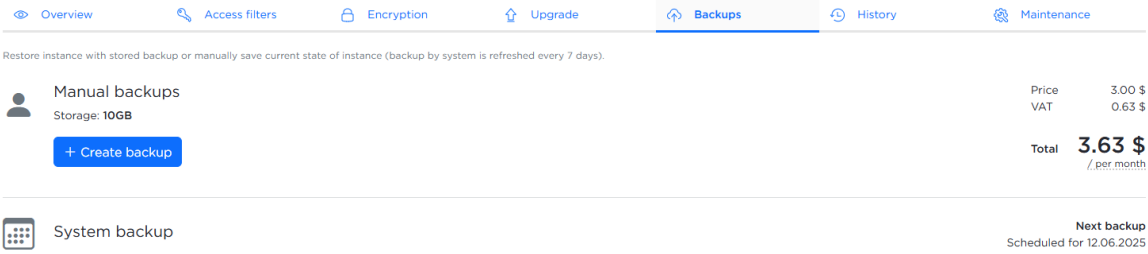
Dopo aver selezionato i file, fare clic su *Salva*. I file verranno caricati, convalidati e applicati per configurare la crittografia TLS tra i componenti Zabbix.



Scheda Backup

La scheda *Backup* consente di visualizzare, creare, ripristinare ed eliminare i backup del proprio nodo Zabbix.

È possibile ripristinare la propria istanza da qualsiasi backup salvato o creare un nuovo backup manuale in qualsiasi momento. La scheda mostra anche la data del prossimo backup pianificato.



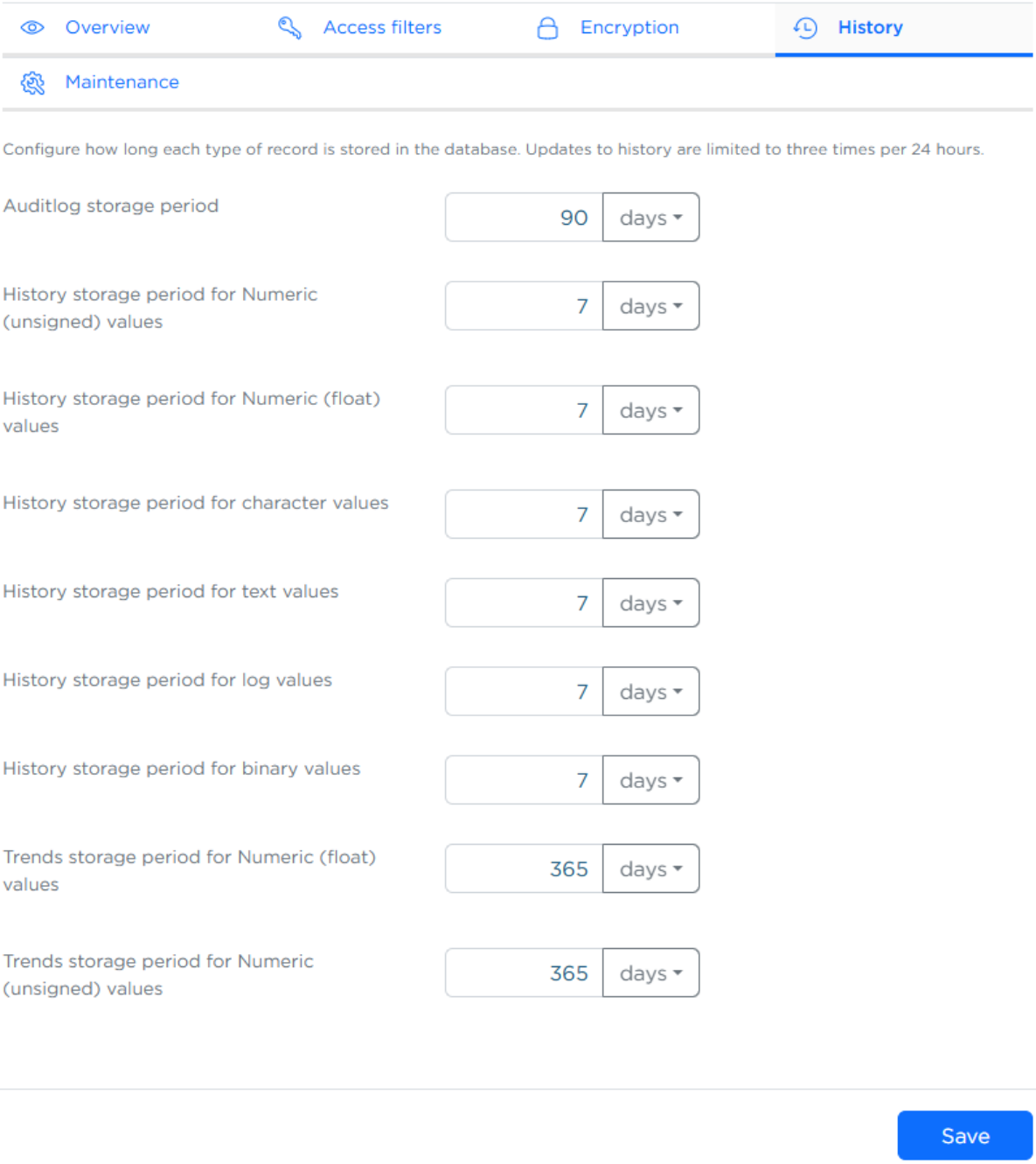
Per creare un backup:

1. Fare clic sul pulsante *Crea backup*.
2. Inserire una descrizione facoltativa nella finestra popup.
3. Fare clic su *Crea* per confermare.

Warning:
I backup aggiuntivi vengono addebitati mensilmente in base all'utilizzo dello spazio di archiviazione.

Scheda Cronologia

La scheda *Cronologia* consente di configurare per quanto tempo vengono conservati i diversi tipi di dati storici.



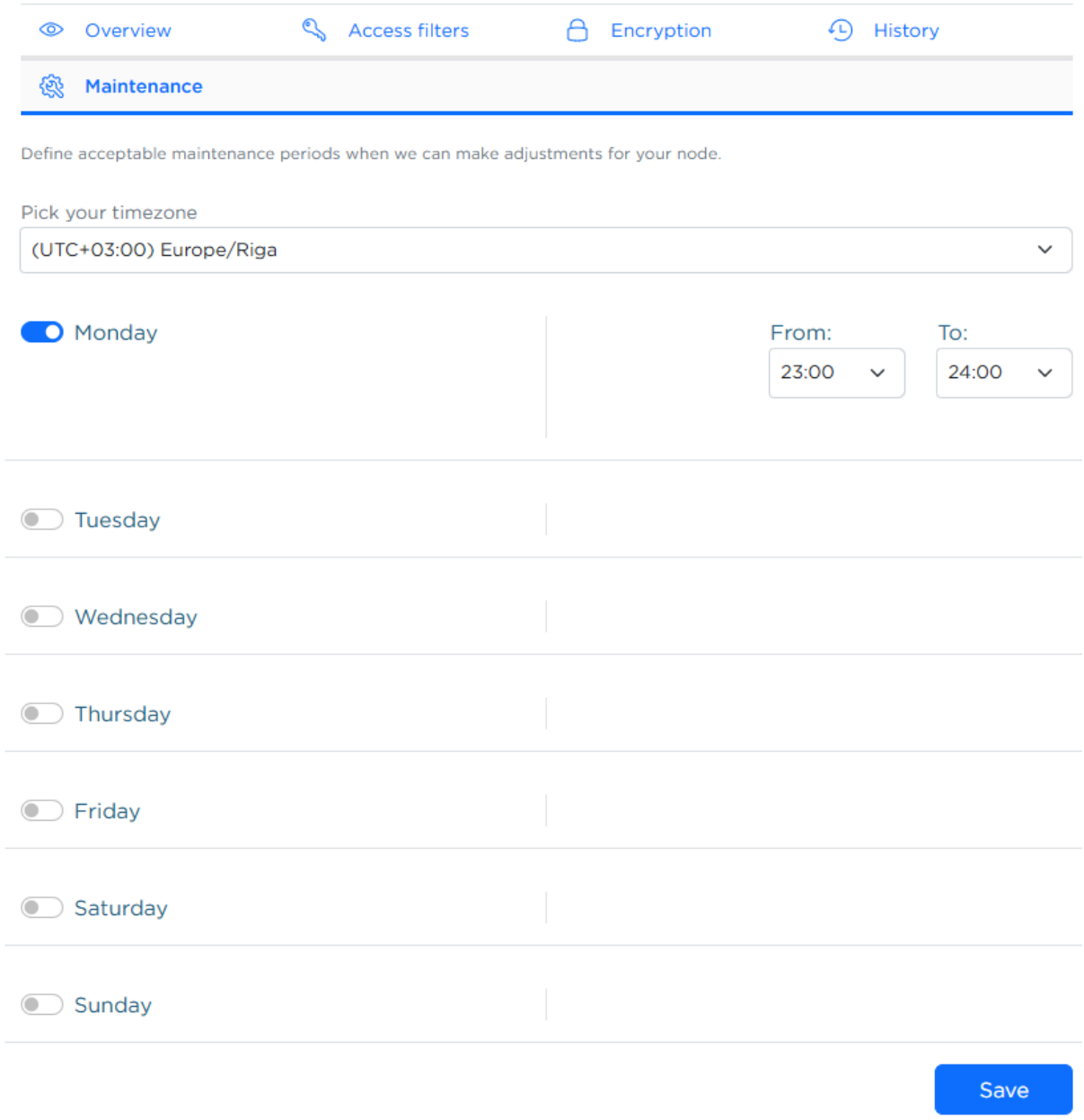
È possibile impostare il periodo di conservazione per ogni tipo di dati (ad esempio, dati numerici, testo e log) e scegliere l'unità di misura (giorni, mesi o anni). Le modifiche possono essere effettuate fino a tre volte ogni 24 ore.

Dopo aver aggiornato i valori, fare clic su *Salva* per applicare le modifiche.

Warning:
Periodi di conservazione più lunghi possono aumentare l'utilizzo del disco e influire sulle prestazioni. Si consiglia di configurare i periodi di conservazione in base ai dati effettivi e alle proprie esigenze di analisi.
 Zabbix Cloud utilizza il partizionamento del database per gestire l'archiviazione dei dati, invece del tradizionale housekeeping di Zabbix. Ciò consente prestazioni più elevate e una gestione della conservazione più prevedibile.

Scheda Manutenzione

La scheda *Manutenzione* consente di definire le finestre di manutenzione consentite durante le quali è possibile eseguire aggiornamenti o modifiche sul proprio nodo.

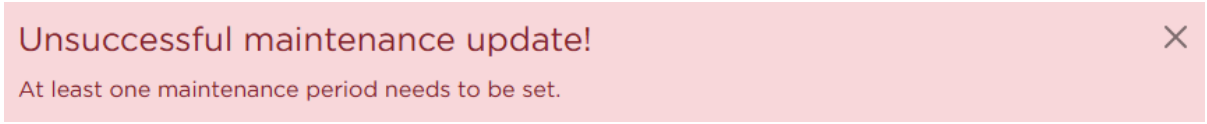


È necessario:

- Selezionare il fuso orario dal menu a discesa.
- Scegliere almeno un giorno e una finestra oraria per la manutenzione.

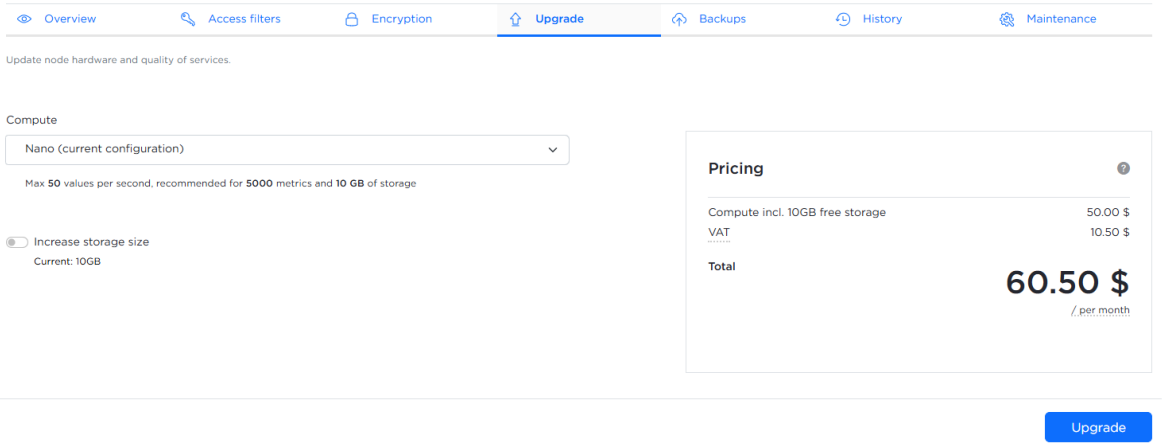
Fare clic su *Salva* per applicare le modifiche.

Se non viene selezionato alcun giorno o alcuna finestra oraria, verrà visualizzato un errore.



Scheda Upgrade

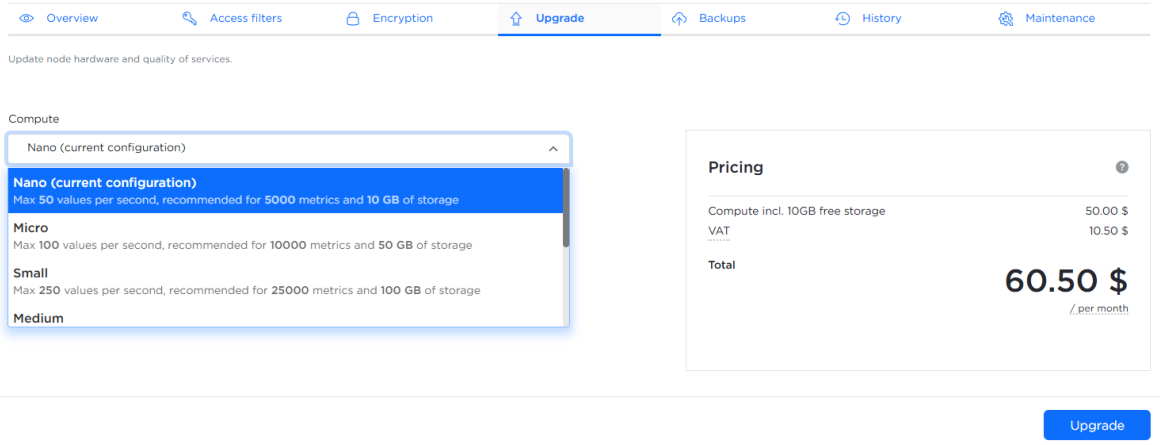
La scheda *Upgrade* consente di aumentare le risorse del proprio nodo o di gestire l'abbonamento.



È possibile aumentare il livello di elaborazione o le dimensioni del disco. Le dimensioni del disco possono essere aumentate solo una volta ogni 24 ore e non possono essere ridotte tramite l'interfaccia utente (contattare il supporto Zabbix per eventuali richieste di riduzione). I nodi in prova gratuita possono essere convertiti in abbonamenti a pagamento.

Per aumentare il livello di elaborazione:

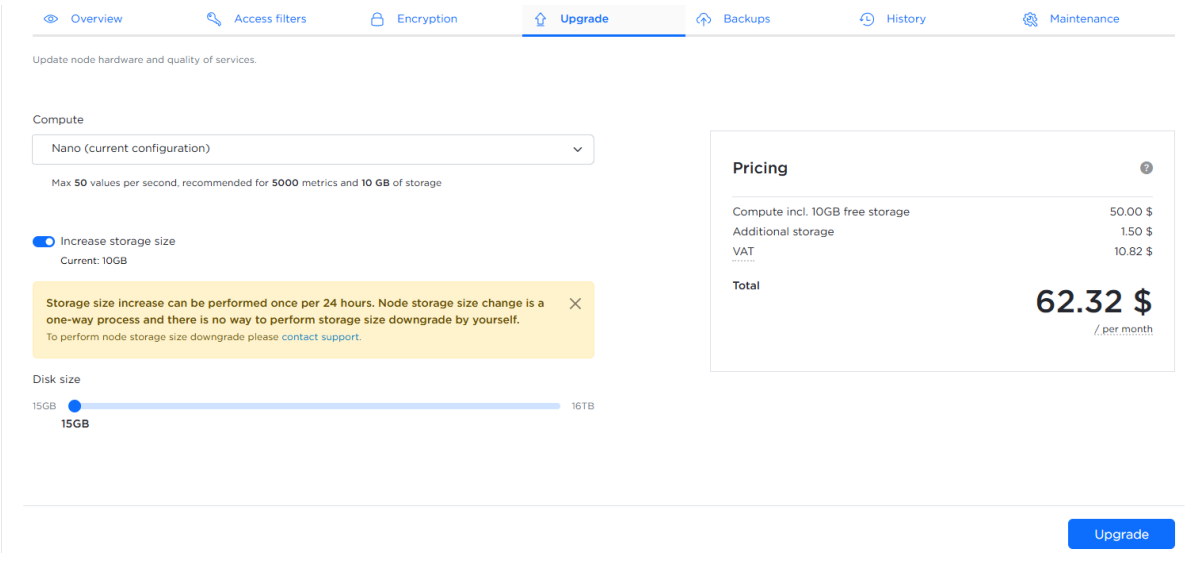
1. Fare clic sul menu a discesa *Compute*.



2. Selezionare il livello di configurazione desiderato.
3. Fare clic sul pulsante *Upgrade*.

Per aumentare le dimensioni del disco:

1. Fare clic sul pulsante di opzione *Increase storage size*.
2. Regolare il cursore *Disk size* sul valore desiderato.



3. Fare clic su *Upgrade*.

Note:
Il nuovo prezzo mensile totale viene mostrato nella tabella *Pricing* prima di confermare le modifiche.

Aggiunta di utenti

Introduzione

Gli esempi seguenti descrivono come aggiungere utenti alla tua organizzazione e ai nodi.

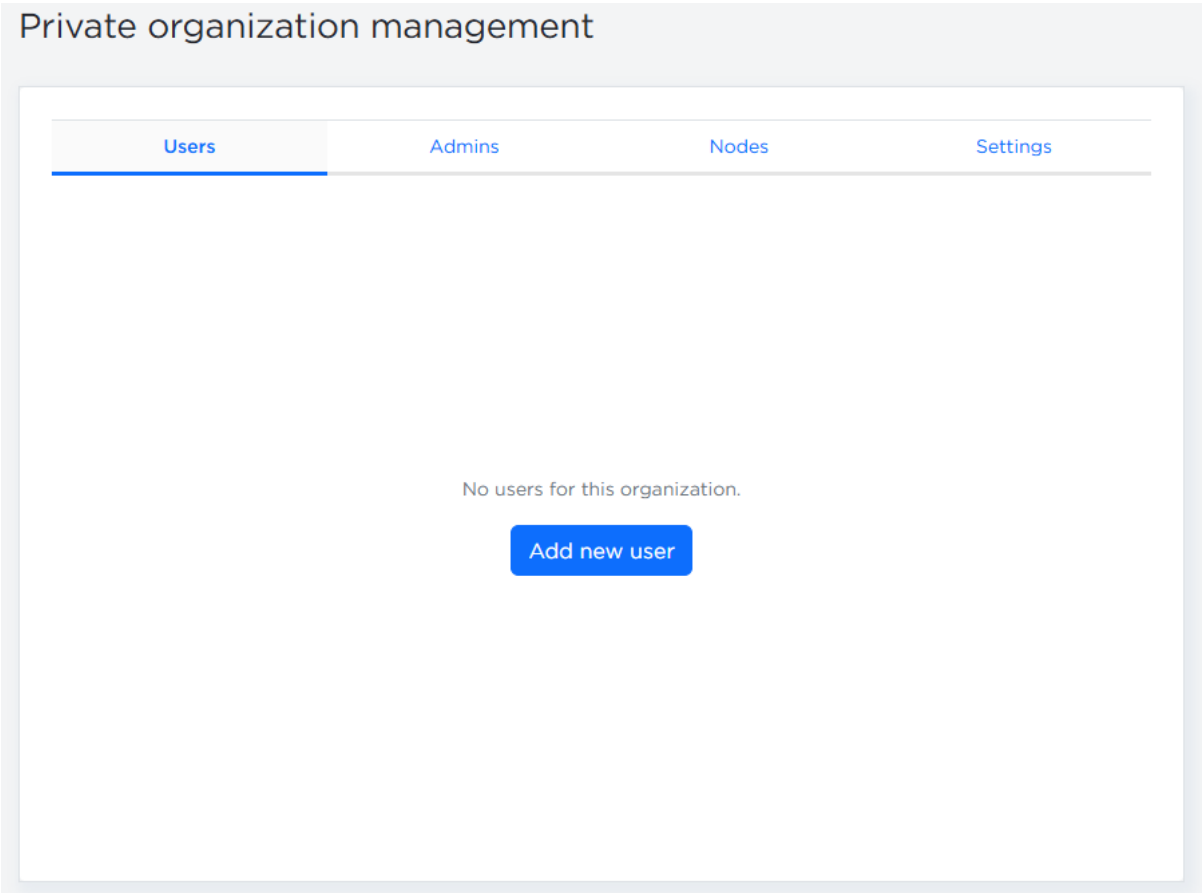
Note:
La copia, la cancellazione e il ripristino della password del frontend possono essere eseguiti solo dal proprietario.

Aggiunta di utenti alle organizzazioni

Analogamente agli utenti in Zabbix, in Zabbix Cloud è possibile creare utenti e assegnare loro ruoli per controllare ciò che possono fare all'interno di un'organizzazione e dei suoi nodi.

Per aggiungere utenti alla tua organizzazione:

1. Apri la pagina *Tutte le organizzazioni*.
2. Fai clic sul pulsante delle impostazioni dell'organizzazione specifica per accedere al relativo menu di gestione.



3. Nella scheda *Utenti* o *Amministratori*, fai clic su *Aggiungi nuovo utente*.
4. Inserisci l'email dell'utente e seleziona un ruolo: *Amministratore* oppure *Utente*.

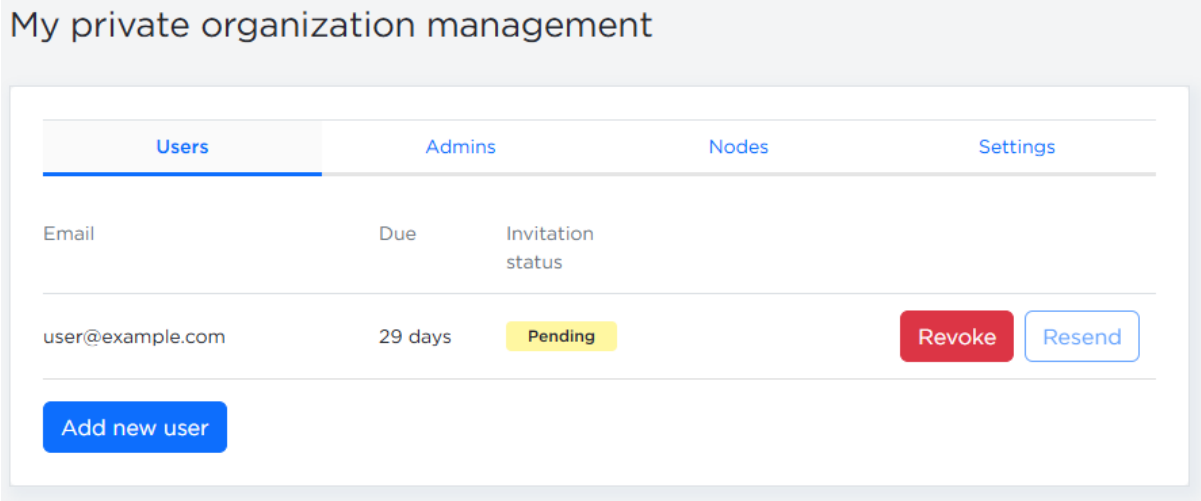
Tieni presente che selezionando il ruolo *Utente* l'utente viene aggiunto all'organizzazione, ma non gli viene concesso l'accesso ad alcun nodo.

Ogni ruolo concede un diverso livello di accesso:

- **Utente** - può accedere ai nodi a cui è assegnato e può eseguire attività di configurazione che non generano addebiti, ad esempio:
  - Gestire i filtri di accesso
  - Configurare la crittografia
  - Impostare i periodi di conservazione di history/trend
  - Pianificare la manutenzione
- **Amministratore** - dispone di tutte le autorizzazioni dell'Utente e può anche eseguire azioni che possono generare addebiti, ad esempio:
  - Creare e ripristinare backup
  - Aumentare o ridurre le risorse del nodo (livello di calcolo e dimensione del disco)
  - Annullare gli abbonamenti
- **Proprietario** - ha il pieno controllo dell'organizzazione. Oltre a tutte le autorizzazioni dell'Amministratore, il Proprietario può:
  - Creare ed eliminare nodi
  - Aggiungere o rimuovere Utenti e Amministratori
  - Rinominare o eliminare l'organizzazione
  - Gestire i dettagli di fatturazione

> **Attention:**
> Il metodo di pagamento del Proprietario viene addebitato per tutti i servizi. Se è richiesta la conferma bancaria, il Proprietario deve autorizzare la transazione, anche se è stata avviata da un Amministratore.

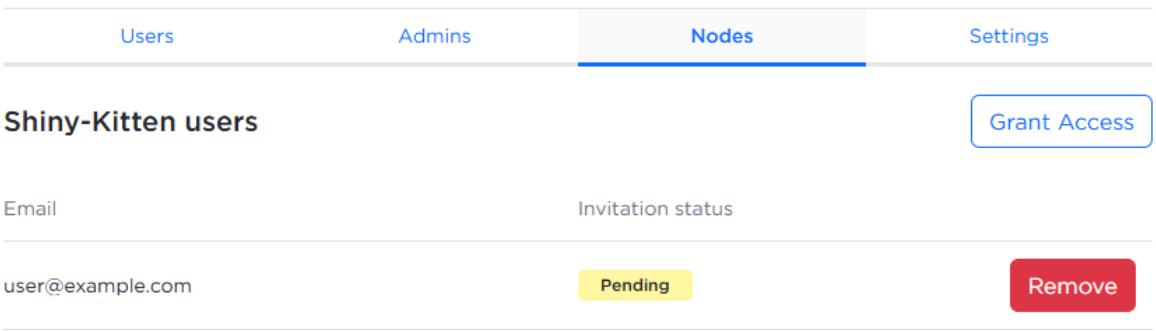
L'utente invitato verrà aggiunto con stato *In sospeso*. Puoi revocare l'invito o inviarlo nuovamente (il reinvio è disponibile una volta ogni 10 minuti: il pulsante sarà disabilitato se questo limite viene raggiunto).



Aggiunta di utenti ai nodi

Per aggiungere l'utente al nodo:

1. Vai alla scheda *Nodi*.
2. Fai clic su *Concedi accesso*.
3. Seleziona dal menu a discesa un utente aggiunto all'organizzazione.
4. Fai clic su *Concedi accesso*.



In alternativa, puoi concedere l'accesso dalla finestra *Configurazione del nodo*:

1. Fai clic sul pulsante delle impostazioni  .
2. Seleziona *Aggiungi utente*.
3. Scegli un utente e fai clic su *Concedi accesso*.

Gli utenti invitati devono accettare l'invito creando un account Zabbix Cloud utilizzando lo stesso indirizzo email a cui è stato inviato l'invito. Se l'utente dispone già di un account Zabbix Cloud associato a quell'indirizzo email, deve accedere e confermare l'invito. Gli inviti sono associati a uno specifico indirizzo email e non possono essere accettati da un account diverso.

Differenze principali di Zabbix Cloud

Introduzione

Zabbix Cloud è una piattaforma basata sul cloud che consente ai suoi utenti di eseguire istanze Zabbix autonome nel cloud (nodi cloud).

Rispetto a una normale distribuzione Zabbix on-premise:

- Non è richiesta alcuna installazione o manutenzione.
- I nodi cloud sono completamente gestiti da Zabbix.
- I nodi cloud possono essere avviati in pochi minuti.

Questo servizio è offerto con un modello pay-as-you-go (è disponibile una **prova gratuita**).

Tabella di confronto Sebbene la maggior parte delle funzionalità sia comune tra le installazioni Zabbix on-premise e i nodi cloud, esistono anche alcune differenze. Di seguito è riportata una tabella di confronto riepilogativa. Le sezioni successive descrivono in dettaglio alcuni elementi selezionati.

Funzionalità / Aspetto	Zabbix Cloud	Zabbix on-premise
Ciclo di rilascio	In linea con le release LTS	È possibile scegliere la versione di Zabbix (stabile, beta, build personalizzate)
Facilità d'uso	Piattaforma pronta all'uso, completamente gestita da Zabbix	Richiede installazione, configurazione e manutenzione
Applicazione delle patch alle vulnerabilità	Applicazione automatica delle patch al sistema operativo e a Zabbix	Applicazione manuale delle patch da parte dell'utente
Certificato HTTPS	Certificato valido pronto all'uso (senza Let's Encrypt)	Autofirmato per impostazione predefinita (è possibile configurare Let's Encrypt o un'altra CA)
Prestazioni del DB (INSERT/UPDATE/SELECT)	Partizionamento automatico per la massima velocità; nessuna operazione DELETE di lunga durata	Partizionamento e pulizia manuali (sono possibili transazioni DELETE di lunga durata)
Ottimizzazione delle prestazioni	Preconfigurata dal team Zabbix Cloud (senza changelog)	Piena visibilità e controllo sui parametri di ottimizzazione
Controllo del firewall	Gestito dal team Zabbix Cloud; le porte GUI/trapper non sono esposte pubblicamente	Pieno controllo sulle regole del firewall e sull'esposizione delle porte
Traffico in uscita sulla porta 25	Bloccato per impostazione predefinita per prevenire spam e abusi; utilizzare invece la porta SMTP 587 o 465	Non limitato da Zabbix; dipende dalla rete e dal provider di hosting
Isolamento e protezione del DB	Altamente isolato, senza accesso diretto al DB/SSH	Dipende dalla rete e dalla sicurezza del proprio host
Alta disponibilità (HA)	Eseguito su una piattaforma cloud ad alta disponibilità, è possibile utilizzare proxy Zabbix per una maggiore resilienza della raccolta dati	HA gestita dall'utente
SNMP trap	Solo tramite un proxy dedicato (nessun ricevitore trap HA/bilanciato)	Direttamente o tramite proxy; sono possibili HA e bilanciamento del carico
Polling SNMP	Difficile senza proxy (ogni dispositivo richiede NAT/una porta personalizzata)	Polling SNMP nativo; proxy facoltativo

Funzionalità / Aspetto	Zabbix Cloud	Zabbix on-premise
Impostazioni di conservazione	Solo History/Trends/Audit tramite l'interfaccia utente (nessuna API o file di configurazione)	Configurabile nel file di configurazione del server Zabbix o tramite API
Script personalizzati	`AlertScriptsPath`, `ExternalScripts`, moduli frontend e moduli della community non supportati	Completamente supportati (percorso di script, moduli, integrazioni)
Monitoraggio ODBC	Solo PostgreSQL (driver `{postgresql}`); il plugin MariaDB è presente come segnaposto	ODBC per PostgreSQL, MySQL, Oracle, ecc. - completamente configurabile
Limitazione delle chiamate ODBC	Non possibile (solo `StartODBCPollers=1`); le query di sincronizzazione pesanti bloccano le altre attività	Numero di poller e pianificazione completamente regolabili
Caricamento del certificato SAML	Non supportato	Supportato tramite interfaccia utente o API
Report pianificati	È necessario creare un media Email personalizzato (i media script non sono supportati)	Supporta sia i media script sia i media Email pronti all'uso
Interfaccia host per i controlli attivi	Indirizzo IP assegnato dalla piattaforma (creato automaticamente)	Interfacce gestite dall'utente; gli indirizzi IP sono sotto il suo controllo

Differenze delle funzionalità Trap SNMP

Le trap SNMP sono supportate solo tramite un proxy Zabbix dedicato. Se è richiesto il monitoraggio dei trap SNMP, non è possibile utilizzare il bilanciamento del carico automatico o l'alta disponibilità per il proxy, perché i trap SNMP devono essere inviati a un singolo indirizzo IP.

Polling SNMP

Senza proxy, il polling SNMP richiede che ogni dispositivo sia esposto tramite NAT e porte personalizzate.

Impostazioni di conservazione

I periodi di conservazione di storico, trend e log di audit possono essere configurati nel cloud solo tramite l'interfaccia web. Queste impostazioni non possono essere configurate tramite `zabbix_server.conf` o l'API.

Le sostituzioni manuali della conservazione dello storico per singolo item non sono supportate (il partizionamento è controllato globalmente). Gli URL dei nodi cloud non accettano gli stessi parametri di query delle installazioni on-premise.

Script personalizzati

I seguenti tipi di script personalizzati non sono supportati da Zabbix Cloud:

- Script di avviso (`AlertScriptsPath`)
- Script esterni (`ExternalScripts`)
- Script del frontend su Zabbix Server

I moduli del frontend sviluppati dalla community non possono essere installati.

Monitoraggio ODBC

Zabbix Cloud supporta il monitoraggio ODBC solo per PostgreSQL. Utilizzare il template ODBC ufficiale di Zabbix e definire la stringa di connessione nel template usando:

```
Driver={postgresql}
```

È inoltre installato un plugin MariaDB, ma al momento non è funzionante. Se utilizzato, definire:

```
Driver={mysql}
```

Attention:
Non esiste alcuna configurazione nota funzionante per Oracle MySQL 8.0 in Zabbix Cloud. Sebbene query semplici (come `SELECT 1`) possano avere esito positivo, query più complesse restituiscono `SQL_ERROR`.

 Non è possibile limitare le chiamate ODBC (`StartODBCPollers=1` soltanto). Una reportistica sincrona pesante può influire sulle prestazioni e può essere eseguita una sola `SELECT` alla volta.

Accesso all'infrastruttura

Zabbix Cloud non fornisce accesso SSH ai nodi sottostanti, né consente connessioni dirette al database (ad esempio, alla porta 3306). Tutte le attività di configurazione, monitoraggio e risoluzione dei problemi devono essere eseguite tramite l'interfaccia utente cloud o l'API, garantendo che i livelli del sistema operativo e del database rimangano isolati e protetti.

Traffico in uscita sulla porta 25

Per prevenire spam e abusi, le connessioni in uscita dai nodi Zabbix Cloud verso la porta TCP 25 sono bloccate per impostazione predefinita. Questo riguarda:

- I tipi di media Email che inviano tramite un server SMTP sulla porta 25. Poiché 25 è la *porta del server SMTP* predefinita per un nuovo tipo di media Email, modificarla impostandola su 587 (con *Sicurezza della connessione* impostata su STARTTLS) oppure su 465 (SSL/TLS), a seconda di ciò che supporta il server SMTP. Il provider email *Office365 relay* non può essere utilizzato, poiché si connette all'endpoint relay di Microsoft sulla porta 25.
- I report pianificati, che richiedono un tipo di media Email.
- I controlli eseguiti direttamente dal nodo cloud sulla porta 25, come i controlli semplici `net.tcp.service[smtp]` o i controlli SMTP nelle regole di network discovery. Questi segnalano il servizio SMTP come non disponibile anche quando è in esecuzione; eseguire invece tali controlli tramite un proxy Zabbix o un agent Zabbix nella propria rete.

Il tipo di media *Cloud Email* predefinito non è interessato.

Interfaccia di active check

Gli active checks nel cloud creano automaticamente un'interfaccia host con un IP non correlato alla tua rete on-premise. Per impostazione predefinita, questo IP può essere esterno. Per mantenere la coerenza, puoi eliminare o modificare manualmente l'interfaccia host dopo la creazione.

Caricamento del certificato SAML

L'autenticazione SAML non è supportata perché pochi provider accettano messaggi SAML non crittografati o non firmati.

Report pianificati

Il tipo di media *Cloud Email* predefinito utilizza un trasporto tramite script e non può essere usato per i report pianificati. Per inviare i report tramite email:

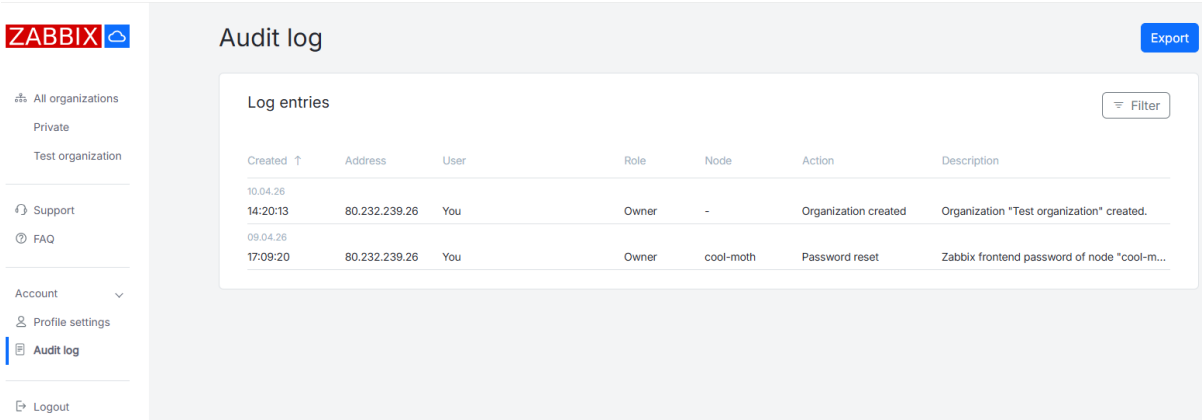
1. Creare un nuovo tipo di media Email.
2. Assegnare il media email al proprio utente in *Impostazioni utente* > *Media*.
3. Selezionare questo tipo di media durante la configurazione dei report pianificati.

Attention:
Non utilizzare la porta 25 come *porta del server SMTP* per questo tipo di media; vedere Traffico in uscita sulla porta 25.

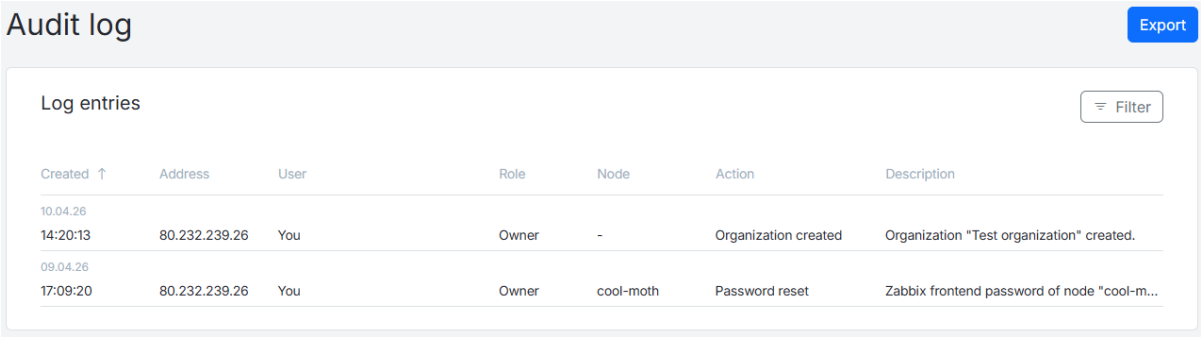
Registro di audit

Panoramica In Zabbix Cloud, il registro di controllo registra le attività degli utenti e del sistema.

Dopo aver effettuato l'accesso a un account Zabbix Cloud, vai su *Account* > *Audit log* per visualizzare i record del registro di controllo. Le voci del registro di controllo disponibili per un utente dipendono dal ruolo dell'utente nell'organizzazione.



Visualizzazione dei record del registro di audit I record del registro di audit sono visualizzati in una tabella.



La tabella contiene le seguenti informazioni:

Column	Description
Created	Data e ora in cui è stata creata la voce del registro. L'ora visualizzata corrisponde al fuso orario dell'utente selezionato nel menu a discesa *User timezone* in *Account > Profile settings*.
User	Nome dell'utente dell'organizzazione che ha eseguito l'azione.
Role	Il ruolo dell'utente che ha eseguito l'azione.
Node	Nome del nodo associato all'azione.
Action	Nome dell'azione eseguita.
Description	Descrizione dell'azione eseguita.

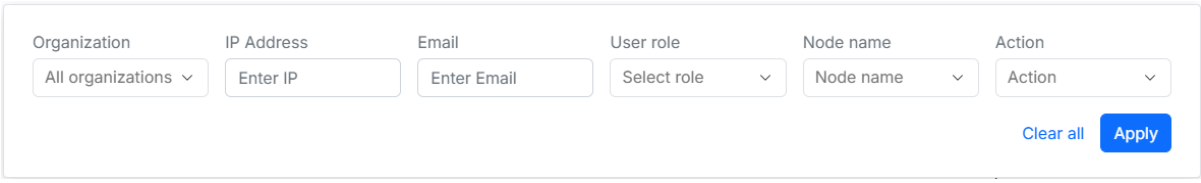
I nomi delle colonne possono essere selezionati per ordinare i record in ordine crescente o decrescente.

I record del registro di audit filtrati possono essere esportati in formato `.csv` usando il pulsante *Export*.

Utilizzo dei filtri

Utilizzare il pulsante *Filter* per aprire il filtro.

Per visualizzare i risultati filtrati, fare clic su *Apply* dopo aver selezionato uno o più parametri di filtro. Per cancellare il filtro, fare clic su *Clear all*.



È possibile filtrare i record in base a uno o più parametri. Sono disponibili i seguenti parametri di filtro:

Parametro	Descrizione
Organization	Selezionare tutte le organizzazioni o un'organizzazione specifica.
IP Address	Specificare un indirizzo IP.
Email	Specificare un indirizzo email associato all'utente.
User role	Selezionare tutti i ruoli o uno specifico: Owner, Admin o User.
Node name	Selezionare tutti i nodi o uno specifico.
Action	Selezionare un tipo di azione specifico.

Il registro di audit tiene traccia delle seguenti azioni:

- *Node creato*
- *Node eliminato*
- *Node aggiornato*
- *Versione del node aggiornata*
- *Password reimpostata*
- *Manutenzione aggiornata*
- *Impostazione della cronologia aggiornata*
- *Data di scadenza del node estesa*
- *Configurazione del proxy aggiornata*
- *Backup creato*
- *Backup eliminato*

- *Backup ripristinato*
- *Backup aggiornato*
- *Filtro di accesso aggiunto*
- *Filtro di accesso aggiornato*
- *Filtro di accesso eliminato*
- *Crittografia abilitata*
- *Crittografia disabilitata*
- *Abbonamento annullato*
- *Abbonamento rinnovato*
- *Pagamento dell'abbonamento non riuscito*
- *Il pagamento richiede conferma*
- *Organizzazione creata*
- *Organizzazione eliminata*
- *Admin invitato*
- *Utente invitato*
- *Admin entrato*
- *Utente entrato*
- *Admin rimosso dall'organizzazione*
- *Utente rimosso dall'organizzazione*
- *Node aggiunto all'organizzazione*
- *Node rimosso dall'organizzazione*
- *Accesso al node concesso*
- *Accesso al node revocato*
- *Impostazioni utente aggiornate*
- *Pagamento creato*
- *Pagamento completato*
- *Pagamento non riuscito*
- *Addebito IVA applicato*
- *Convalida IVA non riuscita*
- *Informazioni di fatturazione aggiornate*
- *Metodo di pagamento aggiornato*
- *Metodo di pagamento rimosso*
- *Node prepagato creato*
- *Node prepagato aggiornato*
- *Incidente iniziato*
- *Incidente terminato*

Configurare un proxy Zabbix Cloud

Panoramica Questa pagina descrive come configurare un proxy Zabbix distribuito su Zabbix Cloud per raccogliere dati di monitoraggio dagli host in un ambiente remoto.

La procedura è destinata principalmente a un server Zabbix e a un proxy creati su Zabbix Cloud.

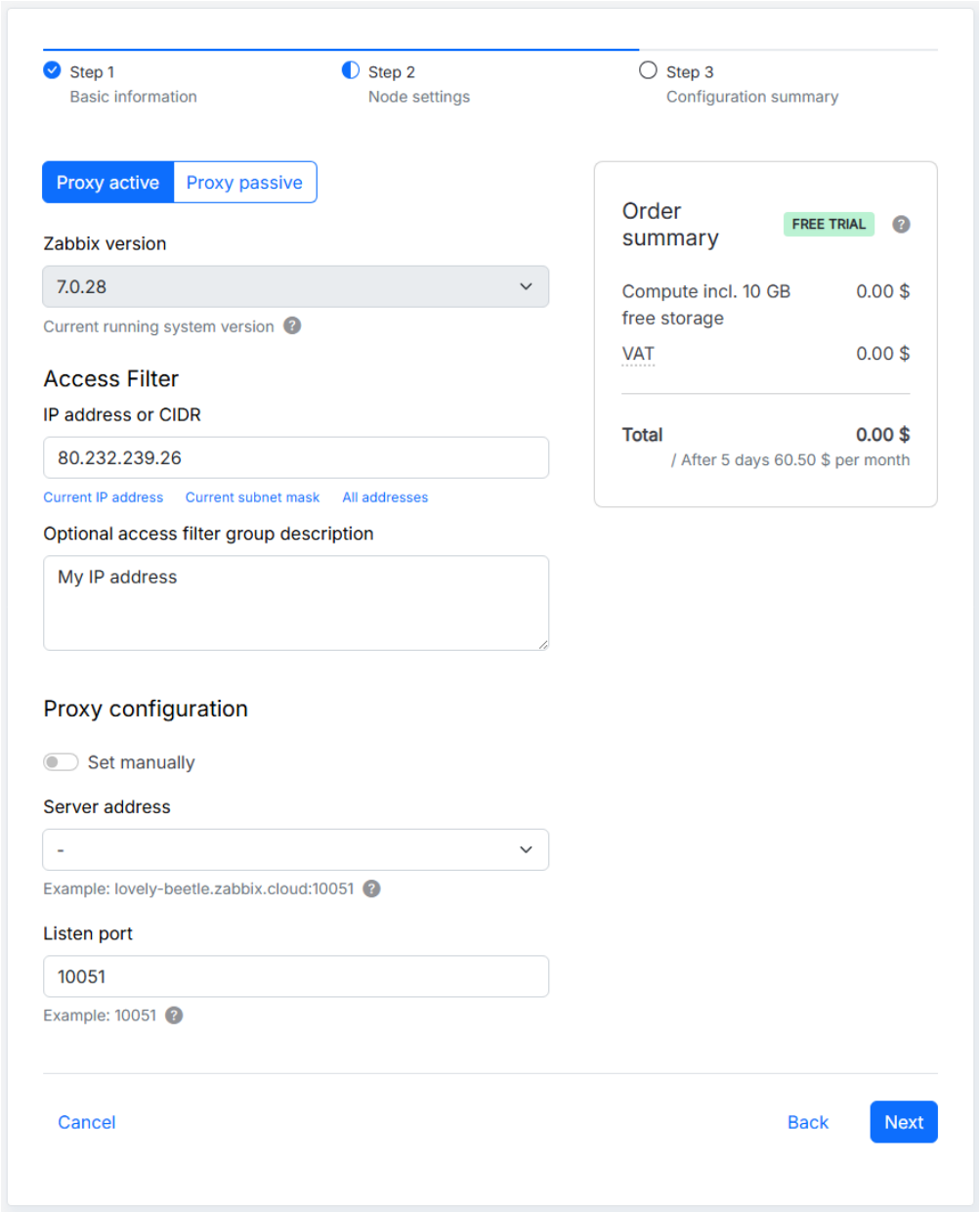
La procedura descrive anche la configurazione necessaria quando il server Zabbix non è ospitato su Zabbix Cloud.

Prerequisiti Un server Zabbix è installato, configurato e in esecuzione, oppure un nodo server Zabbix è in esecuzione su Zabbix Cloud.

Creare un nodo proxy Creare un nodo proxy Zabbix Cloud.

Durante la creazione del nodo:

1. Selezionare *Proxy*.
2. Selezionare *Proxy active* o *Proxy passive* a seconda di come il proxy deve comunicare con il server Zabbix.
3. Configurare l'indirizzo IP del nodo o i filtri di accesso CIDR (è possibile farlo dopo la creazione del nodo).
4. Specificare il server Zabbix a cui il proxy si connetterà.



5. Completare il processo di creazione del nodo.

Per ulteriori informazioni sulla creazione di un nodo proxy, vedere Distribuire Zabbix nel cloud.

Configurare i filtri di accesso Zabbix Cloud utilizza i filtri di accesso per controllare quali indirizzi IP o subnet possono connettersi a un nodo cloud.

Durante l'inizializzazione del nodo, viene creato automaticamente un filtro di accesso per l'indirizzo IP utilizzato per accedere al nodo. Questo non include necessariamente gli indirizzi degli host monitorati o di altri componenti Zabbix. Aggiungere manualmente gli indirizzi necessari.

Per ulteriori informazioni, vedere Configurazione del nodo — Filtri di accesso.

Nodo proxy

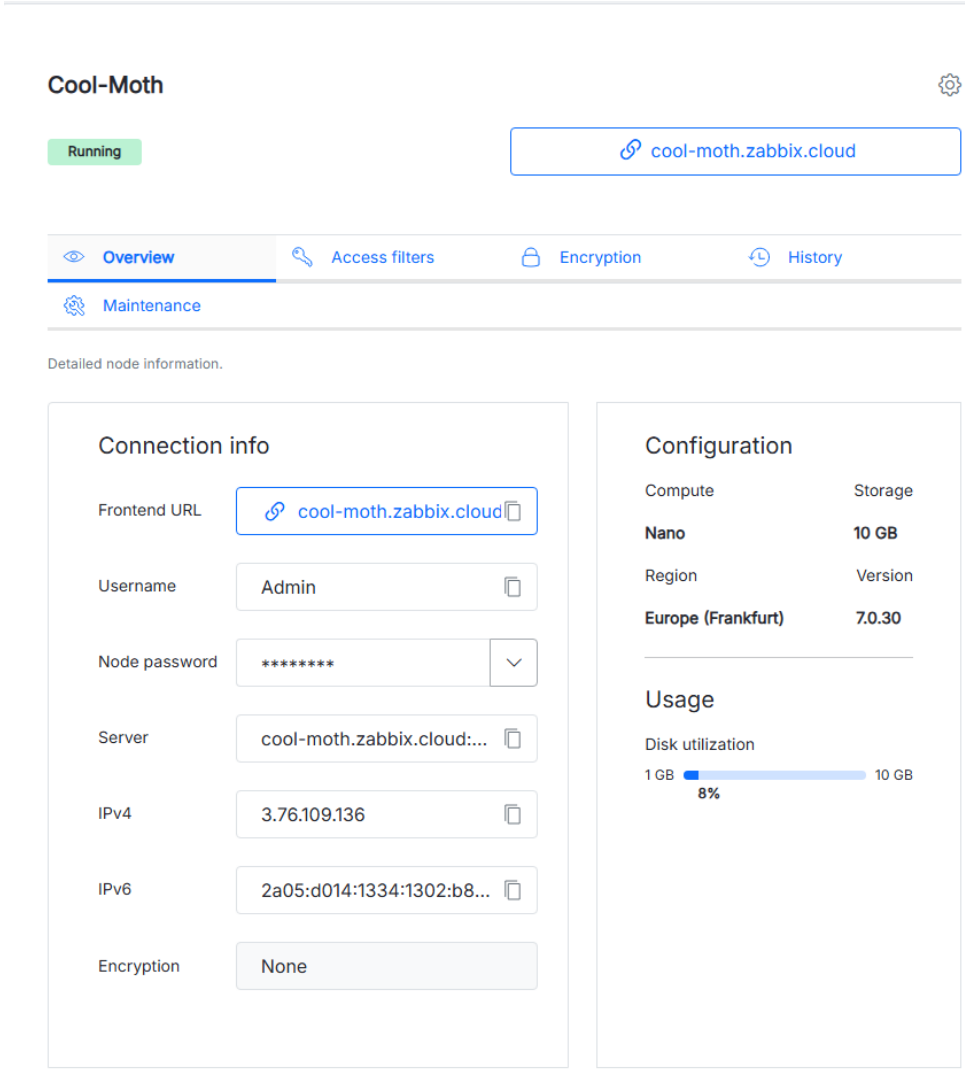
Aggiungere gli indirizzi IP o le subnet da cui si prevedono connessioni al proxy.

Se gli host monitorati utilizzano controlli attivi dell'agent, gli agent avviano le connessioni al proxy; pertanto, l'indirizzo IP o la subnet degli host monitorati deve essere consentito dai filtri di accesso del nodo proxy.

Se *Proxy Passive* è stato impostato come modalità proxy, il server Zabbix si connette al proxy; pertanto, l'indirizzo IP o la subnet del server Zabbix deve essere consentito dai filtri di accesso del nodo proxy.

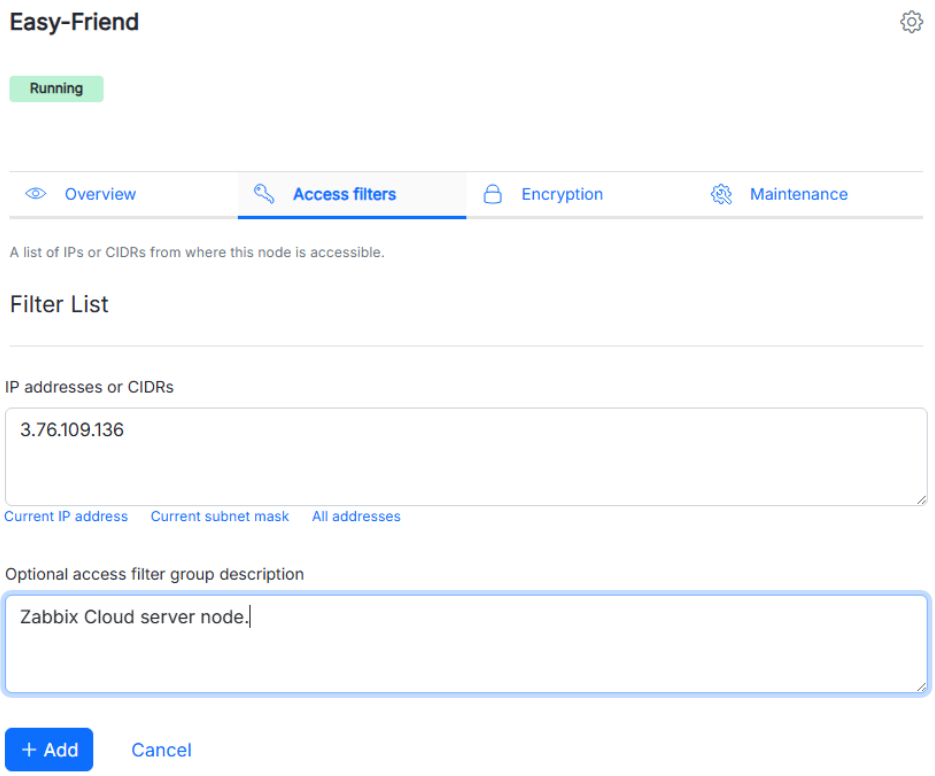
Se è stato selezionato *Proxy passive* e viene utilizzato un nodo server:

1. Accedere alla scheda *Overview* del nodo server e copiare l'indirizzo IPv4 o IPv6 del nodo.

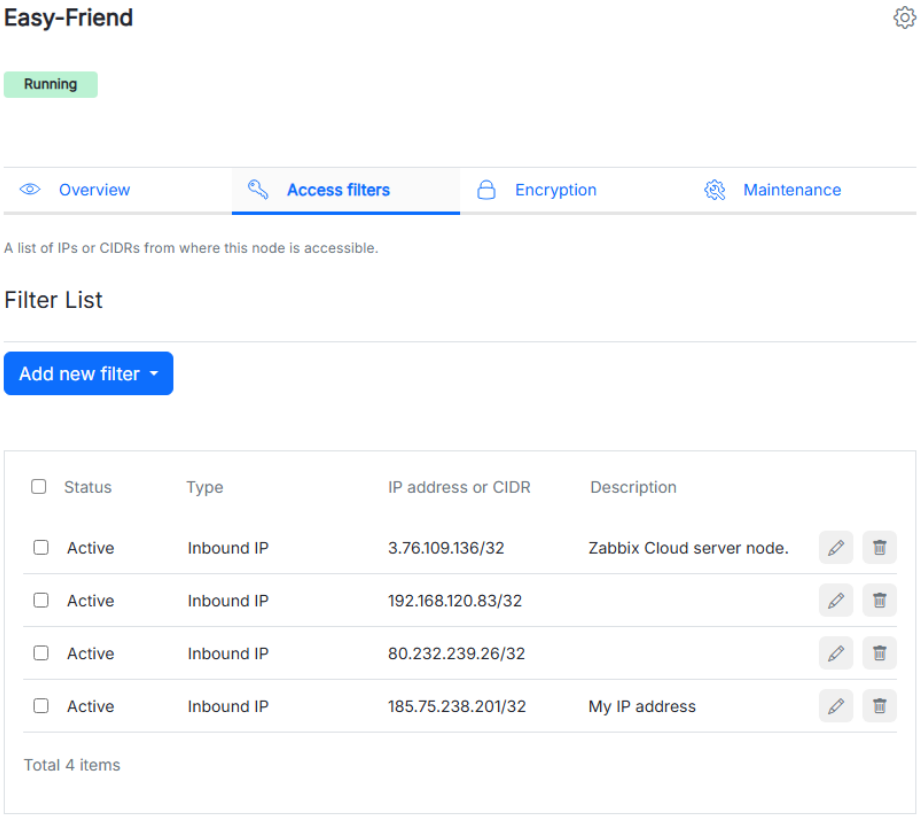


2. Accedere alla scheda *Access filters* del nodo proxy, quindi fare clic su *Add new filter* > *Add*.

3. Incollare l'indirizzo copiato nel campo *IP addresses or CIDRs* e aggiungere una descrizione nel campo *Optional access filter group description*.



4. Fare clic su *Add*.



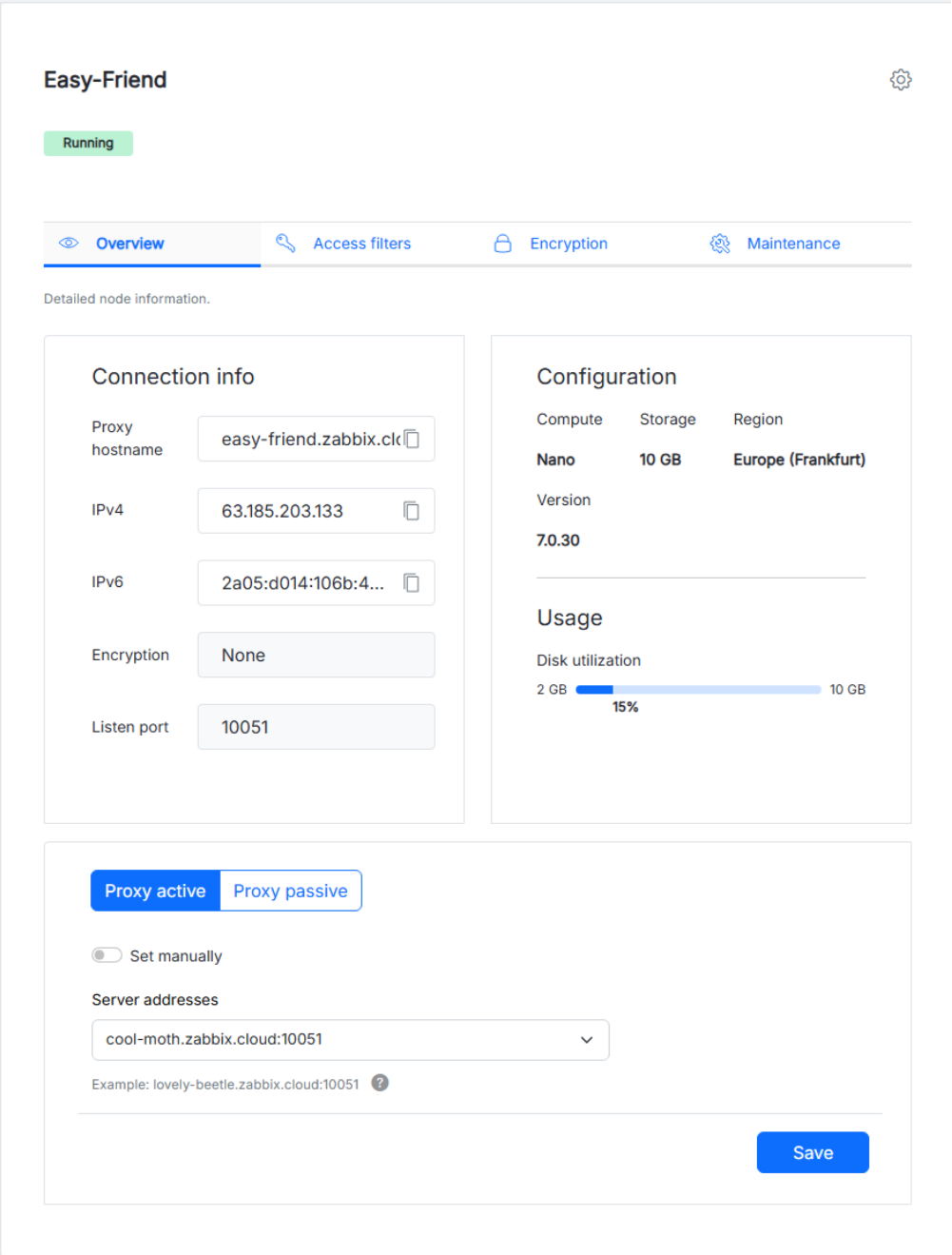
Ripetere i passaggi 2-4 per l'indirizzo IP di un agent Zabbix che utilizza controlli attivi.

Nodo server

Se il server Zabbix è ospitato su Zabbix Cloud, configura i filtri di accesso anche sul nodo server.

Per un proxy in modalità attiva, aggiungi l'indirizzo IP o il CIDR del nodo proxy ai filtri di accesso del nodo server, poiché è il proxy a iniziare la connessione al server.

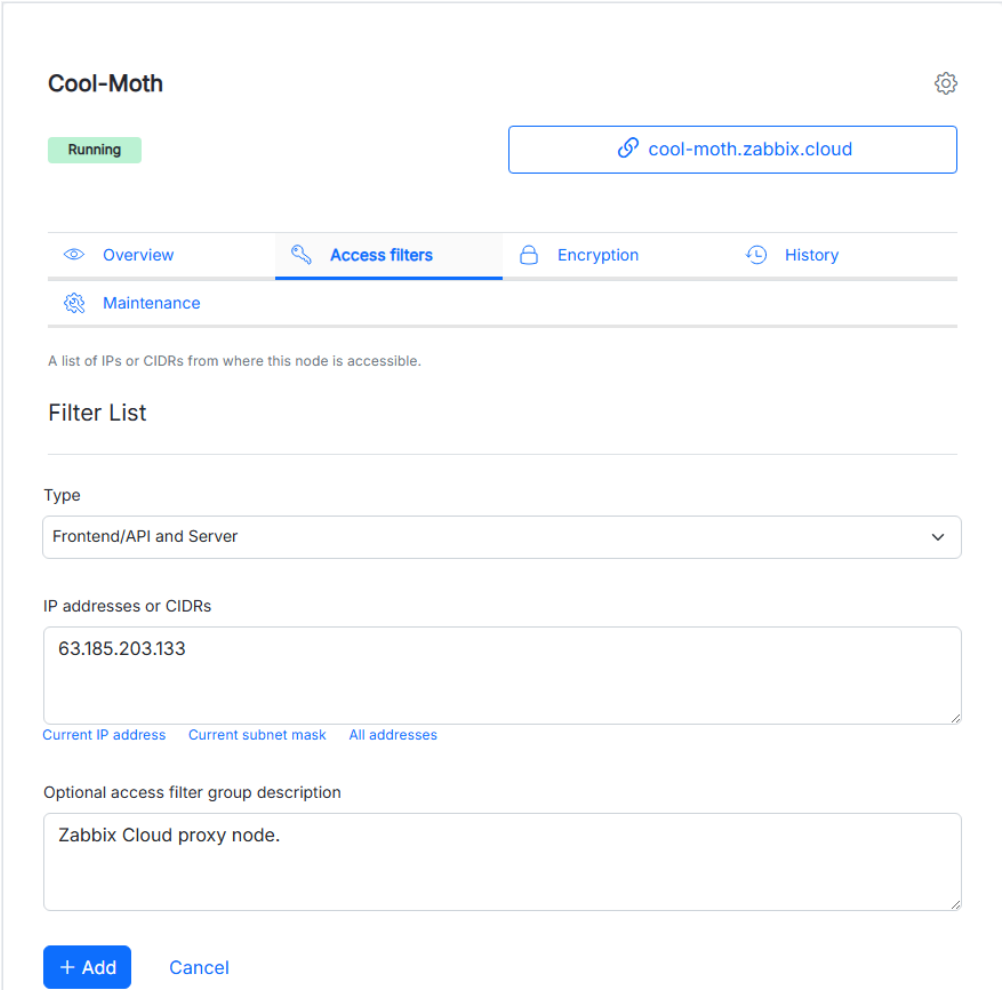
1. Vai alla scheda *Overview* del nodo proxy e copia l'indirizzo IPv4 o IPv6 del nodo.



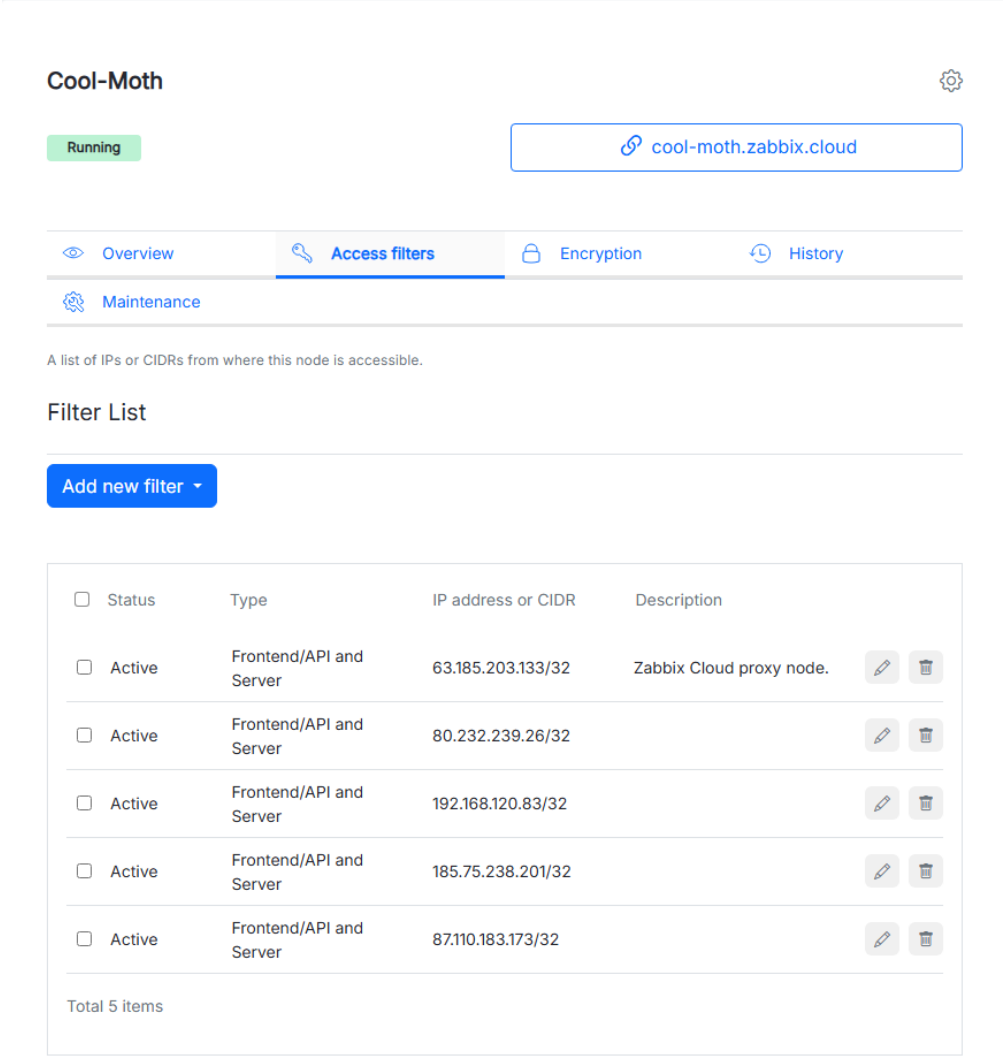
2. Vai alla scheda *Access filters* del nodo server e fai clic su *Add new filter* > *Add*.

3. Incolla l'indirizzo copiato nel campo *IP addresses or CIDRs* e aggiungi una descrizione nel campo *Optional access filter group description*.

4. Seleziona il tipo di filtro appropriato.



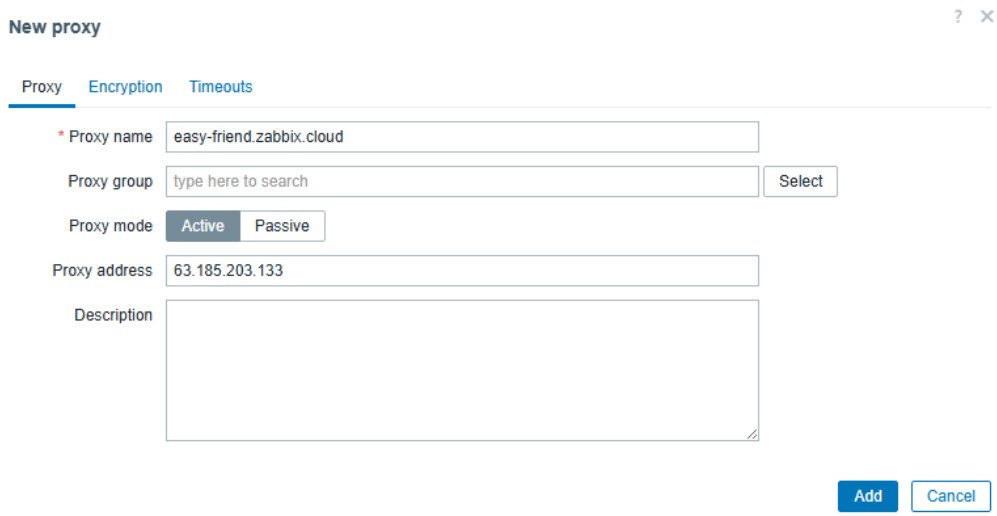
5. Fai clic su *Add*.



Configurare il proxy Il proxy deve essere configurato nel frontend di Zabbix per comunicare con il server Zabbix.

Aprire *Amministrazione* > *Proxy* nel frontend di Zabbix, fare clic su *Crea proxy* e configurare il proxy come richiesto:

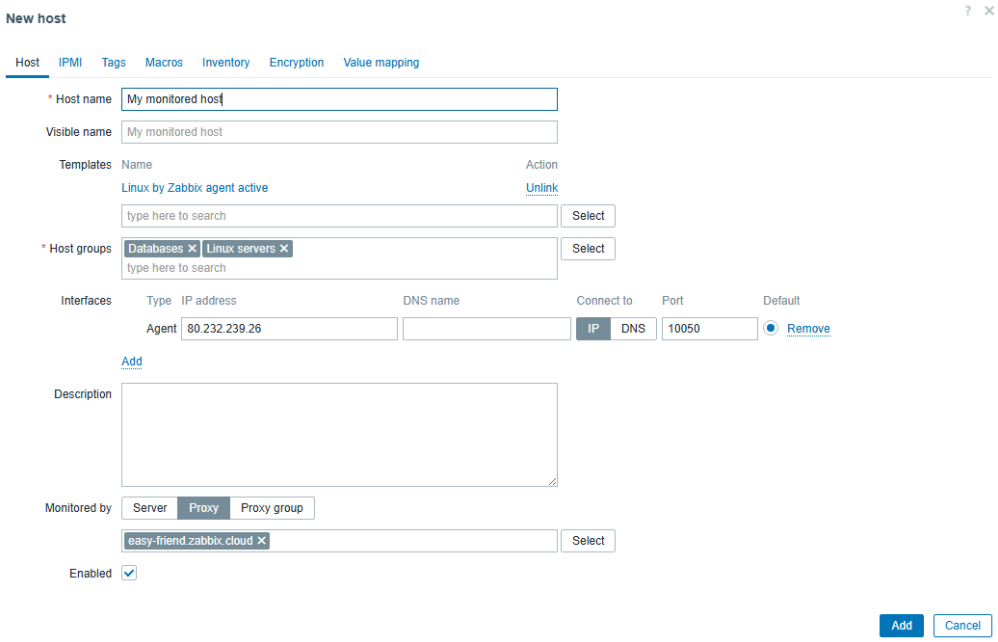
1. Aggiungere il nome del proxy al campo *Nome proxy*. Deve corrispondere al nome host del proxy configurato per il proxy. Immetterlo oppure copiarlo dalla scheda *Panoramica* del proxy.
2. Selezionare la modalità proxy corrispondente a quella selezionata durante la creazione del proxy.
3. Per un proxy attivo, immettere l'indirizzo IP o il nome DNS del proxy.
4. Fare clic su *Aggiungi*.



Per ulteriori informazioni, vedere Monitoraggio distribuito tramite proxy.

Configurare gli host monitorati Aggiungere gli host che devono essere monitorati tramite il proxy.

1. Andare a *Raccolta dati* > *Host*.
2. Creare un host oppure aprire un host esistente.
3. In *Monitorato da*, selezionare il *Proxy* e digitare il nome del proxy per cercarlo e selezionarlo, oppure premere il pulsante *Seleziona* per aprire un elenco di proxy tra cui scegliere.
4. Configurare l'interfaccia dell'host e gli altri parametri.
5. Fare clic su *Aggiungi* o *Aggiorna*.



Un host configurato per essere monitorato da un proxy invia o riceve i dati di monitoraggio tramite tale proxy, a seconda del tipo di item e della modalità dell'agent.

Configurare Zabbix agent Quando un host monitorato utilizza Zabbix agent o Zabbix agent 2, configurare l'agent per comunicare con il componente Zabbix appropriato.

La configurazione dipende dal fatto che l'agent utilizzi controlli *passivi* o *attivi*.

Controlli passivi dell'agent

Con i controlli passivi, il proxy Zabbix avvia le connessioni all'agent e richiede i dati di monitoraggio.

Aprire il file di configurazione dell'agent Zabbix e aggiornare il parametro `Server` per consentire le connessioni dall'indirizzo IP del proxy Zabbix Cloud.

Ad esempio:

```
Server=<proxy_IP>
```

Assicurarsi che l'host monitorato consenta connessioni TCP in ingresso dal proxy sulla porta dell'agent, che per impostazione predefinita è 10050.

Il proxy deve poter raggiungere l'host monitorato tramite la rete.

Controlli attivi dell'agent

Con i controlli attivi, l'agent avvia le connessioni al proxy Zabbix e invia i dati di monitoraggio raccolti.

Aprire il file di configurazione dell'agent Zabbix e aggiornare il parametro `ServerActive` con l'indirizzo del proxy Zabbix Cloud.

Ad esempio:

`ServerActive=<proxy_IP>:<proxy_listen_port>`

Sostituire `<proxy_IP>` con l'indirizzo IP del proxy Zabbix Cloud e `<proxy_listen_port>` con la porta di ascolto configurata quando il nodo proxy è stato creato. La porta di ascolto predefinita è 10051.

Se viene specificato un proxy in `ServerActive`, l'agent ottiene la configurazione dei controlli attivi dal proxy anziché direttamente dal server Zabbix.

Assicurarsi che l'host monitorato possa stabilire una connessione TCP in uscita al proxy sulla porta di ascolto configurata.

Per ulteriori informazioni sui controlli passivi e attivi dell'agent, vedere Controlli passivi e attivi dell'agent.

Server Zabbix esterno Un proxy Zabbix Cloud può comunicare anche con un server Zabbix distribuito al di fuori di Zabbix Cloud.

In questo caso, il server Zabbix esterno deve poter comunicare con il proxy cloud in base alla modalità proxy selezionata.

Per un proxy in modalità attiva, il proxy avvia la connessione al server Zabbix esterno. Configurare il server Zabbix esterno in modo che accetti le connessioni dal proxy e assicurarsi che il proxy possa raggiungere il server tramite la rete.

Per un proxy passivo, il server Zabbix avvia la connessione al proxy. Aggiungere l'indirizzo IP o la subnet del server Zabbix esterno ai filtri di accesso del nodo proxy e assicurarsi che il server possa raggiungere il proxy tramite la rete.

Il server Zabbix esterno non utilizza i filtri di accesso di Zabbix Cloud. Configurare invece le regole firewall e di rete necessarie nell'ambiente esterno.

Gli host monitorati possono trovarsi nello stesso ambiente del server Zabbix esterno, in un'altra rete o in una rete accessibile dal proxy.

Quando gli agent utilizzano controlli attivi, configurare il parametro `ServerActive` in modo che punti al proxy Zabbix Cloud. Quando gli agent utilizzano controlli passivi, configurare il parametro `Server` per consentire le connessioni dal proxy.

Verifica della configurazione Dopo aver configurato il proxy, gli agent, i filtri di accesso e la connettività di rete, verifica che il proxy comunichi con il server Zabbix.

1. Vai a *Amministrazione* > *Proxy*.
2. Controlla lo *Stato* del proxy.
3. Vai a *Monitoraggio* > *Ultimi dati* e verifica che siano disponibili valori recenti degli item.

Se il proxy non comunica con il server, verifica:

- la modalità del proxy;
- l'indirizzo e la porta del server;
- i filtri di accesso del proxy;
- i filtri di accesso del server, quando il server è ospitato su Zabbix Cloud;
- la configurazione dell'agent;
- le regole del firewall e di routing;
- l'indirizzo configurato in `Server` o `ServerActive`.

App mobile

App mobile

L'app mobile di Zabbix consente di monitorare il proprio ambiente Zabbix dal dispositivo mobile. Visualizza i problemi, controllane i dettagli, aggiorna i problemi e ricevi notifiche push quando qualcosa richiede attenzione.

Scarica l'app (prossimamente)

Configurare Zabbix per l'app

Configurare l'adattatore bridge di Zabbix Installare l'adattatore bridge di Zabbix per consentire ai dispositivi mobili di connettersi a Zabbix.

Aggiungi dispositivi Aggiungi il tuo dispositivo mobile nell'interfaccia frontend di Zabbix per connetterlo al tuo server Zabbix.

Configura le notifiche push Ricevi avvisi sul tuo dispositivo mobile quando si verifica un problema.

Scopri l'app

Panoramica dell'app Scopri le nozioni di base sull'utilizzo dell'app mobile Zabbix.

Accesso mobile sicuro Proteggi l'app, i tuoi dispositivi mobili e Zabbix bridge adapter oppure revoca l'accesso dai dispositivi smarriti o compromessi.

Configurare Zabbix per l'app

Per utilizzare l'app mobile Zabbix (disponibile prossimamente), è necessario prima:

1. Configurare l'adattatore bridge di Zabbix.
2. Configurare le notifiche push.
3. Aggiungere il dispositivo mobile nell'interfaccia web di Zabbix.

Le sezioni seguenti illustrano ogni passaggio.

Dopo la configurazione, consultare Accesso mobile sicuro per mantenere sicuri l'app e l'adattatore bridge di Zabbix.

Configurazione dell'adattatore bridge di Zabbix Per consentire al tuo Zabbix server di comunicare con l'app mobile, devi configurare Zabbix bridge adapter.

Il bridge adapter può essere installato sullo stesso sistema del tuo Zabbix server oppure su un sistema diverso, utilizzando i pacchetti:

> **Attention:**
> I passaggi riportati di seguito sono validi per Ubuntu 24.04. Per altri sistemi operativi, usa i comandi per l'installazione del repository e dei pacchetti (ad esempio, `wget`, `dpkg`, `apt`) indicati nella pagina Get Zabbix per il tuo sistema operativo e utilizzali per installare il pacchetto `zabbix-bridge-adapter`.

1. Avvia una nuova sessione shell con privilegi root:

```
sudo -s
```

2. Installa il repository Zabbix:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Installa Zabbix bridge adapter:

```
apt install zabbix-bridge-adapter
```

4. Modifica il file di configurazione di Zabbix server:

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Riavvia Zabbix server, ad esempio (vedi altre opzioni):

```
systemctl restart zabbix-server
```

6. Copia l'ID di Zabbix server da *Reports* > *System information*; lo utilizzerai nel passaggio successivo.

7. Modifica il file di configurazione di Zabbix bridge adapter:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

> **Attention:**
> La directory specificata in `DataDir` deve avere i permessi 0700 e tutti i file archiviati al suo interno devono avere i permessi 0600.

8. Esamina e accetta il contratto di licenza con l'utente finale (EULA) di Zabbix bridge adapter:

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Avvia Zabbix bridge adapter:

```
systemctl start zabbix-bridge-adapter
```

Attention:
All'avvio, Zabbix bridge adapter utilizza l'ID del server per registrare il tuo Zabbix server nel Zabbix Bridge ospitato centralmente. Lo stato della registrazione viene memorizzato in un file di stato nella directory `DataDir`. Se reinstalli il bridge adapter e il file di stato viene perso, la registrazione dello stesso Zabbix server potrebbe non riuscire. Per evitare questo problema, esegui il backup del file di stato prima di reinstallare il bridge adapter e ripristinalo successivamente.

Configurare le notifiche push Per ricevere avvisi sul tuo dispositivo mobile quando qualcosa richiede attenzione nell'ambiente monitorato:

1. Crea un tipo di supporto per le notifiche push (oppure abilita quello predefinito).
2. Crea un'azione (oppure abilita l'azione trigger predefinita *Segnala i problemi agli amministratori di Zabbix*).
3. Aggiungi un dispositivo per aggiungere automaticamente il tipo di supporto al tuo utente (oppure aggiungi manualmente i supporti dell'utente).

Aggiunta di dispositivi Per utilizzare l'app mobile Zabbix sul proprio dispositivo, è necessario aggiungere tale dispositivo nell'interfaccia web di Zabbix. Per le istruzioni, vedere:

- *Utenti* > *Dispositivi* (come *Super admin*, aggiungere i propri dispositivi mobili o quelli di altri utenti)
- *Impostazioni utente* > *Dispositivi* (come qualsiasi utente, aggiungere i propri dispositivi mobili)

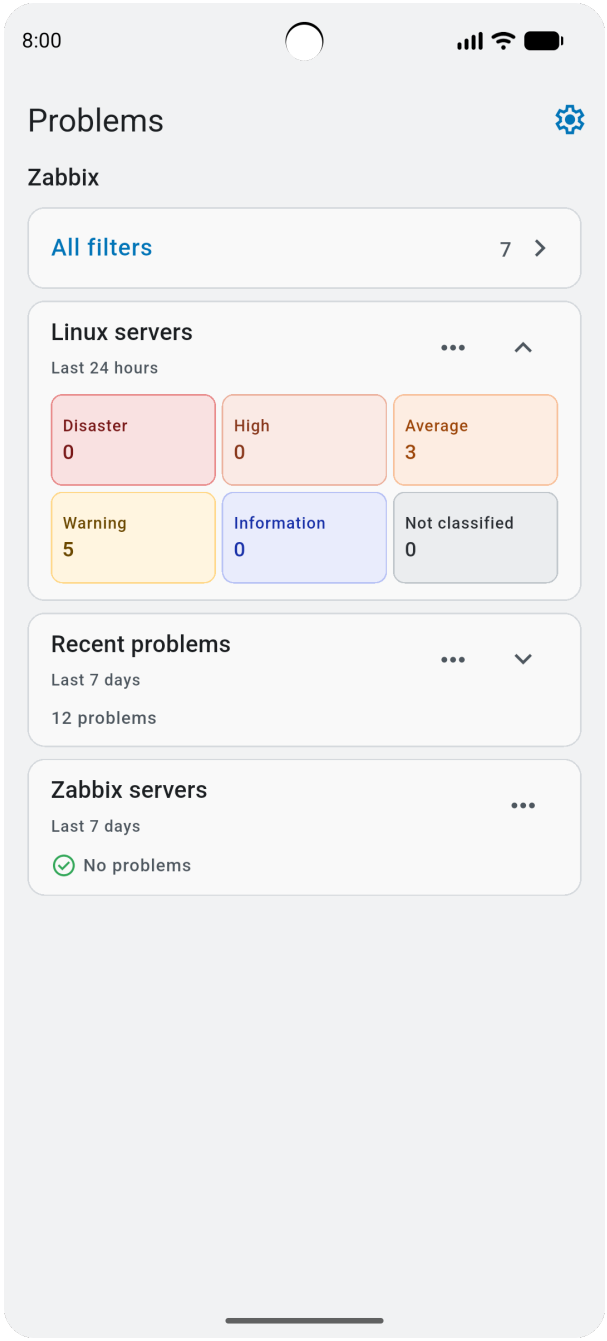
Panoramica dell'app

L'app mobile di Zabbix (disponibile prossimamente) consente di monitorare il proprio ambiente Zabbix dal dispositivo mobile. Visualizza i problemi, controllane i dettagli, aggiorna i problemi e ricevi notifiche push quando è necessario intervenire.

Questa pagina offre una panoramica dell'app mobile e delle relative modalità di utilizzo.

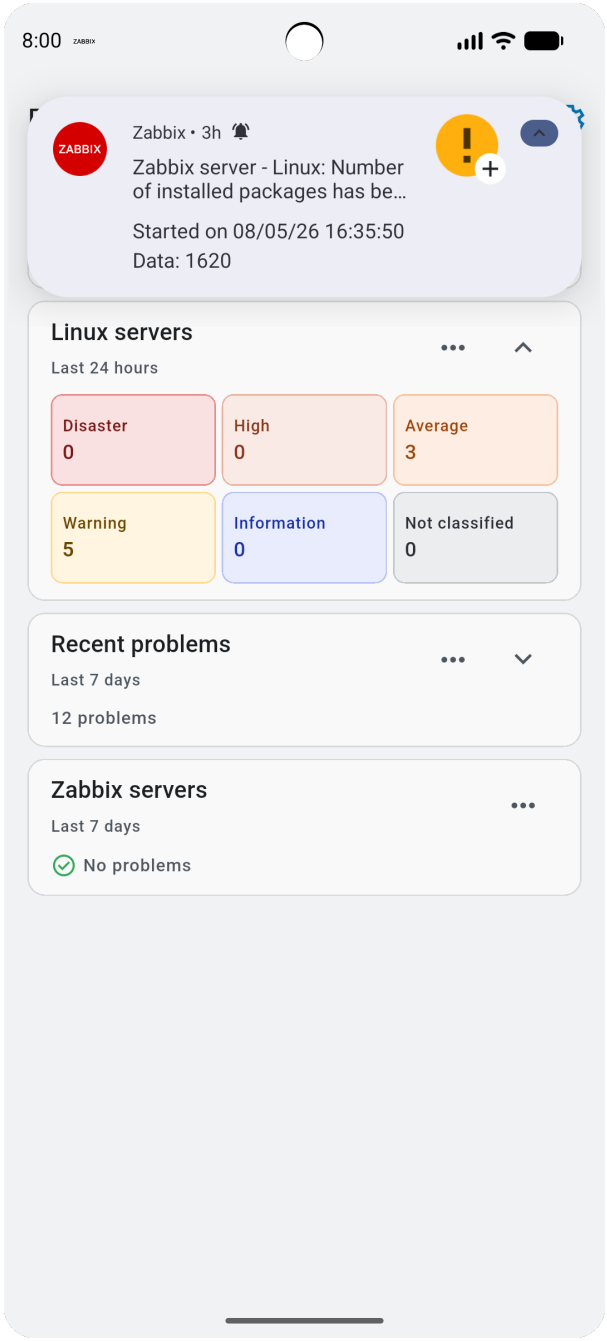
Problemi La schermata *Problemi* mostra una panoramica dei problemi in base ai filtri aggiunti ai preferiti. Contiene:

- Icona *Impostazioni* - toccala per visualizzare le impostazioni dell'app.
- *Tutti i filtri* - toccalo per visualizzare e gestire i filtri.
- Nome del server Zabbix (oppure *Zabbix*, se non è stato configurato durante l'installazione dell'interfaccia web di Zabbix).
- Filtri aggiunti ai preferiti - tocca il loro nome o una gravità per aprire la schermata *Elenco dei problemi*. Tocca il menu con i tre puntini per gestire il filtro (modificarlo, duplicarlo, rimuoverlo dai preferiti o eliminarlo), oppure tocca la freccia per espanderlo o comprimerlo. I filtri aggiunti ai preferiti sono ordinati alfabeticamente e il filtro *Problemi recenti* non può essere modificato né rimosso.



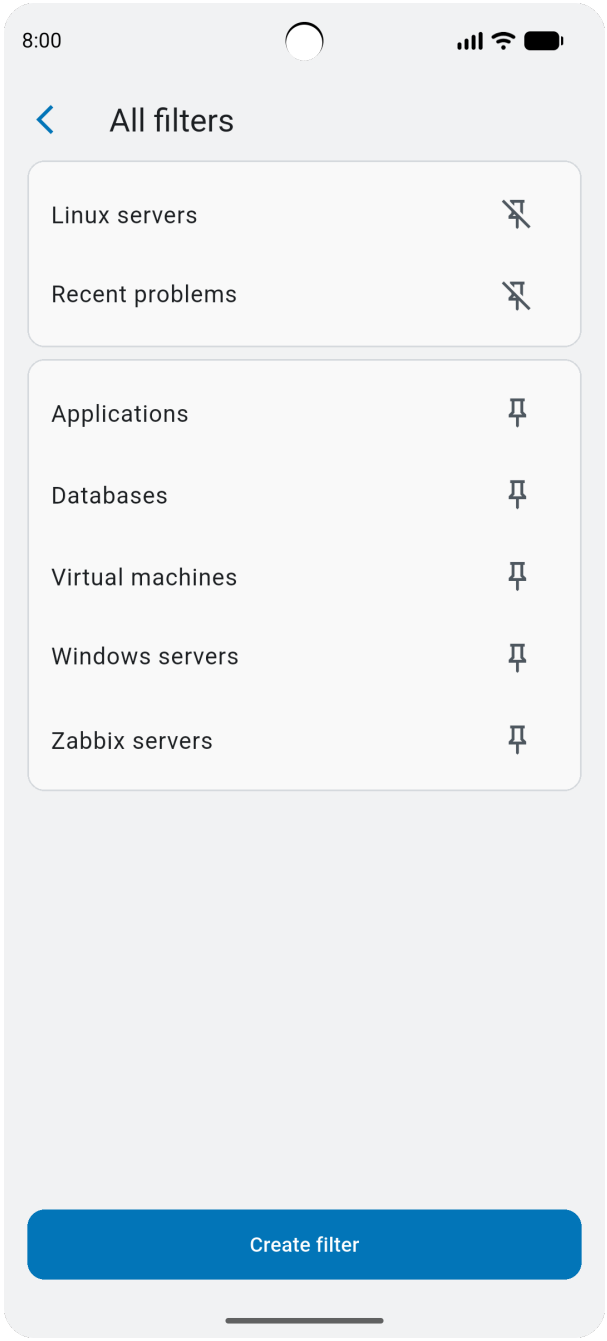
Notifiche Quando Zabbix server rileva un problema, oppure quando un problema viene aggiornato o risolto, può inviare una notifica al tuo dispositivo mobile. La notifica contiene l'host del problema, il nome e un messaggio generato dai template dei messaggi configurati per il tipo di supporto notifica push.

Tocca la notifica per aprire la schermata *Dettagli del problema*.



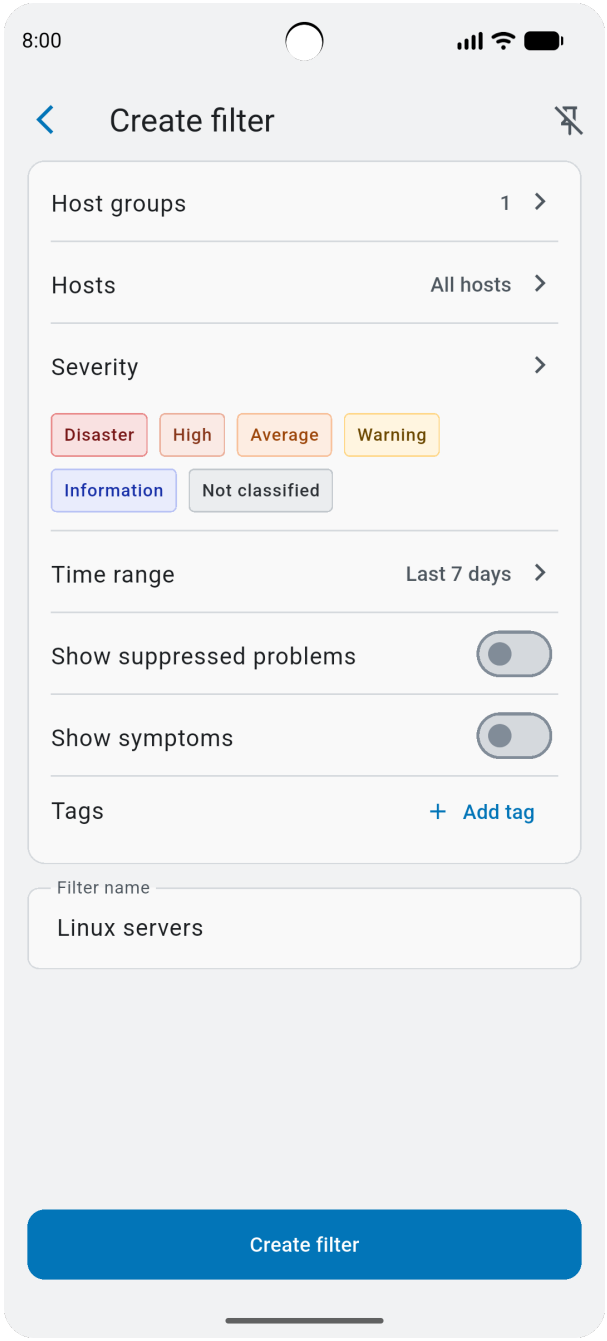
Tutti i filtri La schermata *Tutti i filtri* consente di gestire i filtri. Contiene:

- Nomi dei filtri: toccare per modificare il filtro.
- Icone a forma di puntina: toccare per fissare o rimuovere il filtro dalla schermata *Problemi*.
- *Crea filtro*: toccare per creare un nuovo filtro.



Crea filtro La schermata *Crea filtro* consente di creare un filtro, con opzioni simili a quelle disponibili in *Monitoraggio* > *Problemi* (vedere i dettagli). È inoltre possibile impostare un *Intervallo di tempo* per specificare per quanto tempo visualizzare i problemi non risolti.

Toccare *Crea filtro* per salvare il filtro. I nuovi filtri vengono aggiunti alla schermata *Problemi*; toccare l'icona della puntina per modificare questa impostazione.



Elenco dei problemi La schermata *Elenco dei problemi* mostra i problemi che corrispondono al filtro (ad esempio, *Problemi recenti*).

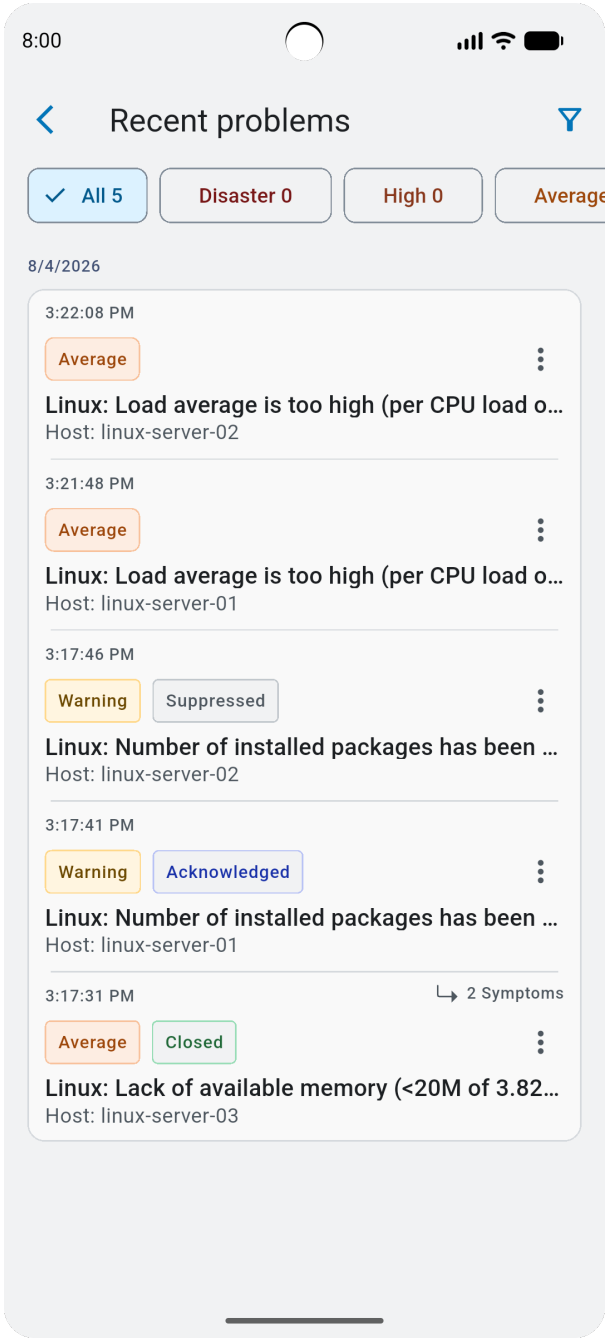
Tocca l'icona del filtro o una gravità del problema per restringere ulteriormente l'elenco.

Per ogni problema, puoi visualizzare:

- Ora e data di inizio.
- Gravità, nome e host.
- Se è stato soppresso (visualizzato se *Mostra problemi soppressi* è abilitato).
- Se è stato riconosciuto o chiuso di recente (risolto automaticamente o chiuso manualmente).
- Se presenta sintomi: tocca per visualizzare l'elenco dei problemi sintomo.

Tocca un problema per aprire la schermata *Dettagli del problema*.

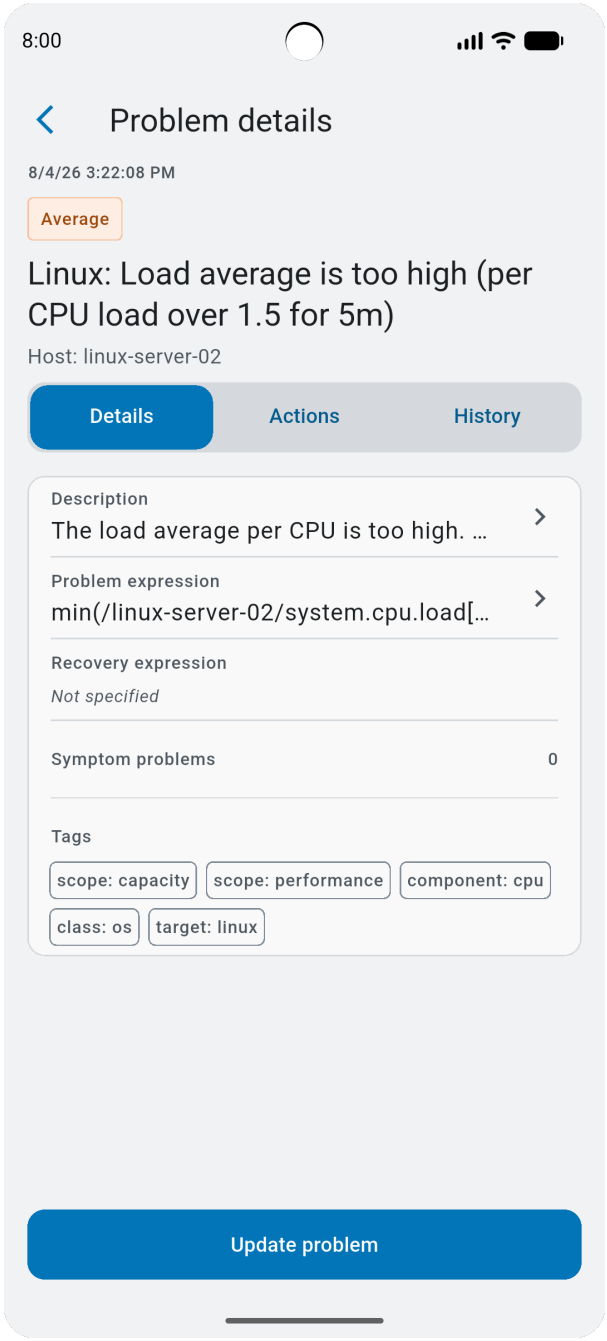
Tocca il menu con i tre puntini per riconoscere rapidamente il problema oppure sopprimerlo per il periodo di tempo desiderato.



Dettagli del problema La schermata *Dettagli del problema* mostra i dettagli del problema, le azioni e la cronologia.

- Scheda *Dettagli* - mostra la descrizione del problema, le espressioni, i problemi sintomo (toccare per visualizzare l'elenco dei problemi) e i tag.
- Scheda *Azioni* - mostra la cronologia del problema: quando è iniziato o è stato riconosciuto, se è stata inviata una notifica e altre azioni.
- Scheda *Cronologia* - mostra l'età del problema, la durata e il tempo di ripristino (oppure *Non disponibile* se il problema è ancora in corso).

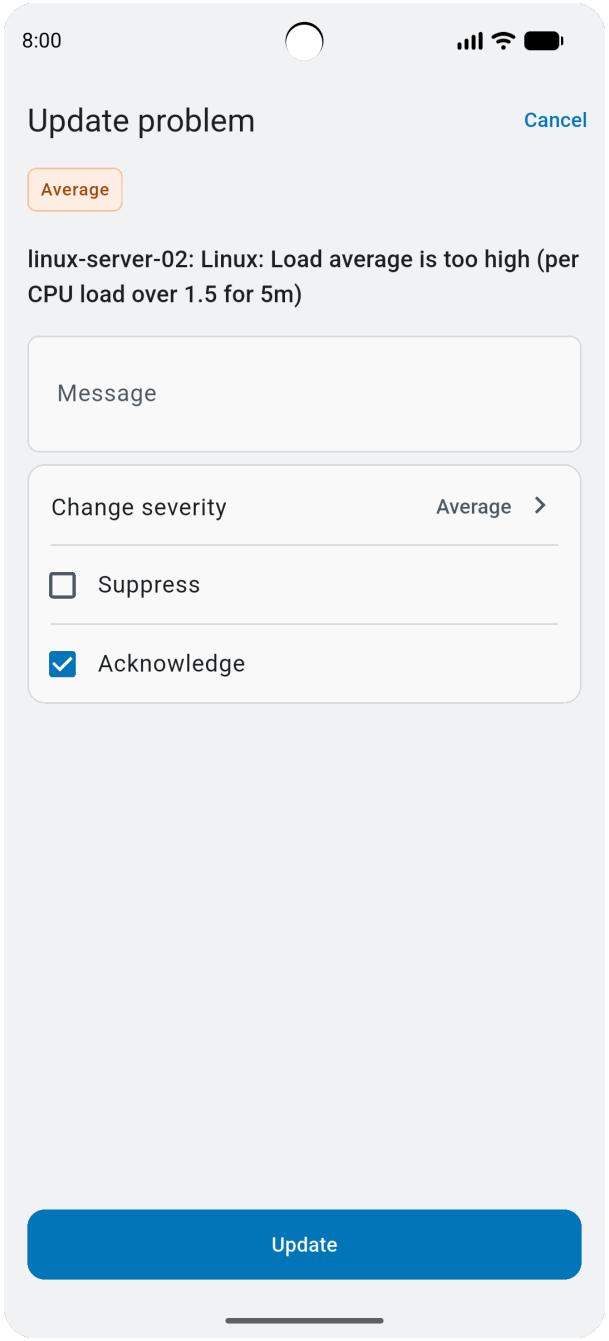
Toccare *Aggiorna problema* per aggiornare il problema.



Aggiorna problema La schermata *Aggiorna problema* consente di aggiornare un problema:

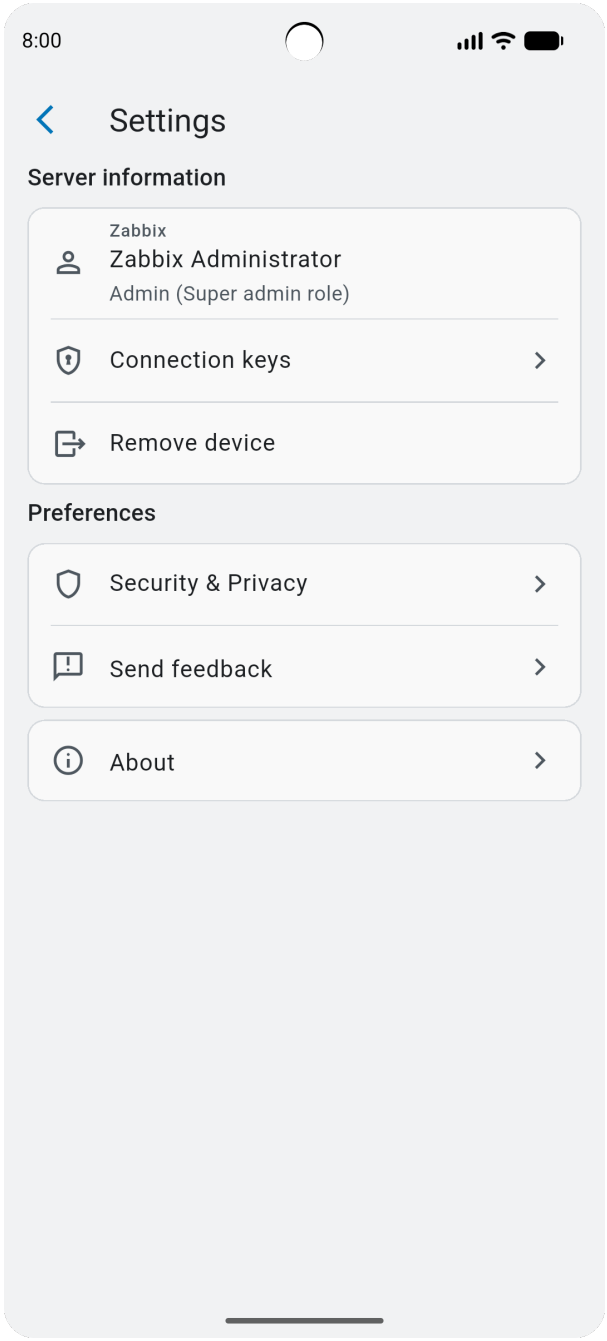
- Aggiungere un messaggio.
- Modificare la gravità del problema.
- Sospendere il problema per il periodo di tempo scelto.
- Confermare o annullare la conferma del problema.
- Chiudere il problema.

Toccare *Aggiorna* per salvare le modifiche.



Impostazioni La schermata *Impostazioni* consente di accedere alle impostazioni dell'app. Contiene:

- Nome del server Zabbix (oppure *Zabbix*, se non configurato durante l'installazione dell'interfaccia web di Zabbix).
- Nome, tipo e ruolo dell'utente a cui è aggiunto il dispositivo mobile.
- *Chiavi di connessione* - aprire per visualizzare e copiare l'ID del server, l'ID del dispositivo e altri dettagli.
- *Rimuovi dispositivo* - toccare per rimuovere il dispositivo mobile da Zabbix.
- *Sicurezza e privacy* - aprire per abilitare il blocco dell'app e gestire la condivisione dei dati diagnostici.
- *Invia feedback* - aprire per inviare feedback a Zabbix.
- *Informazioni* - aprire per visualizzare la versione dell'app e altri dettagli sull'app.



Accesso mobile sicuro

Questa pagina contiene le best practice per mantenere sicuri l'app mobile Zabbix e l'adattatore bridge Zabbix.

Proteggi i dispositivi mobili Per impedire ad altre persone di aprire l'app sul tuo dispositivo mobile, abilita il blocco dell'app nelle *Impostazioni* dell'app.

Se un dispositivo è stato perso o compromesso, rimuovilo nell'interfaccia web di Zabbix. In questo modo le relative credenziali di accesso vengono revocate e rimosse, impedendo al dispositivo di accedere ulteriormente a Zabbix. A seconda delle autorizzazioni del tuo ruolo utente, puoi rimuovere i dispositivi nelle seguenti sezioni:

- *Impostazioni utente > Dispositivi* - qualsiasi utente può rimuovere i propri dispositivi (è necessaria l'autorizzazione *Gestisci i propri dispositivi*).
- *Utenti > Dispositivi* - un *Super amministratore* può rimuovere i dispositivi di qualsiasi utente (è necessaria l'autorizzazione *Gestisci i dispositivi degli utenti*).

Puoi anche limitare le parti di Zabbix a cui è possibile accedere da un dispositivo mobile. A tale scopo, assegna agli utenti un ruolo utente che conceda solo le autorizzazioni richieste dall'app. Le autorizzazioni minime sono le seguenti:

Sezione	Configurazione
Accesso agli elementi dell'interfaccia utente	Deseleziona tutte le caselle di controllo.
Accesso ai servizi	Imposta *Accesso in lettura e scrittura ai servizi* e *Accesso in sola lettura ai servizi* su `None`.
Accesso ai moduli	Deseleziona tutte le caselle di controllo.
Accesso all'API	Deseleziona la casella di controllo *Abilitato*.
Accesso alle azioni	Deseleziona tutte le caselle di controllo. Per consentire agli utenti di aggiornare i problemi nell'app, seleziona anche le azioni necessarie: - *Aggiungi commenti ai problemi* - *Modifica la gravità* - *Riconosci i problemi* - *Sopprimi i problemi* - *Chiudi i problemi* - *Crea e modifica i propri media*
Dispositivi	Nota che i problemi visualizzabili dagli utenti nell'app dipendono anche dal loro accesso agli host. Seleziona la casella di controllo *Abilitato*. Per consentire agli utenti di aggiungere e rimuovere i propri dispositivi mobili, seleziona anche la casella di controllo *Gestisci i propri dispositivi*; in caso contrario, un *Super amministratore* deve gestire i loro dispositivi in *Utenti > Dispositivi*.

Protezione dell'adapter bridge di Zabbix L'adapter bridge di Zabbix archivia i propri dati (il file di stato e le chiavi di crittografia) nella directory specificata nel parametro `DataDir` del file di configurazione dell'adapter bridge di Zabbix. Assicurarsi che la directory disponga dei permessi 0700 e che tutti i file archiviati al suo interno dispongano dei permessi 0600.

L'adapter bridge di Zabbix inoltra le richieste dai dispositivi mobili all'API di Zabbix. Per limitare i metodi API che possono essere chiamati tramite l'adapter bridge di Zabbix, utilizzare il parametro `AllowedAPIMethods`. Ad esempio, consentire solo il set minimo di metodi richiesto dall'app mobile, o anche un numero inferiore, se non si desidera utilizzare alcune funzionalità dell'app mobile (ad esempio, non consentire `event.acknowledge` se gli utenti dell'app mobile non devono aggiornare i problemi).

Centro sviluppatori

Centro sviluppatori

Inizia a sviluppare estensioni personalizzate per Zabbix con indicazioni su plugin, moduli e widget.

Sviluppare su Zabbix

Plugin Una panoramica su come sviluppare e gestire i plugin di Zabbix per estendere le funzionalità o integrare sistemi esterni.

Libreria Python per Zabbix Risorse per utilizzare la libreria ufficiale `zabbix_utils` per interagire con l'API di Zabbix e creare integrazioni.

Moduli del frontend Guide e riferimenti per creare moduli personalizzati che estendono o modificano il frontend di Zabbix per adattarlo a casi d'uso specifici.

Widget Una panoramica della struttura e della logica dei widget, con istruzioni per creare elementi dashboard personalizzati in base alle tue esigenze.

Creazione di un plugin

Questa pagina è destinata a chi è interessato a sviluppare il proprio plugin (raccoglitore di dati indipendente) per Zabbix agent 2.

Un plugin personalizzato estende le funzionalità di Zabbix agent 2 oltre i plugin integrati e i plugin caricabili disponibili.

Ogni plugin è un pacchetto *Go* che definisce la struttura e implementa una o più interfacce (*Exporter*, *Configurator*, *Runner*). Per ulteriori informazioni, vedere Interfacce dei plugin e Diagramma delle connessioni.

Questa guida illustra come creare un plugin caricabile personalizzato.

Per ulteriori indicazioni, consultare anche i repository di:

- Plugin di esempio
- Plugin caricabili esistenti

Cosa creerai Questo è un tutorial passo passo per creare un semplice plugin caricabile chiamato **MyIP**. Questo plugin implementerà una singola chiave item (`myip`) che restituisce l'indirizzo IP esterno dell'host dell'agent Zabbix.

Passaggio 1: Configurazione

1. Un plugin è un modulo Go standard. Inizia inizializzando il file `go.mod` nella directory del plugin per tenere traccia delle dipendenze del plugin:

```
cd path/to/plugins/myip # Passa alla directory del plugin
go mod init myip
```

2. Installa la dipendenza obbligatoria Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Sostituisci `$LATEST_COMMIT_HASH` con l'hash del commit `HEAD` più recente dal repository `golang.zabbix.com/sdk` nel branch di rilascio appropriato. Per esempio:

```
go get golang.zabbix.com/sdk@af85407
```

Tieni presente che il versionamento di `golang.zabbix.com/sdk` al momento non è supportato, ma questo potrebbe cambiare in futuro.

Le dipendenze aggiuntive possono essere installate secondo necessità usando `go get`.

3. Crea un file `main.go` vuoto per il codice sorgente del plugin:

```
touch main.go
```

Ora la configurazione iniziale è completa e il plugin è pronto per lo sviluppo.

Passaggio 2: Struttura del plugin Il modulo `golang.zabbix.com/sdk`, installato nel passaggio precedente, fornisce il framework necessario per lo sviluppo dei plugin e garantisce che tutti i plugin abbiano una struttura coerente.

1. Configurare il flusso di esecuzione di base.

Iniziare definendo il flusso di esecuzione principale del plugin. Aggiungere il codice seguente a `main.go`:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

Questo stabilisce il flusso di esecuzione di base del plugin. La funzione run conterrà in seguito la logica principale del plugin.

2. Esaminare le interfacce del plugin.

Un plugin di Zabbix agent 2 deve essere rappresentato da una struct che implementa le interfacce del pacchetto `golang.zabbix.com/sdk/p`

- *Accessor* - definisce i metodi essenziali che tutti i plugin devono implementare, come l'impostazione del nome del plugin e la gestione dei timeout delle chiavi degli item.
- Una o più delle seguenti interfacce funzionali del plugin:
  - *Exporter* - esegue un polling e restituisce un valore (o più valori), nulla oppure un errore; viene spesso utilizzata insieme all'interfaccia *Collector*.
  - *Collector* - gestisce la raccolta periodica dei dati.
  - *Runner* - definisce le procedure di avvio e arresto del plugin.

- *Watcher* - abilita il polling indipendente delle metriche, ignorando lo scheduler interno dell'agent; è utile per il monitoraggio basato su trap o guidato dagli eventi.
- *Configurator* - definisce il modo in cui il plugin legge e applica le proprie impostazioni di configurazione.

È possibile implementare queste interfacce autonomamente oppure utilizzare quelle predefinite fornite da Zabbix Go SDK, modificandole secondo necessità. Questa esercitazione utilizza le implementazioni predefinite.

3. Creare la struct del plugin.

Ora importare il pacchetto *plugin* e creare una struct `myIP` che includa la struct `plugin.Base`:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

La struct myIP soddisfa attualmente l'interfaccia Accessor. Un metodo per implementare una delle interfacce funzionali del plugin, *Exporter*, verrà aggiunto più avanti nell'esercitazione.

Passaggio 3: Definire le chiavi degli item Il tuo plugin ha bisogno delle chiavi degli item per raccogliere i dati e fornirli a Zabbix server o proxy.

1. Importa il pacchetto *errs* per la gestione degli errori:

```
import "golang.zabbix.com/sdk/errs"
```

2. Registra le chiavi degli item utilizzando la funzione `plugin.RegisterMetrics()` all'interno della funzione `run()`:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // Plugin name
        "myip",                             // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

Per registrare più chiavi degli item, ripeti i parametri *nome della metrica* e *descrizione* per ogni metrica. Ad esempio:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Passaggio 4: Configurare l'handler L'handler facilita la comunicazione tra l'agent e il plugin.

1. Importare il pacchetto *container*:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. All'interno della funzione `run()` aggiungere il codice per creare e configurare un handler:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // Plugin name
        "myip",                             // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
```

```
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Passaggio 5: Implementare la raccolta dei dati La raccolta dei dati viene eseguita tramite l'interfaccia Exporter, che descrive il metodo `Export`:

```
func Export(
  key string,             // La chiave item da raccogliere.
  params []string,        // Argomenti passati alla chiave item (`myip[arg1, arg2]`).
  context ContextProvider // Metadati sulla raccolta dei dati della chiave item.
) (any, error)
```

1. Importare i pacchetti richiesti per le richieste HTTP e la lettura della risposta:

```
import (
    "io"
    "net/http"
)
```

2. Implementare il metodo `Export` per la struct `myIP`:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // Il plugin può utilizzare una logica di raccolta dati diversa in base al parametro `key`.
    // Questa implementazione verifica solo che il `key` fornito sia supportato.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Il log verrà inoltrato al log di agent 2.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Raccogliere i dati e restituirli.

    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
}
```

Passo 6: Compilare e configurare il plugin

1. Per compilare il plugin, eseguire:

```
go mod tidy
go build
```

Questo dovrebbe creare un eseguibile `myip` nella directory corrente.

2. Configurare Zabbix agent 2 per usare il plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

Sostituire `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` con il percorso del `myip` creato nel passaggio 5.

Il nome del plugin nel nome del parametro di configurazione (*MyIP* in questo tutorial) deve corrispondere al nome del plugin definito nella funzione *plugin.RegisterMetrics()*.

3. Per testare il plugin e il relativo item `myip`, eseguire:

```
zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf -t myip
```

L'output dovrebbe contenere un indirizzo IP esterno del vostro host e apparire simile a questo:

```
myip                                          [s|192.0.2.1]
```

Con questo, avete creato un semplice plugin caricabile per Zabbix agent 2. Complimenti!

```
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{}

    // Registra la chiave item `myip`.
```

```
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Nome del plugin
        "myip",                          // Nome della chiave item
        "Restituisce l'indirizzo IP dell'host.", // Descrizione della chiave item
    )
    if err != nil {
        return errs.Wrap(err, "impossibile registrare le metriche")
    }

    // Crea un nuovo handler.
    h, err := container.NewHandler("MyIP") // Nome del plugin
    if err != nil {
        return errs.Wrap(err, "impossibile creare un nuovo handler")
    }

    // Configura il logging per inoltrare i log dal plugin all'agent.
    // Disponibile tramite p.Logger.Infof, p.Logger.Debugf, ecc.
    p.Logger = h

    // Avvia l'esecuzione del plugin.
    // Si blocca finché non viene ricevuta una richiesta di terminazione dall'agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "impossibile eseguire l'handler del plugin")
    }

    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // Il plugin può utilizzare una logica diversa di raccolta dati in base al parametro `key`.
    // Questa implementazione verifica solo che la `key` fornita sia supportata.
    if key != "myip" {
        return nil, errs.Errorf("chiave item sconosciuta %q", key)
    }

    // Il log verrà inoltrato all'agent 2.
    p.Infof(
        "ricevuta richiesta di gestire la chiave %q con %d parametri",
        key,
        len(params),
    )

    // Raccoglie i dati e li restituisce.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "impossibile ottenere l'indirizzo IP")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "impossibile leggere il corpo della risposta")
    }

    return string(body), nil
}
```

Codice sorgente completo

Interfacce plugin

Questa sezione descrive le interfacce dei plugin disponibili.

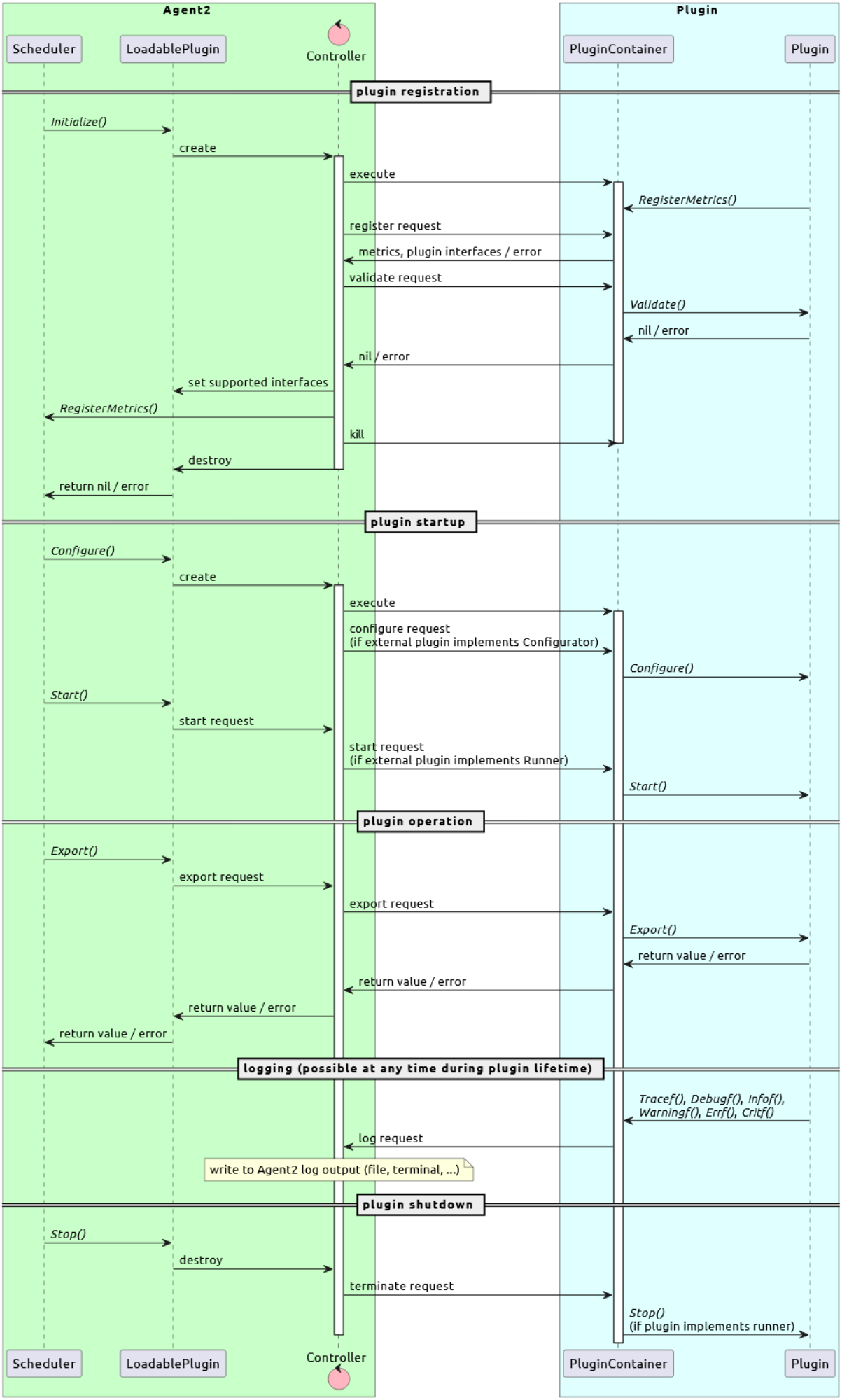
plugin.Esportatore *Exporter* è l'interfaccia più semplice che esegue un sondaggio e restituisce un valore (valori), niente o un errore. Accetta una chiave di elemento, parametri e contesto preparati. L'accesso a tutte le altre interfacce del plugin è esclusivo e nessun metodo può essere chiamato se un plugin sta già eseguendo un'attività. Inoltre, esiste un limite massimo di 100 chiamate simultanee *Export()* per plugin, che può essere ridotto in base ai requisiti di ciascun plugin.

plugin.Configurator L'interfaccia *Configurator* fornisce i parametri di configurazione del plugin dai file di configurazione dell'agente 2 di Zabbix.

plugin.Runner L'interfaccia *Runner* fornisce i mezzi per eseguire l'inizializzazione quando un plugin viene avviato (attivato) e la deinizializzazione quando un plugin viene arrestato (disattivato). Ad esempio, un plugin può avviare/interrompere alcune *goroutine* in background implementando l'interfaccia Runner.

Diagramma di connessione Zabbix agent 2 si connette in modo bidirezionale ai plugin utilizzando socket UNIX su Linux e Named Pipes su Windows.

Il diagramma di connessione seguente illustra il processo di comunicazione tra Zabbix agent 2 e un plugin caricabile, nonché il processo di raccolta delle metriche.


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

Libreria Python per Zabbix

zabbix_utils è la libreria Python ufficiale per Zabbix.

Con zabbix_utils è possibile:

- Utilizzare le API di Zabbix per gestire gli oggetti di Zabbix (creare host, aggiornare item, ottenere eventi, ecc.).
- Raccogliere dati da Zabbix agent (in modo simile a Zabbix get).
- Inviare dati a Zabbix server o proxy (in modo simile a Zabbix sender).

La libreria supporta sia l'esecuzione sincrona che asincrona degli script, consentendo di gestire sia flussi di lavoro semplici sia operazioni parallele su larga scala, come la raccolta massiva di dati, le esportazioni in blocco, l'elaborazione degli avvisi e un efficiente ridimensionamento del sistema.

La libreria è compatibile con le versioni supportate di Zabbix server e con le versioni moderne di Python.

Per iniziare, installare la libreria e quindi seguire la guida quickstart.

Installazione

zabbix_utils può essere installato utilizzando uno qualsiasi dei metodi descritti di seguito.

Requisiti:

- Zabbix 6.0 o versioni successive (testato su 6.0, 7.0, 7.2, 7.4)
- Python 3.8 o versioni successive (testato su 3.8–3.13)
- framework HTTP asincrono aiohttp (richiesto solo per la modalità asincrona)

Python Package Index (PyPI)

Questo è il metodo più comune per la maggior parte degli ambienti:

```
pip install zabbix_utils
pip install zabbix_utils[async] # Solo per il supporto async
```

Repository Zabbix

Utilizzare questo metodo se si preferisce gestire le dipendenze tramite il gestore di pacchetti del sistema.

Scaricare i pacchetti Zabbix per la propria distribuzione Linux ed eseguire quindi i seguenti comandi.

Su RHEL e derivate:

```
dnf install python3-zabbix-utils
dnf install epel-release # Solo per il supporto async
dnf install python3-aiohttp # Solo per il supporto async
```

Su Debian/Ubuntu e derivate:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # Solo per il supporto async
```

Sorgente (GitHub)

Usa questo metodo se preferisci la versione di sviluppo più recente:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # Solo per il supporto async
python3 setup.py install
```

Per installare le dipendenze richieste per la modalità asincrona, puoi anche utilizzare uno dei metodi descritti sopra.

Guida rapida

Dopo aver installato zabbix_utils, puoi usarlo nel tuo script.

Questa guida rapida mostra come:

- Recuperare il nome host da Zabbix agent.
- Creare un host e un item trapper usando le API di Zabbix.
- Inviare un valore all'item.

La guida presenta lo script passo dopo passo, spiegando ogni parte man mano che viene introdotta. Lo script completo è fornito alla fine della pagina.

La guida presuppone inoltre che il tuo Zabbix server, agent e le API siano in esecuzione localmente.

Recuperare il nome host da Zabbix agent

Inizia recuperando il nome host del sistema su cui è in esecuzione Zabbix agent. Ti servirà questo nome host per creare un host utilizzando Zabbix API.

1. Importa la classe `Getter` da zabbix_utils. Questa classe funziona come Zabbix get e ti consente di richiedere dati da Zabbix agent.

2. Crea un'istanza di `Getter` che si connetta a Zabbix agent locale all'indirizzo `127.0.0.1` sulla porta 10050.

3. Chiama il metodo `get()` sull'istanza `Getter` per richiedere il nome host da Zabbix agent.

- Il metodo `get()` accetta come parametro una chiave item e invia una richiesta a Zabbix agent per quell'item.
- Il metodo `get()` restituisce un oggetto e il nome host è memorizzato nell'attributo `value` di quell'oggetto.
- `system.hostname` è una chiave integrata di Zabbix agent che restituisce il nome dell'host.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Connettersi e accedere alla API di Zabbix

Successivamente, connettiti alla API di Zabbix. Questo consente al tuo script di gestire oggetti di Zabbix come host e item.

1. Importa la classe `ZabbixAPI` da zabbix_utils.

2. Crea un'istanza di `ZabbixAPI` e fornisci l'URL della tua interfaccia web di Zabbix.

3. Chiama il metodo `login()` sull'istanza `ZabbixAPI` e fornisci il tuo nome utente e la tua password. L'account deve avere il permesso di accedere alla API di Zabbix.

4. Chiama il metodo `logout()` per chiudere la sessione dopo le operazioni API.

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### API operations go here

api.logout()
```

Creare un host in Zabbix

Ora che hai il nome host da Zabbix agent e sei connesso alla Zabbix API, puoi creare un nuovo host in Zabbix.

1. Chiama il metodo API `host.create()` sull'istanza `ZabbixAPI` e fornisci i dettagli dell'host:

- `host` - impostato sulla variabile `hostname` che contiene il nome host recuperato da Zabbix agent.
- `interfaces` - contiene i dettagli di connessione per Zabbix agent in esecuzione sull'host.
- `groups` - contiene almeno un gruppo di host a cui l'host deve appartenere.

2. Poiché la Zabbix API restituisce l'ID dell'host appena creato, salva questo ID nella variabile `host_id` per usarlo in seguito.

3. Stampa un messaggio per confermare che l'host è stato creato.

```
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}")
```

Creare un nuovo item in Zabbix

Dopo aver creato un host, puoi aggiungervi un item.

1. Definisci una chiave univoca per l'item da creare.

2. Chiama il metodo `item.create()` sull'istanza `ZabbixAPI` e fornisci i dettagli dell'item:

   - `hostid` - impostato sulla variabile `host_id` che contiene l'ID dell'host appena creato.
   - `name` - un nome per l'item.
   - `key_` - impostato sulla variabile `item_key` che hai appena definito.
   - `type` - impostato su 2 (trapper item), necessario per ricevere i valori inviati dal tuo script nei passaggi successivi.
   - `value_type` - impostato su 3 (numerico senza segno), il tipo di dati memorizzati da questo item.

3. Poiché l'API di Zabbix restituisce l'ID dell'item appena creato, salva questo ID nella variabile `item_id` per usarlo in seguito.

4. Stampa un messaggio per confermare che l'item è stato creato.

```
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}")
```

Inviare un valore all'host

Ora che hai creato il tuo host e item, puoi inviargli dei dati.

1. Importa la classe `time` e attendi alcuni secondi prima di inviare i dati.
Questo garantisce che Zabbix abbia elaborato completamente il nuovo host e item.

2. Importa la classe `Sender` da zabbix_utils.
Questa classe funziona come Zabbix sender e consente al tuo script di inviare dati a Zabbix.

3. Crea un'istanza di `Sender` che si connetta al server Zabbix locale all'indirizzo `127.0.0.1` sulla porta `10051`.

4. Chiama il metodo `send_value()` sull'istanza `Sender` per inviare un valore all'item del tuo host e fornisci questi dettagli:

   - `hostname` - impostato sulla variabile `hostname` contenente il nome host recuperato da Zabbix agent.
   - `item_key` - impostato sulla variabile `item_key` che hai appena definito in precedenza nel tuo script.
   - 1 - il valore da inviare.

4. Stampa un messaggio per confermare che il valore è stato inviato.

```
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}")
```

Dopo che questo codice è stato eseguito correttamente, dovresti vedere un messaggio di successo che indica che Zabbix ha ricevuto il tuo valore:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Ora puoi anche controllare l'interfaccia web di Zabbix (*Monitoring* > *Latest data*) per vedere il valore.

Script completo

Di seguito è riportato lo script completo che combina tutti i passaggi: recupero del nome host, creazione del host e dell'item tramite Zabbix API e invio dei dati a Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Recupera il nome host da Zabbix agent
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Connessione e accesso a Zabbix API
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Crea il host in Zabbix
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' creato con ID {host_id}")

### Crea l'item in Zabbix
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' creato con ID {item_id}")

### Disconnessione da API
api.logout()

### Attende che Zabbix elabori il nuovo host
time.sleep(10)

### Invia il valore al host
```

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Risposta del sender: {response}")
```

Utilizzo della API di Zabbix

zabbix_utils consente di utilizzare la API di Zabbix per gestire gli oggetti di Zabbix, inclusa la creazione di host, l'aggiornamento di item, il recupero di eventi e altro ancora.

Le richieste API possono essere effettuate in modalità sincrona o asincrona:

- In modalità sincrona, lo script Python invia una richiesta e attende una risposta prima di continuare, il che è adatto per operazioni semplici, sequenziali e prevedibili.
- In modalità asincrona, lo script invia richieste senza attendere ogni risposta, consentendo ad altre operazioni di procedere in parallelo; questo è più efficiente per richieste lente o grandi quantità di dati.

Gli esempi in questa pagina si concentrano sulla modalità sincrona, sebbene la modalità asincrona segua schemi simili. Ulteriori esempi sono disponibili nel repository GitHub zabbix_utils.

Importazione

Per utilizzare zabbix_utils per lavorare con l'API di Zabbix, importa la classe `ZabbixAPI` nel tuo script:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Se utilizzi già una libreria Python della community, di solito puoi sostituire quell'importazione con `ZabbixAPI` da zabbix_utils.

Accesso

Prima di effettuare richieste API, è necessario creare un'istanza di `ZabbixAPI` ed effettuare l'accesso alla API di Zabbix.

Scegli il metodo più adatto al modo in cui gestisci le credenziali. Puoi utilizzare il tuo nome utente e la tua password oppure i token API.

Durante l'inizializzazione

Quando si crea un'istanza di `ZabbixAPI`, è possibile fornire in un'unica volta l'URL dell'interfaccia web di Zabbix e le proprie credenziali:

```
### Nome utente e password:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### Token API:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Se si preferisce raggruppare i parametri di connessione, è possibile passarli come dizionario Python e decomprimerlo utilizzando la sintassi degli argomenti con nome di Python:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Utilizzo di `login()`

Per prima cosa puoi creare un'istanza di `ZabbixAPI` usando solo l'URL dell'interfaccia web di Zabbix e successivamente utilizzare il metodo `login()` con le tue credenziali:

```
### Nome utente e password:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")
```

```
### Token API:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Utilizzo delle variabili d'ambiente

Puoi innanzitutto memorizzare le tue credenziali come variabili d'ambiente, ad esempio utilizzando un terminale:

```
### Nome utente e password:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
```

Quindi, se avvii il tuo script dalla stessa sessione del terminale, l'istanza `ZabbixAPI` nello script richiede solo l'URL dell'interfaccia web di Zabbix:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Puoi anche memorizzare l'URL in una variabile d'ambiente:

```
### Nome utente e password:
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"
```

Quando l'URL e le credenziali sono tutti impostati come variabili d'ambiente, l'istanza `ZabbixAPI` nello script non richiede alcun argomento:

```
api = ZabbixAPI()
```

Disconnessione

Se hai effettuato l'accesso utilizzando un nome utente e una password, richiama il metodo `logout()` dopo aver completato le operazioni API:

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API actions go here

api.logout()
```

Note:
La chiamata al metodo `logout()` non è necessaria quando si utilizzano token API.

Richieste API

Dopo aver effettuato l'accesso, puoi eseguire qualsiasi richiesta API chiamando i metodi descritti nel riferimento dei metodi delle API di Zabbix.

I metodi API vengono chiamati utilizzando il seguente formato:

```
api_instance.zabbix_object.method(parameters)
```

Note:
Alcuni nomi di metodi o oggetti dell'API di Zabbix utilizzano parole che in Python sono parole chiave riservate (ad esempio, `import`). Per evitare errori in Python, aggiungi un trattino basso al nome del metodo o dell'oggetto quando lo utilizzi in Python (ad esempio, `api.configuration.import_`).

Ad esempio, per visualizzare un elenco di host con `host.get`:

```
### 1. Importa ZabbixAPI da zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2. Crea l'istanza ZabbixAPI ed effettua l'accesso:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3. Recupera gli host che corrispondono a un filtro:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Scorri gli host restituiti e stampa i dettagli di ciascun host:
for host in hosts:
    print(host)

### 5. Disconnettiti dall'API per chiudere la sessione:
api.logout()
```

Modalità asincrona

La modalità asincrona consente al tuo script di inviare richieste API senza attendere che ciascuna venga completata. Questo può rendere il tuo script più efficiente quando deve eseguire molte richieste API o quando alcune richieste richiedono molto tempo.

Quando si utilizza la modalità asincrona, ci sono differenze importanti rispetto alla modalità sincrona:

- Importa il modulo `asyncio` di Python (devi prima installare le dipendenze richieste).
- Importa `AsyncZabbixAPI` invece di `ZabbixAPI`.
- Scrivi il tuo codice all'interno di una funzione `async`.
- Usa `await` quando chiami i metodi API, inclusi `login()` e `logout()`.

> **Attention:**
> Quando crei un'istanza di `AsyncZabbixAPI`, non puoi effettuare l'accesso durante l'inizializzazione.

Ad esempio, per visualizzare un elenco di host con `host.get` in modalità asincrona:

```
### 1. Importa asyncio per la modalità asincrona e AsyncZabbixAPI da zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. Definisci la funzione async principale in cui verranno eseguite tutte le operazioni API:
async def main():

    # 3. Crea un'istanza di AsyncZabbixAPI ed effettua l'accesso (è necessario await):
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. Recupera gli host che corrispondono a un filtro (è necessario await):
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. Scorri gli host restituiti e stampa i dettagli di ciascun host:
```

```
    for host in hosts:
        print(host)

    # 6. Disconnettiti dall'API per chiudere la sessione (è necessario await):
    await api.logout()

### 7. Esegui la funzione async main() usando l'event loop di asyncio:
asyncio.run(main())
```

Esempi

I seguenti esempi mostrano attività comuni dell'API di Zabbix utilizzando la libreria zabbix_utils.

Ulteriori esempi sono disponibili nella directory `examples` del repository GitHub zabbix_utils.

Raccogliere dati da Zabbix agent

Panoramica

zabbix_utils consente di raccogliere dati da Zabbix agent (in modo simile a Zabbix get).

I dati possono essere raccolti in modalità sincrona o asincrona:

- In modalità sincrona, lo script Python richiede i dati e attende di riceverli prima di continuare, il che è adatto a operazioni semplici, sequenziali e prevedibili.
- In modalità asincrona, lo script richiede i dati senza attendere ogni risposta, consentendo ad altre operazioni di procedere in parallelo; questo è più efficiente per richieste lente o grandi quantità di dati.

Gli esempi in questa pagina si concentrano sulla modalità sincrona, sebbene la modalità asincrona segua schemi simili. Ulteriori esempi sono disponibili nel repository GitHub zabbix_utils.

Importazione

Per utilizzare zabbix_utils per raccogliere i valori degli item, importa la classe `Getter` nel tuo script Python:

```
from zabbix_utils import Getter
```

Dati della richiesta

Per richiedere il valore di un item:

1. Creare un'istanza `Getter`, specificando l'indirizzo IP e la porta del proprio agent Zabbix.
2. Chiamare il metodo `get()` sull'istanza `Getter`, specificando la chiave dell'item che si desidera recuperare.

Ad esempio, per richiedere i dati dell'item `system.uname`:

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

Utilizzo di un IP non predefinito

Se il server su cui viene eseguito lo script ha più indirizzi IP, puoi specificare un `source_ip` da usare per `Getter` quando si connette all'agent Zabbix:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

Uso del timeout

Puoi impostare un `timeout` di risposta per il `Getter` per controllare per quanto tempo il tuo script deve attendere una risposta dall'agent Zabbix prima di rinunciare:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

Uso della crittografia

`Getter` non include il supporto integrato per la crittografia, ma è possibile aggiungerlo creando un wrapper tramite librerie di terze parti:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementazione del wrapper TLS PSK per il socket
    # ...

agent = Getter(
    host='192.0.2.1',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Risposta

La risposta di Zabbix agent viene elaborata dalla libreria e restituita come oggetto `AgentResponse`:

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

Modalità asincrona

La modalità asincrona consente allo script di raccogliere valori senza attendere l'arrivo di ciascuno di essi. Questo può rendere lo script più efficiente quando deve raccogliere molti valori o quando alcuni valori richiedono molto tempo per essere raccolti.

Quando si usa la modalità asincrona, ci sono differenze importanti rispetto alla modalità sincrona:

- Importa il modulo `asyncio` di Python (devi prima installare le dipendenze richieste).
- Importa `AsyncGetter` invece di `Getter`.
- Scrivi il codice all'interno di una funzione `async`.
- Usa `await` quando chiami il metodo `get()`.

Ad esempio, per raccogliere un singolo valore usando la modalità asincrona:

```
### 1. Importa asyncio per la modalità asincrona e AsyncGetter da zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter

### 2. Definisci la funzione principale async in cui verranno eseguite tutte le richieste di dati:
async def main():
    agent = AsyncGetter(host='192.0.2.1', port=10050)

    # 3. Recupera il valore system.uname da agent Zabbix (deve essere preceduto da await):
    response = await agent.get('system.uname')

    # 4. Stampa il valore restituito da agent Zabbix:
    print(response.value)

### 5. Esegui la funzione main() asincrona usando il loop di eventi di asyncio:
asyncio.run(main())
```

Inviare dati al server o al proxy Zabbix

zabbix_utils consente di inviare valori di item a un item trapper sul server o proxy Zabbix (in modo simile a Zabbix sender).

È possibile inviare un singolo valore, più valori o persino indirizzare più cluster Zabbix.

I dati possono essere inviati in modalità sincrona o asincrona:

- In modalità sincrona, lo script Python invia i valori e attende una risposta prima di continuare; questa modalità è adatta per operazioni semplici, sequenziali e prevedibili.
- In modalità asincrona, lo script invia i valori senza attendere ogni risposta, consentendo ad altre operazioni di procedere in parallelo; questa modalità è più efficiente per richieste lente o grandi quantità di dati.

Gli esempi in questa pagina si concentrano sulla modalità sincrona, sebbene la modalità asincrona segua schemi simili. Ulteriori esempi sono disponibili nel repository GitHub zabbix_utils.

Importazione

Per utilizzare zabbix_utils per l'invio dei valori degli item, importa la classe `Sender` nel tuo script:

```
from zabbix_utils import Sender
```

Per inviare più valori, puoi anche importare la classe `ItemValue`:

```
from zabbix_utils import Sender, ItemValue
```

Inviare un singolo valore

Per inviare un valore di un item:

1. Creare un'istanza di `Sender`, specificando l'indirizzo IP e la porta del server o proxy Zabbix.
2. Chiamare il metodo `send_value()` sull'istanza `Sender` utilizzando il seguente formato:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Ad esempio, per inviare 1 al trapper item `service.status` del host `Linux server`:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Utilizzo di un IP non predefinito

Se il server che esegue lo script dispone di più indirizzi IP, è possibile specificare un `source_ip` che `Sender` utilizzerà durante l'invio dei valori al server o al proxy Zabbix:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Uso del timeout

Puoi impostare un `timeout` di risposta per `Sender` per controllare per quanto tempo il tuo script deve attendere una risposta dal server o dal proxy Zabbix prima di interrompere il tentativo:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Utilizzo del file di configurazione dell'agent

È possibile consentire a `zabbix_utils` di leggere i parametri `Server` o `ServerActive` da un file di configurazione locale dell'agent Zabbix o dell'agent 2. In questi casi, non è necessario specificare i parametri di connessione quando si crea un'istanza di `Sender`:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Se `ServerActive` contiene uno o più cluster Zabbix con più istanze del server, `Sender` invia i dati al primo server disponibile in ogni cluster. Se `ServerActive` non è impostato, viene utilizzato l'indirizzo specificato in `Server` con la porta predefinita (10051).

Utilizzo della crittografia

`Sender` non include il supporto integrato per la crittografia, ma è possibile fornirlo creando un wrapper tramite librerie di terze parti:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementazione del wrapper TLS PSK per il socket
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Risposta per valore singolo

La risposta restituita da Zabbix server o proxy viene elaborata dalla libreria e restituita come oggetto `TrapperResponse`:

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

Modalità asincrona

La modalità asincrona consente al tuo script Python di inviare valori senza attendere una risposta dal server o dal proxy Zabbix. Questo può rendere lo script più efficiente quando deve inviare molti valori o quando alcuni valori richiedono molto tempo per essere inviati.

Quando si utilizza la modalità asincrona, ci sono alcune differenze importanti rispetto alla modalità sincrona:

- Importa il modulo `asyncio` di Python (devi prima installare le dipendenze richieste).
- Importa `AsyncSender` invece di `Sender`.
- Scrivi il codice all'interno di una funzione `async`.
- Usa `await` quando chiami il metodo `send_value()`.

Ad esempio, per inviare un singolo valore utilizzando la modalità asincrona:

```
### 1. Importa asyncio per la modalità asincrona e AsyncSender da zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Definisci la funzione async principale in cui verranno eseguite tutte le operazioni di invio dei da
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Stampa la risposta restituita dal server o dal proxy Zabbix:
    print(response)

### 4. Esegui la funzione async main() utilizzando il ciclo di eventi di asyncio:
asyncio.run(main())
```

Inviare più valori

Per inviare più valori:

1. Preparare un array di oggetti `ItemValue`, ciascuno utilizzando lo stesso formato del metodo `send_value()`.
2. Creare un'istanza di `Sender`, specificando l'indirizzo IP e la porta del proprio server o proxy Zabbix.
3. Chiamare il metodo `send()` (invece di `send_value()`) sull'istanza `Sender`, specificando l'array di oggetti con i valori da inviare.

Ad esempio, per inviare cinque valori a host diversi:

```
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)
```

Utilizzo di una dimensione personalizzata dei blocchi

Se devi inviare più valori di quanti un item trapper possa accettare in una singola richiesta, puoi suddividerli in blocchi.

Per impostazione predefinita, la dimensione del blocco è di 250 valori. Puoi modificarla impostando il parametro `chunk_size` durante la creazione di un'istanza di `Sender`.

Ad esempio, per inviare cinque valori in tre blocchi (2-2-1), imposta il parametro `chunk_size` su 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)
```

Inviare valori a più cluster Zabbix

Per inviare valori a più cluster Zabbix:

1. Preparare un array di cluster Zabbix. Se un cluster ha più nodi, il valore verrà inviato al primo nodo **disponibile** di ciascun cluster.
2. Creare un `Sender`, specificando il proprio array di cluster Zabbix.
3. Chiamare il metodo `send_value()` sull'istanza `Sender` utilizzando lo stesso formato del metodo `send_value()`.

Ad esempio, per inviare un valore al primo nodo disponibile in ciascun cluster:

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]

sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

Risposta per più valori

Per impostazione predefinita, `Sender` restituisce un risultato aggregato dell'invio dei valori tra tutti gli host o cluster:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Se hai bisogno di informazioni più dettagliate, puoi esaminare i risultati per ciascun cluster e ciascun chunk utilizzando l'attributo `response.details`:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

```
if response.failed == 0:
    print(f"Valore inviato con successo in {response.time}")
else:
    print(response.details)
    # {
    #     127.0.0.1:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000051",
    #             "chunk": 1
    #         }
    #     ],
    #     zabbix.example.local:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000057",
    #             "chunk": 1
    #         }
    #     ]
    # }

    for node, chunks in response.details.items():
        for resp in chunks:
            print(f"Elaborati {resp.processed} di {resp.total} su {node.address}:{node.port}")
            # Elaborati 1 di 1 su 127.0.0.1:10051
            # Elaborati 1 di 1 su zabbix.example.local:10051
```

Logging di debug

Per risolvere i problemi durante l'utilizzo della API di Zabbix, la raccolta dei dati da Zabbix agent o l'invio dei dati a Zabbix server o proxy, è possibile abilitare il logging di debug.

La libreria zabbix_utils utilizza il modulo standard `logging` di Python.

Per visualizzare messaggi di debug dettagliati dalla libreria:

1. Importare il modulo `logging` di Python nello script.
2. Configurare il sistema di logging in modo che produca messaggi in un formato leggibile e impostare il livello di logging su 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)
```

Con il logging di debug abilitato, vengono stampate informazioni dettagliate su richieste e risposte, ad esempio:

```
[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
```

Moduli

Cos'è un modulo frontend PHP?

- Un modulo è un'entità con un ID univoco, nome, descrizione, autore e altri campi definiti nel suo file manifest, insieme a PHP, Javascript e altri file situati in un'unica directory all'interno della directory *modules* della tua installazione frontend Zabbix (ad esempio, *zabbix/ui/modules*).
- Un modulo deve rispettare semplici regole per garantire il corretto funzionamento.
- Un modulo deve essere installato (decompresso) e abilitato nel frontend dall'amministratore.

A cosa può servire un modulo

- Aggiunta di nuove funzionalità tramite sezioni frontend personalizzate;
- Creazione di tipi di widget dashboard personalizzati (vedi moduli widget);
- Sostituire o estendere la funzionalità esistente.

Per cosa non può essere utilizzato un modulo

- Registrazione di un nuovo metodo API o modifica di uno esistente.

Come funzionano i moduli

- Ad ogni richiesta HTTP viene avviato un modulo abilitato, prima di eseguire il codice dell'azione.
- Il modulo registrerà nuove azioni o ridefinirà quelle esistenti.
- Il modulo aggiungerà nuove sezioni frontend e rimuoverà o ridefinirà quelle esistenti.
- Il modulo si collegherà agli eventi frontend come onBeforeAction e onTerminate, se necessario.
- L'azione richiesta viene infine eseguita eseguendo il codice dell'azione, quello predefinito o quello definito dal modulo.

Dove andare dopo Sia che tu preferisca imparare facendo o leggere prima le linee guida, queste pagine contengono le informazioni e i passaggi necessari per creare i tuoi moduli:

- Tutorial passo passo per scrivere il tuo primo modulo
- Struttura del file del modulo
- Specifiche del modulo widget
- Esempi di moduli da riutilizzare

Struttura del file del modulo

Tutto il codice relativo a un modulo è memorizzato in una singola directory all'interno della directory dei **moduli** del frontend di Zabbix (ad esempio *zabbix/ui/modules/*).

```
example_module_directory/ (obbligatorio)
manifest.json (obbligatorio) Metadati e definizione delle azioni.
Module.php Inizializzazione del modulo e gestione degli eventi.
actions/ File del controller delle azioni.
SomethingView.php
SomethingCreate.php
SomethingDelete.php
data_export/
ExportAsXml.php
ExportAsExcel.php
views/ File delle viste.
example.something.view.php
example.something.delete.php
assets/ Qualsiasi file aggiuntivo da utilizzare nelle viste. Deve essere specificato in manifest.json.
```

```
js/ File JavaScript utilizzati nelle viste.
example.something.view.js.php
css/ File CSS utilizzati nelle viste.
example.something.css
image.png Immagini utilizzate nelle viste.
example.something.file Qualsiasi file da utilizzare nelle viste.
```

Albero dei file del modulo

Scrivere un module Un processo di scrittura di un module di esempio consiste nei seguenti passaggi (dove disponibile, fare clic sul nome del file o della cartella per visualizzare ulteriori dettagli sul passaggio):

1. Creare una nuova directory per il module all'interno di **zabbix/ui/modules/**.
2. Aggiungere il file manifest.json con i metadati del module.
3. Creare la cartella *views* e definire una o più view del module.
4. Creare la cartella *actions* e definire una o più action del module.
5. Creare il file Module.php (o Widget.php per i widget della dashboard) e definire le regole di inizializzazione e gestione degli eventi.
6. Creare la cartella *assets* per i file JavaScript (da inserire in *assets/js*), gli stili CSS (da inserire in *assets/css*) o qualsiasi altro file aggiuntivo.
7. Assicurarsi di specificare nel file manifest.json le view, le action e i file asset richiesti.
8. Registrare il module nel frontend di Zabbix e iniziare a usarlo.

Note:
È possibile registrare e abilitare un module non appena si crea il file manifest.json. Una volta abilitato il module, è possibile visualizzare immediatamente tutte le modifiche apportate ai file del module aggiornando il frontend di Zabbix.

manifest.json

Ogni modulo necessita del file manifest.json. Il file deve trovarsi nella directory principale del modulo (ad esempio, *zabbix/ui/modules/module_name/manifest.json*).

Come minimo indispensabile, manifest.json deve specificare questi campi:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

I parametri supportati in **manifest.json** (fai clic sul nome del parametro per una descrizione dettagliata):

Parameter	Description	Required
manifest_version	Versione del manifest del modulo.	Yes
id	ID univoco del modulo.	
name	Nome del modulo che verrà visualizzato nella sezione Amministrazione.	
namespace	Namespace PHP per le classi del modulo.	
version	Versione del modulo.	
type	Tipo del modulo. Per i widget deve essere impostato su *widget*	Yes for widgets, otherwise no
widget	Configurazione del widget. Usata solo per i widget.	
actions	Azioni da registrare con il modulo.	
assets	Fogli di stile CSS e file JavaScript da includere.	No
author	Autore del modulo.	
config	Valori predefiniti per le opzioni personalizzate del modulo.	
description	Descrizione del modulo.	
url	Un collegamento alla descrizione del modulo.	

manifest_version

Versione manifest del modulo. Attualmente, la versione supportata è **2.0**.

Type: Double

Esempio:

```
"manifest_version": 2.0
```

id

Module ID. Deve essere univoco. Al fine di evitare futuri conflitti di nome, è raccomandato utilizzare prefissi per i moduli (autore, nome della società o altro). Per esempio, se il modulo è un esempio per lezioni e il nome del modulo è ”My module”, allora l’ID sarà ”example_my_module”.

Type: String

Example:

```
"id": "example_my_module"
```

name

Nome del modulo che verrà mostrato nella sezione Administration.

Type: String

Example:

```
"name": "My module"
```

namespace

PHP Namespace per le classi del modulo.

Type: String

Example:

```
"namespace": "ClockWidget"
```

version

Versione del modulo. La versione verrà mostrata nella sezione Administration.

Type: String

Example:

```
"version": "1.0"
```

type

Tipo del modulo. Obbligatorio per i Widget e deve contenere la stringa ”widget”.

Type: String

Default: ”module”

Example:

```
"type": "widget"
```

actions

Azioni da registrare con il modulo. La definizione di una *class* object key per ciascuna azione è obbligatoria, altre action keys sono opzionali.

Type: Object

Object keys supportate se *type* è *module*:

- **write.your.action.name** (object) - nome dell’azione, deve essere scritto in minuscolo [a-z], separando le parole con dei punti. Supporta le chiavi:
  - **class** (string; required) - nome della classe dell’azione.
  - **layout** (string) - layout dell’azione. Valori supportati: *layout.json*, *layout.htmlpage* (default), *null*.
  - **view** (string) - view dell’azione.

Esempio:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

Object keys supportate se *type* è *widget*:

- **widget.{id}.view** (object) - file e nome della classe per la widget view. Sostituisci **{id}** con il valore id del widget (per esempio, *widget.example_clock.view*). Supporta le chiavi:
  - **class** (string; required) - nome della classe dell'azione per la modalità di visualizzazione widget da utilizzare per estendere la classe di default CControllerDashboardWidgetView. Il file sorgente della classe deve essere posto nella cartella *actions*.
  - **view** (string) - vista widget. Deve essere posto nella cartella *views*. Se il file di vista è chiamato *widget.view.php*, che è il default atteso, questo parametro può essere omesso. Se invece si usa un nome differente, specificatelo qui.
- **widget.{id}.edit** (object) - nome del file per la configurazione della vista widget. Sostituisci **{id}** con il valore id del widget (ad esempio, *widget.example_clock.edit*). Supporta le chiavi:
  - **class** (string; required) - nome della classe action per la vista della modalita di configurazione del widget. Il sorgente della classe deve essere posto nella cartella *actions*.
  - **view** (string) - vista configurazione del widget. Deve essere posta nella cartella *views*. Se il nome del file di vista è *widget.edit.php*, che è il default atteso, allora questo parametro può essere omesso. Se invece si usa un nome differente, specificatelo qui.

Esempio:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

assets

File CSS e JavaScript da includere

Tipo: Object

Object keys supportate: - **css** (array) - File CSS da includere. I file devono essere posizionati in *assets/css*. - **js** (array) - File JavaScript da includere. I file devono essere posizionati in *assets/js*.

Esempio:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

Autore

Autore del modulo. L'autore verrà mostrato nella sezione Amministrativa.

Type: String

Example:

```
"author": "John Smith"
```

config

Valori predefiniti per le opzioni del modulo. L'oggetto può contenere qualsiasi chiave personalizzata. Se specificati, questi valori verranno scritti nel database durante la registrazione del modulo. Le nuove variabili aggiunte successivamente verranno scritte alla prima chiamata. Successivamente, i valori delle variabili potranno essere modificati solo direttamente nel database.

Tipo: Oggetto

Esempio:

```
"config": {
"username": "Admin",
"password": "",
"auth_url": "https://example.com/auth"
}
```

descrizione

Descrizione del modulo.

Type: String

Esempio:

```
"description": "This is a clock widget."
```

widget

Configurazione del widget. Utilizzata se *type* è impostato su *widget*.

Tipo: Oggetto

Chiavi dell'oggetto supportate:

- **name** (stringa) - utilizzato nell'elenco dei widget e come intestazione predefinita. Se vuoto, verrà utilizzato il parametro "nome" del modulo.
- **size** (oggetto) - dimensioni predefinite del widget. Chiavi supportate:
- *width* (intero) - larghezza predefinita del widget.
- *height* (intero) - altezza predefinita del widget.
- **form_class** (stringa) - classe con il form dei campi del widget. Deve trovarsi nella directory *includes*. Se la classe è *WidgetForm.php*, come previsto per impostazione predefinita, questo parametro può essere omesso. Se si utilizza un nome diverso, specificarlo qui.
- **js_class** (stringa) - nome di una classe JavaScript per la modalità di visualizzazione del widget per estendere la classe CWidget predefinita. La classe verrà caricata con la dashboard. Il file sorgente della classe deve trovarsi nella directory assets/js. Vedi anche: assets.
- **use_time_selector** (booleano) - determina se il widget richiede un selettore di tempo per la dashboard. Valori supportati: *true*, *false* (predefinito).
- **refresh_rate** (intero) - frequenza di aggiornamento del widget in secondi (predefinita: 60).

Esempio:

```
"widget": {
"name": "",
"size": {
"width": 12,
"height": 5
},
"form_class": "WidgetForm",
"js_class": "CWidget",
"use_time_selector": false,
"refresh_rate": 60
}
```

url

Un collegamento alla descrizione del modulo. Per i widget, questo collegamento verrà aperto facendo clic sull'icona della guida ? nella finestra *Aggiungi widget* o *Modifica widget*. Se **url** non è specificato, facendo clic sull'icona della guida si aprirà la pagina generale Widget della dashboard.

Tipo: Stringa

Esempio:

```
"url": "http://example.com"
```

Azioni

Le azioni sono responsabili della 'logica di business' del modulo. Un'azione di solito è composta da un controller e una vista azione.

Un modulo può:

- Richiamare azioni già definite nel frontend di Zabbix.
- Sovrascrivere le azioni predefinite con azioni personalizzate.
- Definire azioni completamente nuove.

Per sovrascrivere un comportamento predefinito di un'azione con un comportamento personalizzato, definire un'azione con lo stesso nome nella configurazione del modulo. Quando l'azione viene richiamata, verrà eseguita l'azione del modulo al posto dell'azione Zabbix predefinita.

I file delle azioni devono essere memorizzati nella cartella *actions*. Le azioni devono essere specificate nel file manifest.json.

Controller

Flusso di lavoro del controller delle azioni:

1. Verificare che tutti i parametri passati in una richiesta HTTP siano validi:

- Chiamare il metodo *checkInput()* del controller;
- Utilizzare le regole di validazione definite in CNewValidator.php;
- Chiamare il metodo *validateInput()*.

2. Verificare i permessi dell'utente.
3. Preparare i dati in base ai parametri passati: se *checkInput()* restituisce true, Zabbix chiama il metodo *doAction()* del controller.
4. Preparare l'array **$data** per la vista. Utilizzare *CControllerResponseData* e il metodo *setResponse()* per memorizzare la risposta nell'array **$data**.

Esempio:

```
/**
 * Validate input parameters.
 *
 * @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;
}

/**
 * Check user permissions.
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Execute action and generate response object.
 */
protected function do Action(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
```

```
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

> **Note:**
> È possibile visualizzare l'elenco completo delle classi controller disponibili nel codice sorgente di Zabbix.

Viste

Il file della vista riceve i dati da un controller e quindi ne prepara l'aspetto HTML.

> **Note:**
> La definizione di una o più viste per un modulo del frontend è facoltativa, a meno che il modulo non sia un widget.
 I widget della dashboard richiedono almeno due viste: una per la modalità di modifica e una per la modalità di visualizzazione (devono essere archiviate nella directory *views*).

È possibile utilizzare nella vista classi HTML Zabbix predefinite (da */zabbix/ui/include/classes/html*), nonché aggiungere nuove classi HTML e CSS. Le nuove classi devono essere archiviate nella cartella *assets* del modulo.

Esempio:

```
...
(new CColHeader(_('Name')))
```

Questo aggiungerà una nuova colonna con nome *Name* e applicherà lo stile alla riga superiore della tabella come nelle altre pagine di Zabbix.

Action view

Questo è il file di esempio per la definizione di una vista azione.

```
<?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();
```

Risorse

La cartella *assets* può contenere file e sottocartelle che non appartengono ad altre directory. Puoi usarlo per:

- Stili JavaScript (devono essere all'interno di *assets/js*);
- Stili CSS (devono essere all'interno di *assets/css*);
- Immagini;
- Fonts;
- Qualsiasi altra cosa che devi includere.

assets/js

La directory *assets/js* è riservata e deve contenere solo file JavaScript. Per essere utilizzati dal widget, specificare questi file in *manifest.json*.

Per esempio:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css è riservato e dovrebbe contenere solamente file di stile CSS. Per essere utilizzato dal widget, crea dei riferimenti a questi file nel *manifest.json*.

Ad esempio:

```
"assets": {
    "css": ["mywidget.css"]
}
```

Stili CSS

I file CSS possono contenere un attributo personalizzato `theme` per definire stili diversi per temi specifici del frontend.

Temi disponibili e relativi valori dell'attributo:

- **Blu** - [theme='blue-theme']
- **Blu (classico)** - [theme='blue-classic-theme']
- **Scuro** - [theme='dark-theme']
- **Blu scuro** - [theme='dark-blue-theme']
- **Scuro (classico)** - [theme='dark-classic-theme']
- **Chiaro ad alto contrasto** - [theme='hc-light']
- **Scuro ad alto contrasto** - [theme='hc-dark']

Esempio:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

Registrare un nuovo modulo

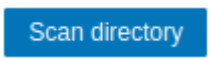
Questa sezione spiega come aggiungere un nuovo modulo al frontend di Zabbix.

Prerequisiti Prima di procedere, assicurati che:

- Il modulo si trovi nella directory *modules* dell'installazione del frontend Zabbix (ad esempio, *zabbix/ui/modules*).
- Il modulo abbia almeno una versione base del file manifest.json.
- Hai accesso alla sezione del menu Amministrazione di Zabbix (richiede il ruolo utente Super admin).

> **Note:**
> Il frontend non installerà né riconoscerà i moduli incompatibili.

Aggiunta di un modulo Apri la pagina *Administration* > *General* > *Modules* e fai clic su *Scan directory*.



Individua il tuo modulo nell'elenco e attivalo.

Per attivare un modulo, fai clic sul collegamento ipertestuale *Disabled* - lo stato del modulo cambierà in *Enabled*.

Fai clic sul nome del modulo per visualizzare ulteriori informazioni sul modulo, come autore, versione o una breve descrizione (se definita nel manifest).

Anteprima del widget I moduli widget, una volta aggiunti, diventano immediatamente visibili nell'elenco dei widget della dashboard.

Puoi aprire una dashboard, passare alla modalità di modifica e aggiungere il widget alla dashboard come di consueto.

Quando apporti alcune modifiche al widget, aggiorna la dashboard per vedere come appare il widget con gli aggiornamenti più recenti.

Widget

I widget sono componenti del frontend di Zabbix utilizzati per le dashboard. Salvo diversa indicazione, tutte le linee guida per i componenti sono applicabili anche ai widget.

Tuttavia, un widget è notevolmente diverso da un componente. Per creare un widget:

- specificare il tipo "widget" nel file manifest.json ("type": "widget");
- includere almeno due viste: una per la modalità di presentazione del widget e una per la modalità di configurazione del widget (example.widget.view.php e example.widget.edit.php);
- e un controller per la presentazione dei widget (WidgetView.php);
- utilizzare ed estendere le classi widget predefinite.

Configurazione

Questa pagina descrive le classi che possono essere utilizzate per creare una vista di configurazione del widget con campi di configurazione personalizzati. La vista di configurazione del widget è la parte del widget che consente all'utente di configurare i parametri del widget per la presentazione.

Widget

La classe widget primaria, estende la classe base di tutti i widget della dashboard - *CWidget*. Necessaria per sovrascrivere il comportamento predefinito del widget.

La classe *Widget* deve trovarsi nella directory principale del widget (ad esempio, *zabbix/ui/modules/my_custom_widget*).

Esempio di Widget.php

```
<?php

namespace Modules\MyCustomWidget;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

public const MY_CONSTANT = 0;

public function getTranslationStrings(): array {
return [
'class.widget.js' => [
'No data' => _('No data')
]
];
}
}
```

WidgetForm

La classe *WidgetForm* estende la classe predefinita *CWidgetForm* e contiene un insieme di campi *CWidgetField* necessari per definire la struttura di archiviazione della configurazione del widget nel database e gestire la validazione dell'input.

La classe *WidgetForm* deve trovarsi nella directory *includes*. Se la classe ha un nome diverso, il nome deve essere specificato nel parametro *widget/form_class* nel file *manifest.json*.

Esempio di includes/WidgetForm.php

```php
<?php

namespace Modules\MyCustomWidget\Includes;

use Modules\MyCustomWidget\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox,
    CWidgetFieldColor
};


class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

La classe *CWidgetFormView* è necessaria per specificare la logica di presentazione dei campi definiti nella classe *WidgetForm*, determinando il loro aspetto e comportamento durante il rendering nella vista di configurazione.

La classe *CWidgetFormView* supporta i seguenti metodi:

- *addField()* - riceve come parametro un'istanza della classe *CWidgetField*; ogni classe *CWidgetField* dispone di una classe *CWidgetFieldView* corrispondente da utilizzare nella vista di configurazione del widget.
- *addFieldset()* - riceve un'istanza della classe *CWidgetFieldsGroupView*, che combina i campi in un contenitore comprimibile.
- *addFieldsGroup()* - riceve un'istanza di *CWidgetFormFieldsetCollapsibleView*, che combina visivamente (con un bordo) i campi in un gruppo.
- *includeJsFile()* - consente di aggiungere un file JavaScript alla vista di configurazione del widget.
- *addJavaScript()* - consente di aggiungere codice JavaScript inline che verrà eseguito non appena la vista di configurazione del widget viene caricata.

La classe *CWidgetFormView* deve trovarsi nella directory *views*.

Esempio di views/widget.edit.php

```php
<?php

/**
 * My custom widget form view.
 *
```

```
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

Una classe JavaScript può essere utilizzata per aggiungere comportamento dinamico e interattività alla vista di configurazione del widget. Ad esempio, è possibile inizializzare un selettore di colori, definito nella classe *CWidgetFormView*.

La classe JavaScript deve essere caricata insieme al modulo, pertanto deve essere referenziata nella classe *CWidgetFormView* utilizzando i metodi *includeJsFile()* e *addJavaScript()*.

Nell'esempio seguente, un'istanza singleton della classe viene creata immediatamente e memorizzata con il nome *window.my_custom_widget_form*. Pertanto, aprendo il modulo una seconda volta, l'istanza verrà ricreata.

La classe JavaScript deve trovarsi nella directory *views*.

Esempio di views/widget.edit.js.php

```
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

    init({color_palette}) {
        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }
    }
};
```

CWidgetField

La classe *CWidgetField* è una classe di base da cui ereditano tutte le classi dei campi dei moduli (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList*, ecc.). Le classi che estendono *CWidgetField* sono responsabili della ricezione, del

salvataggio e della convalida dei valori di configurazione del widget.

Sono disponibili le seguenti classi *CWidgetField*.

CWidgetField class	Database field type	Description
CWidgetFieldCheckBox	int32	Casella di controllo singola.
CWidgetFieldCheckBoxList	array of int32	Più caselle di controllo sotto un singolo campo di configurazione.
CWidgetFieldColor	string	Campo per la selezione del colore.
CWidgetFieldDatePicker	string	Campo per la selezione della data.
CWidgetFieldIntegerBox	int32	Campo per l'inserimento di un numero intero. Può essere utilizzato per configurare i valori minimo e massimo.
CWidgetFieldLatLng	string	Casella di testo per l'inserimento di latitudine, longitudine e livello di zoom della mappa, separati da virgole.
CWidgetFieldMultiSelect	-	Classe di base per i campi multiselezione, estesa da tutte le classi `CWidgetFieldMultiSelect*`.
CWidgetFieldMultiSelectAction	ID	Campo multiselezione per selezionare le azioni dall'elenco delle azioni definite in *Alerts* > *Actions*.
CWidgetFieldMultiSelectGraph	ID	Campo multiselezione per selezionare i grafici personalizzati.
CWidgetFieldMultiSelectGraphPrototype	ID	Campo multiselezione per selezionare i prototipi di grafici personalizzati.
CWidgetFieldMultiSelectGroup	ID	Campo multiselezione per selezionare i gruppi di host.
CWidgetFieldMultiSelectHost	ID	Campo multiselezione per selezionare gli host.
CWidgetFieldMultiSelectHostInventory	array of int32	Campo multiselezione per selezionare i campi dell'inventario degli host.
CWidgetFieldMultiSelectItem	ID	Campo multiselezione per selezionare gli item.
CWidgetFieldMultiSelectItemPrototype	ID	Campo multiselezione per selezionare i prototipi di item.
CWidgetFieldMultiSelectMap	ID	Campo multiselezione per selezionare le mappe.
CWidgetFieldMultiSelectMediaType	ID	Campo multiselezione per selezionare i tipi di supporto.
CWidgetFieldMultiSelectOverrideHost	ID	Campo multiselezione per selezionare un'origine dati per gli host da una dashboard o da un altro widget.
CWidgetFieldMultiSelectService	ID	Campo multiselezione per selezionare i servizi.
CWidgetFieldMultiSelectSla	ID	Campo multiselezione per selezionare gli SLA.
CWidgetFieldMultiSelectUser	ID	Campo multiselezione per selezionare gli utenti.
CWidgetFieldNavTree	array of (multiple mixed)	Memorizza una struttura ad albero di navigazione delle mappe da utilizzare nel widget *Map navigation tree*.
CWidgetFieldNumericBox	string	Campo per l'inserimento di un numero con virgola mobile.
CWidgetFieldPatternSelect	-	Classe di base per i campi di selezione dei pattern, estesa da tutte le classi `CWidgetPatternSelect*`.
CWidgetFieldPatternSelectHost	string	Campo multiselezione per selezionare gli host. Supporta i pattern dei nomi degli host.
CWidgetFieldPatternSelectItem	string	Campo multiselezione per selezionare gli item. Supporta i pattern dei nomi degli item.
CWidgetFieldRadioButtonList	int32	Gruppo di pulsanti di opzione costituito da uno o più pulsanti di opzione.
CWidgetFieldRangeControl	int32	Dispositivo di scorrimento per selezionare un valore di tipo intero.
CWidgetFieldReference	string	Memorizza un identificatore univoco per questo widget nella dashboard; viene utilizzato per la comunicazione tra widget.
CWidgetFieldSelect	int32	Casella di selezione a discesa.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* preimpostato con le gravità dei trigger.
CWidgetFieldSparkline	array of (multiple mixed)	Consente di configurare un grafico sparkline.
CWidgetFieldTags	array of (string, int32, string)	Consente di configurare una o più righe di filtri per tag.
CWidgetFieldTextArea	string	Area di testo per l'inserimento di testo su più righe.
CWidgetFieldTextBox	string	Casella di testo per l'inserimento di testo su una sola riga.
CWidgetFieldThresholds	array of (string, string)	Consente di configurare coppie di colori e numeri.
CWidgetFieldTimePeriod	array of string	Campo per la selezione del periodo di tempo.
CWidgetFieldTimeZone	string	Elenco a discesa con i fusi orari.
CWidgetFieldUrl	string	Casella di testo per l'inserimento di URL.

Note:
La lunghezza massima per i campi `int32` è di 10 caratteri, per i campi `ID` è di 20 caratteri e per i campi `string` è di 65535 caratteri. Per i tipi compositi (ad esempio, `array of (string, int32, string)`), il limite si applica a ogni elemento singolarmente. Le restrizioni sulla lunghezza specifiche del campo possono essere impostate utilizzando metodi come `setMaxLength()`.

Presentazione

Questa pagina descrive i componenti che possono essere utilizzati per creare una vista di presentazione del widget. La vista di presentazione del widget è la parte del widget che riceve i dati in base alla sua configurazione e li visualizza sulla dashboard in un contenitore.

La vista di presentazione è composta da tre parti:

- Azione del widget
- Vista del widget
- JavaScript

Azione del widget

La classe di azione del widget (*WidgetView*) contiene metodi per le operazioni con i widget nella modalità di visualizzazione della presentazione. La maggior parte delle azioni dei widget utilizza e/o estende la classe controller predefinita *CControllerDashboard-WidgetView*.

La classe di azione del widget deve trovarsi nella directory *actions* ed essere specificata nel parametro *actions* (*actions/widget.{id}.view/class*) nel file *manifest.json*.

Esempio di actions/WidgetView.php (implementato nel widget nativo di Zabbix *Informazioni di sistema*)

```
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()
            ]
        ]));
    }
}
```

Vista del widget

La classe della vista del widget (*CWidgetView*) è responsabile della costruzione della vista di presentazione del widget.

La classe della vista del widget deve trovarsi nella directory *views*. Se il file che contiene la classe della vista del widget ha un nome diverso da quello predefinito (*widget.view.php*), allora deve essere specificato nel file *manifest.json* nel parametro *actions* (*actions/widget.{id}.view/view*).

Esempio di views/widget.view.php

```
<?php

/**
 * My custom widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
```

```
    )
    ->show();
```

JavaScript

La classe JavaScript è responsabile della definizione del comportamento del widget, ad esempio l'aggiornamento dei dati del widget, il ridimensionamento del widget, la visualizzazione degli elementi del widget, ecc.

Tutte le operazioni JavaScript utilizzano e/o estendono la classe JavaScript di base di tutti i widget della dashboard - *CWidget*. La classe *CWidget* contiene un insieme di metodi con l'implementazione predefinita del comportamento del widget. A seconda della complessità del widget, questi metodi possono essere utilizzati così come sono oppure estesi.

La classe *CWidget* contiene i seguenti metodi:

- Metodi che definiscono il ciclo di vita del widget: *onInitialize()*, *onStart()*, *onActivate()*, *onDeactivate()*, *onDestroy()*, *onEdit()*.
- Metodi che gestiscono l'aggiornamento e la visualizzazione dei dati del widget: *promiseUpdate()*, *getUpdateRequestData()*, *processUpdateResponse(response)*, *processUpdateErrorResponse(error)*, *setContents(response)*.
- Metodi che modificano l'aspetto del widget: *onResize()*, *hasPadding()*.

La classe JavaScript deve trovarsi nella directory *assets/js* ed essere specificata nel parametro *assets* (*assets/js*) nel file *manifest.json*.

Metodi del ciclo di vita

I metodi del ciclo di vita del widget vengono invocati dalla dashboard, e in diverse fasi del ciclo di vita del widget durante la sua esistenza all'interno della dashboard.

Il metodo ***onInitialize()*** definisce lo stato iniziale e/o i valori del widget, senza eseguire alcuna manipolazione di HTML o dei dati. Questo metodo viene richiamato quando un widget viene creato (viene istanziato un oggetto widget), in genere aggiungendo il widget a una pagina della dashboard o caricando la pagina della dashboard.

Esempio:

```
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;
}
```

Il metodo ***onStart()*** definisce la struttura HTML del widget, senza eseguire alcuna manipolazione dei dati. Questo metodo viene invocato prima della prima attivazione della pagina della dashboard, cioè prima che la dashboard e i relativi widget siano completamente visualizzati all'utente.

Esempio:

```
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`
        );
    }
}
```

Il metodo ***onActivate()*** rende il widget attivo e interattivo abilitando listener di eventi personalizzati (per rispondere alle azioni dell'utente) e avviando il ciclo di aggiornamento del widget (per mantenere il suo contenuto aggiornato). Questo metodo viene richiamato quando la pagina della dashboard viene attivata, cioè quando viene visualizzata completamente nell'interfaccia utente.

Si noti che prima che il metodo *onActivate()* venga richiamato, il widget si trova nello stato inattivo (`WIDGET_STATE_INACTIVE`). Dopo il richiamo riuscito, il widget passa allo stato attivo (`WIDGET_STATE_ACTIVE`). Nello stato attivo, il widget è reattivo, ascolta gli eventi, aggiorna periodicamente il proprio contenuto e può interagire con altri widget.

Esempio:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

Il metodo ***onDeactivate()*** interrompe qualsiasi attività e interattività del widget disattivando i listener di eventi personalizzati e arrestando il ciclo di aggiornamento del widget. Questo metodo viene richiamato quando la pagina della dashboard viene disattivata, cioè quando si passa a un'altra pagina o viene eliminata, oppure quando il widget viene eliminato dalla pagina della dashboard.

Si noti che prima che il metodo *onDeactivate()* venga richiamato, il widget si trova nello stato attivo (`WIDGET_STATE_ACTIVE`). Dopo il richiamo riuscito, il widget passa allo stato inattivo (`WIDGET_STATE_INACTIVE`).

Esempio:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

Il metodo ***onDestroy()*** esegue le operazioni di pulizia prima che il widget venga eliminato dalla dashboard, che possono includere la chiusura di una connessione al database stabilita durante l'inizializzazione del widget, la rimozione dei dati temporanei per liberare memoria di sistema ed evitare perdite di risorse, la deregistrazione degli event listener relativi agli eventi di ridimensionamento o ai clic sui pulsanti per prevenire la gestione non necessaria degli eventi e perdite di memoria, ecc. Questo metodo viene richiamato quando il widget o la pagina della dashboard che lo contiene viene eliminata.

Si noti che prima che il metodo *onDestroy()* venga richiamato, un widget in stato attivo (`WIDGET_STATE_ACTIVE`) viene sempre disattivato tramite il richiamo del metodo *onDeactivate()*.

Esempio:

```
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);
    }
}
```

Il metodo ***onEdit()*** definisce l'aspetto e il comportamento del widget quando la dashboard passa alla modalità di modifica. Questo metodo viene richiamato quando la dashboard passa alla modalità di modifica, in genere quando un utente interagisce con il pulsante *Modifica* del widget o con il pulsante *Modifica dashboard* della dashboard.

Esempio:

```
onEdit() {
    this._deactivateGraph();
}
```

Metodi del processo di aggiornamento

I metodi del processo di aggiornamento del widget sono responsabili del recupero dei dati aggiornati dal server Zabbix o da qualsiasi altra origine dati e della loro visualizzazione nel widget.

Il metodo ***promiseUpdate()*** avvia il processo di aggiornamento dei dati recuperandoli, in genere tramite richieste web o chiamate API. Questo metodo viene richiamato quando viene visualizzata una pagina della dashboard e successivamente a intervalli periodici, finché la pagina della dashboard non viene cambiata con un'altra pagina della dashboard.

Di seguito è riportato un esempio dell'implementazione predefinita del metodo *promiseUpdate()* utilizzata dalla maggior parte dei widget nativi di Zabbix. Nell'implementazione predefinita, il metodo *promiseUpdate()* segue uno schema generale per il recupero dei dati dal server. Crea un nuovo oggetto `Curl` con l'URL appropriato e i parametri della richiesta, invia una richiesta `POST` usando il metodo *fetch()* con l'oggetto dati costruito dal metodo *getUpdateRequestData()*, ed elabora la risposta (o una risposta di errore) rispettivamente con *processUpdateResponse(response)* o *processUpdateErrorResponse(error)*. Questa implementazione è adatta alla maggior parte dei widget, poiché in genere recuperano i dati in formato JSON e li gestiscono in modo coerente.

```
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);
        });
    }
```

Il metodo ***getUpdateRequestData()*** prepara i dati della richiesta al server per l'aggiornamento del widget raccogliendo varie proprietà e i relativi valori (identificatori del widget, impostazioni dei filtri, intervalli di tempo, ecc.) dallo stato e dalla configurazione del widget, e costruendo un oggetto dati che rappresenta le informazioni necessarie da inviare al server nella richiesta di aggiornamento. Questo metodo viene richiamato solo come parte del metodo predefinito *promiseUpdate()*, cioè durante il processo di aggiornamento del widget.

Implementazione predefinita:

```
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
    };
}
```

Il metodo ***processUpdateResponse(response)*** gestisce la risposta ricevuta dal server dopo la richiesta di aggiornamento e, se il processo di aggiornamento è andato a buon fine e senza errori, svuota i dati del widget e visualizza i nuovi contenuti con il metodo *setContents()*. Questo metodo viene richiamato solo come parte del metodo predefinito *promiseUpdate()*, cioè durante il processo di aggiornamento del widget.

Implementazione predefinita:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

Il metodo ***processUpdateErrorResponse(error)*** gestisce la risposta ricevuta dal server dopo la richiesta di aggiornamento se la risposta è un errore e visualizza il/i messaggio/i di errore. Questo metodo viene richiamato solo come parte del metodo predefinito

promiseUpdate(), cioè durante il processo di aggiornamento del widget.

Implementazione predefinita:

```
processUpdateErrorResponse(error) {
    this._updateMessages(error.messages, error.title);
}
```

Il metodo ***setContents(response)*** visualizza il contenuto del widget se il processo di aggiornamento del widget è stato completato con successo e senza errori, il che può includere la manipolazione di elementi DOM, l'aggiornamento di componenti dell'interfaccia utente, l'applicazione di stili o formattazione, ecc. Questo metodo viene richiamato solo come parte del metodo predefinito *processUpdateResponse(response)*, cioè durante il processo di gestione della risposta ricevuta dal server dopo la richiesta di aggiornamento.

Implementazione predefinita:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

Metodi di modifica della presentazione

I metodi di modifica della presentazione del widget sono responsabili della modifica dell'aspetto del widget.

Il metodo ***onResize()*** è responsabile dell'adattamento degli elementi visivi del widget per adeguarsi alle nuove dimensioni del widget, operazione che può includere la riorganizzazione degli elementi, la regolazione delle dimensioni degli elementi, il troncamento del testo, l'implementazione del lazy loading per migliorare la reattività durante il ridimensionamento, ecc. Questo metodo viene richiamato quando il widget viene ridimensionato, ad esempio quando l'utente ridimensiona manualmente il widget o quando viene ridimensionata la finestra del browser.

Esempio:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

Il metodo ***hasPadding()*** è responsabile dell'applicazione di un padding verticale di 8 px nella parte inferiore del widget quando è configurato per mostrare la propria intestazione. Questo metodo viene richiamato quando la pagina della dashboard viene attivata, cioè quando diventa la pagina visualizzata nell'interfaccia utente.

Implementazione predefinita:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

Per alcuni widget è necessario utilizzare tutto lo spazio disponibile del widget per configurare, ad esempio, un colore di sfondo personalizzato. Di seguito è riportato un esempio dell'implementazione del metodo *hasPadding()* utilizzato nel widget *Item value* nativo di Zabbix.

```
hasPadding() {
    return false;
}
```

Tutorial

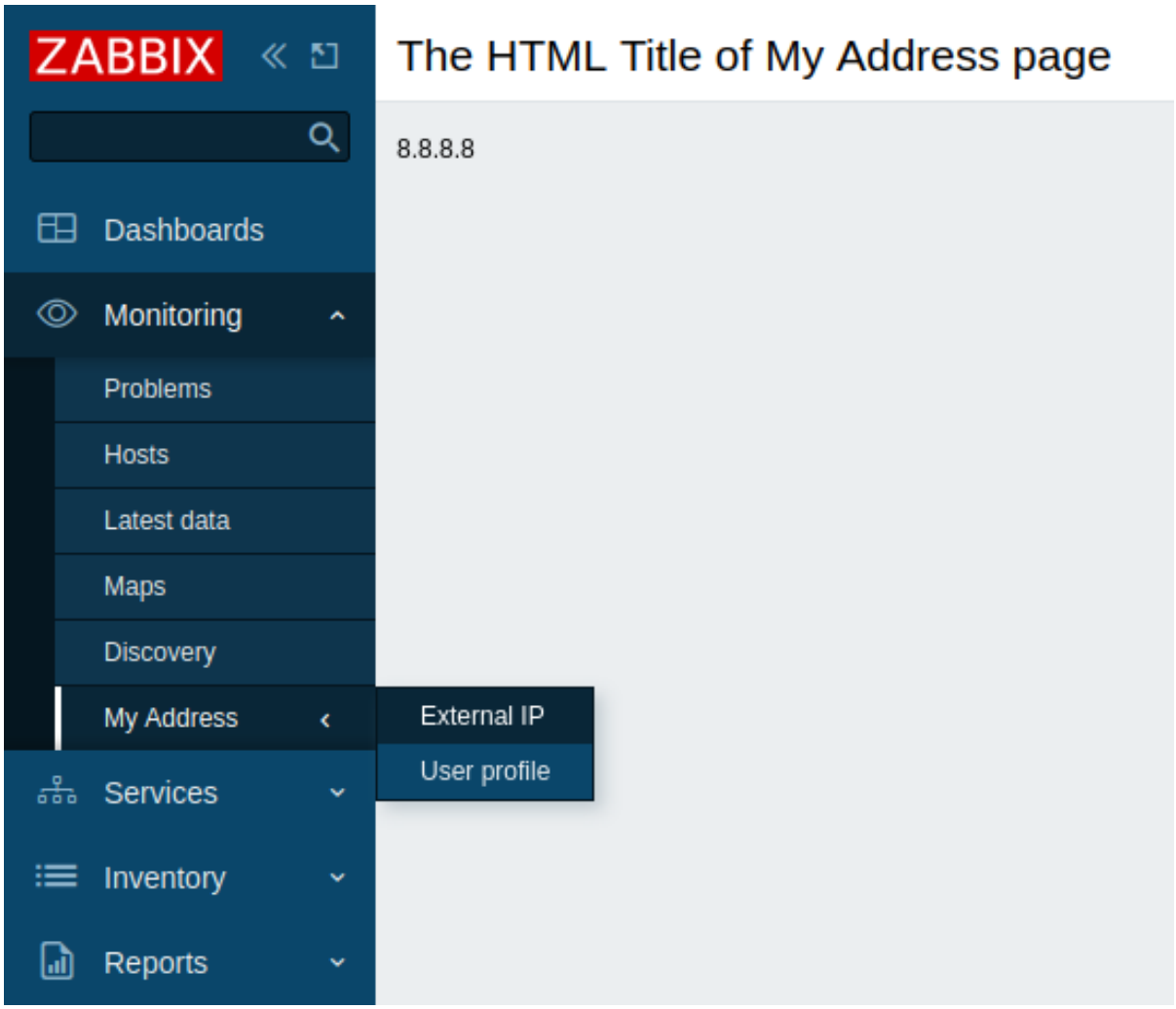
Questa sezione contiene pratici tutorial passo-passo per illustrare come creare un modulo personalizzato e un widget in Zabbix.

Crea un modulo (tutorial)

Questo è un tutorial passo-passo che mostra come creare un semplice modulo frontend Zabbix. Puoi scaricare tutti i file di questo modulo come archivio ZIP: MyAddress.zip.

Cosa realizzerai

Durante questo tutorial, creerai innanzitutto un modulo frontend che aggiunge una nuova sezione di menu *My Address* e poi lo convertirai in un modulo frontend più avanzato che effettua una richiesta HTTP a *https://api.seeip.org* e visualizza la risposta - l'indirizzo IP del tuo computer - in una nuova pagina nella sezione di menu *My Address* appena creata. Ecco come apparirà il modulo completato:



Parte I - Nuova sezione del menu

Aggiungere un modulo vuoto al frontend di Zabbix

1. Creare una directory *MyAddress* nella directory *modules* dell'installazione del frontend di Zabbix (ad esempio, *zabbix/ui/modules*).
2. Creare un file *manifest.json* con i metadati di base del modulo (vedere la descrizione dei parametri supportati).

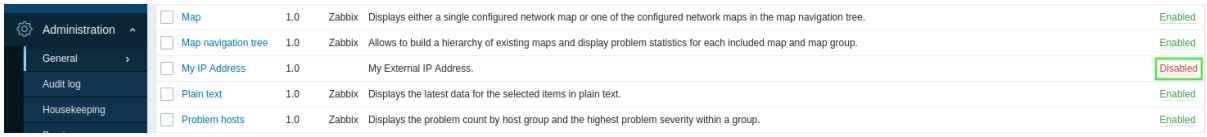
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Nel frontend di Zabbix, andare alla sezione *Administration* > *General* > *Modules* e fare clic sul pulsante *Scan directory*.



4. Individuare il nuovo modulo *My IP Address* nell'elenco e fare clic sul collegamento ipertestuale "Disabled" per modificare lo stato del modulo da "Disabled" a "Enabled" (se il modulo non è elencato, vedere la sezione troubleshooting).



Il modulo è ora registrato nel frontend. Tuttavia, non è ancora visibile, perché è necessario definire la funzionalità del modulo. Una volta aggiunto contenuto alla directory del modulo, le modifiche saranno immediatamente visibili nel frontend di Zabbix dopo l'aggiornamento della pagina.

Creare una sezione di menu

1. Creare un file *Module.php* nella directory *MyAddress*.

Questo file implementa una nuova classe *Module* che estende la classe predefinita *CModule*. La classe *Module* inserirà una nuova sezione di menu *My Address* nel menu principale.

Il metodo *setAction()* specifica un'azione da eseguire quando si fa clic sulla sezione di menu. Per iniziare, puoi usare l'azione predefinita *userprofile.edit*, che aprirà la pagina *User profile*. Nella parte III di questo tutorial, imparerai a creare un'azione personalizzata.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

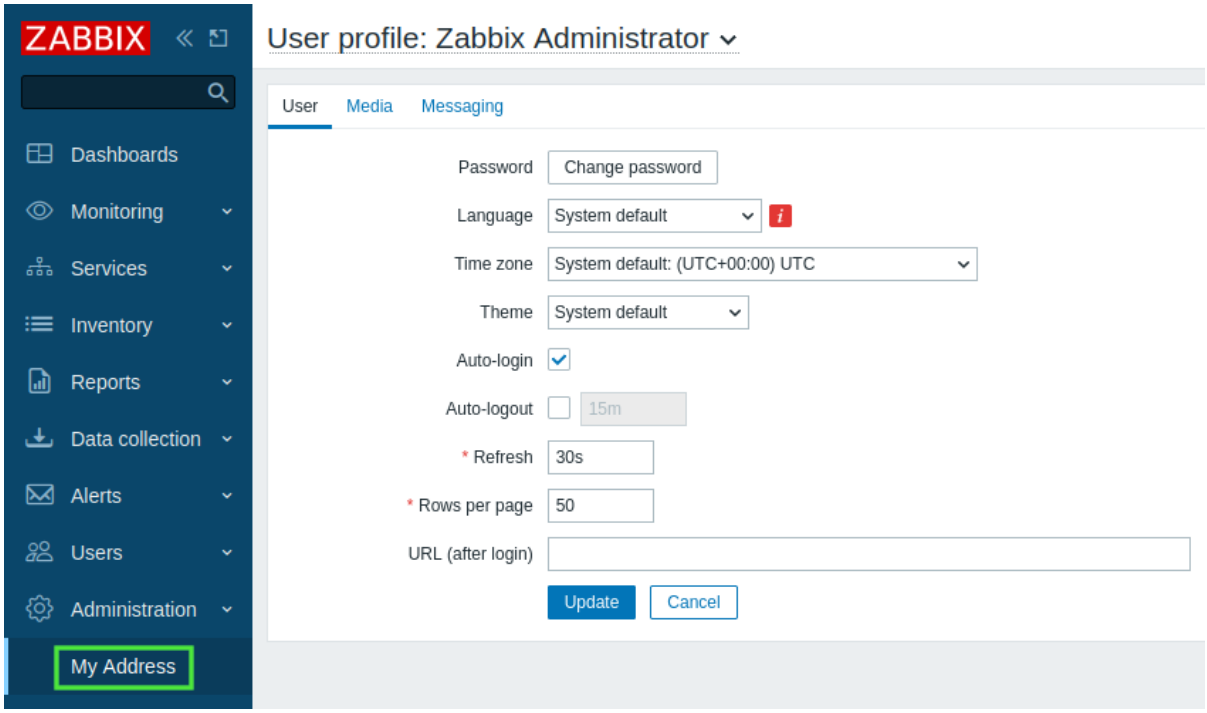
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Puoi sostituire `'userprofile.edit'` con altre azioni, ad esempio `'charts.view'` (apre i grafici personalizzati), `'problems.view'` (apre *Monitoring* > *Problems*), oppure `'report.status'` (apre il report *System information*).

3. Aggiorna il frontend di Zabbix. Ora in fondo al menu principale di Zabbix è presente una nuova sezione *My Address*. Fai clic su *My Address* per aprire la pagina *User profile*.



Parte II - Modifica della posizione della sezione di menu

In questa parte, sposterai la sezione di menu *My Address* nella sezione *Monitoring* e poi vi aggiungerai un menu annidato. Di conseguenza, gli utenti potranno accedere a due pagine del sottomenu dalla sezione di menu *Monitoring* > *My Address*.

1. Apri e modifica il file *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

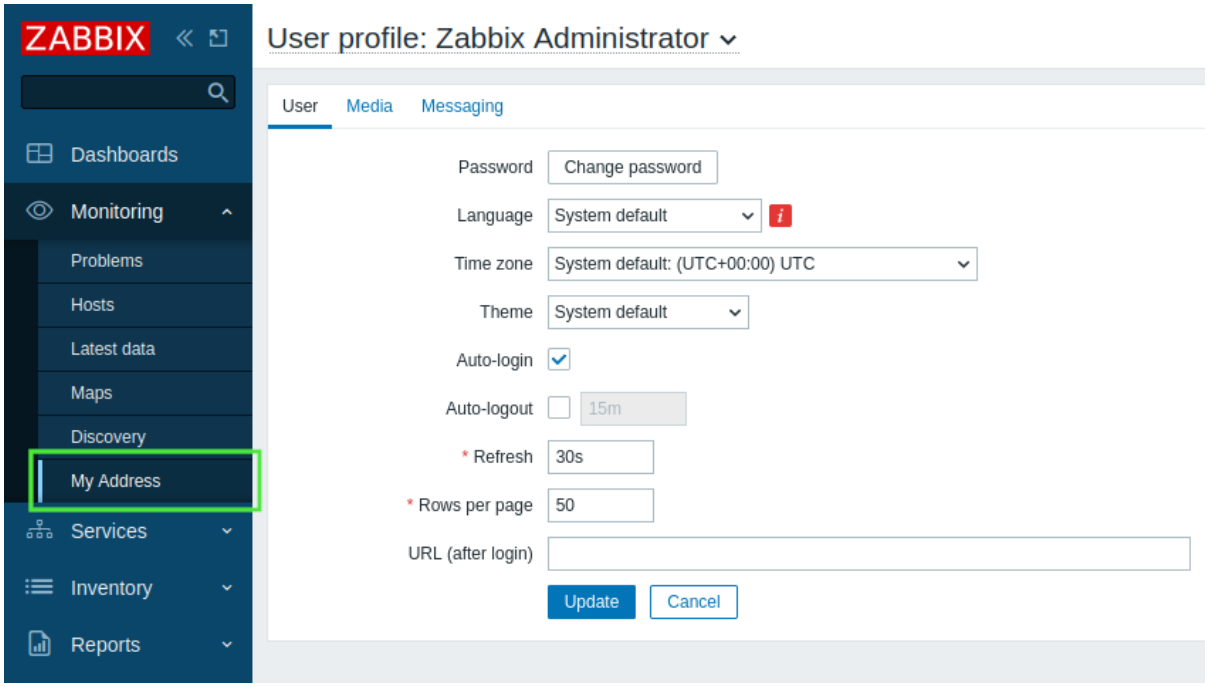
namespace Modules\MyAddress;

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. Aggiorna il frontend di Zabbix. Espandi la sezione di menu *Monitoring* e osserva che la sezione *My address* si trova ora sotto la sezione *Discovery*.



3. Per aggiungere pagine annidate alla sezione di menu *My Address*, apri di nuovo e modifica il file *Module.php*.

Questo passaggio crea due sottosezioni:

- *External IP* che esegue una nuova azione 'my.address' che verrà definita nei passaggi successivi;
- *User profile* che esegue l'azione predefinita 'userprofile.edit' per aprire la pagina *User profile*.

Nota che, per il menu annidato, devi usare la classe *CMenu* oltre alle classi utilizzate nei passaggi precedenti.

ui/modules/MyAddress/Module.php

```
<?php

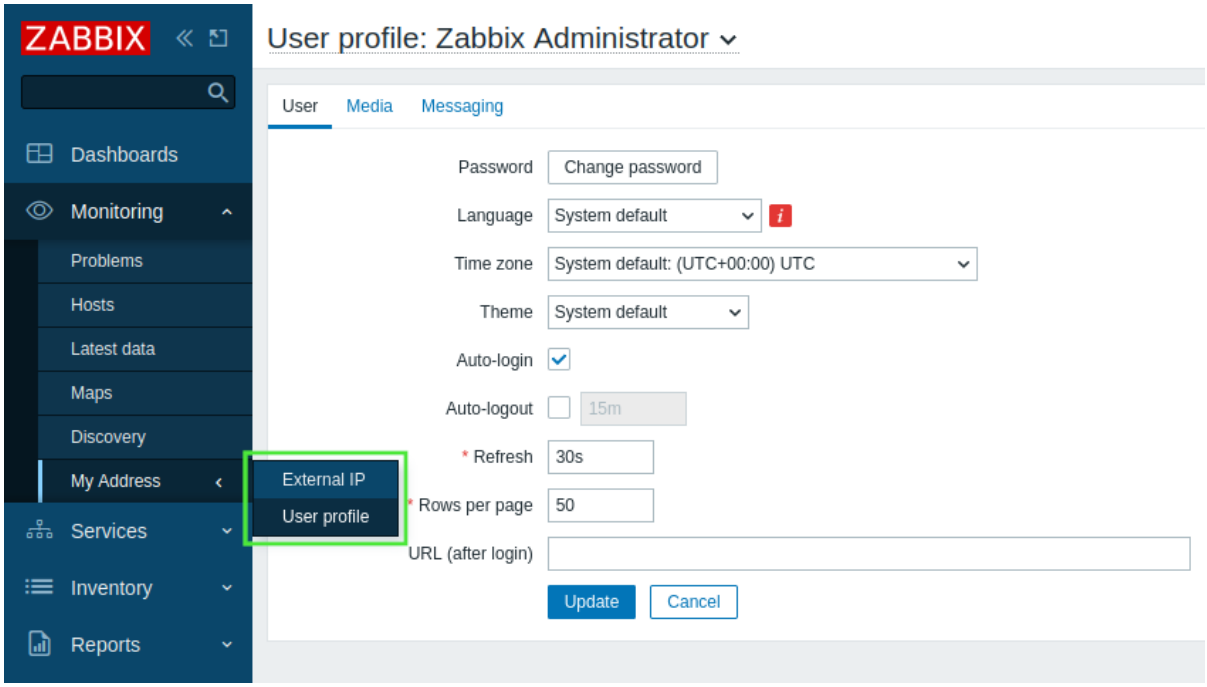
namespace Modules\MyAddress;

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. Aggiorna il frontend di Zabbix. Osserva che la sezione di menu *My address* ora contiene un menu di terzo livello con due pagine - *External IP* e *User profile*.



Parte III - Azione del modulo

Un'azione viene implementata in due file - *actions/MyAddress.php* e *views/my.address.php*. Il file ***actions/MyAddress.php*** si occupa dell'implementazione della logica di business, mentre il file ***views/my.address.php*** è responsabile della view.

1. Crea una directory *actions* nella directory *MyAddress*.
2. Crea un file *MyAddress.php* nella directory *actions*.

La logica dell'azione sarà definita nella classe *MyAddress*. Questa classe di azione implementerà quattro funzioni: *init()*, *checkInput()*, *checkPermissions()* e *doAction()*. Zabbix frontend chiama la funzione *doAction()* quando viene richiesta l'azione. Questa funzione è responsabile della logica di business del modulo.

Attention:
I dati devono essere organizzati come un array associativo. L'array può essere multidimensionale e può contenere qualsiasi dato previsto dalla view.

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. Crea una nuova directory *views* nella directory *MyAddress*.
4. Crea un file *my.address.php* nella directory *views* e definisci la view del modulo.

Nota che la variabile `$data` è disponibile nella view senza doverla definire esplicitamente. Il framework passa automaticamente l'array associativo alla view.

ui/modules/MyAddress/views/my.address.php

```
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

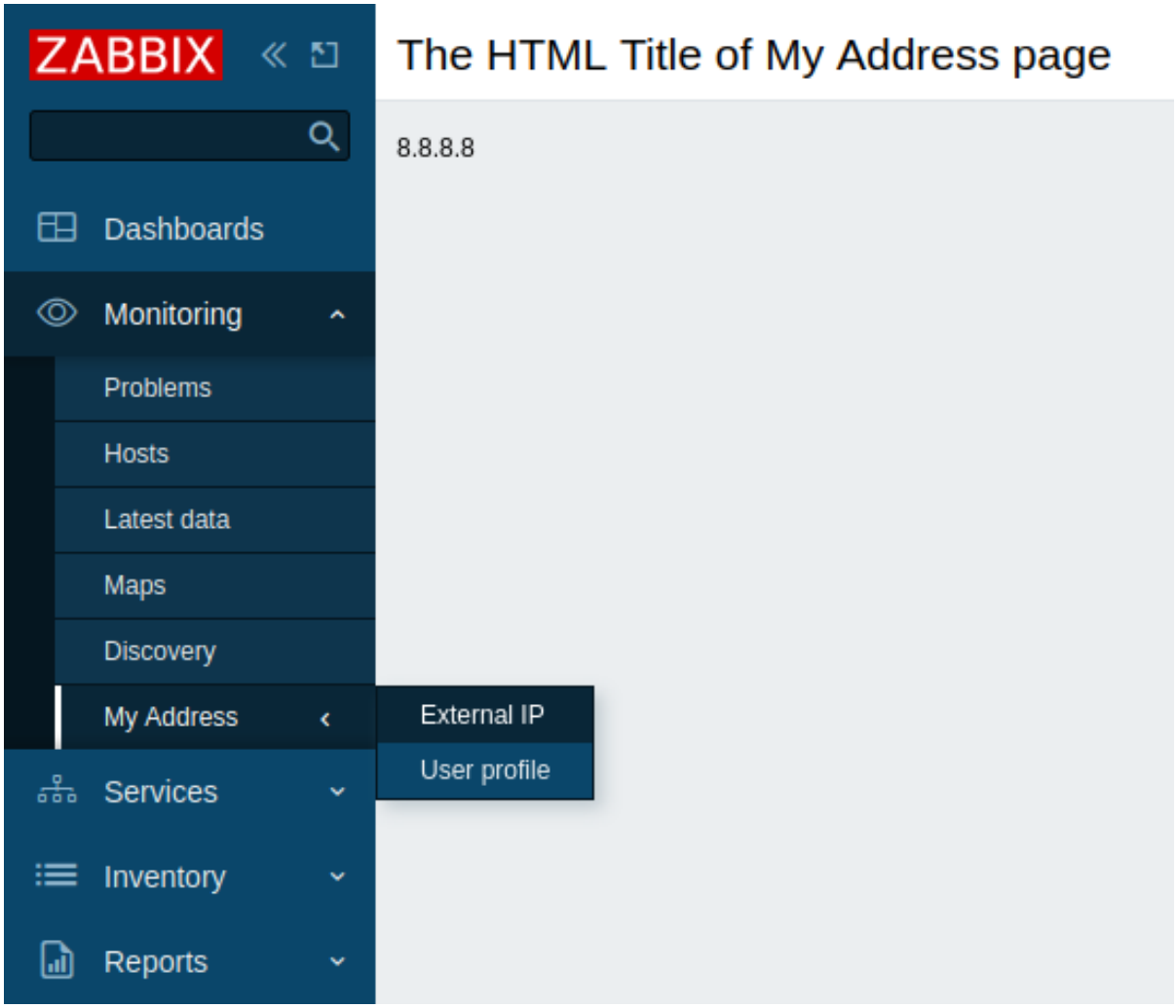
5. L'azione del modulo deve essere registrata nel file *manifest.json*. Apri *manifest.json* e aggiungi un nuovo oggetto `actions` che contenga:

- la chiave dell'azione con il nome dell'azione scritto in minuscolo (a-z) e con le parole separate da punti (ad esempio, `my.address`);
- il nome della classe dell'azione (`MyAddress`) come valore per la chiave `class` dell'oggetto `my.address`;
- il nome della view dell'azione (`my.address`) come valore per la chiave `view` dell'oggetto `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

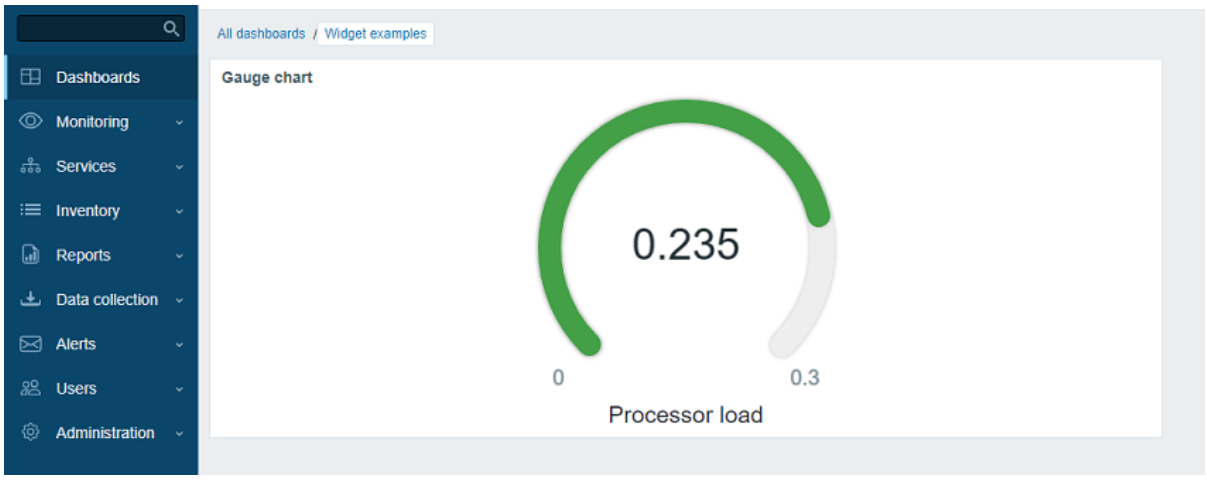
6. Aggiorna Zabbix frontend. Fai clic su *My address* > *External IP* per vedere l'indirizzo IP del tuo computer.



Crea un widget (tutorial)

Questo è un tutorial passo passo che mostra come creare un semplice widget per la dashboard. Puoi scaricare tutti i file di questo widget in un archivio ZIP: lesson_gauge_chart.zip.

Cosa creerai Durante questo tutorial, creerai innanzitutto un widget di base "Hello, world!" e poi lo convertirai in un widget più avanzato che visualizza il valore di un item come grafico a indicatore. Ecco come apparirà il widget completato:



Parte I - "Hello, world!" In questa sezione imparerai come creare gli elementi widget minimi richiesti e aggiungere un nuovo widget al frontend di Zabbix.

Aggiungere un widget vuoto al frontend di Zabbix

1. Creare una directory `lesson_gauge_chart` nella directory `modules` dell'installazione del frontend di Zabbix (ad esempio, *zabbix/ui/modules*).

Note:
Tutti i widget personalizzati sono considerati moduli esterni e devono essere aggiunti alla directory `modules` dell'installazione del frontend di Zabbix (ad esempio, *zabbix/ui/modules*). La directory `zabbix/ui/widgets` è riservata ai widget integrati di Zabbix e viene aggiornata insieme alla UI di Zabbix.

2. Creare un file `manifest.json` con i metadati di base del widget (vedere la descrizione dei parametri supportati).

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"
}
```

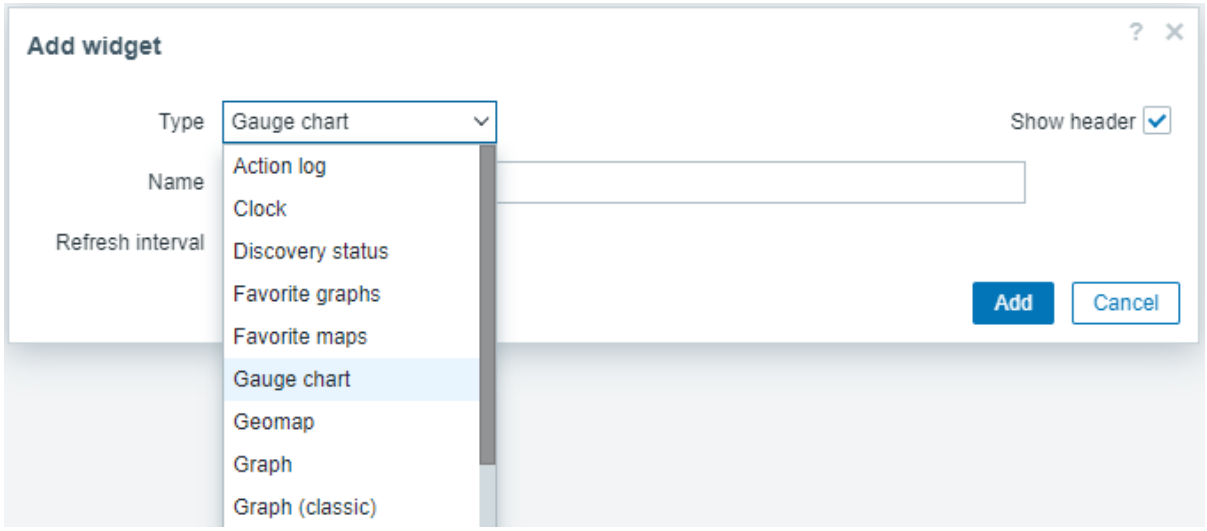
3. Nel frontend di Zabbix, andare alla sezione *Administration* > *General* > *Modules* e fare clic sul pulsante *Scan directory*.



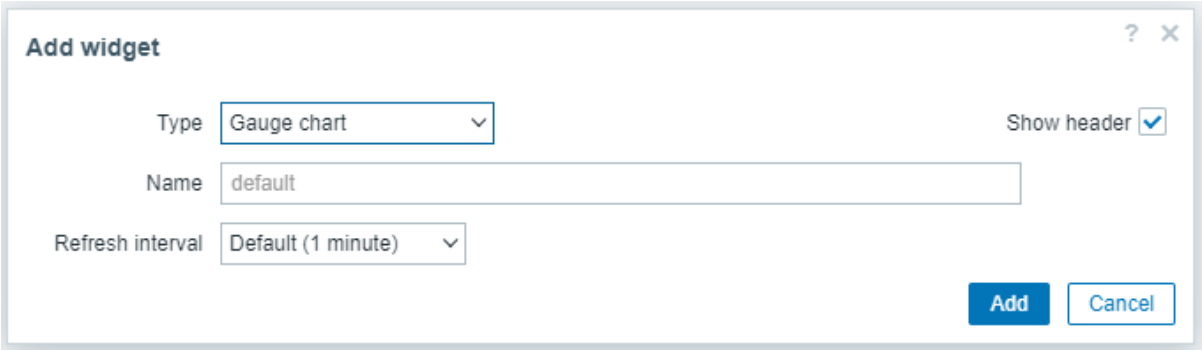
4. Individuare il nuovo modulo *Gauge chart* nell'elenco e fare clic sul collegamento ipertestuale "Disabled" per modificare lo stato del modulo da "Disabled" a "Enabled" (se il modulo non è elencato, vedere la sezione troubleshooting).



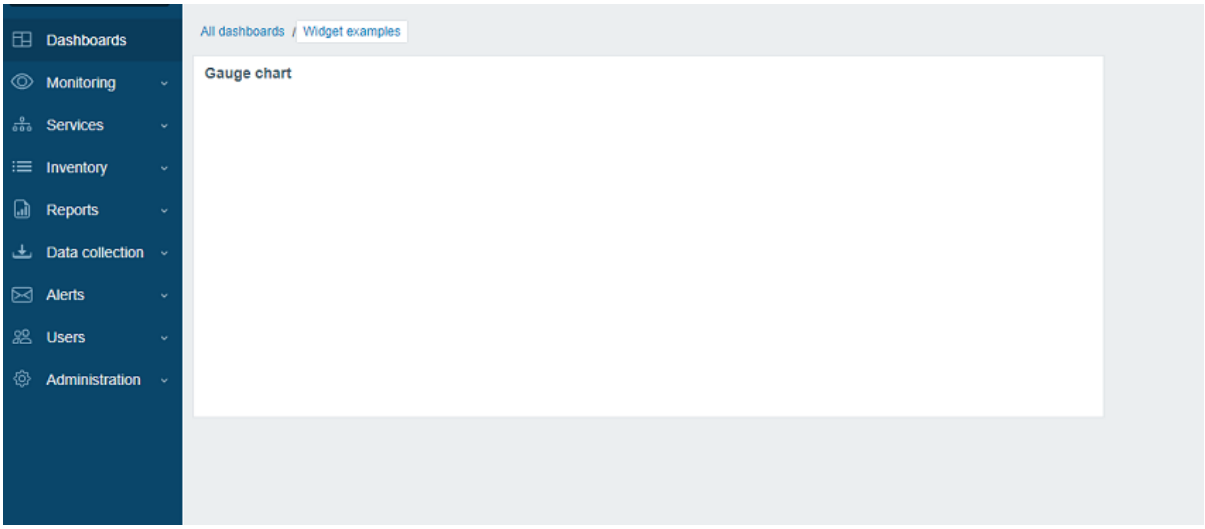
5. Aprire una dashboard, passare alla modalità di modifica e aggiungere un nuovo widget. Nel campo *Type*, selezionare "Gauge chart".



6. A questo punto, la configurazione del widget *Gauge chart* contiene solo i campi comuni del widget *Name* e *Refresh interval*. Fare clic su *Add* per aggiungere il widget alla dashboard.



7. Dovrebbe apparire un widget vuoto nella dashboard. Fare clic su *Save changes* nell'angolo in alto a destra per salvare la dashboard.



Aggiungere una vista del widget

> **Note:**
> Il file **view** del widget deve trovarsi nella directory `views` (per questo tutorial, `ui/modules/lesson_gauge_chart/views/`). Se il file ha il nome predefinito `widget.view.php`, non è necessario registrarlo nel file `manifest.json`. Se il file ha un nome diverso, specificarlo nella sezione `actions/widget.lesson_gauge_chart.view` del file `manifest.json`.

1. Creare una directory `views` nella directory `lesson_gauge_chart`.
2. Creare un file `widget.view.php` nella directory `views`.

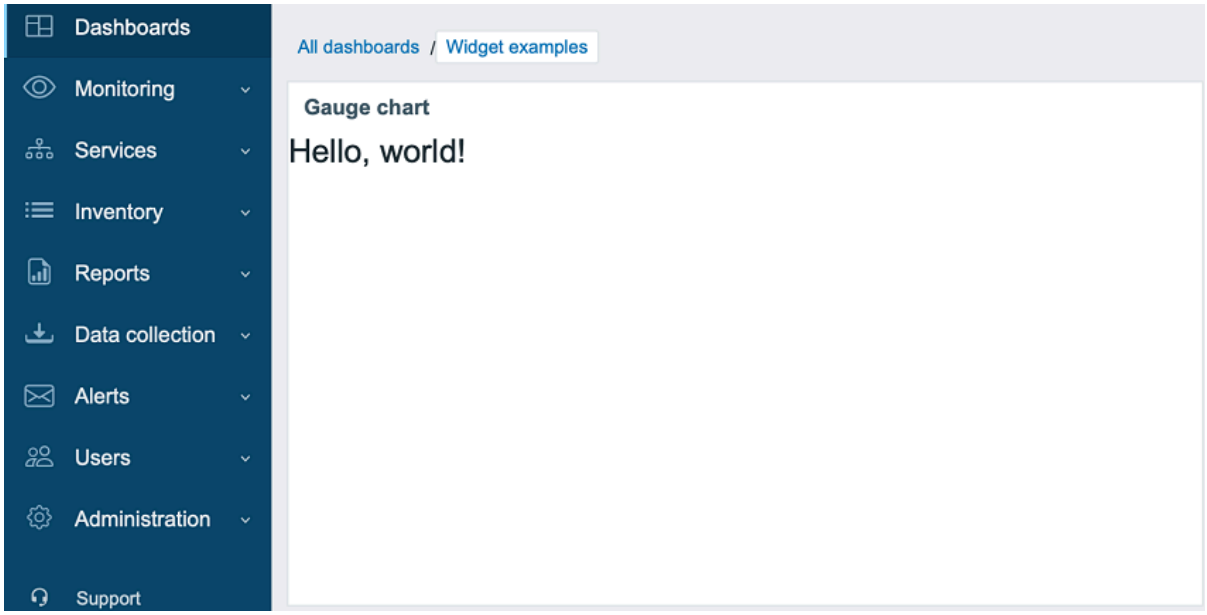
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Aggiornare la dashboard. Il widget *Gauge chart* ora mostra "Hello, world!".



Parte II - Grafico degli indicatori Aggiungere impostazioni a una vista di configurazione e usarle in una vista del widget

In questa sezione, imparerai come aggiungere un campo di configurazione del widget e mostrare il valore inserito nella vista del widget come testo.

La configurazione del widget è composta da un form (`Zabbix\Widgets\CWidgetForm`) e da una vista del form del widget (`widget.edit.php`). Per aggiungere campi (`Zabbix\Widgets\CWidgetField`), devi creare una classe `WidgetForm`, che estenderà `Zabbix\Widgets\CWidgetForm`.

Il form contiene un insieme di campi (`Zabbix\Widgets\CWidgetField`) di vari tipi, utilizzati per convalidare i valori inseriti dall'utente. Il campo del form (`Zabbix\Widgets\CWidgetField`) per ogni tipo di elemento di input converte il valore in un formato unico da memorizzare nel database.

> **Note:**
> Il file **form** del widget deve trovarsi nella directory `includes` (per questo tutorial, `ui/modules/lesson_gauge_chart/includes/`). Se il file ha il nome predefinito `WidgetForm.php`, non è necessario registrarlo nel file `manifest.json`. Se il file ha un nome diverso, specificarlo nella sezione `widget/form_class` del file `manifest.json`.

1. Crea una nuova directory `includes` nella directory `lesson_gauge_chart`.
2. Crea un file `WidgetForm.php` nella directory `includes`.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. Aggiungi un campo *Description* al form di configurazione del widget. Si tratta di un normale campo di testo, in cui l'utente può inserire qualsiasi set di caratteri. Per questo puoi usare la classe `CWidgetFieldTextBox`.

`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. Nella directory `views`, crea un file di vista di configurazione del widget `widget.edit.php` e aggiungi una vista per il nuovo campo *Description*. Per la classe di campo `CWidgetFieldTextBox`, la vista è `CWidgetFieldTextBoxView`.

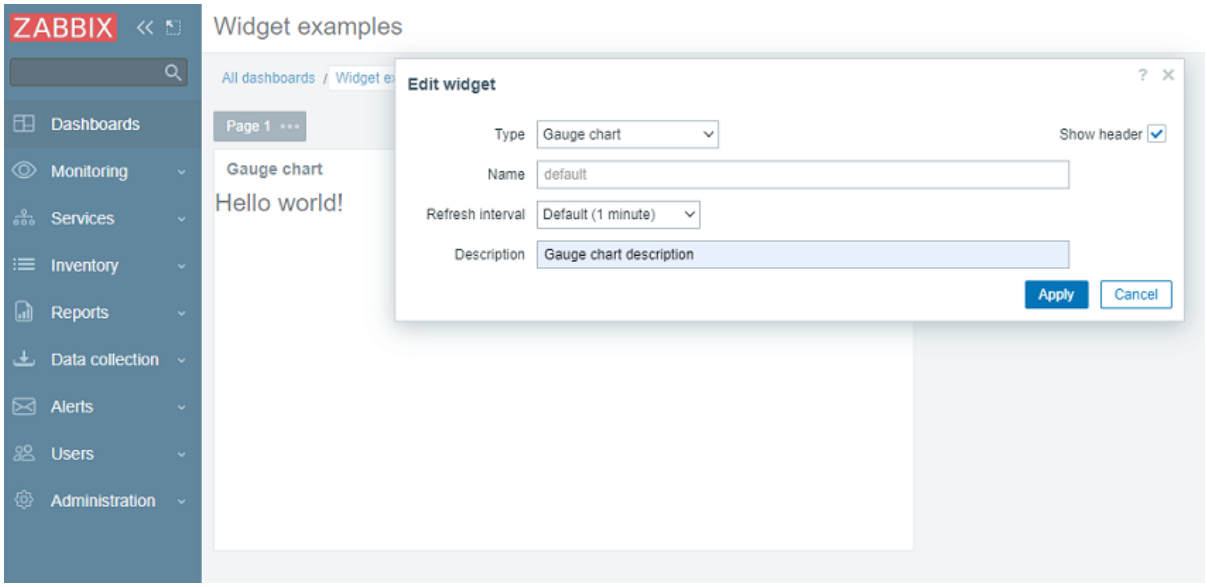
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Vai alla dashboard e fai clic sull'icona a ingranaggio nel widget per aprire il form di configurazione del widget.
6. Il form di configurazione del widget ora contiene un nuovo campo di testo *Description*. Inserisci un valore qualsiasi, ad esempio *Gauge chart description*.



7. Fai clic su *Apply* nel form di configurazione del widget. Quindi fai clic su *Save changes* nell'angolo in alto a destra per salvare la dashboard. Nota che la nuova descrizione non è visibile da nessuna parte e il widget continua a mostrare "Hello, world!".

Per fare in modo che la nuova descrizione appaia nel widget, il valore del campo *Description* deve essere recuperato dal database e passato alla vista del widget. Per questo, devi creare una classe di azione.

8. Crea una nuova directory `actions` nella directory `lesson_gauge_chart`.
9. Crea un file `WidgetView.php` nella directory `actions`. La classe di azione `WidgetView` estenderà la classe `CControllerDashboardWidgetView`.

I valori dei campi di configurazione del widget sono memorizzati nella proprietà `$fields_values` della classe di azione.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```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. Apri `manifest.json` e registra `WidgetView` come classe di azione nella sezione `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. Ora puoi usare il valore del campo description, contenuto in `$data['description']`, nella vista del widget. Apri `views/widget.view.php` e sostituisci il testo statico "Hello, world!" con `$data['description']`.

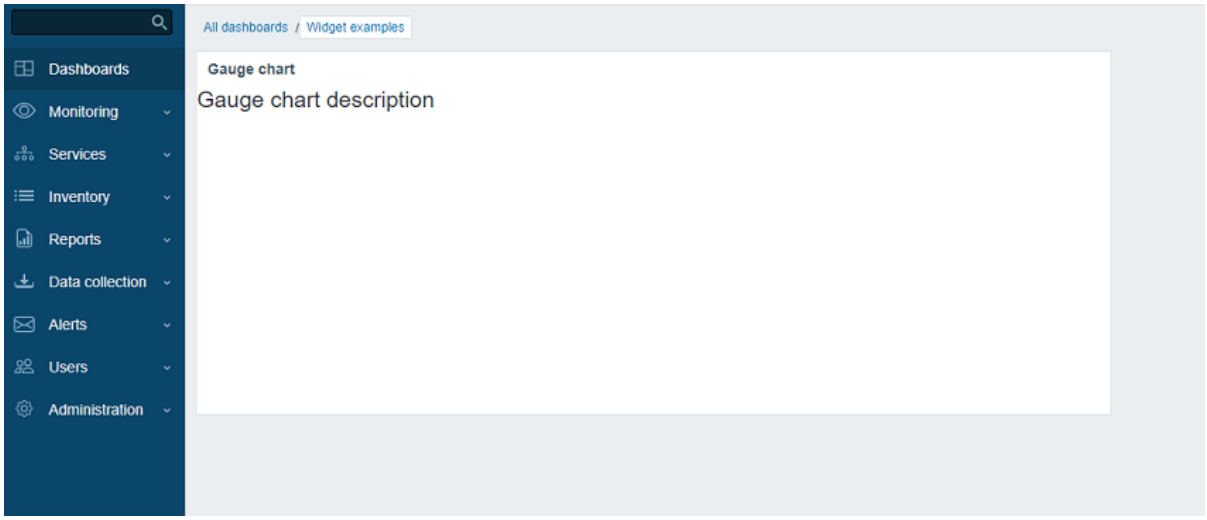
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```php
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. Aggiorna la pagina della dashboard. Ora dovresti vedere il testo della descrizione del widget al posto di "Hello, world!".



Recuperare il valore di un item tramite API

Il widget dovrebbe mostrare l'ultimo valore di un item scelto dall'utente. Per farlo, è necessario aggiungere la possibilità di selezionare gli item nella configurazione del widget.

In questa sezione, imparerai come aggiungere un campo di selezione item al modulo del widget e come aggiungere la parte visiva di questo campo alla vista di configurazione. Poi, il controller del widget potrà recuperare i dati dell'item e il suo valore tramite una richiesta API. Una volta ricevuto, il valore potrà essere visualizzato nella vista del widget.

1. Apri `includes/WidgetForm.php` e aggiungi il campo `CWidgetFieldMultiSelectItem`. Questo consentirà di selezionare un item nel modulo di configurazione.

`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. Apri `views/widget.edit.php` e aggiungi il componente visivo del campo alla vista di configurazione.

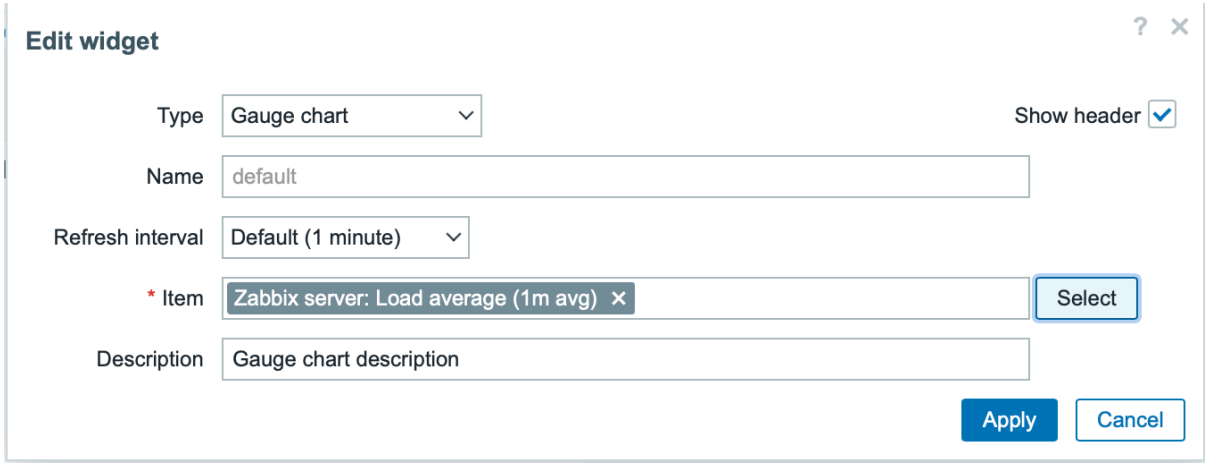
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```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. Torna alla dashboard e fai clic sull'icona a ingranaggio nel widget per aprire il modulo di configurazione del widget.
4. Il modulo di configurazione del widget ora contiene un nuovo campo di input *Item*. Seleziona l'host "Zabbix server" e l'item "Load average (1m avg)".



5. Fai clic su *Apply* nel modulo di configurazione del widget. Poi fai clic su *Save changes* nell'angolo in alto a destra per salvare la dashboard.
6. Apri e modifica `actions/WidgetView.php`.

Da questo momento, l'ID dell'item sarà disponibile nel controller del widget in `$this->fields_values['itemid']`. Il metodo del controller `doAction()` raccoglie i dati dell'item (nome, tipo di valore, unità) usando il metodo API `item.get` e l'ultimo valore dell'item usando il metodo API `history.get`.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
```

```
            ]
        ]);

        $value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'value' => $value,
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

7. Apri `views/widget.view.php` e aggiungi il valore dell'item alla vista del widget.

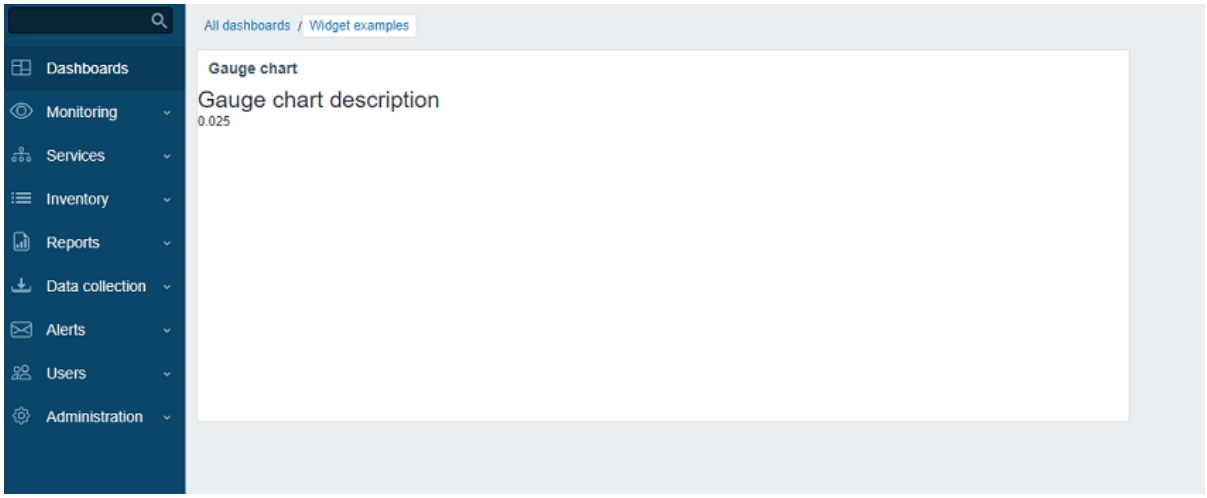
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. Aggiorna la pagina della dashboard. Il widget mostrerà l'ultimo valore dell'item.



Aggiungere impostazioni di configurazione avanzate a una vista di configurazione

In questa sezione imparerai come aggiungere una sezione *Configurazione avanzata* espandibile/comprimibile con parametri opzionali, come colore, valori minimo e massimo, unità e il campo *Description* creato in precedenza.

1. Crea un file `Widget.php` nella directory principale del widget `lesson_gauge_chart` per creare una nuova classe `Widget`.

La classe `Widget` estenderà la classe base `CWidget` per aggiungere/sovrascrivere le impostazioni predefinite del widget (in questo caso - le traduzioni). Il JavaScript, fornito di seguito, mostra la stringa "No data" in caso di dati mancanti. La stringa "No data" è presente nei file di traduzione dell'interfaccia utente di Zabbix.

Se sono presenti costanti del widget, è consigliabile specificarle anche nella classe `Widget`.

`ui/modules/lesson_gauge_chart/Widget.php`

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. Apri `includes/WidgetForm.php` e aggiungi i nuovi campi *Color* (selettore colore), *Min* (campo numerico), *Max* (campo numerico) e *Units* (selezione), e definisci la palette di colori predefinita per il selettore colore, in modo che possa essere utilizzata nei passaggi successivi.

`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. Apri `views/widget.edit.php` e aggiungi i componenti visivi dei campi alla vista di configurazione.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
```

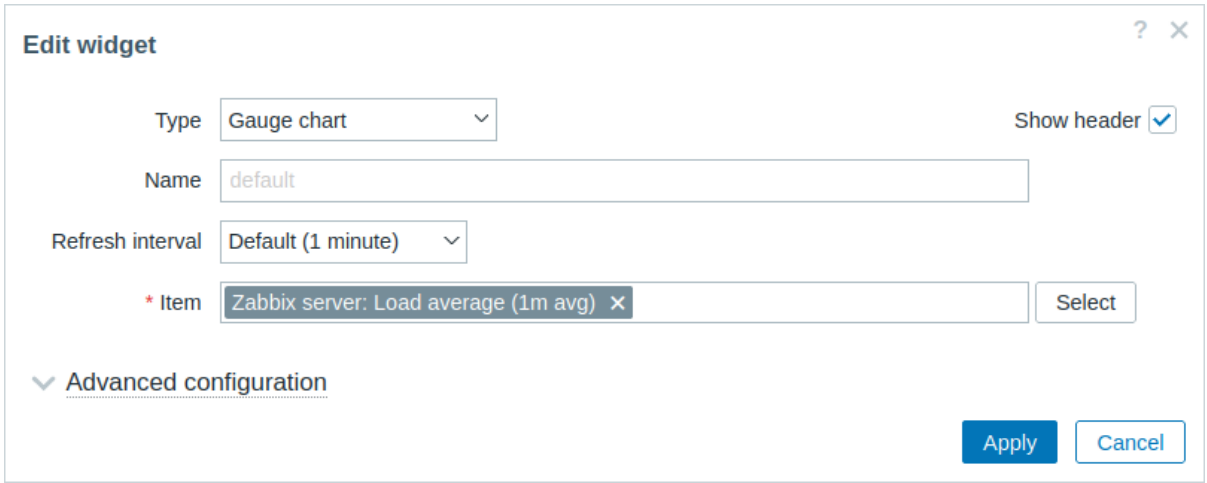
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

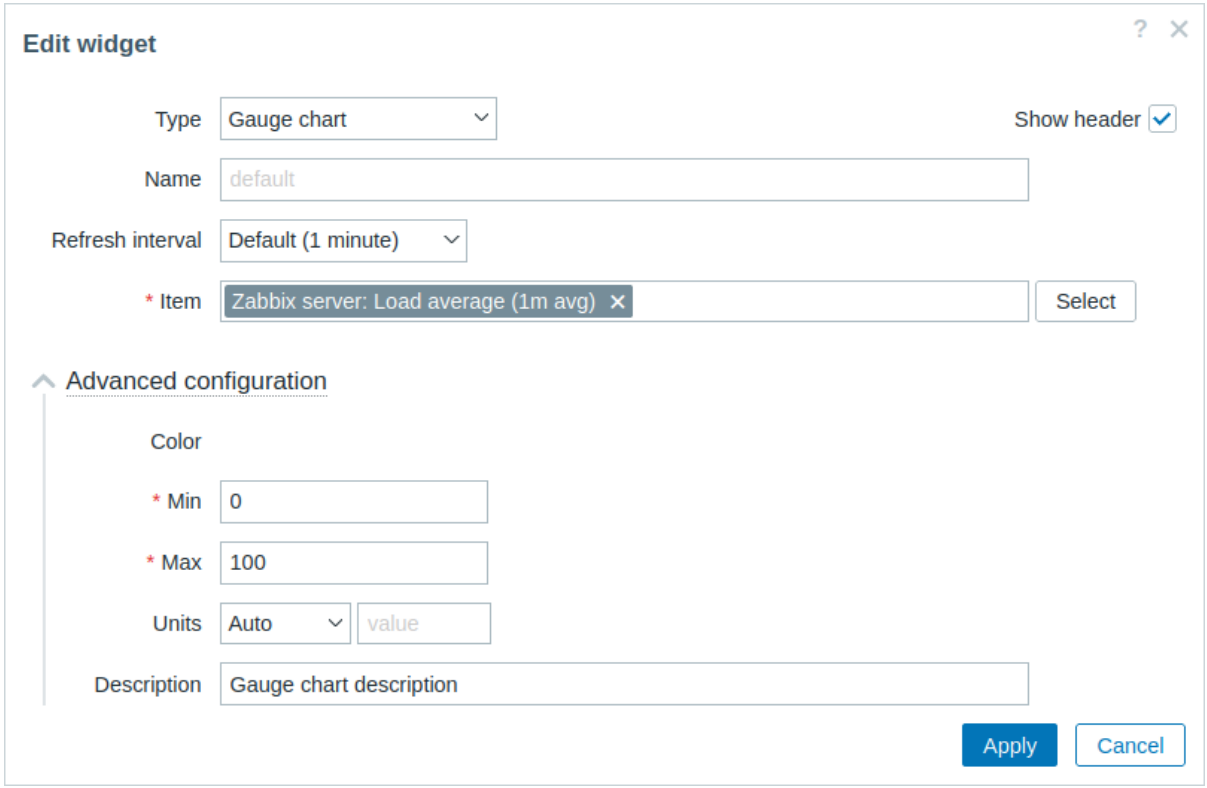
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                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:
Il metodo `addField()` della classe `CWidgetFormView` accetta una stringa di classe CSS come secondo parametro.

4. Torna alla dashboard, passa alla modalità di modifica e fai clic sull'icona a ingranaggio nel widget per aprire il modulo di configurazione del widget. Il modulo di configurazione del widget ora contiene una nuova sezione espandibile/comprimibile *Advanced configuration*.



5. Espandi la sezione *Advanced configuration* per vedere i campi aggiuntivi di configurazione del widget. Nota che il campo *Color* non ha ancora un selettore colore. Questo perché il selettore colore deve essere inizializzato con JavaScript, che verrà aggiunto nella prossima sezione - *Add JavaScript to the widget*.



Aggiungere JavaScript al widget

In questa sezione imparerai come aggiungere un grafico a tachimetro, realizzato con JavaScript, che mostra se l'ultimo valore è normale oppure troppo alto/troppo basso.

1. Crea un file `widget.edit.js.php` nella directory `views`.

JavaScript sarà responsabile dell'inizializzazione del selettore di colori nella vista di configurazione.

`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. Apri `views/widget.edit.php` e aggiungi il file `widget.edit.js.php` con il JavaScript alla vista di configurazione. Per farlo, usa il metodo `includeJsFile()`. Per aggiungere JavaScript inline, usa il metodo `addJavaScript()`.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

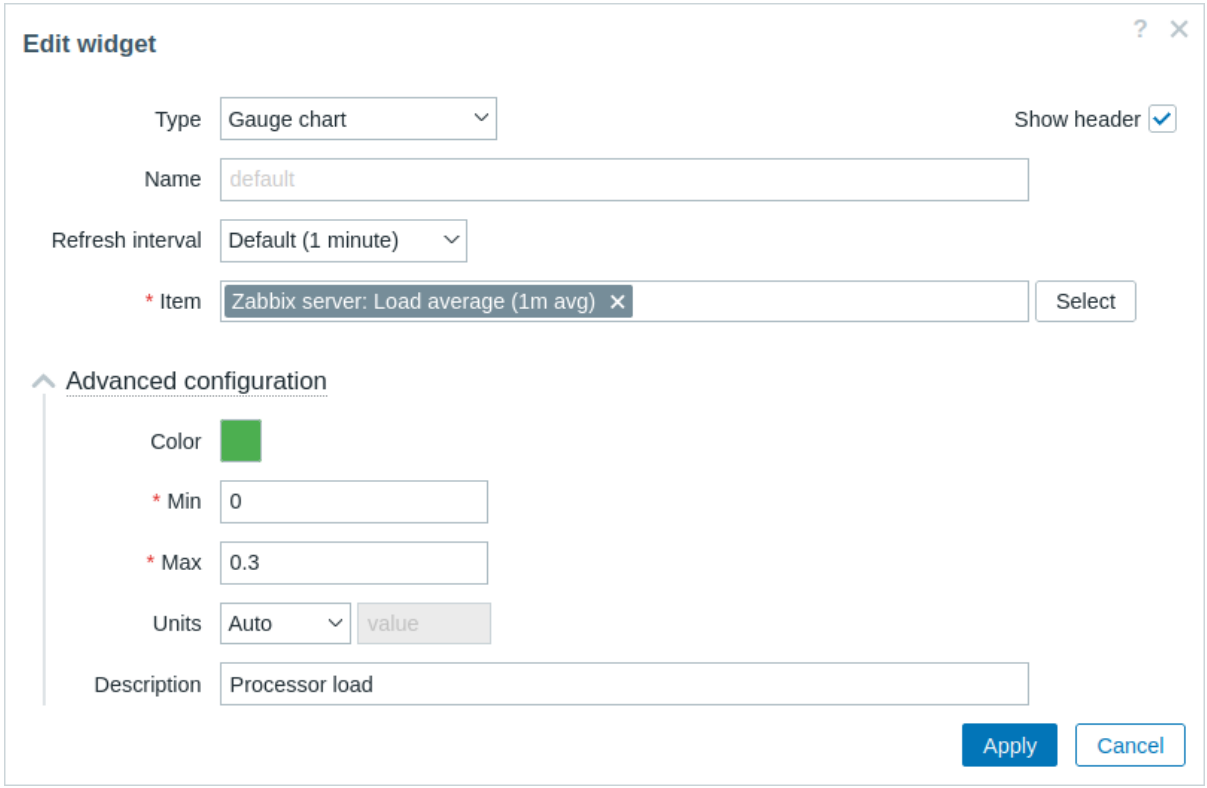
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

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. Torna alla dashboard, fai clic sull'icona a ingranaggio nel widget per aprire il modulo di configurazione del widget. Quindi espandi la sezione *Advanced configuration* per vedere il selettore di colori inizializzato. Compila i campi con dei valori e seleziona un colore per il grafico a tachimetro.



4. Fai clic su *Apply* nel modulo di configurazione del widget. Quindi fai clic su *Save changes* nell'angolo in alto a destra per salvare la dashboard.

5. Apri `actions/WidgetView.php` e aggiorna il controller.

La proprietà `$this->fields_values` ora contiene i valori di tutti i campi di *Advanced configuration*. Completa il controller per consentire il passaggio della configurazione e del valore dell'item selezionato alla vista del widget.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
```

```
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. Apri e modifica `views/widget.view.php`.

Devi creare un contenitore per il grafico a tachimetro, che disegnerai nei passaggi successivi, e un contenitore per la descrizione.

Per passare valori a JavaScript come oggetto JSON, usa il metodo `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. Crea una nuova directory `assets` nella directory `lesson_gauge_chart`. Questa directory verrà usata per memorizzare JavaScript, CSS e, potenzialmente, altre risorse come font o immagini.
8. Per il JavaScript della vista del widget, crea una directory `js` nella directory `assets`.
9. Crea un file `class.widget.js` nella directory `assets/js`.

Questa classe JavaScript del widget estenderà la classe JavaScript base di tutti i widget della dashboard, `CWidget`.

La dashboard si basa su un'implementazione corretta di un widget e comunica qualsiasi informazione rilevante al widget chiamando i rispettivi metodi JavaScript. La dashboard si aspetta inoltre che il widget generi eventi quando si verifica qualche interazione. Pertanto, la classe `CWidget` contiene un insieme di metodi con l'implementazione predefinita del comportamento del widget, che può essere personalizzata estendendo la classe.

In questo caso, è necessaria una certa personalizzazione, quindi la logica personalizzata verrà implementata per il seguente comportamento del widget:

- inizializzazione del widget, responsabile della definizione dello stato iniziale del widget (vedi il metodo `onInitialize()`);
- visualizzazione del contenuto del widget (cioè il disegno del grafico a tachimetro) se il processo di aggiornamento del widget è andato a buon fine e senza errori (vedi il metodo `processUpdateResponse(response)` e i metodi correlati `_resizeChart()` e `_updatedChart()`)
- ridimensionamento del widget (vedi il metodo `onResize()` e il metodo correlato `_resizeChart()`)

Per gli altri aspetti del widget del grafico a tachimetro, verrà usata l'implementazione predefinita del comportamento del widget. Per saperne di più sui metodi JavaScript della classe `CWidget`, vedi: JavaScript.

Poiché questo JavaScript è richiesto per la vista del widget, deve essere caricato con la pagina della dashboard. Per abilitare il caricamento del JavaScript, dovrai aggiornare i parametri `assets/js` e `js_class` nel file `manifest.json` come mostrato nel passaggio 10.

`ui/modules/lesson_gauge_chart/assets/js/class.widget.js`

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }

    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');

            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
```

```
        }

        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;

        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;

                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);
```

```
                ctx.beginPath();
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';

                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. Apri `manifest.json` e aggiungi:

- il nome del file (`class.widget.js`) all'array nella sezione `assets/js`;
- il nome della classe (`WidgetLessonGaugeChart`) al parametro `js_class` nella sezione `widget`.

La classe `WidgetLessonGaugeChart` verrà ora caricata automaticamente con la dashboard.

`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"]
    }
}
```

Aggiungere stili CSS al widget

In questa sezione imparerai come aggiungere stili CSS personalizzati per rendere il widget più gradevole.

1. Per gli stili del widget, crea una nuova directory `css` nella directory `assets`.
2. Crea un file `widget.css` nella directory `assets/css`. Per stilizzare gli elementi del widget, usa il selettore `div.dashboard-widget-{id}`. Per configurare il CSS per l'intero widget, usa il selettore `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;
}
```

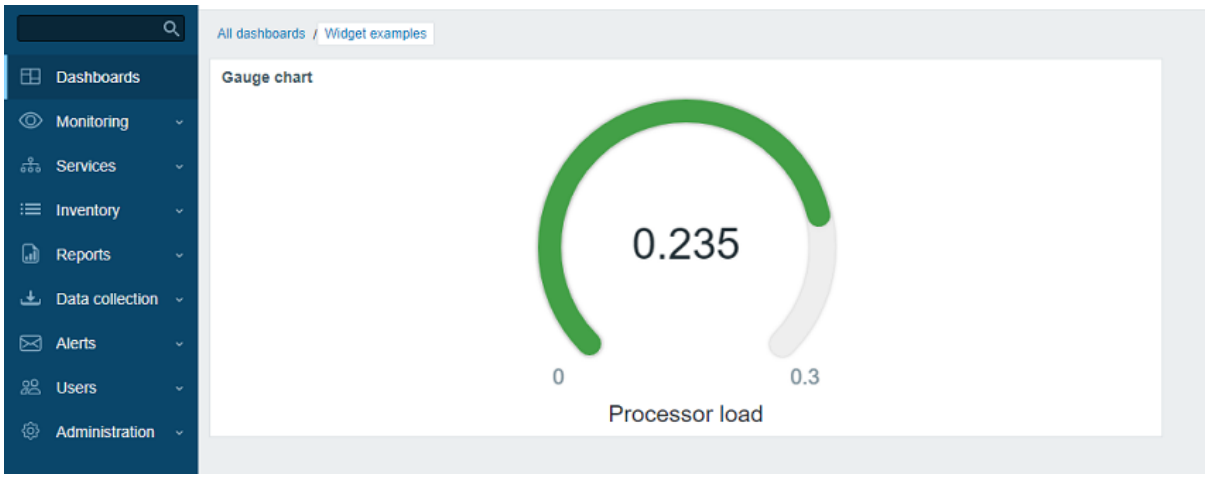
3. Apri `manifest.json` e aggiungi il nome del file CSS (`widget.css`) all'array nella sezione `assets/css`. Questo consentirà di caricare gli stili CSS definiti in `widget.css` insieme alla pagina del dashboard.

`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. Aggiorna la pagina del dashboard per vedere la versione finale del widget.



Esempi

Questa sezione fornisce file di esempio di moduli e widget, che puoi utilizzare come base per i tuoi moduli personalizzati.

Per utilizzare un modulo:

1. Scarica l'archivio ZIP.
2. Decomprimi il contenuto in una directory separata all'interno della directory *modules* dell'installazione del frontend Zabbix (ad esempio, *zabbix/ui/modules*).
3. Registra il modulo nel frontend Zabbix.

Esempio di modulo

- Quando crei un gruppo di host, concedi le autorizzazioni di lettura ai gruppi di utenti configurati - hg_auto_perm.zip

Widget examples

- Widget Minimale - widget_min.zip
- "Hello, world" widget che utilizza solo CSS - hello_world_css.zip
- "Hello, world" widget che utilizza solo JavaScript - hello_world_js.zip
- "Hello, world" widget che utilizza PHP - hello_world_php.zip

> **Note:**
> Puoi usare anche gli Zabbix native widgets come esempi.

Modifiche allo sviluppo delle estensioni

Modifiche dalla 7.4 alla 8.0

Pagine man di Zabbix

Documentazione di riferimento per gli strumenti e i servizi a riga di comando di Zabbix, con l'elenco dei comandi disponibili, delle opzioni e degli esempi di utilizzo.

zabbix_agent2

```
ZABBIX_AGENT2(8)            Manuale dell'amministratore di sistema           ZABBIX_AGENT2(8)

NOME
       zabbix_agent2 - Zabbix agent 2

SINOSSI
       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

DESCRIZIONE
       zabbix_agent2  è  un'applicazione per monitorare i parametri di vari
       servizi.

OPZIONI
       -c, --config config-file
              Usa il file di configurazione alternativo invece di quello predefinito.

       -R, --runtime-control runtime-option
              Esegue funzioni amministrative in base a runtime-option.

           Opzioni di controllo runtime
               userparameter_reload
                   Ricarica i parametri utente dal file di configurazione

               log_level_increase
                   Aumenta il livello di log

               log_level_decrease
                   Diminuisce il livello di log

               help
                   Elenca le opzioni di controllo runtime disponibili

               metrics
                   Elenca le metriche disponibili
               version
                   Visualizza la versione

       -p, --print
              Stampa gli item noti ed esce. Per ogni item vengono usati
              valori predefiniti generici oppure vengono forniti valori
              predefiniti specifici per il test. Questi valori predefiniti
              sono elencati tra parentesi quadre come parametri della chiave
              item. I valori restituiti sono racchiusi tra parentesi quadre e
              preceduti dal tipo del valore restituito, separato da una barra
              verticale. Per i parametri utente il tipo è sempre t, poiché
```

```
              l'agent non può determinare tutti i possibili valori restituiti.
              Gli item visualizzati come funzionanti non sono garantiti come
              funzionanti dal server Zabbix o da zabbix_get quando interrogano
              un demone agent in esecuzione, poiché i permessi o l'ambiente
              potrebbero essere diversi. I tipi di valore restituito sono:

           d   Numero con parte decimale.

           m   Non supportato. Questo può essere causato dall'interrogazione di
               un item che funziona solo in modalità attiva, come un item di
               monitoraggio dei log, oppure di un item che richiede più valori
               raccolti. Anche problemi di permessi o parametri utente errati
               possono causare lo stato non supportato.

           s   Testo. Lunghezza massima non limitata.

           t   Testo. Uguale a s.

           u   Intero senza segno.

       -t, --test item-key
              Verifica un singolo item ed esce. Vedere --print per la descrizione dell'output.

       -v, --verbose
              Abilita l'output dettagliato durante il test di un item o la stampa degli item noti.

       -T, --test-config
              Convalida il file di configurazione ed esce.

       -h, --help
              Visualizza questo aiuto ed esce.

       -V, --version
              Mostra le informazioni sulla versione ed esce.

FILE
       /usr/local/etc/zabbix_agent2.conf
              Posizione predefinita del file di configurazione di Zabbix agent 2 (se non
              modificata in fase di compilazione).

VEDERE ANCHE
       Documentazione

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_AGENT2(8)
```

zabbix_agentd

```
ZABBIX_AGENTD(8)            Manuale dell'amministratore di sistema           ZABBIX_AGENTD(8)

NOME
       zabbix_agentd - demone dell'agent Zabbix

SINOSSI
       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

DESCRIZIONE
       zabbix_agentd è un demone per il monitoraggio di vari parametri del server.

OPZIONI
       -c, --config config-file
              Usa il file di configurazione alternativo invece di quello predefinito.

       -f, --foreground
              Esegue l'agent Zabbix in primo piano.

       -R, --runtime-control runtime-option
              Esegue funzioni amministrative in base a runtime-option.

           Opzioni di controllo a runtime
               userparameter_reload
                   Ricarica i parametri utente dal file di configurazione

               log_level_increase[=target]
                   Aumenta il livello di log; interessa tutti i processi se target non
                   è specificato

               log_level_decrease[=target]
                   Diminuisce il livello di log; interessa tutti i processi se target non
                   è specificato

           Destinazioni del controllo del livello di log
               process-type
                   Tutti i processi del tipo specificato (controlli attivi, collector,
                   listener)

               process-type,N
                   Tipo di processo e numero (ad es. listener,3)
               pid Identificatore di processo, fino a 65535. Per valori maggiori specificare
                   target come "process-type,N"

       -p, --print
              Stampa gli item conosciuti ed esce. Per ciascun item vengono usati
              valori predefiniti generici oppure, per i test, vengono forniti
              valori predefiniti specifici. Questi valori predefiniti sono elencati
              tra parentesi quadre come parametri della chiave item. I valori
              restituiti sono racchiusi tra parentesi quadre e preceduti dal tipo
              del valore restituito, separato da un carattere pipe. Per i parametri
              utente il tipo è sempre t, poiché l'agent non può determinare tutti
              i possibili valori restituiti. Gli item visualizzati come funzionanti
              non sono garantiti come funzionanti dal server Zabbix o da
              zabbix_get quando interrogano un demone agent in esecuzione, poiché
              i permessi o l'ambiente possono essere diversi. I tipi di valori
              restituiti sono:

           d   Numero con parte decimale.

           m   Non supportato. Ciò può essere causato dall'interrogazione di un item che
               funziona solo in modalità attiva, come un item di monitoraggio log o un
               item che richiede più valori raccolti. Problemi di permessi o parametri
               utente non corretti possono anch'essi causare lo stato non supportato.

           s   Testo. Lunghezza massima non limitata.

           t   Testo. Uguale a s.
```

```
           u   Intero senza segno.

       -t, --test item-key
              Verifica un singolo item ed esce. Vedere --print per la descrizione dell'output.

       -T, --test-config
              Convalida il file di configurazione ed esce.

       -h, --help
              Visualizza questo aiuto ed esce.

       -V, --version
              Mostra le informazioni sulla versione ed esce.

FILE
       /usr/local/etc/zabbix_agentd.conf
              Posizione predefinita del file di configurazione dell'agent Zabbix (se non
              modificata in fase di compilazione).

VEDERE ANCHE
       Documentazione

       zabbix_get(1),  zabbix_proxy(8),  zabbix_sender(1),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_AGENTD(8)
```

zabbix_get

```
ZABBIX_GET(1)               Manuale generale dei comandi         ZABBIX_GET(1)

NOME
       zabbix_get - utility get di Zabbix

SINOSSI
       zabbix_get  -s  host-name-or-IP  [-p  port-number]  [-I IP-address] [-t
       timeout] -k item-key
       zabbix_get -s host-name-or-IP  [-p  port-number]  [-I  IP-address]  [-t
       timeout]   --tls-connect  cert  --tls-ca-file  CA-file  [--tls-crl-file
       CRL-file] [--tls-agent-cert-issuer cert-issuer]  [--tls-agent-cert-sub-
       ject  cert-subject]  --tls-cert-file  cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k item-key
       zabbix_get -s host-name-or-IP  [-p  port-number]  [-I  IP-address]  [-t
       timeout]     --tls-connect    psk    --tls-psk-identity    PSK-identity
       --tls-psk-file PSK-file  [--tls-cipher13  cipher-string]  [--tls-cipher
       cipher-string] -k item-key
       zabbix_get -h
       zabbix_get -V

DESCRIZIONE
       zabbix_get  è  un'utility  da  riga  di  comando  per ottenere dati da
       Zabbix agent.

OPZIONI
       -s, --host host-name-or-IP
              Specifica il nome host o l'indirizzo IP di un host.

       -p, --port port-number
              Specifica il numero di porta dell'agent in esecuzione sull'host.
              Il valore predefinito è 10050.
```

```
-I, --source-address IP-address
       Specifica l'indirizzo IP di origine.

-t, --timeout seconds
       Specifica il timeout. Intervallo valido: 1-600 secondi (predefi-
       nito: 30)

-k, --key item-key
       Specifica la chiave dell'item di cui recuperare il valore.

-P, --protocol value
       Protocollo usato per comunicare con l'agent. Valori:

            auto        si connette usando il protocollo JSON, in caso
                        di errore ripiega e riprova con il protocollo in
                        testo semplice (predefinito)

            json        si connette usando il protocollo JSON

            plaintext   si connette usando il protocollo in testo sem-
                        plice in cui viene inviata solo la chiave
                        dell'item (release 6.4.x e precedenti)

--tls-connect value
       Come connettersi all'agent. Valori:

            unencrypted
                        si connette senza crittografia (predefinito)

            psk         si connette usando TLS e una chiave precondivisa

            cert        si connette usando TLS e un certificato

--tls-ca-file CA-file
       Percorso completo di un file contenente i certificati della CA
       di livello superiore per la verifica del certificato del peer.

--tls-crl-file CRL-file
       Percorso completo di un file contenente certificati revocati.

--tls-agent-cert-issuer cert-issuer
       Emittente del certificato dell'agent consentito.

--tls-agent-cert-subject cert-subject
       Soggetto del certificato dell'agent consentito.

--tls-cert-file cert-file
       Percorso completo di un file contenente il certificato o la
       catena di certificati.

--tls-key-file key-file
       Percorso completo di un file contenente la chiave privata.

--tls-psk-identity PSK-identity
       Stringa PSK-identity.

--tls-psk-file PSK-file
       Percorso completo di un file contenente la chiave precondivisa.

--tls-cipher13 cipher-string
       Stringa di cifratura per OpenSSL 1.1.1 o versioni successive per
       TLS 1.3. Sovrascrive i criteri predefiniti di selezione della
       suite di cifratura. Questa opzione non è disponibile se la ver-
```

```
              sione di OpenSSL è inferiore a 1.1.1.

       --tls-cipher cipher-string
              Stringa di priorità GnuTLS (per TLS 1.2 e superiori) oppure
              stringa di cifratura OpenSSL (solo per TLS 1.2). Sovrascrive i
              criteri predefiniti di selezione della suite di cifratura.

       -h, --help
              Visualizza questo aiuto ed esce.

       -V, --version
              Mostra le informazioni sulla versione ed esce.

STATO DI USCITA
       Lo stato di uscita è 0 se zabbix_get recupera correttamente un valore
       da Zabbix agent. Se il recupero del valore non riesce, lo stato di
       uscita è 1.

ESEMPI
       zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
       zabbix_get   -s   127.0.0.1  -p  10050  -k  "system.cpu.load[all,avg1]"
       --tls-connect    cert     --tls-ca-file     /home/zabbix/zabbix_ca_file
       --tls-agent-cert-issuer   "CN=Signing   CA,OU=IT   operations,O=Example
       Corp,DC=example,DC=com" --tls-agent-cert-subject "CN=server1,OU=IT  op-
       erations,O=Example  Corp,DC=example,DC=com"  --tls-cert-file /home/zab-
       bix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
       zabbix_get  -s  127.0.0.1  -p  10050   -k   "system.cpu.load[all,avg1]"
       --tls-connect   psk   --tls-psk-identity   "PSK   ID   Zabbix   agentd"
       --tls-psk-file /home/zabbix/zabbix_agentd.psk

VEDI ANCHE
       Documentazione

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                      ZABBIX_GET(1)
```

zabbix_js

```
ZABBIX_JS(1)                Manuale generale dei comandi                ZABBIX_JS(1)

NOME
       zabbix_js - Utilità JS di Zabbix

SINOSSI
       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

DESCRIZIONE
       zabbix_js è un'utilità da riga di comando che può essere utilizzata per
       testare script incorporati.

OPZIONI
       -s, --script script-file
              Specifica il nome del file dello script da eseguire. Se '-' viene
              specificato come nome file, lo script verrà letto da stdin.

       -p, --param input-param
```

```
              Specifica il parametro di input passato allo script. Il parametro
              viene passato letteralmente come stringa ed è disponibile nello
              script tramite la variabile 'value'. Mutuamente esclusivo con -i.

       -i, --input input-file
              Specifica un file contenente il parametro di input passato allo
              script. Il contenuto del file viene passato letteralmente come
              stringa ed è disponibile nello script tramite la variabile
              'value'. Se '-' viene specificato come nome file, l'input verrà
              letto da stdin. Mutuamente esclusivo con -p.

       -w, --webdriver url
              Specifica l'URL del webdriver.

       -l, --loglevel log-level
              Specifica il livello di log (intervallo: 0-5).

       -t, --timeout timeout
              Specifica il timeout in secondi. Intervallo valido: 1-600 secondi
              (predefinito: 10)

       -h, --help
              Visualizza questa guida ed esce.

       -V, --version
              Mostra le informazioni sulla versione ed esce.

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

VEDERE ANCHE
       Documentazione

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                       ZABBIX_JS(1)
```

zabbix_proxy

```
ZABBIX_PROXY(8)             Manuale del System Manager            ZABBIX_PROXY(8)

NOME
       zabbix_proxy - daemon proxy di Zabbix

SINOSSI
       zabbix_proxy [-c file-di-configurazione]
       zabbix_proxy [-c file-di-configurazione] -R opzione-runtime
       zabbix_proxy [-c file-di-configurazione] -T
       zabbix_proxy -h
       zabbix_proxy -V

DESCRIZIONE
       zabbix_proxy è un daemon che raccoglie i dati di monitoraggio dai dispositivi e
       li invia a Zabbix server.

OPZIONI
       -c, --config file-di-configurazione
              Usa il file-di-configurazione alternativo invece di quello predefinito.

       -f, --foreground
```

```
       Esegui Zabbix proxy in primo piano.

-R, --runtime-control opzione-runtime
       Esegui funzioni amministrative in base a opzione-runtime.

    Opzioni di controllo runtime
        config_cache_reload
            Ricarica la cache di configurazione. Ignorato se la cache è attualmente
            in fase di caricamento. Il Zabbix proxy attivo si connetterà a
            Zabbix server e richiederà i dati di configurazione. Il file di
            configurazione predefinito (a meno che non sia specificata l'opzione -c)
            verrà usato per individuare il file PID e il segnale verrà inviato al processo
            elencato nel file PID.

        snmp_cache_reload
            Ricarica la cache SNMP.

        housekeeper_execute
            Esegue l'housekeeper. Ignorato se l'housekeeper è attualmente in esecuzione.

        diaginfo[=sezione]
            Registra informazioni diagnostiche interne della sezione specificata.
            La sezione può essere historycache, preprocessing, locks.
            Per impostazione predefinita vengono registrate le informazioni diagnostiche
            di tutte le sezioni.

        prof_enable[=target]
            Abilita il profiling, interessa tutti i processi se target non è specificato

        prof_disable[=target]
            Disabilita il profiling, interessa tutti i processi se target non è specificato

        log_level_increase[=target]
            Aumenta il livello di log, interessa tutti i processi se target non è specificato.

        log_level_decrease[=target]
            Diminuisce il livello di log, interessa tutti i processi se target non è specificato.

    Target del controllo del livello di log
        process-type
            Tutti i processi del tipo specificato (availability manager,
            browser poller, configuration syncer, data sender, discovery manager,
            history syncer, housekeeper, http poller, icmp pinger, internal poller,
            ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller,
            http agent poller, snmp poller, preprocessing manager, preprocessing worker,
            self-monitoring, snmp trapper, supervisor, task manager, trapper,
            unreachable poller, vmware collector)

        process-type,N
            Tipo di processo e numero (ad esempio, poller,3)

        pid Identificatore del processo

    Target del controllo del profiling
        process-type
            Tutti i processi del tipo specificato (availability manager,
            browser poller, configuration syncer, data sender, discovery manager,
            history syncer, housekeeper, http poller, icmp pinger, internal poller,
            ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller,
            http agent poller, snmp poller, preprocessing manager, self-monitoring,
            snmp trapper, task manager, trapper, unreachable poller,
            vmware collector)
```

```
               process-type,N
                   Tipo di processo e numero (ad esempio, history syncer,1)

               pid Identificatore del processo

               scope
                   (rwlock, mutex, processing) può essere usato con process-type
                   (ad esempio, history syncer,1,processing)

       -T, --test-config
              Valida il file di configurazione ed esce.

       -h, --help
              Mostra questo aiuto ed esce.

       -V, --version
              Mostra le informazioni sulla versione ed esce.

FILE
       /usr/local/etc/zabbix_proxy.conf
              Posizione predefinita del file di configurazione di Zabbix proxy (se non
              modificata in fase di compilazione).

VEDI ANCHE
       Documentazione

       zabbix_agentd(8), zabbix_get(1), zabbix_sender(1), zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                   ZABBIX_PROXY(8)
```

zabbix_sender

```
ZABBIX_SENDER(1)            Manuale dei comandi generali           ZABBIX_SENDER(1)

NOME
       zabbix_sender - utilità sender di Zabbix

SINTASSI
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host -k key -o value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] -k key -o value
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file]
       [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject
       cert-subject] --tls-cert-file cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k key -o
       value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
       CRL-file] [--tls-server-cert-issuer cert-issuer]
       [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
       --tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
```

```
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k
key -o value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string]
[-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender -h
zabbix_sender -V
```

DESCRIZIONE

zabbix_sender è un'utilità da riga di comando per inviare dati di monitoraggio al server o al proxy Zabbix. Sul server Zabbix deve essere creato un item di tipo Zabbix trapper con la chiave corrispondente. Si noti che i valori in ingresso saranno accettati solo dagli host specificati nel campo Allowed hosts di questo item.

OPZIONI

-c, --config config-file

Usa config-file. Zabbix sender legge i dettagli del server dal file di configurazione di agentd. Per impostazione predefinita, Zabbix sender non legge alcun file di configurazione. Sono supportati solo i parametri Hostname, ServerActive, SourceIP, TLSConnect, TLSCAFile, TLSCRLFile, TLSServerCertIssuer, TLSServerCertSubject, TLSCertFile, TLSKeyFile, TLSPSKIdentity e TLSPSKFile. Hostname definito tramite il parametro HostnameItem non verrà utilizzato; in questo caso il nome host deve essere specificato tramite la riga di comando (vedere l'opzione -s). Per l'invio dei dati vengono utilizzati tutti gli indirizzi definiti nel parametro di configurazione ServerActive dell'agent. Se l'invio dei dati in batch non riesce verso un indirizzo, i batch successivi non vengono inviati a tale indirizzo.

-z, --zabbix-server server

Nome host o indirizzo IP del server Zabbix. Se un host è monitorato da un proxy, è necessario utilizzare il nome host o l'indirizzo IP del proxy. Se usato insieme a --config, sostituisce le voci del parametro ServerActive specificate nel file di configurazione di agentd.

-p, --port port

Specifica il numero di porta su cui è in esecuzione il trapper del server Zabbix. Il valore predefinito è 10051. Se usato insieme a --config, sostituisce le voci della porta del parametro

ServerActive specificate nel file di configurazione di agentd.

-I, --source-address IP-address
Specifica l'indirizzo IP di origine. Se usato insieme a --config, sostituisce il parametro SourceIP specificato nel file di configurazione di agentd.

-t, --timeout seconds
Specifica il timeout. Intervallo valido: 1-300 secondi (predefinito: 60)

-s, --host host
Specifica il nome host a cui appartiene l'item (come registrato nel frontend Zabbix). L'indirizzo IP e il nome DNS dell'host non funzioneranno. Se usato insieme a --config, sostituisce il parametro Hostname specificato nel file di configurazione di agentd.

-k, --key key
Specifica la chiave dell'item a cui inviare il valore.

-o, --value value
Specifica il valore dell'item.

-i, --input-file input-file
Carica i valori dal file di input. Specificare - come <input-file> per leggere i valori dallo standard input. Ogni riga del file contiene, separati da spazi bianchi: <hostname> <key> <value>. Ogni valore deve essere specificato su una riga separata. Ogni riga deve contenere 3 elementi separati da spazi bianchi: <hostname> <key> <value>, dove "hostname" è il nome dell'host monitorato registrato nel frontend Zabbix, "key" è la chiave dell'item di destinazione e "value" è il valore da inviare. Specificare - come <hostname> per usare il nome host dal file di configurazione dell'agent o dall'argomento --host.

Esempio di una riga di un file di input:

"Linux DB3" db.connections 43

Il tipo di valore deve essere impostato correttamente nella configurazione dell'item del frontend Zabbix. Zabbix sender invierà fino a 250 valori in una connessione. Il contenuto del file di input deve essere codificato in UTF-8. Tutti i valori del file di input vengono inviati in ordine sequenziale dall'alto verso il basso. Le voci devono essere formattate secondo le seguenti regole:

- Sono supportate voci tra virgolette e non tra virgolette.
- Il doppio apice è il carattere di quotatura.
- Le voci contenenti spazi bianchi devono essere racchiuse tra virgolette.
- I caratteri doppio apice e barra rovesciata all'interno di una voce tra virgolette devono essere preceduti da una barra rovesciata.
- L'escape non è supportato nelle voci non racchiuse tra virgolette.

- Le sequenze di escape per il ritorno a capo (\n) sono supportate nelle stringhe tra virgolette.

- Le sequenze di escape per il ritorno a capo vengono rimosse dalla fine di una voce.

-T, --with-timestamps
       Questa opzione può essere usata solo con l'opzione --input-file.

       Ogni riga del file di input deve contenere 4 elementi separati da spazi bianchi: <hostname> <key> <timestamp> <value>. Il timestamp deve essere specificato nel formato Unix timestamp. Se l'item di destinazione ha trigger che fanno riferimento a esso, tutti i timestamp devono essere in ordine crescente; in caso contrario, il calcolo dell'evento non sarà corretto.

       Esempio di una riga di un file di input:

        "Linux DB3" db.connections 1429533600 43

       Per ulteriori dettagli, vedere l'opzione --input-file.

       Se viene inviato un valore con timestamp per un host che si trova in un periodo di manutenzione di tipo “no data”, il valore verrà eliminato; tuttavia, è possibile inviare un valore con timestamp relativo a un periodo di manutenzione scaduto e il valore verrà accettato.

-N, --with-ns
       Questa opzione può essere usata solo con l'opzione --with-timestamps.

       Ogni riga del file di input deve contenere 5 elementi separati da spazi bianchi: <hostname> <key> <timestamp> <ns> <value>.

       Esempio di una riga di un file di input:

        "Linux DB3" db.connections 1429533600 7402561 43

       Per ulteriori dettagli, vedere l'opzione --input-file.

-r, --real-time
       Invia i valori uno alla volta non appena vengono ricevuti. Può essere usato durante la lettura dallo standard input.

--tls-connect value
       Come connettersi al server o al proxy. Valori:

            unencrypted
                        connessione senza crittografia (predefinito)

            psk   connessione tramite TLS e una chiave precondivisa

            cert   connessione tramite TLS e un certificato

--tls-ca-file CA-file
       Percorso completo di un file contenente i certificati delle CA di livello superiore per la verifica del certificato del peer.

--tls-crl-file CRL-file
       Percorso completo di un file contenente i certificati revocati.

```
       --tls-server-cert-issuer cert-issuer
              Emittente del certificato del server consentito.

       --tls-server-cert-subject cert-subject
              Subject del certificato del server consentito.

       --tls-cert-file cert-file
              Percorso completo di un file contenente il certificato o la
              catena di certificati.

       --tls-key-file key-file
              Percorso completo di un file contenente la chiave privata.

       --tls-psk-identity PSK-identity
              Stringa PSK-identity.

       --tls-psk-file PSK-file
              Percorso completo di un file contenente la chiave precondivisa.

       --tls-cipher13 cipher-string
              Stringa di cifratura per OpenSSL 1.1.1 o versioni successive per
              TLS 1.3. Sostituisce i criteri predefiniti di selezione della
              suite di cifratura. Questa opzione non è disponibile se la
              versione di OpenSSL è precedente alla 1.1.1.

       --tls-cipher cipher-string
              Stringa di priorità GnuTLS (per TLS 1.2 e versioni successive) o
              stringa di cifratura OpenSSL (solo per TLS 1.2). Sostituisce i
              criteri predefiniti di selezione della suite di cifratura.

       -v, --verbose
              Modalità dettagliata; usare -vv per ulteriori dettagli.

       -h, --help
              Visualizza questo aiuto ed esce.

       -V, --version
              Visualizza le informazioni sulla versione ed esce.

STATO DI USCITA
       Lo stato di uscita è 0 se i valori sono stati inviati e tutti sono stati
       elaborati correttamente dal server. Se i dati sono stati inviati, ma
       l'elaborazione di almeno uno dei valori non è riuscita, lo stato di
       uscita è 2. Se l'invio dei dati non è riuscito, lo stato di uscita è 1.

ESEMPI
       zabbix_sender -c  /etc/zabbix/zabbix_agentd.conf  -k  mysql.queries  -o
       342.45
              Invia 342.45 come valore dell'item mysql.queries dell'host
              monitorato. Usa l'host monitorato e il server Zabbix definiti nel
              file di configurazione dell'agent.

       zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host"  -k
       mysql.queries -o 342.45
              Invia 342.45 come valore dell'item mysql.queries dell'host
              Monitored Host usando il server Zabbix definito nel file di
              configurazione dell'agent.

       zabbix_sender -z 192.168.1.113 -i data_values.txt
              Invia i valori dal file data_values.txt al server Zabbix con
              indirizzo IP 192.168.1.113. I nomi host e le chiavi sono definiti
              nel file.
```

```
       echo "- hw.serial.number  1287872261  SQ4321ASDF"  |  zabbix_sender  -c
       /usr/local/etc/zabbix_agentd.conf -T -i -
              Invia un valore con timestamp dalla riga di comando al server
              Zabbix specificato nel file di configurazione dell'agent. Il
              trattino nei dati di input indica che anche il nome host deve
              essere utilizzato dallo stesso file di configurazione.

       echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113
       -p 10000 -i -
              Invia un valore vuoto di un item al server Zabbix con indirizzo
              IP 192.168.1.113 sulla porta 10000 dalla riga di comando. I
              valori vuoti devono essere indicati da doppi apici vuoti.

       zabbix_sender -z 192.168.1.113  -s "Monitored Host" -k mysql.queries -o
       342.45  --tls-connect  cert  --tls-ca-file  /home/zabbix/zabbix_ca_file
       --tls-cert-file      /home/zabbix/zabbix_agentd.crt      --tls-key-file
       /home/zabbix/zabbix_agentd.key
              Invia 342.45 come valore dell'item mysql.queries dell'host
              Monitored Host al server con indirizzo IP 192.168.1.113 usando
              TLS con un certificato.

       zabbix_sender -z 192.168.1.113  -s "Monitored Host" -k mysql.queries -o
       342.45  --tls-connect  psk  --tls-psk-identity  "PSK  ID Zabbix agentd"
       --tls-psk-file /home/zabbix/zabbix_agentd.psk
              Invia 342.45 come valore dell'item mysql.queries dell'host
              Monitored Host al server con indirizzo IP 192.168.1.113 usando
              TLS con una chiave precondivisa (PSK).

VEDERE ANCHE
       Documentazione

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SENDER(1)
```

zabbix_server

```
ZABBIX_SERVER(8)            Manuale del System Manager           ZABBIX_SERVER(8)

NOME
       zabbix_server - demone del server Zabbix

SINOSSI
       zabbix_server [-c file-di-configurazione]
       zabbix_server [-c file-di-configurazione] -R opzione-runtime
       zabbix_server [-c file-di-configurazione] -T
       zabbix_server -h
       zabbix_server -V

DESCRIZIONE
       zabbix_server è il demone principale del software Zabbix.

OPZIONI
       -c, --config file-di-configurazione
              Usa il file-di-configurazione alternativo invece di quello predefinito.

       -f, --foreground
              Esegui Zabbix server in primo piano.

       -R, --runtime-control opzione-runtime
```

```
   Esegui funzioni amministrative in base a opzione-runtime.

Opzioni di controllo runtime
    config_cache_reload
        Ricarica la cache di configurazione. Ignorato se la cache è
        attualmente in caricamento. Verrà usato il file di configurazione
        predefinito (a meno che non sia specificata l'opzione -c) per
        trovare il file PID e il segnale verrà inviato al processo
        elencato nel file PID.

    snmp_cache_reload
        Ricarica la cache SNMP.

    housekeeper_execute
        Esegui l'housekeeper. Ignorato se l'housekeeper è attualmente in esecuzione.

    trigger_housekeeper_execute
        Esegui l'housekeeper dei trigger (rimuove i problemi per i trigger
        eliminati). Ignorato se l'housekeeper dei trigger è attualmente in esecuzione.

    secrets_reload
        Ricarica i segreti da Vault. Ignorato se la cache di configurazione
        o i segreti sono attualmente in caricamento.

    diaginfo[=sezione]
        Registra informazioni diagnostiche interne della sezione specificata.
        La sezione può essere historycache, preprocessing, alerting, lld,
        valuecache, locks e connector. Per impostazione predefinita, vengono
        registrate le informazioni diagnostiche di tutte le sezioni.

    prof_enable[=target]
        Abilita il profiling, interessa tutti i processi se target non è specificato

    prof_disable[=target]
        Disabilita il profiling, interessa tutti i processi se target non è specificato

    ha_status
        Mostra lo stato del cluster ad alta disponibilità. Può essere eseguito
        solo sul nodo attivo.

    ha_remove_node
        ha_remove_node[=nome|ID] Rimuove un nodo ad alta disponibilità in base
        al suo nome o ID elencato in ha_status. Può essere eseguito solo sul
        nodo attivo.

    ha_set_failover_delay[=ritardo]
        Imposta il ritardo di failover dell'alta disponibilità. Il ritardo può
        essere specificato usando suffissi temporali. Può essere eseguito solo
        sul nodo attivo.

    log_level_increase[=target]
        Aumenta il livello di log, interessa tutti i processi se target non è specificato

    log_level_decrease[=target]
        Diminuisce il livello di log, interessa tutti i processi se target non è specificato

Target del controllo del livello di log
    process-type
        Tutti i processi del tipo specificato (alerter, alert manager,
        alert syncer, 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, http agent poller, icmp
```

```
                    pinger, internal poller, ipmi manager, ipmi poller, java poller,
                    lld manager, lld worker, odbc poller, poller, agent poller,
                    preprocessing manager, preprocessing worker, proxy poller,
                    proxy group manager, report manager, report writer, self-monitoring,
                    service manager, snmp poller, snmp trapper, supervisor, task manager,
                    timer, trapper, trigger housekeeper, unreachable poller, vmware collector)

                process-type,N
                    Tipo di processo e numero (ad esempio, poller,3)

                pid Identificatore del processo

            Target del controllo del profiling
                process-type
                    Tutti i processi del tipo specificato (alerter, alert manager,
                    alert syncer, 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, http agent poller, icmp
                    pinger, internal poller, ipmi manager, ipmi poller, java poller,
                    lld manager, lld worker, odbc poller, poller, agent poller,
                    preprocessing manager, proxy poller, proxy group manager,
                    report manager, report writer, self-monitoring, service manager,
                    snmp poller, snmp trapper, task manager, timer, trapper,
                    trigger housekeeper, unreachable poller, vmware collector)

                process-type,N
                    Tipo di processo e numero (ad esempio, history syncer,1)

                pid Identificatore del processo

                scope
                    (rwlock, mutex, processing) può essere usato con process-type
                    (ad esempio, history syncer,1,processing)

        -T, --test-config
               Valida il file di configurazione ed esce.

        -h, --help
               Mostra questo aiuto ed esce.

        -V, --version
               Mostra le informazioni sulla versione ed esce.

FILE
        /usr/local/etc/zabbix_server.conf
               Posizione predefinita del file di configurazione di Zabbix server
               (se non modificata durante la compilazione).

VEDI ANCHE
        Documentazione

        zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1),
        zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SERVER(8)
```

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)       Manuale dell'amministratore di sistema      ZABBIX_WEB_SERVICE(8)
```

```
NOME
       zabbix_web_service - Servizio web di Zabbix

SINOSSI
       zabbix_web_service [-c file-di-configurazione]
       zabbix_web_service [-c file-di-configurazione] -T
       zabbix_web_service -h
       zabbix_web_service -V

DESCRIZIONE
       zabbix_web_service è un'applicazione per fornire servizi web ai compo-
       nenti di Zabbix.

OPZIONI
       -c, --config file-di-configurazione
              Usa il file-di-configurazione alternativo invece di quello predefinito.

       -T, --test-config
              Convalida il file di configurazione ed esce.

       -h, --help
              Visualizza questo aiuto ed esce.

       -V, --version
              Mostra le informazioni sulla versione ed esce.

FILE
       /usr/local/etc/zabbix_web_service.conf
              Posizione predefinita del file di configurazione del servizio web di Zabbix (se
              non modificata in fase di compilazione).

VEDERE ANCHE
       Documentazione

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06             ZABBIX_WEB_SERVICE(8)
```

Avviso sul copyright

La documentazione di Zabbix NON è distribuita con la licenza AGPL-3.0. L'uso della documentazione di Zabbix è soggetto ai seguenti termini:

È possibile creare una copia stampata di questa documentazione esclusivamente per uso personale. La conversione in altri formati è consentita purché il contenuto effettivo non venga alterato o modificato in alcun modo. Non è consentito pubblicare o distribuire questa documentazione in alcuna forma o su alcun supporto, salvo nel caso in cui la documentazione venga distribuita in modo analogo a quello utilizzato da Zabbix per diffonderla (ossia elettronicamente per il download su un sito web di Zabbix) oppure su un supporto USB o simile, a condizione tuttavia che la documentazione venga distribuita insieme al software sullo stesso supporto. Qualsiasi altro utilizzo, come la diffusione di copie stampate o l'uso di questa documentazione, in tutto o in parte, in un'altra pubblicazione, richiede il previo consenso scritto di un rappresentante autorizzato di Zabbix. Zabbix si riserva tutti i diritti su questa documentazione non espressamente concessi sopra.